



Dipl.-Ing. Rudolf Aschauer

# **Qualität als Schlüssel zur Rechtsrelevanz bei der Übernahme von Geodaten**

**DISSERTATION**

Zur Erlangung des akademischen Grades  
Doktor der technischen Wissenschaften  
Eingereicht an der

**Technischen Universität Graz**

Betreuer  
ao.Univ.-Prof.i.R. Dr.phil. tit.Univ.-Prof. Norbert Bartelme

Institut für Geodäsie

Graz, Oktober 2017

### EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

---

Datum

---

Unterschrift

# Zusammenfassung

Geodateninfrastrukturen (GDI) mit rechtlichen Inhalten werden insbesondere von öffentlichen Stellen aufgebaut und auch in Kartendiensten publiziert. Jedoch haben die GDI aufgrund mangelnder Qualität keine Rechtsrelevanz. Häufig arbeitet eine Vielzahl von Zivilingenieuren an diesen Registern, wobei jede Person nur ein kleines Teilgebiet abdeckt und die Geodaten von einer Zentralstelle zusammengeführt werden. Die vorliegende Arbeit beschreibt Maßnahmen, die sicherstellen, dass die Geodaten die Rechtsverhältnisse korrekt abbilden. Hauptsächlich treten Regelverstöße auf, die durch Korrekturprogramme beseitigt werden können. Planungsautonome Teile entziehen sich der Überwachung durch Regeln oder andere Datenbanken, sodass organisatorische Maßnahmen und Änderungen im rechtlichen Rahmen zu korrekten Daten führen. Entsprechende Wege werden aufgezeigt. Eine besondere Herausforderung bei diesem Vorhaben stellt die technische Umsetzung dar, weshalb die Qualitätsmodelle der Geoinformatik mit den Qualitätsmodellen des Softwareengineering verflochten wurden.

## Abstract

A large number of civil engineers or departments work on geospatial data infrastructures with legal contexts, which are published in web services. Each planner covers only a small sub-area and the geodata are brought together by a central office. However, these digital maps are of no legal relevance due to a lack of quality. This paper describes action plans that ensure that the geodata correctly represent the legal situation. The overwhelming majority of mistakes can be eliminated by software. Autonomous definitions of planners are beyond the control of rules or other databases, thus changes in the organization and the legal framework are necessary to guarantee correct data. A particular challenge in this project is the technical implementation, which is why the quality models of geoinformatics were interwoven with the quality models of software engineering.

# Danksagung

Danken möchte ich meinen Töchtern Paula, Theresia, Katharina und Anna, die immer aufmunternde Worte für mich hatten und mit ihrem wissenschaftlichen Fleiß in ihrem Studium ein leuchtendes Beispiel für mich sind und waren. Anna hat darüber hinaus viel Zeit in stilistische Fragen meinerseits investiert.

Für den fachlichen Input danke ich besonders ao.Univ.-Prof.i.R. Dr.phil. tit.Univ.-Prof. Norbert Bartelme, der diese Arbeit von universitärer Seite betreut hat. Es war mir eine große Freude und Ehre von ihm lernen zu können. Für die vielen guten Vorschläge bezüglich der Vorgehensweise bei wissenschaftlichen Arbeiten danke ich ao.Univ.-Prof. Mag. Dr.iur. Nora Melzer-Azodanloo.

Meinen Kollegen Dipl.-Ing<sup>in</sup> Manuela Weissenbeck, Mag<sup>a</sup> Schunter-Angerer, Dipl.-Ing. Michael Redik und Mag. Christian Freiberger möchte ich für viele sachliche Hinweise im Bereich Raumordnung und in rechtlichen Fragen danken.

Danke

# Inhaltsverzeichnis

Inhaltsverzeichnis .....	III
Abkürzungsverzeichnis .....	VII
Tabellenverzeichnis .....	IX
Abbildungsverzeichnis .....	X
Begrifflichkeit .....	XI
1. Einleitung.....	12
2. Analyse von Abgaben .....	14
2.1. Allgemeines .....	14
2.2. Zusammenfassende Darstellung der Ergebnisse .....	14
2.3. Lehren für zukünftige Schnittstellen – Kritik an der derzeitigen Schnittstelle..	18
3. Die Rechtsrelevanz des digitalen Flächenwidmungsplans .....	24
3.1. Das derzeitige Verfahren .....	24
3.2. Probleme mit den Geodatenätzen.....	26
3.3. Folgen der Probleme .....	28
3.4. Rechtliche Grundlagen .....	28
3.5. Beispiele anderer Länder.....	30
3.6. Lösungsansatz FWP .....	31
3.7. Erweiterter Lösungsansatz des Teilbereiches Ersichtlichmachungen.....	34
3.8. Zu schaffende Voraussetzungen .....	40
3.8.1. Vereinfachung der Planzeichen.....	40
3.8.2. Anforderungen an die EDV .....	42
4. Software-Engineering .....	44
4.1. Vorgehensmodelle .....	46
4.1.1. Code and Fix .....	46
4.1.2. Wasserfallmodell.....	48
4.1.3. Nichtlineare Vorgehensmodelle .....	50
4.1.4. Verwendetes Vorgehensmodell .....	51
4.1.5. Vorgehensmodell versus Prozessmodell.....	53
4.2. Prozessmodelle .....	54
4.2.1. Phasenmodell .....	54
4.2.2. Unified-Software-Development-Prozess.....	56
4.2.3. Agile Prozesse .....	58
4.2.3.1. Extreme Programming (XP) .....	61
4.3. Bewertung und angewandter Prozess .....	65
4.4. Metrik.....	68
4.4.1. Anzahl an Linien – Lines of Code.....	69

---

4.4.2.	MacCabe-Metrik – Cyclomatic complexity .....	69
4.4.3.	Halstead-Metrik.....	71
4.4.4.	Maintainability-Index.....	72
4.4.5.	Objektorientierte Metriken.....	75
4.4.6.	Maßzahlen der Zuverlässigkeit.....	76
4.5.	Qualität von Software.....	80
4.5.1.	Korrektheit.....	81
4.5.1.1.	Vermeidung von Komplexität .....	81
4.5.2.	Robustheit.....	88
4.5.3.	Zuverlässigkeit .....	92
4.5.4.	Effizienz.....	94
4.5.4.1.	Allgemeines .....	94
4.5.4.2.	Strategie .....	95
4.5.4.3.	Organisatorische Maßnahmen.....	96
4.5.4.5.	Schlüsselstellen in der Software .....	97
4.5.4.6.	Nutzung des Arbeitsspeichers .....	99
4.5.4.7.	Spatial Reference and Geoprocessing.....	100
4.5.4.8.	Vermeidung unnötig großer Datensätze.....	101
4.5.4.9.	Korrekte Geometrie .....	102
4.5.4.10.	Web-Service.....	107
4.5.5.	Sicherheit.....	120
4.5.5.1.	Problembeschreibung und Begriffsbestimmung .....	120
4.5.5.2.	Symmetrische und asymmetrische Kryptografie .....	121
4.5.5.3.	Sicherheit und Recht .....	122
4.5.6.	Verfügbarkeit.....	127
4.5.7.	Wartbarkeit .....	131
4.5.8.	Flexibilität.....	135
4.5.9.	Gebrauchstauglichkeit.....	140
4.5.10.	Softwaretest .....	145
4.5.10.1.	Allgemeines .....	145
4.5.10.2.	Äquivalenzklassen von Tests .....	146
4.5.10.3.	Methodik.....	149
4.5.10.4.	Vorgehensweise .....	152
5.	Qualität der abzugebenden Geodaten .....	154
5.1.	Vollständigkeit .....	154
5.2.	Logische Konsistenz.....	157
5.3.	Positionsgenauigkeit.....	161
5.4.	Zeitgenauigkeit .....	163
5.5.	Thematische Genauigkeit.....	164
5.6.	Zweck.....	165
5.7.	Verwendung.....	166
5.8.	Herkunft.....	167

---

6.	Qualitätsbegleitende Maßnahmen und Gegebenheiten.....	169
6.1.	Passiv aggressives Verhalten, Kritikfähigkeit.....	169
6.2.	Selbstverpflichtung.....	170
6.3.	Fertig.....	172
7.	Die entwickelte Software .....	174
7.1.	Allgemeines .....	174
7.2.	Use-Case-Diagramm .....	175
7.3.	Erweiterungen in der Software zur Steigerung der Rechtsrelevanz .....	178
7.4.	Wesentliche Komponenten.....	179
7.4.1.	Hochladen.....	179
7.4.2.	Prüfen, Reparieren Verwalten .....	180
7.4.2.1.	Allgemeines .....	180
7.4.2.2.	Aufruf, Environment-Variablen .....	181
7.4.2.3.	Objektorientiert.....	182
7.4.2.4.	Prüfungen gegenüber der Schnittstellenverordnung .....	184
7.4.2.5.	Vollständigkeit, Lagerichtigkeit .....	188
7.4.3.	Fehlerkorrektur.....	189
7.4.3.1.	Geometriereparatur, Elimination von Leerzeichen .....	190
7.4.3.2.	Sliver-Polygone .....	190
7.4.3.3.	Koordinatensystem.....	193
7.4.3.4.	Überstände beseitigen.....	194
7.4.3.5.	Multipart to singlepart.....	194
7.4.3.6.	Umhüllende Erzeugen.....	195
7.4.3.7.	Semantische Korrekturen .....	195
7.4.4.	Semantische Transformationen.....	199
7.4.5.	Mailing .....	203
7.4.6.	Paralleles Aufbereiten der Geodaten .....	204
7.4.7.	Warnungen.....	205
7.4.8.	Geodaten verwalten.....	205
7.4.9.	Implementierung der Gemeindestrukturreform (GSR).....	207
7.4.10.	Änderungen von Teilbereichen des Plans.....	210
7.5.	Abgrenzung zu anderen Softwareprodukten .....	211
7.6.	Abgrenzung zur künstlichen Intelligenz (KI).....	213
8.	Fazit .....	216
	Anhang - Fehlerbericht.....	220
1.	Vorgehensweise .....	220
2.	Beobachtungsperiode und Endergebnisse Flächenwidmungspläne .....	220
3.	Örtlichen Entwicklungspläne.....	221
4.	Beschreibung der Analysen und Ergebnisse.....	222
4.1.	Fehler in den abgeleiteten Gesamtdaten.....	222
4.2.	Fehler der Einzelpläne .....	224

---

4.3.	Fehler innerhalb eines Planes .....	228
4.3.1.	Abgleich mit der Baulandbilanz .....	228
4.3.2.	Liegen prüfbare Elemente vor?.....	229
4.3.3.	Unvollständiger Fileaufbau.....	230
4.3.4.	Anzahl der Elemente eines Geodatensatzes ist nicht bestimmbar .....	230
4.3.5.	Groß- und Kleinschreibung in der Ebenenbezeichnung .....	231
4.3.6.	Fehlerhafter Geometrietyt .....	231
4.3.7.	Leere Ebenen .....	232
4.3.8.	Objekte außerhalb der Gemeinde.....	233
4.3.9.	Nicht definierte Ebenen.....	235
4.3.10.	Keine zu prüfenden Daten vorhanden .....	238
4.3.11.	Nicht erlaubter Attributtyp .....	238
4.3.12.	Nicht erlaubte Attributbezeichnungen .....	240
4.3.13.	Fehlende Spalten .....	242
4.3.14.	Falsche Attributwerte .....	243
	Literaturverzeichnis .....	249

## Abkürzungsverzeichnis

ArcSDE	Spatial Database Engine – Aufsatz auf eine relationale Datenbank zum Speichern von Geodaten, entwickelt von der Firma ESRI
AROB	Ausschuss für Raumordnung und Baurecht des Landtages
BEV	Bundesamt für Eich- und Vermessungswesen
BMNM31	Koordinatensystem Bundesmeldenetz M31
BMNM34	Koordinatensystem Bundesmeldenetz M34
BRMS	Business Rule Management Systems
CAD	Computer-Aided Design
DNS	Domain-Name-System
DKM	Digitale Katastralmappe
ESRI	Environment System Research Institute – Softwarehersteller von Geoinformationssystemen
ETRS 89	Europäisches Terrestrisches Referenzsystem 1989
ETRS LAEA	Koordinatensystem ETRS89 Lambert Equal Area
fGDB	File Geodatabase von ESRI
FME	Feature Manipulation Engine
FWP	Flächenwidmungsplan
GDI	Geodateninfrastruktur
GI	Geographic Information (deutsch oft Geoinformation)
GKM31	Koordinatensystem Gauß Krüger M31
GKM34	Koordinatensystem Gauß Krüger M34
GSR	Gemeindestrukturreform
INSPIRE	Infrastructure for Spatial Information in the European Community
KI	Künstliche Intelligenz
LGBI	Landesgesetzblatt
LOC	Lines of Code
MTBF	mean time between failure
OEP	Örtlicher Entwicklungsplan
ÖREB	Öffentlich-rechtliche Eigentumsbeschränkungen – Schweizer Register
RIS	Rechtsinformationssystem
ROKAT	Raumordnungskataster
STERZ	Steirisches Rechteverwaltungs- und Zutrittssystem

---

Stmk LREG	Steiermärkische Landesregierung
TDD	Test-Driven-Development
TROG	Tiroler Raumordnungsgesetz
UTM33N	Koordinatensystem Universal Transverse Mercator Streifen 33N

## Tabellenverzeichnis

Tabelle 1: Fehler bei der Datenabgabe.....	15
Tabelle 2: Mehrdeutige Kodierung .....	20
Tabelle 3: zyklomatische Komplexität und Fehlerrisiko [24] .....	83
Tabelle 4: Beschriftungsbeispiel Widmung .....	114
Tabelle 5: Risikoanalyse .....	130
Tabelle 6: Wertung des Maintainability-Index [27].....	135
Tabelle 7: Wartungsprogramme .....	139
Tabelle 8: Beispiel TDD .....	150
Tabelle 9: Fehlerkorrektur – Realisierung von verzögertem Lernen und Vergessen.....	198
Tabelle 10: Aufbau der Gültigkeit.....	206
Tabelle 11: Anzahl an positiven sowie negativen Abgaben und Systemabstürzen .....	225
Tabelle 12: Fehlerhafte Layerbezeichnungen .....	236
Tabelle 13: Fehlerhafte Spaltennamen .....	240
Tabelle 14: Fehlende Spalten.....	242
Tabelle 15: Fehlerklassen der Textattribute .....	245

# Abbildungsverzeichnis

Abbildung 1: Gemeinden mit digitalen FWPs.....	220
Abbildung 2: Histogramm der Uploads pro für Gemeinde.....	226
Abbildung 3: Histogramm des Auftretens von diversen Attributfehlern .....	246
Abbildung 4: Tortendiagramm der Korrigierbarkeit von Attributfehlern .....	247
Abbildung 5: Schema des Verfahrens .....	33
Abbildung 6: Unterschiedlicher Zeitbedarf von Entwicklern .....	45
Abbildung 7: Wasserfallmodell.....	48
Abbildung 8: Iteratives Vorgehensmodell.....	50
Abbildung 9: Unified-Software-Development-Prozess .....	57
Abbildung 10: Zyklomatische Komplexität am Beispiel .....	70
Abbildung 11: Wartbarkeitsindex und Kommentarzeilen .....	73
Abbildung 12: Wartbarkeit und Programmlänge.....	74
Abbildung 13: Entstehung von Sliver-Polygonen .....	101
Abbildung 14: Das Geometrieelement für Linien .....	103
Abbildung 15: Das Geometrieelement für Flächen.....	104
Abbildung 16: Verschiedene Typen an Linien.....	113
Abbildung 17: Risikodiagramm .....	130
Abbildung 18: Schnittstelle 2003–2007.....	143
Abbildung 19: Räumliche Äquivalenzklasse mit keinem oder einem Objekt.....	147
Abbildung 20: Räumliche Äquivalenzklasse mit einem oder mehreren Objekten .....	148
Abbildung 21: Use-Case-Diagramm der Datenübernahme .....	175
Abbildung 22: Softwareerweiterungen für rechtsrelevante Gleichstücke .....	178

## Begrifflichkeit

Agile Prozesse	Softwareentwicklung auf Basis von Werten, wie Kundeneinbindung, Positive Aufnahme von Änderungen, kurze Produktzyklen, persönliche Gespräche, Einfachheit und funktionierende Software als Primärziel
Architektur (Softwarearchitektur)	Grundsätzliche Organisation eines Systems in Komponenten, Beziehungen, Schnittstellen sowie prinzipielle Richtlinien und Software-entscheidungen
Business Rule Management Systems	Regelbaserendes System, dass im Sinn von „wenn A, dann B“ agiert.
Pair Programming	Zwei Entwickler arbeiten gemeinsam an einem Code, wobei die Tastatur im Takt von wenigen Minuten gewechselt wird.
Pickle-Datei	Datei, welche Objekte persistent speichert
Refactoring	Software wird überarbeitet, mit dem Ziel die Struktur des Programmes zu verbessern. Die Funktionalität wird dabei nicht geändert.
Test Driven Development (TDD)	Bei dieser Form der Entwicklung von Software wird zuerst die Testroutine und dann der Produktivcode geschrieben.

# 1. Einleitung

Geodateninfrastrukturen (GDI) werden insbesondere von öffentlichen Stellen aufgebaut, indem komplexe Planwerke von einer Vielzahl von Ingenieuren zusammengetragen werden. Das wohl größte Vorhaben dieser Art ist „Infrastructure for Spatial Information in the European Community (INSPIRE)“, welches am 15. Mai 2007 vom Europäischen Parlament und vom Rat der Europäischen Union als Richtlinie 2007/2/EG in Kraft gesetzt wurde. Aber auch kleinere Organisationen, wie das Amt der Steiermärkischen Landesregierung, baut GDIs auf. Ein Beispiel ist der Flächenwidmungsplan, welcher von Gemeinden oder von ihnen beauftragten Raumplanern erstellt wird. Diese Projekte haben gemeinsam, dass viele unterschiedliche Planersteller an einem gemeinsamen meist überregionalen Planwerk arbeiten und jeder Planer ein bestimmtes Gebiet abdeckt.

Diese meist öffentlich-rechtlichen Festlegungen bestimmen den Preis einer Parzelle maßgeblich. Kreuzer, Fischer & Partner ermittelten für die Jahre 2007 bis 2011 eine Wertsteigerung von € 48 pro Quadratmeter durch den Verwaltungsakt der Umwandlung von Grünland in Bauland als österreichischen Durchschnittswert [109]. Neben dem Flächenwidmungsplan gibt es noch viele behördliche Festlegungen, welche den Wert von Grundstücken mitbestimmen, wie Naturschutzgebiete, Tieffluggzonen oder Brunnenschutzgebiete. Sie liegen in Datensammlungen vor, wie sie im vorherigen Absatz beschrieben wurden. Internetdienste, welche diese öffentlichen rechtlichen Festlegungen zeigen, schließen die Haftung in der Regel aus. Die Internetseiten sind mit Sätzen wie: „Keine Haftung für Verfügbarkeit, Vollständigkeit und Richtigkeit der Darstellung [110]“ versehen, was Qualitätsmängel der Geodaten vermuten lässt. Rechtsrelevante Information muss auf anderen, aufwendigen Wegen erhoben werden. Es entsteht ein volkswirtschaftlicher Schaden und die Rechtssicherheit leidet.

Das steht in krassem Gegensatz zu privatrechtlichen Einschränkungen auf Grund und Boden, die im Grundbuch dokumentiert sind und digital abgefragt werden können, wobei der Raumbezug mit der flächendeckend vorliegenden Katastralmappe hergestellt wird. Vom Bundesministerium für Justiz wurden einige Stellen autorisiert, welche den Bürgern diese rechtsrelevante Information gegen Entgelt bereitstellen [108]. Die Rechte von Privaten an Grundstücken, wie Wegrechte oder Hypotheken, sind so im Gegensatz zu öffentlich-rechtlichen Einschränkungen leicht zugänglich.

Kann man diese heterogenen, oft fehlerhaften und von einer Vielzahl von Produzenten erstellten öffentlich-rechtlichen Geodaten in einer Art und Weise prüfen und vielleicht sogar korrigieren, sodass ein einheitliches Gesamtbild entsteht? Wie kann die Rechtsmaterie ausreichend berücksichtigt werden, um die Zielsetzung der Rechtsrelevanz von Geodaten in Zukunft auch auf eine Vielzahl von Registern auszudehnen, welche von heterogenen Stellen erstellt werden, und lassen sich die theoretischen Modelle zur Qualitätssicherung von Geodaten und Verarbeitungssoftware in praktischer Form umsetzen, sodass sie zu ausreichender Qualität führen?

## 2. Analyse von Abgaben

### 2.1. Allgemeines

Sollen  $n$  Systeme bidirektional miteinander kommunizieren, benötigt man  $n \cdot (n-1)/2$  unterschiedliche Implementierungen. Arbeitet man aber mit einer standardisierten Schnittstelle, reicht diese für den Datenaustausch aus [39, S. 104]. Für den Aufbau einer GDI sind Normen und Schnittstellen essentiell, da sie die Basis der Kommunikation bilden, wie etwa Straßen für den Verkehr. Eines der bekanntesten Projekte zur Schaffung einer gemeinsamen GDI ist das Vorhaben „Infrastructure for Spatial Information in the European Community (INSPIRE)“, welches am 15. Mai 2007 vom Europäischen Parlament und vom Rat der Europäischen Union als Richtlinie 2007/2/EG in Kraft gesetzt wurde. Neben Metadaten und Diensten sind auch Geodaten in definierte Strukturen überzuführen. Diese Datenmodelle können komplex sein (z. B. INSPIRE Data Specification on Cadastral Parcels – Technical Guidelines 3.1 [105]) und führen dementsprechend zu hohen Aufwendungen und Fehlern.

Die Komplexität der Datenmodelle erschwert nicht nur die Erstellung der Informationen, sondern wirkt sich negativ bei der Verwendung der Daten aus. Es stellt sich die Frage, ob die Modelle tatsächlich vereinheitlicht werden müssen oder nicht ein kleiner Auszug der Daten für eine europaweite Sicht reichen würde. Was bereitet Datenproduzenten Schwierigkeiten und könnten die Normen nicht entsprechend der praktisch auftretenden Probleme simplifiziert werden?

### 2.2. Zusammenfassende Darstellung der Ergebnisse

Die detaillierten Ergebnisse der Analyse können im Anhang nachgelesen werden. In diesem Kapitel ist eine kompakte Übersicht über die Probleme beim Erstellen von Geodaten zu finden.

Vom Amt der Steiermärkischen Landesregierung wird die GDI – Flächenwidmungsplan aufgebaut, indem die Pläne der einzelnen Gemeinden gesammelt werden. Die Datenlieferungen wurden 2,6 Jahre lang beobachtet. In diesem Zeitraum kam es zu 331 Planabgaben.

Es liegt eine valide Datenbasis vor, mit der Aussagen über Problemfelder getroffen werden können.

Tabelle 1: Fehler bei der Datenabgabe

Anhang	Problemfeld	Anzahl/Korrigierbar- in Prozent
4.3.2	Prüfbare Elemente	0/0
4.3.3	Unvollständiger Fileaufbau	1/0 - 0 %
4.3.4	Kaputte innere Filestruktur	1/0 - 0 %
4.3.5	Groß- u. Kleinschreibung Ebenenname	60/60 - 100 %
4.3.6	Fehlerhafter Geometriotyp	21/14 - 67 %
4.3.7	Leere Geodatenätze	308/280 <sup>1</sup> - 91 %
4.3.8	Außerhalb der Gemeinde	132/132 - 100 %
4.3.9	Nicht definierte Ebenen	28/28 - 100 %
4.3.10	Prüfung zum Zweck der Programmstabilität analog zu 4.3.2	0/0
4.3.11	Unzulässiger Attributtyp	130/130 - 100 %
4.3.12	Attributbezeichnung	601/601 - 100 %
4.3.13	Fehlende Spalte	10/0 - 0 %
4.3.14	Falsche Attributeinträge	10511/7428 - 71 %
	Summe	11803/8673 – 73 %

Viele Verstöße können durch automatische Bereinigungen korrigiert werden, wie aus den Prozentzahlen ersichtlich ist. Addiert man die Fehler und Korrekturmöglichkeiten, erkennt man, dass 73% der Regelverstöße automatisch zu lösen sind. Im Anhang wird detailliert darauf eingegangen. Es erscheint sinnvoll, die automatischen Korrekturen zentral umzusetzen, da den Planern viel Arbeitszeit durch lästige Reparaturarbeiten erspart wird. Viele Schreibprogramme unterstützen die Anwender mit Rechtschreibkorrekturen. In Analogie

<sup>1</sup> Für inhaltliche Fehler wurde eine Größenordnung von 8% gefunden (siehe Anhang 4.2). Deshalb wird davon ausgegangen, dass 8 % der leeren Ebenen Einträge aufweisen müssten und nicht einfach gelöscht werden können. Welche Ebenen das sind, kann durch Abgleich mit anderen Datenbanken herausgefunden werden.

dazu ist beim Aufbau einer GDI ein System sinnvoll, welches Vorschläge zur Datenkorrektur entsprechend einer vorgegebenen Schnittstelle liefert.

Für die Umsetzung der Korrekturen wurden zwei unterschiedliche Ansätze erprobt. In Anlehnung an Business Rule Management Systems (BRMS) wurden konkrete Regeln zur Ingangsetzung von Lösungsalgorithmen implementiert. Jeder Arbeitsschritt konnte über eine Erklärungskomponente begründet und dokumentiert werden. Darüber hinaus wurde auch ein selbstlernendes System implementiert, welches die Abgaben der Planer beobachtet und deren Korrekturen aufzeichnet. Dieses Wissen wird für gleich gelagerte Fälle herangezogen. Wurde ein Regelverstoß korrigiert, konnte für die Art der Korrektur kein sachlicher Grund mehr angegeben werden. Es kann lediglich gesagt werden, auf welche Planer diese Vorgangsweise zurückzuführen ist. Dieses Manko ist schwerwiegend, da im behördlichen Umfeld großer Wert auf begründetes Vorgehen gelegt wird. Ein Beispiel für diesen behördlichen Grundsatz ist das zentrale Werkzeug des Bescheids, für den eine Begründung rechtlich verpflichtend vorgesehen ist. Expertensysteme besitzen eine Erklärungskomponente (siehe [107]) und sind damit gegenüber selbstlernenden Systemen überlegen. Zu dem stellt sich die Frage, ob die Art der Korrektur geistiges Eigentum des Planers ist, welches nun zum Vorteil eines wirtschaftlichen Konkurrenten eingesetzt wird. Dieselbe Fragestellung tritt beim Einsatz von Robotern auf und ist in beiden Fällen unregelt. Ein Rechtsausschuss der Europäischen Union hat einen Bericht zur Robotik am 12.1.2017 veröffentlicht und fordert die Union zu einer entsprechenden Regelung auf [106].

Strukturelle Fehler, wie Verstöße gegen den Attributtyp, Geometriotyp, fehlerhafte Ebenennamen etc., sind gering im Vergleich zu Verstößen gegen die Attributdomains. Das war überraschend, da angenommen wurde, dass die Probleme bei der komplexen Datenstruktur liegen würden. Offensichtlich macht die einfache Massenaarbeit der Attribuierung mehr Schwierigkeiten als der Bau der Datenstruktur. Möglicherweise kommt es zu so wenigen Strukturmängeln, da leere Musterdateien vom Land bereitgestellt werden und Topologien von der verwendeten Software der Planer unterstützt werden.

20% der Attributfehler waren eine Folge geänderter Vorgaben. Gewohnte Abkürzungen zu ändern, ist problematisch, auch wenn dadurch eine Vereinheitlichung erfolgt. Bei der Umsetzung neuer Vorgangsweisen ist anfangs mit größeren Problemen zu rechnen.

Überflüssige Spalten stellen das größte strukturelle Problem dar. Meist handelt es sich um Reste, welche bei der Überführung aus anderen Datenformaten generiert wurden. Sie enthalten keine weiter benötigte Information und können gelöscht werden.

Geodaten werden üblicherweise anhand von Regeln, welche in einer Schnittstellenbeschreibung niedergeschrieben sind, geprüft. Inhaltliche Fehler werden damit nicht registriert. Deshalb werden die abgegebenen FWPs von den Planern ein weiteres Mal optisch kontrolliert, wofür ein Web-Service bereitsteht, das die von den Planern abgegebene Arbeit zeigt. Bei 8 % der Gemeinden erfolgte nach der ersten positiven Abgabe eine inhaltlich korrigierte Nachlieferung. Man kann davon ausgehen, dass nicht alle inhaltlichen Fehler entdeckt wurden. Die Prüfung gegenüber einer Schnittstelle ist also bei weitem nicht ausreichend um richtige Daten zu erhalten.

Im Kapitel 7.4.2.5 wird ein technisches Verfahren vorgestellt, mit dem durch unscharfen Abgleich mit anderen Datenbanken eine Verbesserung der inhaltlichen Datenqualität erzielt werden kann. Die technischen Möglichkeiten sind bescheiden, da Datenbanken zum Abgleich in kaum ausreichender Zahl bereitstehen. Im Kapitel 3.6 wird am Beispiel des FWPs gezeigt, wie dieses Problem organisatorisch und rechtlich gelöst werden kann.

Neben der Identifikation von Problemen bei der Planerstellung lieferte die Analyse die wertvolle Erkenntnis, dass bei 1,5% aller Abgaben Probleme mit der Übernahmeroutine auftraten. Die Zuverlässigkeit (siehe Kapitel 4.5.3) ist nicht ausreichend, um die digitalen Daten ins Zentrum von Rechtsbeurteilungen und Rechtsgeschäften zu rücken. Eine vereinfachte Architektur ist notwendig, um die Komplexität zu reduzieren. Um entsprechende Qualität zu erreichen, wurden Vorgehensmodelle für GI-Projekte mittlerer Größe validiert und ein adaptierter Weg wird eingeschlagen. Metriken überwachen den Entwicklungsprozess und ein Qualitätsmodell für Geodaten findet Anwendung.

### 2.3. Lehren für zukünftige Schnittstellen – Kritik an der derzeitigen Schnittstelle

Verschiedene GI-Produkte erlauben Layernamen, Spaltenbezeichnungen oder Geometrietypen, welche von anderen Produkten nicht gelesen oder verarbeitet werden können. So gibt es GI-Produkte, die Shape-Dateien erzeugen, deren Spaltennamen Punkte und Sonderzeichen enthalten können. Andere Produkte können nur Spalten lesen, die aus den Zeichen a–z, A–Z, 0–9 sowie \_ bestehen, nicht mit einer Ziffer beginnen und nicht länger als 12 Zeichen sind. Auf den kleinsten gemeinsamen Nenner sollte Rücksicht genommen werden, möchte man keine Anbieter ausschließen.

Grundsätzlich sollte man bei allen internen Festlegungen, wie Ebenenbezeichnungen, Spaltenbezeichnungen etc., die nur von Entwicklern gesehen werden, mit den ASCII-Zeichen auskommen. Die Struktur bleibt trotzdem gut lesbar. Diese Zeichen werden von allen Produkten verstanden.

Es ist sinnvoll, auch hier Strukturierungen einzuführen. In der vorliegenden Schnittstelle sind die Namen von Ebenen immer in Kleinbuchstaben geschrieben. Die Endung des Spaltennamens verweist auf den Geometrietyp, sodass dieser schon von außen sichtbar ist. Spaltennamen werden in Großbuchstaben definiert. Ordnung wird als angenehm empfunden und erleichtert das Arbeiten.

Änderungen in einer Schnittstelle bereiten den Nutzern große Probleme, wie der Übergang von Verk auf VERK zeigt. Das ist besonders dann der Fall, wenn die Änderungen nur gering ausfallen. Grobe Neuerungen fallen schnell auf und werden in der Regel nicht falsch gemacht. Ein Beispiel sind Sanierungsgebiete, deren Festlegung von der Widmungsebene nutz\_f in die Ebene für Beschränkungen beschr\_f verlegt wurde. Darüber hinaus wurde die Kodierung geändert.

Sonderzeichen in der Attribuierung sind problematisch, da vorausgesetzt wird, dass der Datenempfänger weiß, mit welcher Codierung die Datenerstellung erfolgte. Entsprechende Erweiterungen im Shape entschärfen das Problem, da der ESRI-Shape nun den verwendeten Zeichensatz als Metadateninformation mitliefert. Die entsprechende Datei ist jedoch kein obligatorischer Teil des Formates und es gibt GI-Produkte, welche diese Datei nicht schreiben.

In der Attribuierung kommen Klammern vor. Häufig werden schließende Klammern vergessen. Auch geforderte Einheitszeichen werden weggelassen. Es ist beim Schnittstellentwurf darauf zu achten, dass diese Fehler zweifelsfrei behoben werden können und zu keinen Doppeldeutigkeiten führen.

Viele Fehler traten bei der Groß- und Kleinschreibung im Bereich der Attribuierung auf. Innerhalb einer Norm sollten unterschiedliche Dinge auch über die Groß- und Kleinschreibung hinaus unterschiedlich benannt werden. Verstöße gegen die Groß- und Kleinschreibung können dann automatisch bereinigt werden. Natürlich macht es Sinn, dem Nutzer der Daten, mittels Groß- und Kleinschreibung in der Attribuierung, zusätzliche kategorisierende Information zu geben.

In der Schnittstelle des FWP's werden alle Baulandausweisungen, Verkehrsflächen und landwirtschaftlichen Nutzflächen in Großbuchstaben kodiert, Sondernutzungen in Freiland hingegen in Kleinbuchstaben. Groß- und Kleinschreibung strukturiert in einem rechtlichen Sinn und erhöht die Lesbarkeit für den Kartenleser wesentlich. Groß- und Kleinschreibung ist eine der großen Verbesserungen der Schnittstelle 2007 gegenüber der Schnittstelle 2003.

Es ist sicher von Vorteil, wenn die verwendeten Kürzel für Widmungen und Zusatzwidmungen auch über die einzelnen Ebenen hinweg möglichst eindeutig sind. Wenn sie sinnhaft das Gleiche darstellen, sollten sie gleich kodiert werden. Es ist sinnvoll, Schnellstraßen in der Widmungsebene (nutz\_f) mit S zu kodieren und dieses Kürzel auch in der Ebene für Projekte von geplanten Schnellstraßen zu verwenden (nutzproj\_f). Damit wird Verwechslungen vorgebeugt. Das ist in der vorliegenden Schnittstelle sehr konsequent berücksichtigt. Natürlich findet man unter den mehr als 700 Codes auch Ausnahmen, an der die Problematik veranschaulicht werden soll. Der Code „DK“ in den Ebene ersl\_versorg\_f (Ersichtlichmachung flächenhafter Versorgungseinrichtungen) und ersl\_versorg\_p (Ersichtlichmachung punkthafter Versorgungseinrichtungen) wird für Dampfkraftwerke verwendet. Das ist sinnvoll, da es sich um eine Objektqualität handelt. Die Ebene ersl\_versorg\_l (Ersichtlichmachung linienhafter Versorgungseinrichtung) kodiert mit „DK“ jedoch Datenkabel, was zu Verwechslungen führen kann. Hier ist die Problematik besonders groß, da es sich um artverwandte Ebenen handelt. Sie enthalten Versorgungseinrichtungen in verschiedener Geometrie.

Das wurde zum Anlass genommen, ein Programm zu schreiben, welches die Schnittstelle auf diese Umstände untersucht. Nachstehend werden über die Ebenen hinweg gleichlautende Kodierungen aufgelistet. Es ist nun abzuwägen, ob die Ebene als Unterscheidungsmerkmal ausreicht oder ein neues Planzeichen eingeführt werden sollte. Dabei handelt es sich um eine subjektive Sicht, wobei jede Wertung auch gut begründet anders gefällt werden kann.

Tabelle 2: Mehrdeutige Kodierung

Ebene	Spalte	Codierung	Bedeutung	Bewertung
Ers1_ver-sorg_f Ers1_ver-sorg_p	ZSW	<b>DK</b>	Dampfkraftwerk	ändern
Ers1_ver-sorg_l	ZSW	<b>DK</b>	Datenkabel	beibehalten
Nutz_f Nutz-proj_f	Widmung	<b>S</b>	Schnellstraße	ok
Beschr_f	Widmung	<b>S</b>	Sanierungsgebiet	ok
Ers1_ver-kehr_f	ZSW	<b>S</b>	Schnellstraße	ok
Ers1_ver-sorg_p	ZSW	<b>S</b>	Sendeanlage	ok
Nutz_f, Nutz-proj_f,...	Widmung	<b>A</b>	Autobahn	ok
Beschr_f	Widmung	<b>A</b>	Aufschließungsgebiet	ok
Ers1_ver-sorg_f Ers1_ver-sorg_p	ZSW	<b>F</b>	Fernheizwerk	ändern
Ers1_ver-sorg_l	Widmung	<b>F</b>	Hochspannungsfreileitung	beibehalten
Ers1_flur_f	Widmung	<b>F</b>	Flurbereinigungsgebiete	ok
Ers1_oeff_f	ZSW	<b>Bg</b>	Bezirksgericht	ok
Ers1_sich_l	Widmung	<b>BG</b>	Bergbaugebiete	ok

Ersl_grenze_l	ZSW	<b>BG</b>	Bezirksgrenze	ok
Ersl_wasg_l	Widmung	<b>H</b>	Heilquellen- und Heilmoorschutzgebiete	ändern
Ersl_versorg_p Ersl_versorg_l	ZSW	<b>H</b>	Hochbehälter	beibehalten
Ersl_versorg_f Ersl_versorg_p	ZSW	<b>M</b>	Molchstation	ändern
Ersl_mili_l	Widmung	<b>M</b>	militärisches Sperrgebiet	beibehalten
Ersl_versorg_f Ersl_versorg_p	ZSW	<b>P</b>	Pumpstation	ok
Nutz_f	Widmung	<b>P</b>	Parkplatz	ok
Ersl_versorg_f Ersl_versorg_p	ZSW	<b>T</b>	Transformator	ok
Ersl_mila_f	ZSW	<b>T</b>	Truppenübungsplatz	ok
Nutz_f	Widmung	<b>U</b>	Verkehrsfläche mit Überlagerung	ok
Ersl_versorg_f Ersl_versorg_p	ZSW	<b>U</b>	Umspannwerk	ok
Ersl_versorg_f Ersl_versorg_p	ZSW	<b>WS</b>	Wetterstation	ok
Ersl_wasg_l	WIDMUNG	<b>WS</b>	artesisches Wasserschongebiet	ok
Beschr_f	WIDMUNG	<b>E</b>	erhaltenswerte Gebäudegruppe	ok
Ersl_versorg_l	WIDMUNG	<b>E</b>	Hochspannungserdkabel	ok
Ersl_Mila_f	ZSW	<b>G</b>	Garnisonsübungsplatz	ok
Ersl_Versorg_l	ZSW	<b>G</b>	Gasleitung allgemein	ok
Ersl_mila_f	ZSW	<b>K</b>	Kaserne	ok

Ersl_imm_1	WID- MUNG	<b>K</b>	Kurbezirk	ok
------------	--------------	----------	-----------	----

Summarisch ist festzustellen, dass die Planzeichenverordnung ein gelungenes Werk darstellt, mit dem bestens gearbeitet werden kann. Nachbesserungsbedarf ist nur in unbedeutenden Bereichen in geringem Ausmaß notwendig. Der Großteil der Kürzel ist historisch gewachsen und stellt im weitesten Sinne ein raumplanerisches Kulturgut dar. Die Vielzahl an Ersichtlichmachungen bewirkt eine schwere Lesbarkeit des FWP und verteuert die Erstellung des Planwerks. Das wird durch die Tatsache unterstrichen, dass 34 Ebenen für Ersichtlichmachungen notwendig sind und lediglich 3 Ebenen für den Planungsautonomen Teil ausreichen.<sup>2</sup>

Alle Ebenen sind sicher für die Planung notwendig, müssen aber für die Datenübermittlung in eine vorgegebene Struktur überführt werden, was Kosten und Fehler verursacht. Automatische Algorithmen sind kaum mehr in der Lage, ein lesbares Bild zu erzeugen. Dementsprechend wird über eine Überarbeitung nachgedacht. Z. B. sind Richtfunkstrecken im FWP abgebildet, was notwendig erscheint, da Gebäudehöhen auf den Funkbedarf Rücksicht nehmen müssen. Andererseits kann seit der Schnittstelle 2007 die Bauhöhe über das Attribut MAXH der Flächennutzung geregelt werden. Die zusätzliche Ebene begründet die Beschränkung. Es geht also um die prinzipielle Frage, wie teuer es sein darf, den Normunterworfenen die Sinnhaftigkeit der Beschränkung zu erklären. Eine Diskussion unter Einbindung der Politik als Bürgervertretung ist notwendig.

Betrachtet man andere behördliche Rechtsmittel, scheint ein Verzicht auf Ersichtlichmachungen schwer machbar. Ein wichtiges Beispiel ist etwa der Bescheid, wo Begründungen zwingend vorgeschrieben sind. Damit wird die Möglichkeit von Einsprüchen erleichtert.

---

<sup>2</sup> Zusätzlich sind 2 Ebenen für die Katasterdarstellungen notwendig und eine für kartografische Zwecke.

Erlangt ein digitaler Plan Rechtsgültigkeit oder verfügt die Behörde über ein digitales Gleichstück<sup>3</sup> des rechtsgültigen Plans, können die Ersichtlichmachungen zu- und weggeschaltet werden und es können beliebige Maßstäbe eingestellt werden, was die Lesbarkeit unproblematisch werden lässt.

---

<sup>3</sup> Gleichstücke sind idente Kopien eines Originals. Der Erstellungsprozess, wie etwa Fotokopie, Mehrfachausdruck oder Blaupause ist belanglos. Kommt es zu Widersprüchen vermeintlicher Gleichstücke, ist häufig durch den Gesetzgeber oder erstinstanzliche Urteile geregelt, welche Ausfertigung als richtig anzusehen ist. Der FWP liegt sowohl im Gemeindeamt als auch im Amt der Steiermärkischen Landesregierung auf. Bei Unterschieden wird der Version im Gemeindeamt der Vorzug gegeben, sofern sich der richtige Sachverhalt nicht aus dem Wortlaut, einer schriftlichen Erläuterung des Planes, ergibt

### 3. Die Rechtsrelevanz des digitalen Flächenwidmungsplans

Möchte man derzeit rechtssichere Information aus dem FWP erhalten, muss man auf das Gemeindeamt gehen. Werden die nachfolgend angeführten rechtlichen Überlegungen konsequent umgesetzt, kommt es zu einer qualitativen Steigerung der Geodaten, welche es ermöglicht alle Rechtsgeschäfte im Zusammenhang mit FWPs auf Basis von Geodaten zu erledigen. Auf ein entsprechendes Web-Service kann jederzeit von überall zugegriffen werden. Es kommt zu einer Demokratisierung, da niemand von der Information durch örtliche oder zeitliche Beschränkungen ausgeschlossen wird. Vergleiche von Grundstücken auf einer soliden rechtlichen Basis werden vereinfacht. Besicherungen von Krediten durch Grundstücke benötigen keinen Gang auf das Gemeindeamt, da qualitätsgesicherte Information aus dem Internet gewonnen werden kann.

#### 3.1. Das derzeitige Verfahren

Der Verfahrensablauf des FWPs ist im steirischen Raumordnungsgesetz 2010 geregelt, welches im LGBl. Nr. 49/2010 kundgemacht wurde und zuletzt durch die Veröffentlichung im LGBl. Nr. 44/2012 geändert wurde. Da es sich um Landesrecht handelt, divergieren die Vorgehensweisen in den Bundesländern im Allgemeinen leicht. Grob unterscheiden sich vom steirischen Weg lediglich die Vorgehensweisen der Bundesländer Wien und Vorarlberg, wo die Planungen von Abteilungen der Länder erfolgen.

Am Beginn des Verfahrens eines FWPs steht die Kundmachung der Verfahrenseröffnung durch den Bürgermeister einer Gemeinde. Dabei werden nicht nur die Gemeindebürger, sondern auch das Amt der Steiermärkischen Landesregierung über das geplante Vorhaben der Erstellung eines neuen FWPs informiert. Die Informierten können nun Planungsinteressen anmelden und der Gemeinderat beschließt daraufhin die Durchführung eines FWPs. Meist, aber nicht zwingend vorgeschrieben, wird für die Planung ein externer Raumplaner bestellt. Es erfolgen ein Vorentwurf und eine Prüfung der Umweltauswirkungen. Der Gemeinderat beschließt den Entwurf, der öffentlich zur allgemeinen Ansicht aufgelegt wird. Auch das Land erhält einen Papierplan, der geprüft wird. Dieser Plan wird als Auflage be-

zeichnet. Bürger und Land können Einwendungen zur Auflage einbringen, die im Anschluss abgearbeitet werden. Der Planer erstellt eine Genehmigungsvorlage des Plans, welche nach vorheriger Prüfung vom Gemeinderat beschlossen wird. Auch dieser Plan geht an die Landesregierung. Der Papierplan wird von der Behörde überprüft und im Raumordnungsbeirat erörtert. Dieser Beirat setzt sich aus Vertretern der Landtagsklubs von Regierungsmitgliedern sowie Vertretern der Kammer für Land- und Forstwirtschaft, der Wirtschaftskammer, der Arbeiterkammer und der steirischen Sektion des Städte- und Gemeindebunds zusammen. Die Landesregierung folgt in der Regel der Empfehlung des Raumordnungsbeirates und beschließt oder verwirft den Plan. Der Bürgermeister macht den von der Landesregierung beschlossenen Plan kund, womit der FWP Rechtskraft erlangt. Die Kundmachung ist zumindest mit einem Aushang des Plans im Gemeindeamt verbunden. In seltenen Fällen wird die Genehmigung verweigert, was zur Folge hat, dass die Planungen in der Gemeinde neu zu erfolgen haben.

Das Land erhält sowohl den Auflageentwurf als auch die Genehmigungsvorlage in Papierform. Die Planzeichenverordnung 2007 sieht vor, dass neben den Papierplänen auch digitale Geodaten in genormter Form des rechtsgültigen Planes abzugeben sind. Das ist die von der Landesregierung beschlossene Genehmigungsvorlage in digitaler Form. Diese Geodaten werden im Verfahren nicht weiter verwendet. Die Abgabe dieser Daten erfolgt in aller Regel nach der Kundmachung.

Die Planung erfolgt gemeindeweise, wobei das ganze Gemeindegebiet eingeteilt wird, was zum sogenannten Stammplan führt. Änderungen am Stammplan sind möglich. Neben den verpflichtenden Papierplänen können Änderungen auch digital abgegeben werden. Die digitale Abgabe von Änderungen ist jedoch nicht verpflichtend und erfolgt nur selten. Die Kundmachung eines neuen Stammplans einer Gemeinde setzt den alten Stammplan und alle Änderungen außer Kraft.

Fehlen digitale Daten von Änderungen, werden die betroffenen Bereiche von einem Mitarbeiter des Amts der Stmk LREG erfasst und mit dem Änderungsdokument verlinkt.

Mit dieser Arbeit wurde unter anderem die Verarbeitung von digitalen Änderungsdaten umgesetzt und die neue Planzeichenverordnung 2016 verpflichtet die Gemeinden zukünftig zur Abgabe von digitalen Änderungsdaten. Werden die Geodaten nicht abgegeben, hat das keine Strafbestimmung zur Folge.

Diese Dissertation war entscheidend für die digitale Vorschreibung von Änderungsdaten und ist damit ein Wegbereiter für aktuelle digitale FWPs.

Die Probleme der Planer mit der alten Schnittstelle wurden quantitativ erhoben und flossen teilweise in die neue Planzeichenverordnung ein, wodurch die Planerstellung einfacher wurde. Darüber hinaus werden Datenfehler von der entwickelten Software korrigiert und die neue Planversion wird dem Planer zur Kontrolle übermittelt. Viel Routinearbeit wird dadurch vom Computer erledigt, was den Erstellungsaufwand der Änderungsdaten, die nun digital zu liefern sind, kompensiert.

Die Planzeichenverordnung 2007, auf deren Grundlage Geodaten übermittelt werden, wurde im LGBl. Nr. 12/2008 kundgemacht und geändert mit LGBl. Nr. 57/2011. Zukünftig werden Pläne nach der Planzeichenverordnung 2016 abgegeben, welche im LGBl. Nr. 117/2016 veröffentlicht wurde.

### 3.2. Probleme mit den Geodatensätzen

Neben den gemeindeweiten Stammpänen werden auch Änderungen von Teilgebieten durchgeführt. Seit dem Jahr 1974 wurden ca. 17.000 Änderungen vorgenommen, was 30 Fälle pro Stammpan bedeutet. Raumplaner laden nur selten Änderungsdaten hoch, die dann vollautomatisch wie eine Briefmarke über den FWP geklebt werden.<sup>4</sup> Eine Verpflichtung, Geodaten hochzuladen, besteht für Änderungsdaten derzeit nicht. Nur für gemeindeweite Stammpäne ist die digitale Datenlieferung verpflichtend. Deshalb gibt es einen zweiten Weg, um Änderungen zu dokumentieren. Von Amts wegen werden diese Gebiete gekennzeichnet und mit den Änderungsschriftstücken verlinkt. Das erfolgt aus personellen Gründen jedoch nicht vollständig.

Auch die gemeindeweiten Stammpäne sind nicht immer auf Letztstand. Es kann vorkommen, dass bereits eine gemeindeweite Planung Rechtskraft erhalten hat, die Geodaten jedoch nicht geliefert wurden. Manchmal wird überhaupt nicht digital geliefert. Darüber hinaus beinhalten Lieferungen Fehler. Das wesentliche Problem der Geodaten besteht in der mangelnden Bedeutung im Rechtsprozess, weshalb auf sie weniger Wert gelegt wird. Der

---

<sup>4</sup> Die Verarbeitung der Änderungen wurde ebenfalls im Rahmen dieser Dissertation entwickelt.

FWP wird vom Raumplaner auf einem hauseigenen System gezeichnet. Nicht immer wird die Schnittstelle des Amtes der Steiermärkischen Landesregierung von der Software unterstützt, weshalb die Daten im Nachhinein konvertiert werden, wobei Fehler passieren. Es wird nicht immer die letztgültige Version verwendet oder es werden Teilaspekte weggelassen (Ersichtlichmachungen). Bei der Überführung von CAD-Daten in Geodaten kommt es zu Problemen mit Flächenschließungen.

Von ähnlichen Erfahrungen berichten auch andere. Noack beschreibt seine Erfahrungen mit Zonierungsplänen im Schweizer Kanton Basel Landschaft so:

„Leider kann auch bei Planungen, welche von ausgewiesenen Planungsbüros gemacht werden, nicht immer davon ausgegangen werden, dass die abgelieferten digitalen Geodaten auch wirklich dem ausgedruckten Plan entsprechen“ [3, S. 11].

Vor allem im Bereich der OEPs werden selbst die Stammpläne nicht immer digital abgegeben, weshalb von Amtes wegen von Zeit zu Zeit kontrolliert wird, ob für jedes Verfahren die digitalen Geodaten vorliegen. Ist das nicht der Fall, werden die Daten nachgefordert. Die Nachlieferung erfolgt nicht immer.

Für die Gemeinden entsteht das Problem, dass digitale Daten lediglich beim Raumplaner vorliegen. Wechselt die Gemeinde den Planer, kann dieser nicht auf digitale Daten aufsetzen, was zu einer Verteuerung führt. Die Gemeindestrukturreform, in der viele Gemeinden fusioniert wurden, hat dieses Problem wesentlich verschärft. Es existiert keine Strafbestimmung bei Nichtlieferung der Geodaten.

Die im Rahmen der Dissertation entwickelte Prüfroutine entsendet automatisch eine E-Mail an die Gemeinden, falls die Geodaten positiv abgegeben wurden. Die Auszahlung des Honorars an den Planer kann an diese Mail gekoppelt werden, was im Interesse der Gemeinde als auszahlende Stelle liegt. Im Fall eines Wechsels des Planers stellt das Land dem neuen Planer die Geodaten bereit.

Viele über den eigentlichen Planungsinhalt gezeichnete Ersichtlichmachungen, etwa Ortsbildschutzgebiete, sind Planungen des Amtes der Steiermärkischen Landesregierung. Diese Festlegungen werden vom Referat Statistik und Geoinformation der Abteilung 17 den Planern bereitgestellt. Der Großteil dieser Daten wird in anderen Fachabteilungen erstellt. Die

Übergabe an das GIS-Steiermark ist nicht zwingend. Auch für diese Daten können dementsprechend die Richtigkeit und die Vollständigkeit nicht garantiert werden und müssen erst vom Raumplaner überprüft werden. Manche Ebenen oder Teile fehlen überhaupt und sind vom Planer zu erfassen. Das sind aufwendige Arbeiten.

### 3.3. Folgen der Probleme

Digitale FWPs und OEPs werden im Internet dargestellt. Es kann nicht garantiert werden, dass diese Daten mit den eigentlichen rechtsgültigen Papierplänen übereinstimmen. In wenigen Ausnahmefällen ist selbst der Stammpfad nicht aktuell. Es kann auch nicht davon ausgegangen werden, dass alle Änderungen eingearbeitet sind. Durch einen entsprechenden Hinweis wird auf diesem Umstand hingewiesen. Nichtsdestotrotz handelt es sich um ein Service einer öffentlichen Dienststelle, wo der Bürger höhere Ansprüche in Bezug auf Richtigkeit und Vollständigkeit hat.

Der elektronische FWP kann damit nicht seine Wirkung entfalten und dem Bürger Amtswege ersparen. Solange der elektronische FWP im Internet kein Gleichstück der Rechtsinformation ist, stellt dieser elektronische Plan lediglich eine Vorinformation dar und weiterführende Recherchen sind notwendig.

Für den Planer ist die Überarbeitung der Ersichtlichmachungen ein zeitintensiver Prozess.

### 3.4. Rechtliche Grundlagen

Eintragungen in ein (elektronisches) Register können rechtlich unterschiedliche Wirkungen entfalten. „Die deklaratorischen Eintragungen reflektieren eine bereits bestehende Rechtslage“ [4]. Das ist zu unterscheiden von der konstitutiven Wirkung einer Registereintragung, wo das Gesetz festlegt, dass erst die Eintragung in ein Register die Rechtswirksamkeit zur Folge hat. Eine konstitutive Wirkung hat der Eintrag in das Grundbuch, wodurch man Eigentümer einer Liegenschaft wird. Hat eine Eintragung in ein Register konstitutive Wirkung, regelt der Gesetzgeber, was in welcher Form in das Register einzutragen ist.

Der Vertrauensschutz – auch als öffentlicher Glaube bezeichnet – beschreibt, inwieweit auf die Richtigkeit und Vollständigkeit der Eintragungen vertraut werden kann.

Publizitätspflicht wird bei manchen Rechtsvorgängen gefordert, bei denen der Staat an einer öffentlichen Bekanntgabe interessiert ist. Das ist etwa beim Grundbuch oder Grundstückskataster der Fall. Aber auch Flächenwidmungspläne unterliegen der Publizitätspflicht. Man unterscheidet zwischen der positiven und der negativen Publizität. Ersteres meint, dass man sich auf das Eingetragene berufen kann und Zweiteres, dass bei einem nicht vorhandenen Eintrag das Recht auch nicht vorliegt.

Liehard und Zumstein weisen darauf hin, dass die Veröffentlichung im ÖREB-Kataster<sup>5</sup>, der ebenfalls nur deklaratorischen Charakter besitzt mit der Rechtswirkung der Publizität, des Vertrauensschutzes und der Beweiskraft verbunden ist [5] und Noack [6] führt aus, dass der ÖREB-Kataster rechtlich analog zum FWP zu sehen ist.

Laut Noack sind all diese Rechtsfolgen mit der Veröffentlichung des FWPs im Internet verbunden, sofern es sich um ein Register im rechtlichen Sinn handelt. Aufgrund der Datenqualität wäre die entsprechende Web-Map sofort zu schließen. Ein Register im rechtlichen Sinn benötigt jedoch einen gesetzlichen Auftrag, der für diese Datensammlung nicht vorliegt. Die rechtliche Interpretation benötigt Hintergrundwissen, welches nicht von jedem Bürger vorausgesetzt werden kann, weshalb die mangelnde Datenqualität problematisch bleibt.

Ein elektronischer FWP benötigt eine elektronische Unterschrift. Da es sich dabei um sicherheitsrelevante Aspekte handelt, werden das E-Government-Gesetz und die Signaturverordnung des Bundes im Kapitel 4.5.5.3 abgehandelt.

---

<sup>5</sup> Schweizer Register der **Ö**ffentlichen **R**echten **E**igentums**b**eschränkungen auf Grundstücke (ÖREB-Kataster)

Das Eigentum an einem Grundstück erwirbt man durch einen privatrechtlichen Vertrag zwischen Verkäufer und Käufer, wobei beide gleichberechtigt sind. Diese Rechte werden im Grundbuch geführt. Daneben gibt es Festlegungen von Gesetzgebern und Behörden (öffentlich-rechtliche Festlegungen), welche sich auf Grundstücke auswirken.

Für die öffentlich-rechtlichen Bestimmungen besteht bis heute keine zentrale Auskunftsstelle. Genau diese Lücke wird der ÖREB-Kataster schließen [4].

### 3.5. Beispiele anderer Länder

Die Flächenwidmungsinformationen, welche z. B. die Stadt Wien<sup>6</sup> oder Graz<sup>7</sup> als WebMap anbieten, stimmen mit der rechtlichen Situation überein. Das wird erreicht, indem die beschlossenen Papierpläne von den Gemeinden selbst geplottet werden und die dem Plan zugrunde liegenden Daten dann für die Kartendienste verwendet werden. Die beiden Magistrate erstellen die Flächenwidmungspläne und sind gleichzeitig an Geodatendiensten interessiert. Innerhalb des Amtes wird mit diesen Geodaten gearbeitet und sie werden auch für Auskünfte etc. verwendet. Nichtsdestotrotz findet man auf der Website der Stadt Wien einen Haftungsausschluss.

Neben diesen positiven Beispielen sticht sicher auch das Land Tirol mit seiner innovativen Lösung hervor. Hier wurde für den Bereich Flächenwidmung eine vollständige E-Government-Lösung geschaffen. Das gesamte Verfahren läuft ohne Medienbruch digital ab. Eine Ausnahme ist der analoge Plan. Gleichzeitig mit der Vorlage des analogen Planes erfolgt jedoch die Abgabe von Geodaten. Werden keine Geodaten über das Portal hochgeladen, wird der Ablauf des Verfahrens gehemmt, was sicherstellt, dass digitale Daten abgegeben werden [7].

Weiter heißt es in der Tiroler Plangrundlagen- und Planzeichenverordnung 2016 LGBl.Nr. 74/2013 zuletzt geändert durch LGBl.Nr. 112/2016 § 14 Abs. (1):

„Die Landesregierung hat die dem analogen Flächenwidmungsplan zugrunde liegenden digitalen Daten vor der Übernahme in den eFWP auf ihre Vollständigkeit und weiters daraufhin zu überprüfen, ob die Datenstruktur der Plangrundlagen- und Planzeichenverordnung 2004, LGBl. Nr. 13, in der Fassung der Verordnung LGBl. Nr. 2/2012 entspricht. Dabei ist insbesondere eine eindeutige Unterscheidbarkeit der Sonderflächen nach § 43 TROG 2016 hinsichtlich ihres jeweiligen Verwendungszweckes sicherzustellen.“

---

<sup>6</sup> <https://www.wien.gv.at/flaechenwidmung/public/>

<sup>7</sup> [https://geodaten.graz.at/WebOffice/synserver?project=flaewi\\_3&client=core](https://geodaten.graz.at/WebOffice/synserver?project=flaewi_3&client=core)

Amtsintern werden die digitalen Daten auf Folie ausgedruckt und über den analogen Plan gelegt, um sicherzustellen, dass es zu keinen Abweichungen der Geodaten und des analogen Plans kommt (vgl. Tiroler Plangrundlagen- und Planzeichenverordnung 2013 § 13 Abs. 2 LGBl. Nr. 74/2013). Die Richtigkeit und Vollständigkeit der Daten ist nun garantiert. Es erfolgen noch einige Verfahrensschritte, die hier nicht näher ausgeführt werden. Am Ende des Verfahrens steht die Kundmachung. Um ab diesen Zeitpunkt ausschließlich mit elektronischen Daten arbeiten zu können, wird ihnen Rechtskraft verliehen, indem die Kundmachung elektronisch erfolgt.

Das wurde im Tiroler Raumordnungsgesetz 2011 § 69 Abs. 1 LGBl. Nr. 56/2011 so geregelt:

„Die elektronische Kundmachung des Flächenwidmungsplanes auf der Grundlage der digitalen Daten obliegt der Landesregierung.“

Wurde eine Gemeinde in den elektronischen FWP überführt, werden Änderungen digital durchgeführt. Revisionen der gesamten Gemeinde sind nicht periodisch vorgesehen.

Einsehbar ist die Tiroler Lösung auf der Internetseite „eFWP Ein Dienst des Landes Tirol“ [97].

Da nicht mehr die Gemeinden, sondern das Amt der Tiroler Landesregierung die Flächenwidmungspläne kundmacht, hat die Aufsichtsbehörde rechtsgültige digitale Geodaten. Die Überführung der analogen Pläne in digitale Geodaten erfolgt jedoch durch eine zeitaufwendige optische Überprüfung von Landesbediensteten.

### 3.6. Lösungsansatz FWP

Thomas Noack fasste in seiner Masterarbeit an der ETH Zürich bereits im Jahr 2006 zusammen:

„Die Plandarstellung muss als hoheitlicher Akt durch eine kantonale Fachstelle aus den von einem Planungsbüro vorbereiteten Geodaten erfolgen. Somit ist sichergestellt, dass die Geodaten dem ausgedruckten Plan entsprechen. Die so erstellten

Pläne werden anschließend von der Gemeinde beschlossen, öffentlich aufgelegt und mit den zugehörigen Geodaten vom Regierungsrat genehmigt.“ [8]

Die hier beschriebene Vorgehensweise scheint für die Steiermark ideal zu sein. Die Gemeinde bzw. ein vor ihr beauftragter Raumplaner lädt die Geodaten und den Wortlaut von FWP bzw. OEP in einem Portal hoch und die Geodaten werden strukturell und auch inhaltlich durch Abgleich mit anderen Ebenen von der Software geprüft. Der Planer bzw. die Gemeinde erhält eine E-Mail mit den Ergebnissen dieser elektronischen Prüfung. Wird diese elektronische Vorprüfung bestanden, wird aus den Geodaten ein FWP bzw. ein OEP in Form einer PDF-Datei erstellt. Für jeden Upload wird eine eindeutige Nummer vergeben, welche am Plan aufgedruckt wird. Diese ID wird auch dem Wortlaut beigefügt. Plandatei sowie Wortlaut werden elektronisch signiert und sind damit immer eindeutig identifizierbar und zuordenbar. Der Prozess wird im Kapitel 4.5.5 genauer beschrieben. Das PDF-Dokument des Planes sowie der signierte Wortlaut werden der Gemeinde automatisch zugestellt. Das PDF-Dokument kann in der Gemeinde ausgeplottet werden und wird gemeinsam mit dem Wortlaut dem Gemeinderat zum Beschluss vorgelegt. Wird der Plan von der Gemeinde beschlossen, erfolgt ein Schreiben an das Amt der Landesregierung über den positiven Beschluss im Gemeinderat. Das Schreiben muss auch die eindeutige Nummer des Uploads enthalten. Sind mehrere Pläne im Umlauf, wird damit Verwechslungen vorgebeugt. Das PDF sowie der Wortlaut werden nun auch im Land gedruckt und zur aufsichtsrechtlichen Genehmigung vorgelegt. Wird diese erteilt, erfolgt die Kundmachung durch das Land.

Die Abbildung 5 zeigt diesen Ablauf in grafischer Form.

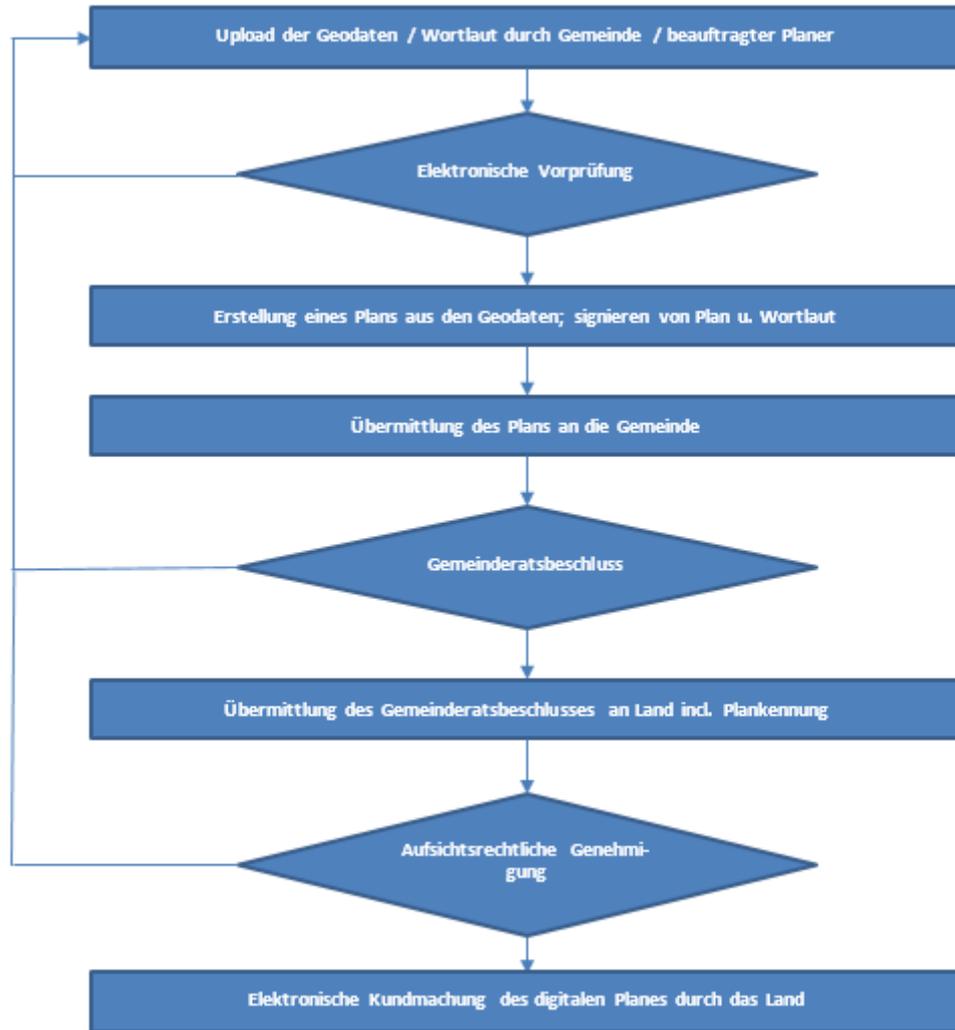


Abbildung 1: Schema des Verfahrens

Die Kundmachung der Verfahrenseröffnung durch den Bürgermeister, die Information der Bürger und des Landes, die Sammlung von Planungsinteressen sowie die Umweltverträglichkeitsprüfung erfolgen wie gehabt. Um den Planern bzw. der Gemeinde schon während der Entwurfsphase den konkreten Plan zeigen zu können, soll das Portal auch im Testmodus betrieben werden können. Dabei werden die Geodaten einer elektronischen Vorprüfung unterzogen und ein PDF-Plan wird erstellt, welcher der Gemeinde bzw. dem Planer zugesandt wird. Ein Wortlaut ist im Testmodus nicht notwendig.

Wird der beschlossene Plan im Land ausgeplottet, kann die aufsichtsrechtliche Genehmigung völlig analog ablaufen. Natürlich können diese noch nicht kundgemachten Geodaten auch über einen Webservice landesintern verwendet werden, was im Verfahren einen Vorteil bringt. Prüfungen können elektronisch oder halbelektronisch erfolgen. So kann der

FWP elektronisch über den Regionalplan gelegt werden und die Überlagerung mit Sachprogrammen ist möglich. Der Prüfaufwand sinkt dabei dramatisch. Prüfungen, die keine raumplanerische Interpretation benötigen, sind auch vollautomatisch durchführbar. Ein Beispiel dafür ist die Überprüfung, ob Sicherheitszonen um Munitionslager frei von Bauland ist. Die Korrektheit gegenüber im Land vorliegenden Zonen, wie etwa den Gefahrenzonen der Wildbachverbauung, kann hier oder bereits in der elektronischen Vorprüfung automatisch erfolgen.

Eine Qualitätsprüfung des planungsautonomen Teils der Gemeinde kann elektronisch unterstützt werden. Ein Beispiel ist die Auswertung, ob Bauland außerhalb eines 500-Meter-Bereiches um Haltestellen vorliegt. Handelt es sich um Neuausweisungen, kann es sich um einen Versagungsgrund handeln.

Das vorgeschlagene Verfahren genügt dem Prinzip der Publizität, da die Pläne im Internet für alle zugänglich sind. Es handelt sich um elektronische Gleichstücke der Rechtsdokumente, welche wie gehabt im Gemeindeamt aufliegen. Da es sich um Gleichstücke der rechtsgültigen Dokumente handelt, kann der Bürger auf die Richtigkeit vertrauen. Das Verfahren kann auch die Vollständigkeit und Aktualität garantieren. Vertrauensschutz besteht.

Soll die Datensammlung deklaratorische oder sogar konstitutive Wirkung erlangen, bedarf es einer Änderung des Raumordnungsgesetzes, wo die rechtlichen Wirkungen der Datensammlung festgelegt werden. Diese Arbeit soll aufzeigen, dass die benötigte Datenqualität für rechtsrelevante Geodaten ohne langfristige Zusatzaufwendungen lediglich durch entsprechende Softwareadaptionen und Verfahrensänderungen herstellbar ist. Die Betrachtung aller relevanten Faktoren soll es dem Gesetzgeber erleichtern eine entsprechende Verfahrensänderung im Raumordnungsgesetz zu beschließen.

### 3.7. Erweiterter Lösungsansatz des Teilbereiches Ersichtlichmachungen

Der FWP besteht aus planungsautonomen Teilen und Ersichtlichmachungen. Unter den planungsautonomen Teilen werden die Elemente verstanden, welche vom Gemeinderat bzw. einem beauftragten Planer festgelegt werden. Darüber hinaus werden im FWP auch zusätzliche Eintragungen durchgeführt, welche für den Leser des FWPs von Bedeutung sind. Diese Eintragungen werden Ersichtlichmachungen genannt, weil sie spezielle Fachthemen

wie die Hochwassersituation, Naturschutzgebiete, Ortsbildschutzzonen, Lärmzonen, Gestank etc. ersichtlich machen. In der Planzeichenverordnung ist festgelegt, welche Fachbereiche im FWP aufzunehmen sind. Diese Inhalte können die Planungsautonomie der Gemeinde einschränken. Es darf z. B. kein neues Bauland in den roten Gefahrenzonen der Wildbach- und Lawinerverbauung ausgewiesen werden.

Die Arbeiten der Planerstellung und Verfassung von Rechtstexten für rechtliche Festlegungen mit Raumbezug erfolgen im Amt der Steiermärkischen Landesregierung in verschiedenen Fachabteilungen, wie etwa den Bezirkshauptmannschaften, der Abteilung für Land und Forstwirtschaft, der Wasserwirtschaft und viele mehr. Dafür stehen einerseits die Rechtsmittel der Verordnungen und andererseits des Bescheides zur Verfügung.

Verordnungen sind Teil der Rechtsnormen. Entsprechend dem Stufenbau der Rechtsordnung stehen Verordnungen unter den Gesetzen. Verordnungen präzisieren die Gesetze und werden im Unterschied zu Gesetzen nicht vom Gesetzgeber (z. B. Landtag) erlassen, sondern von der Administrative als Teil der Exekutive (z. B. Landesregierung). Verordnungen richten sich nicht an eine bestimmte Person, sondern sind für alle gültig. Bescheide haben immer einen oder mehrere Adressaten.

Die Erstellung einer Verordnung mit Raumbezug erfolgt, indem von fachkundigen Beamten entsprechend den Gesetzen (z. B. dem Steirischem Ortsbildschutzgesetz Stammfassung LGBl. Nr. 54/1977) in enger Kooperation mit Kommissionen (z. B. Ortsbildkommission) und der betroffenen Gemeinde etc. Gebiete (z. B. Ortsbildschutzgebiete) festgelegt werden. Der Beamte erstellt einen Verordnungstext. Wurden früher die betroffenen Gebiete der Verordnung verbal beschreiben, wird heute meist eine Karte als Anlage beigefügt. Die Kartenerstellung erfolgt durch den Beamten selbst, sofern er über entsprechende GIS-Kenntnisse verfügt, oder über einen GI-Spezialisten in der Abteilung bzw. durch das GIS-Steiermark, falls es in der Abteilung nicht gelöst werden kann. Danach wird die Verordnung mit den Karten als Anlagen an den Landesverfassungsdienst übermittelt, welcher prüft, ob alle Gesetze eingehalten wurden. Liegt eine positive Stellungnahme des Landesverfassungsdienstes vor, kann die Verordnung in eine Regierungssitzung über das politische Büro eingebracht werden. Wird die Vorlage beschlossen, erfolgt die Einbringung des Verordnungstextes und der Karte vom Landesverfassungsdienst in das Rechtsinformationssystem des Bundes. Die eingebrachten Dokumente müssen als PDF-Dateien oder Word-Dokumente vor-

liegen. Zulässig sind Formate bis A3. Karten werden im Amt der Steiermärkischen Landesregierung fast ausschließlich mit ArcMap oder ArcGIS Pro erstellt, womit PDF-Dateien problemlos erzeugt werden können. Sind größere Gebiete detailreich darzustellen, bedient man sich sogenannter „Mapbooks“. Ein Übersichtsblatt zeigt in einem kleinen Maßstab eine Blatteinteilung der einzelnen A3 Blätter, welche danach folgen. Zur Erstellung von „Mapbooks“ existieren in ArcGIS sowie ArcGIS Pro Werkzeuge und eine eigene Toolbox.

Um zu rechtsgültigen Geodaten zu kommen, wurde eine Arbeitsgruppe eingerichtet,<sup>8</sup> welche die nachstehende Lösung erarbeitete. Die Geodaten müssen nicht selbst Rechtsrelevanz erlangen. Es ist ausreichend, dass die Geodaten vollständig dem tatsächlichen Rechtsstand entsprechen. Das kann durch eine einfache Verwaltungsänderung erreicht werden. Alle Beilagen in Kartenform werden vom GIS-Steiermark in Kooperation mit der Fachabteilung erstellt und mit einer speziellen Bildmarke versehen, über welche nur das GIS-Steiermark verfügt. Die Verordnung geht danach wie üblich zum Landesverfassungsdienst, wo nun auch geprüft wird, ob diese Bildmarke vorliegt. Liegt sie nicht vor, wird die Verordnung nicht positiv bewertet. Das GIS-Steiermark verfügt dadurch immer über die Geodaten, aus denen die rechtsrelevanten Karten gezeichnet wurden. In diesem Stadium weiß das GIS-Steiermark jedoch noch nicht, ob die Landesregierung die Verordnung auch erlässt. Ist dies der Fall, werden die Daten vom Landesverfassungsdienst in das Rechtsinformationssystem des Bundes (RIS) geladen. Zu diesem Zeitpunkt wird auch das GIS-Steiermark informiert, sodass die Geodaten als beschlossen gekennzeichnet werden können. Das erfolgt, indem die Fläche, welche die Raumausdehnung beschreibt, in eine FeatureClass der rechtsrelevanten Geodaten integriert wird und dem Feature der Link zur entsprechenden Verordnung im RIS beigelegt wird.

Die Lösung ist einfach und soll nicht nur für Ebenen angewandt werden, welche für FWPs und OEPs interessant sind, sondern für alle Verordnungen mit Raumbezug. Es entsteht mit der Zeit eine Sammlung an Geoinformation mit Rechtsbezug und die Vollständigkeit ab einen Stichtag sowie die Richtigkeit können garantiert werden.

---

<sup>8</sup> Mitglieder: DI Manuela Weissenbeck, Mag. Christian Freiberger, Mag Elke Schunter-Angerer, DI Rudolf Aschauer

Um das Vorgehen abzurunden, ist ein Erlass des Landesamtsdirektors anzustreben. Der beschriebene Weg wäre dann für die gesamte Landesverwaltung verpflichtend.

Privatrechtliche Beschränkungen an Grund und Boden werden im Grundbuch geführt und der Bürger hat die Möglichkeit, alle diese Einschränkungen gebündelt abzufragen. Im öffentlich-rechtlichen Bereich ist das wesentlich schwieriger. Ob ein Grundstück im Ortsbildschutzgebiet, in einem Naturschutzgebiet oder in einem Ausschlussgebiet für Windkraftwerke etc. liegt, kann den Wert eines Grundstückes und die Kaufentscheidung beeinflussen. Öffentlich-rechtliche räumliche Festlegungen sind also von Bedeutung. Die Information für ein Grundstück zusammenzutragen, ist mühevoll und für den Bürger kaum vollständig möglich. Deshalb ist mit dieser Initiative ein wesentlicher Schritt in Richtung eines transparenteren Staates gesetzt. Diese wesentlichen Informationen können in Form eines Web-Map-Services angeboten werden.

Um das System sinnvoll einsetzen zu können, ist es notwendig, Vollständigkeit anzustreben. Viele Geodatenätze sind bereits mit einer entsprechenden Genauigkeit und Zuverlässigkeit vorhanden. Einige müssten überarbeitet werden.

Diese Bemühungen umfassen nur Landesdaten. Viele öffentlich-rechtliche Beschränkungen werden jedoch von Bundesbehörden oder Gemeinden erlassen. Teilweise werden diese Daten vom Amt der Steiermärkischen Landesregierung übernommen und können dem Dienst beigelegt werden. Beispiel dafür sind die Gefahrenzonen der Wildbach- und Lawinnenverbauung. Sie beschreiben unter anderem die Überflutungszonen durch kleine Gewässer. Die Überflutungsflächen großer Flüsse werden vonseiten des Landes festgelegt. Nur gemeinsam entsteht ein Gesamtbild der Gefährdungssituation. Eine vollständige Sicht auf alle räumlichen Beschränkungen des Bundes ist leider nicht vorhanden.

Aufgrund der verschiedenen Zuständigkeiten, der Kompetenzbereiche und eines fehlenden gesetzlichen Auftrages ist es schwer, einen Kataster aller öffentlich-rechtlichen Beschränkungen zu bauen. Dem gegenüber stehen jedoch auch positive Beispiele, wie der ÖREK Schweiz.

„In der Schweiz gibt es über 150 verschiedene öffentlich-rechtliche Eigentumsbeschränkungen. In der ersten Phase werden auf Bundesebene die 17 wichtigsten Ei-

gentumsbeschränkungen aus acht Bereichen in den Kataster aufgenommen. Kantone und Gemeinden fügen laufend weitere Eigentumsbeschränkungen hinzu. Diese erfüllen in jedem Fall die folgenden Kriterien: Sie sind eindeutig definiert (im Gelände oder auf einem Plan), betreffen eine gewisse Anzahl von Grundstücken und sind rechtskräftig.“ [9]

Wann es in Österreich zu einer entsprechenden Initiative kommt, ist fraglich. Auf jedem Fall geht die Steiermark mit dem hier skizzierten Weg in die richtige Richtung. Das Amt der Steiermärkischen Landesregierung kann dann ihre rechtlichen räumlichen Festlegungen mit allgemeiner Wirkung entweder als Geodatensätze oder als Service gebündelt bereitstellen. Entschließen sich andere Körperschaften zu ähnlichen Initiativen und stellen ihre räumlich rechtlichen Festlegungen ebenfalls bereit, können diese Dienste oder Geodaten zusammengeführt werden und ein gemeinsames Bild entsteht.

Auch in Form von Bescheiden werden räumliche Festlegungen getroffen. Beispiele sind etwa Genehmigungsbescheide für den Schotterabbau oder Wasserentnahmerechte. Sie werden von den zuständigen Behörden entsprechend den Rechtsgrundlagen erlassen und entfalten ihre rechtliche Wirkung. Im Land Steiermark gibt es keinen zentralen Punkt, wo alle Bescheide zusammenlaufen, wie es der Landesverfassungsdienst bei Verordnungen ist. Dadurch ist es schwierig alle mit diesen Bescheiden verbundenen Räume zentral zu erfassen. Möchte man auch diese Daten trotzdem zentral führen,

- sind die entsprechenden Stellen zu identifizieren;
- sind die Abteilungen von der Notwendigkeit einer räumlich zentralen Erfassung zu überzeugen;
- ist eine Weisung des Landesamtdirektors notwendig, da in die Ablauforganisation eingegriffen wird;
- sind technische Unterstützungsleistungen sinnvoll.

Technische Unterstützungsleistungen sind etwa Applikationen, mit denen die räumlichen Ausdehnungen über ein Web-GIS erfasst werden. Im Bereich der Rodungen wurde eine solche Vorgehensweise bereits realisiert und wird von allen Förstern angenommen. So sinnvoll es auch wäre, all diese behördlichen Beschränkungen oder Genehmigungen der Grund-

stücksnutzung zentral zur Abfrage im Internet anzubieten, so schwierig wird dieses Unterfangen werden. Diese Einschätzung wird von Spangl geteilt, der sich wiederum auf Studien bezieht:

„Mehrere Schweizer Studien (z. B. Zufferey et al. (2006), Knoepfel und Wey (2005) und Meyer (2004)) sind zu dem Schluss gekommen, dass es unmöglich ist, alle öffentlich rechtlichen Eigentumsbeschränkungen in der Schweiz zu erfassen [...]“  
[10]

Beschränkt man sich auf ausgewählte Ebenen, wie es der Schweizer ÖREK-Kataster macht, ist das Projekt lösbar. So erschien in der Zeitschrift „cadastre N° 14, April 2014“ die Mitteilung:

„Der Aufbau des Katasters der öffentlich-rechtlichen Eigentumsbeschränkungen (ÖREB-Kataster) geht voran. Im Januar 2014 sind die ersten fünf von acht Pilotkantonen mit ihren kantonalen ÖREB-Geoportalen online gegangen.“ [11]

Die in Rahmen der OEPs und der FWP's notwendigen Ersichtlichmachungen sind in der Planzeichenverordnung festgehalten und die notwendigen Datenquellen können dingfest gemacht werden.

Der FWP beinhaltet derzeit 34 verschiedenen Ebenen mit Ersichtlichmachungen, wobei einige Ebenen mehrere unterschiedliche Quellen haben. Es ist zu hinterfragen, ob alle Ebenen zeitgemäß sind, wie etwa das Grundzusammenlegungs- bzw. Flurbereinigungsgebiet. Deshalb ist eine intensive Diskussion mit den betroffenen Fachbereichen notwendig und vonseiten des Landes auch mittelfristig vorgesehen. Weniger Ebenen würden das Problem der Bereitstellung rechtsrelevanter Ersichtlichmachungen vereinfachen und den ausgedruckten Plan für den Bürger auch leichter lesbar machen.

Ersichtlichmachungen werden den Planern derzeit vom Amt der Steiermärkischen Landesregierung bereitgestellt, von ihnen überarbeitet und mit dem FWP bzw. OEP rückübermittelt. Liegen vollständige und richtige Daten im Amt vor, ist die Rückübermittlung obsolet. Die aktuellen Ersichtlichmachungen können digital tagesaktuell über die Planungsinhalte der Gemeinde gelegt werden.

Alle bisherigen Überlegungen bezüglich Ersichtlichmachungen beziehen sich lediglich auf amtsinterne Abläufe, welche ohne Gesetzesänderungen möglich sind. Natürlich lassen sich elektronische Register gesetzlich verankern. Beispiel dafür sind das Firmenbuch und das Grundbuch. In Analogie zu diesen elektronischen Registern kann das auch mit allen anderen Rechtsnormen erfolgen. Der Gesetzgeber regelt die Wirkung des Eintrages, den Vertrauensschutz und die Publizität.

Kommt es bei Ersichtlichmachungen zu Widersprüchen mit den Daten der zuständigen Stelle, gelten die dort aufliegenden Sachverhalte. Primärer Inhalt des FWP's sind Kategorisierung in Bauland, Freiland und Verkehrsland, welche vom Gemeinderat beschlossen und vom Land genehmigt wird.

### 3.8. Zu schaffende Voraussetzungen

#### 3.8.1. Vereinfachung der Planzeichen

Die Erstellung der PDF-Pläne muss aufgrund der personellen Ressourcen des Landes vollautomatisch erfolgen. Darüber hinaus ist die vollautomatische Abwicklung auch notwendig, um den Verfahrenslauf nicht zu verlangsamen. Der Raumplaner bzw. die Gemeinde muss die PDF-Datei innerhalb weniger Minuten nach der Datenübermittlung erhalten. Nur ein solcher Service wird auf Akzeptanz stoßen.

Die in der derzeitigen Planzeichenverordnung eingeforderten Geodaten beschreiben den Planinhalt vollständig, sind jedoch nicht ausreichend, um FWP's bzw. OEP's in der derzeitigen kartografischen Qualität zu erstellen. Beschriftungstexte werden händisch gesetzt, um den knappen Platz bestmöglich auszunutzen. Die automatischen Algorithmen zur Textplatzierung, wie sie z. B. von MapLex bereitgestellt werden, reichen nicht aus, um ein ansprechendes Kartenbild mit vollständiger Beschriftung zu erhalten. Einige Planzeichen können von der Render-Engine, welche im Land verwendet wird, nicht in der notwendigen Qualität gezeichnet werden. Ein Beispiel soll das zeigen. Eine zeitliche Folgenutzung entsprechend ROG Steiermark § 22 (4) wird durch eine 3 mm starke nach innen gerichtete Randlinie dargestellt. Der Innenbereich wird in der derzeitigen Nutzungsfarbe gerendert. Die Randlinie kann stark gekrümmten Flächen nicht adäquat folgen, weshalb ein Buffer nach innen gerechnet werden muss. Die so entstehende Fläche wird anstatt der Randlinie dargestellt.

Es entsteht eine zusätzliche Ebene, welche die Verhältnisse verkompliziert. Bei sehr kleinen Flächen wird der Innenbereich völlig zugedeckt, sodass die derzeitige Nutzung nicht ersichtlich ist. Die Randlinie ist dünner darzustellen.

Ziel ist es also, eine Planzeichenverordnung zu schaffen, welche kartografisch ansprechend ist und sich automatisch in einen Plan umsetzen lässt, ohne dass menschliche Nachbearbeitung erforderlich ist. Es wäre in diesem Zusammenhang zu hinterfragen, ob alle Ersichtlichmachungen tatsächlich notwendig sind. Die zu beschriftenden Elemente sollten möglichst reduziert werden, da das viele Schriftgut die Karte schwer lesbar macht. Ein möglicher Ansatz ist es, sich primär auf die von der Gemeinde festgelegten Widmungen zu beschränken. Alle anderen Beschriftungen sollten nur bei ausreichendem Platz gesetzt werden. Das sind etwa die Grundstücksnummern. Der Kartenleser kann die fehlenden Grundstücksnummern aus der Lage zu beschrifteten Nachbargrundstücken ableiten. Alle Planzeichen sind auf die automatische Generierbarkeit zu prüfen und gegebenenfalls durch eine neue Symbolik zu ersetzen.

Am Ende dieses Prozesses kann sich die Notwendigkeit ergeben, dass gewisse Elemente des Designs in zusätzlichen Ebenen abzugeben sind. ESRI sieht für die kartografisch händisch nachbearbeiteten Beschriftungen „Annotation“ vor, und für händisch kartografisch überarbeitete Punkte, Linien und Flächen „Representation“, wobei es sich um Complex-Features handelt. Beide Strukturen benötigen als Datencontainer zumindest die FileGeodatabase. Die Verordnung solcher Datenstrukturen würde es ermöglichen, ohne große Änderungen der bisherigen Planzeichen und Beschriftungspraxis zu arbeiten. Es würde jedoch eine starke Firmenabhängigkeit entstehen. Alle Planenden müssten ESRI-Produkte verwenden, um die benötigten Complex-Features erstellen zu können. Die zeitaufwendige händische Nachbearbeitung zum Zweck der guten Lesbarkeit wäre nach wie vor vom Planer zu leisten. Es spricht also viel für eine deutliche Vereinfachung der Planzeichenverordnung, gepaart mit einem ausgereizten Programm zur optimalen Darstellung der Pläne. Als Hilfestellung dient eine Vielzahl an vorhandenen FWPs und OEPs mit allen in der Praxis auftretenden Spezialfällen.

Annotation und Representation sind Strukturen aus dem Hause ESRI. Andere Firmen haben Ähnliches vorgesehen und natürlich sind auch diese Strukturen geeignet. Dies ändert jedoch nichts an den vorgestellten Problemen. Die Abhängigkeit würde dann gegenüber einer anderen Firma bestehen.

### 3.8.2. Anforderungen an die EDV

Neben den rechtlichen Problemstellungen gibt es hohe Anforderungen an die Geoinformatik. Das System muss eine hohe Gebrauchstauglichkeit aufweisen (siehe Kapitel 4.5.9). Der Nutzer soll einem einfachen System gegenüberstehen, dessen Funktionalität er erkennt. Das kann erreicht werden, wenn man an alten Strukturen und Begrifflichkeiten festhält und nur solche Änderungen einführt, die eine merkbare Komfortsteigerung bringen. Die benötigten Programme sollen nicht nur eine hohe Gebrauchstauglichkeit gegenüber den primären Stakeholdern Raumplaner, Gemeinde und Land zeigen, sondern müssen vor allem auch den Bürger im Auge behalten, der einen einfachen Zugang zu rechtlich korrekter Information benötigt. Das kann über einen WebMap-Service erfolgen, wo die einzelnen Festlegungen mit Erläuterungen und dem zugrunde liegenden Rechtstext verlinkt sind. Aufgabe des Systems ist es, dem Bürger konkrete Erleichterungen zu bringen.

Komplexere EDV-technische Abläufe sind nur für das Wartungspersonal der Software zumutbar.

Das System soll eine hohe Zuverlässigkeit bieten (siehe Kapitel 4.5.3 Zuverlässigkeit). Sowohl Bürger, Raumplaner und Gemeinden als auch das Land müssen sich auf das Programm verlassen können. Diese Eigenschaft beschreibt nicht nur das Einhalten der Spezifikation (siehe Kapitel 4.5.1 Korrektheit), sondern fordert abseits der Festlegungen sinnvolles Verhalten.

Die Analyse hat einen hohen Prozentsatz an Systemabstürzen aufgezeigt. Niemand wird sich auf ein System verlassen, das offensichtlich nicht funktioniert. Die Beherrschung von Ausnahmesituationen, wie nicht bereite Server, volle Speicher, fehlende Lizenzen bis hin zu Programmfehlern, ist essenziell. Das Programm muss robust geschrieben werden (siehe Kapitel 4.5.2 Robustheit).

Doch die Meldung, dass derzeit aufgrund von fehlenden Lizenzen etc. kein Betrieb möglich ist, befriedigt nicht. Organisatorische Maßnahmen und auch eine funktionierende Softwarearchitektur müssen die Verfügbarkeit (siehe Kapitel 4.5.6 Verfügbarkeit) garantieren. So müssen für den Betrieb als Rechtssystem fixe Lizenzen bereitstehen, ausreichende Speichermedien vorhanden sein, die nur vom System beschrieben werden und die Ausfallsicherheit muss gewährleistet werden.

Der Schutz vor ungewollter Manipulation der Daten ist sicherzustellen (siehe Kapitel 4.5.5 Sicherheit). Das betrifft die Kommunikation zwischen den Beteiligten und die Abwehr von Hackern etc., aber auch den Schutz vor nichtdokumentierter Manipulation. Mitunter verlangen Systeme Wartungsarbeiten. Diese sollten alle durch entsprechende Routinen abgebildet werden, damit keine Fehler entstehen können und es nachvollziehbar bleibt, was gemacht wurde, denn diese Programme sollten ihre Arbeit dokumentieren.

Es ist eine Vielzahl an Programmen zu entwickeln und es darf die Wirtschaftlichkeit nicht außer Auge gelassen werden. Eine hohe Runtime-Flexibilität (siehe Kapitel 4.5.8 Flexibilität) soll es ermöglichen, durch Parameter gesteuert, unterschiedliches Verhalten zu erreichen. Sinnvolle Klassenbildungen und universelle Methoden erleichtern die Wiederverwertbarkeit (siehe Kapitel 4.5.7 Wartbarkeit und 4.5.8 Flexibilität).

Diese Qualitätskriterien zu erfüllen, ist schwierig und benötigt ein geplantes Vorgehen. Software-Engineering bietet diese Möglichkeit, wobei unter vielen verschiedenen Konzepten ein für das Projekt passendes zu wählen ist (siehe Kapitel 4.3 Bewertung und angewandter Prozess).

## 4. Software-Engineering

In der Frühzeit der digitalen Datenverarbeitung waren die Problemstellungen, die mit Computern gelöst wurden, von geringer Komplexität. Die Programme waren kurz und die Kosten für die Softwareentwicklung und Wartung im Vergleich zur Hardware gering.

„In den späten Sechzigerjahren zeigte sich erstmals, dass sich manche Software-Projekte selbst mit gigantischem Aufwand nicht zu einem befriedigenden Ende bringen ließen.“ [12, S. 46]

Dijkstra formulierte:

„The major cause of our software crisis is, that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.“ [13, S. 4]

Man erkannte, dass technische Lösungen, wie die Entwicklung leistungsfähiger Programmiersprachen, nicht ausreichen und suchte nach einer optimalen Portionierung und Sequenz von Softwaretätigkeiten. Die gefundenen Aufteilungen und Abfolgen wurden als Vorgehensmodelle bezeichnet.

Vorgehensmodelle führen zu einer deutlich besseren Planbarkeit der benötigten Zeit und zu Kostenreduktionen. Im Testbereich bringen sie einschätzbare Qualität. Normen und Standards werden erfüllt und Entwickler übernehmen mehr Verantwortung für ihr Produkt. Die Wiederverwertbarkeit und die Wartbarkeit steigen deutlich. Softwareentwicklung gilt seit der Softwarekrise nicht mehr als Metier, in welchem Kreativität besonders wichtig ist und wo der Softwareentwickler im Umfeld der Künstler gesehen und gesucht wurde, sondern als eine Ingenieurdisziplin. Anarchie wird durch ein Verwaltungssystem abgelöst. Für den Erfolg ist entscheidend, eine Balance zwischen kreativen Freiheiten und notwendigen Vorgaben zu finden. Viele verschiedene Modelle wurden entwickelt. Für erfolgreiche Softwaretätigkeiten muss das Modell selbst kritisierbar sein.

Man hat versucht, die Modelle untereinander zu vergleichen, indem Projektteams gleiche Arbeiten vorgelegt wurden, welche im Rahmen unterschiedlicher Vorgehensmodelle zu bewältigen waren. Ein Vergleich der Modelle war schwer möglich, da große individuelle Unterschiede bei Projektteilnehmern auftraten.

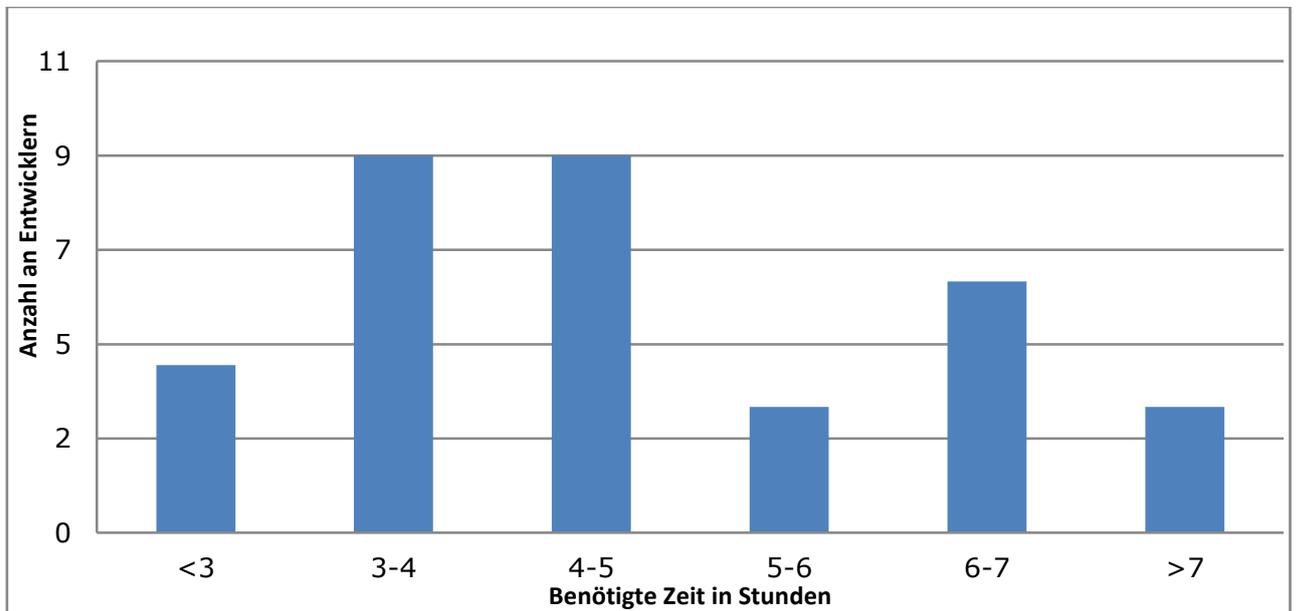


Abbildung 2: Unterschiedlicher Zeitbedarf von Entwicklern

Das zeigt in eindrucksvoller Weise eine Studie der Universität Stuttgart. Abbildung 6 ist der Studie entnommen und stellt die benötigte Zeit in Stunden, die Studenten für die Lösung eines Softwareproblems benötigt haben, dar. Unterschiede um den Faktor 2,7 zwischen dem langsamsten und schnellsten Studenten wurde gemessen [14].

So kommt es eher zu einer evolutionären Entwicklung verschiedener Vorgehensmodelle. Unterschiedliche Probleme benötigen entsprechende Modelle. Wesentlich ist, dass ein Softwareprojekt mehr ist als das Schreiben von Code. Gerne wird im Software-Engineering die Analogie zum Hausbau gezogen, wo nicht nur ein Maurer benötigt wird, sondern auch ein Architekt, ein Statiker und möglicherweise auch ein Vermesser, abhängig von der Größe und von der Komplexität des Projektes.

Auch im wissenschaftlichen Bereich ist die Anwendung dieser Strategien sinnvoll, denn die Forschungstätigkeit soll auch zu hochwertigem Softwarecode in abschätzbarer Zeit führen, wobei jedoch dem kreativen Element eine wesentlich höhere Bedeutung zukommt, als

sie kundenorientierte Projekte haben. Projekte der Wirtschaft haben häufig den Hauptfokus auf der Kostenminimierung bei einer definierten Qualität gelegt. Auch sind Forschungsarbeiten nicht immer auf ein streng definiertes Ziel hin orientiert. Ein wesentliches Element des Software-Engineerings fehlt damit. Zeitschätzungen sind unmöglich. Häufig stößt man auf neue lohnenswerte Forschungsfelder. Trotzdem sind weite Strecken mit den Methoden und Werkzeugen des Software-Engineerings zu bewältigen.

„Software Engineering spielt in der Informatik eine ähnliche Rolle wie die Hygiene in der Medizin: Sie nützt nichts, sondern verhindert vielmehr Schäden und sollte generell beachtet werden.“ [12, S. 51]

## 4.1. Vorgehensmodelle

### 4.1.1. Code and Fix

In der Geoinformatik sind viele Probleme durch Programmierstätigkeit zu lösen, wobei jedoch die Anwendung des Programmes vom Entwickler selbst erfolgt. Die Programme sind kurz, was sich sowohl auf die Lebensdauer als auch auf die Codezeilen<sup>9</sup> bezieht. Eine einmalige Datenübernahme ist z. B. durch ein Programm zu realisieren, da die Lieferung aus vielen strukturierten Einzelgeodatensätzen besteht. Hier ist es durchaus sinnvoll, den Code direkt zu entwickeln, die Entwicklung zu testen und iterativ dem Bedarf anzupassen. Diese Vorgehensweise wird auch als „Code and Fix“ bzw. „Build and Fix“ bezeichnet. Ludewig zählt als positive Elemente auf [12, S. 153]:

1. „Das Vorgehen entspricht unserem Drang, schnell voranzukommen und das uns gestellte Problem zu lösen.“
2. „Die Arbeit liefert schnell ein lauffähiges Programm.“

---

<sup>9</sup> Die Anzahl der Codezeilen als Maß ist im Allgemeinen nicht zielführend, da sie sehr stark vom Programmierstil abhängt. Eine Größenordnung ist jedoch festlegbar. Ein „10-Zeiler“ wird gerne als Synonym für ein kurzes einfaches Programm gebraucht, wenn es auch aus 100 Zeilen besteht.

3. „Die auszuführenden Tätigkeiten, Codieren und spontan Testen, sind relativ einfach.“

Er stellt diesem Vorgehensmodell entgegen [12,S. 154]:

1. „Das Projekt ist nicht planbar, weil nie rational entschieden wurde, was in welcher Qualität hergestellt werden soll.“
2. „Da die Anforderungen nicht systematisch erhoben worden sind, werden sie vom Resultat in der Regel nicht erfüllt.“
3. „Allen Prüfungen fehlen die Sollvorgaben.“
4. „Die entstehenden Programme sind meist schlecht strukturiert und nur mit großem Aufwand wartbar.“
5. „Der Aufwand für Korrekturen ist unangemessen hoch, da Mängel erst spät (nämlich im Einsatz) entdeckt werden und bekämpft werden können.“
6. „Wichtige Konzepte und Entscheidungen sind nicht dokumentiert, sondern lediglich in den Köpfen der Entwickler vorhanden. Dieses Wissen kann kaum transformiert werden.“

Das in dieser wissenschaftlichen Arbeit entstehende Softwareprojekt soll Akzeptanz bei den Raumplanern finden. Nur mit ihnen ist es möglich, das Ziel von rechtsrelevanten Flächenwidmungsplänen im Internet zu erreichen bzw. ihm nahezukommen. Verlässlichkeit und Qualität der Software sind also zentrale Kernpunkte. Treten Fehler auf, und davon ist bei jeder Software auszugehen, sind diese schnell zu beseitigen. So sind Punkt 4, 5, und 6 besonders schmerzlich und ein anderes Vorgehensmodell ist zu wählen.

Die Übernahme der Raumordnungspläne stellt eine zentrale Software dar. Sie muss unabhängig von Personen laufen. Dokumentationen sind unabdingbar. Diese Art des Softwareentwurfes (Code and Fix) zwingt nicht zur Dokumentation.

#### 4.1.2. Wasserfallmodell

Vom Wasserfallmodell spricht man, wenn die Entwicklungsschritte (Systemanalyse, Softwarespezifikation, Architektur, Codierung, Integration der Teile und Test, Installation, Inbetriebnahme und Wartung) vorgegeben sind und in einer definierten Reihenfolge abgearbeitet werden. Das wird häufig grafisch dargestellt und erinnert an einen Wasserfall (siehe Abbildung 7).

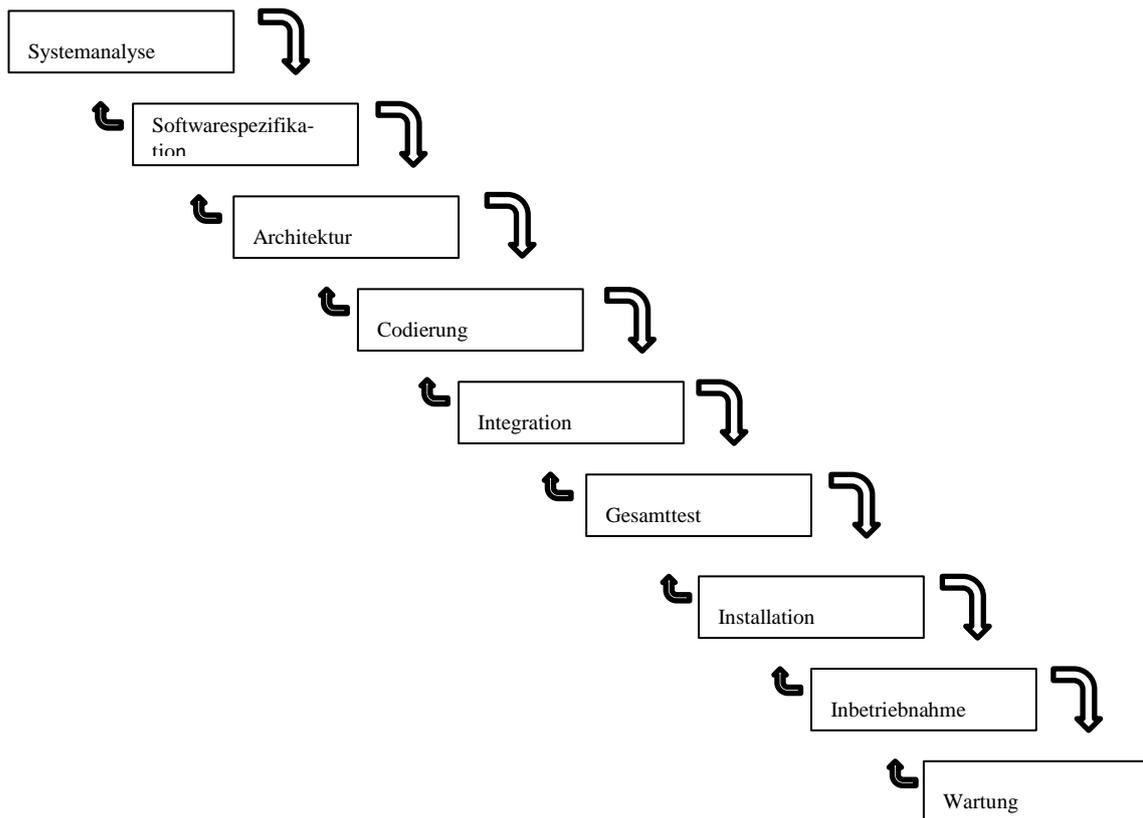


Abbildung 3:Wasserfallmodell

Dieses Modell ist die Antwort auf das „Drauflos-Programmieren“ der 1960er Jahre und wurde von Winston W. Royce (1970) populär gemacht [15]. Jede Phase hat einen definierten Anfangspunkt und Endpunkt, welche meist mit Schriftstücken festgehalten und dokumentiert werden. Beispiel ist etwa das Lastenheft (DIN 69901-5). Es sind also wesentlich mehr Arbeitsschritte zu erledigen als beim Code and Fix.

Die gewünschte Aktivität wird in Abbildung 7 durch dicke Pfeile symbolisiert. Sie bildet einen Fortschritt. Die Praxis hat gezeigt, dass Rückschritte notwendig sind. Den Vorteil dieser Methode wird jeder kennen, der schon einmal eine größere Software mit „Code and

Fix“ programmiert hat und den Überblick verloren hat. Das Wasserfallmodell führt zu Phasen und ist damit zeitlich und finanziell leicht zu steuern. Klar abgegrenzte Phasen gibt es häufig nicht. Das System kann sich nur schwer an Änderungen während des Projektablaufes anpassen. Fehler in einer frühen Phase sind schwer auszubessern, weshalb überproportionaler Aufwand in jede Phase zu stecken ist. Während des Entwurfes kann unter Umständen leicht erkannt werden, dass ohne viel Arbeit ein Vorteil erreicht werden kann bzw. ein geringer Vorteil mit extremem Aufwand programmiert werden muss. Insbesondere wissenschaftliche Projekte eignen sich daher nicht für das Wasserfallmodell, da generell Neuland betreten wird und in frühen Projektphasen nicht gesagt werden kann, was wie erreicht wird.

Ein Sonderfall des Wasserfallmodells ist das *Einbahnmodell*. Bei diesem sind überhaupt keine Zyklen erlaubt. Es ist gut planbar, jedoch völlig unflexibel.

Hierarchischen Organisationen kommt aufgrund der guten Planbarkeit das Wasserfallmodell (Einbahnmodell) entgegen. Die vorgesetzte Stelle kann gut über Kosten entscheiden und den Projektablauf anhand der Meilensteine kontrollieren. Wird von einer Phase in die andere übergegangen, ist aufgrund der Hierarchie mitunter eine Rückkoppelung nicht möglich. Das Dokument ist auf Biegen und Brechen umzusetzen. Eine einfache Lösung wird auf Ebene der Programmierer gesehen, kann jedoch nur durch eine leichte Spezifikation des Pflichtenheftes durchgeführt werden. Das ist jedoch aus der Position des Programmierers schwierig.

Die katholische Soziallehre hat für ein analoges Problem das Subsidiaritätsprinzip entwickelt.

„Was der Einzelne für sich und aus eigenen Kräften leisten kann, darf ihm nicht durch eine übergeordnete Instanz abgenommen werden. Eine übergeordnete gesellschaftliche Institution darf nicht die Aufgaben einer untergeordneten übernehmen und ihr die Kompetenz rauben. Sie ist vielmehr dazu da, subsidiär (also beistehend) einzugreifen, wo der Einzelne oder kleinere Institutionen durch einen Aufgabe überfordert sind.“ [16]

Die einzelne Phase sollte sich nur auf die Teile beziehen, die unabdingbar sind. Dann stellt das System einen brauchbaren Leitfaden dar. Darüber hinaus werden Mitarbeiter nicht

durch Vorschriften frustriert, was dem Arbeitsklima zuträglich ist. Nur zufriedene Mitarbeiter liefern gute Arbeit.

Mit der Methode Code and Fix und der Wasserfallmethode sind die extremen Standpunkte bestimmt.

#### 4.1.3. Nichtlineare Vorgehensmodelle

Neben den linearen Vorgehensweisen Code and Fix und dem Wasserfallmodell haben sich viele weitere Modelle etabliert. Beispiele sind etwa die evolutionäre Entwicklung, die iterative Entwicklung sowie die inkrementelle Entwicklung, wobei es sich bei diesen Modellen um nichtlineare Vorgehensweisen handelt.

Ina Schäfer von der TU Braunschweig folgend bestehen iterative Vorgehensmodelle aus einer Folge von Zyklen, wobei am Ende jedes Zyklus ein neues ausführbares Softwareprodukt steht, welches das vorherige verbessert und erweitert [17, S. 35]. Der Aufgabenstellung folgen eine Analyse, der Entwurf, die Implementierung und die Validierung. Die Validierung beurteilt die Vorversion, die entsprechend verbessert wird.

Abbildung 8 zeigt das iterative Vorgehensmodell.

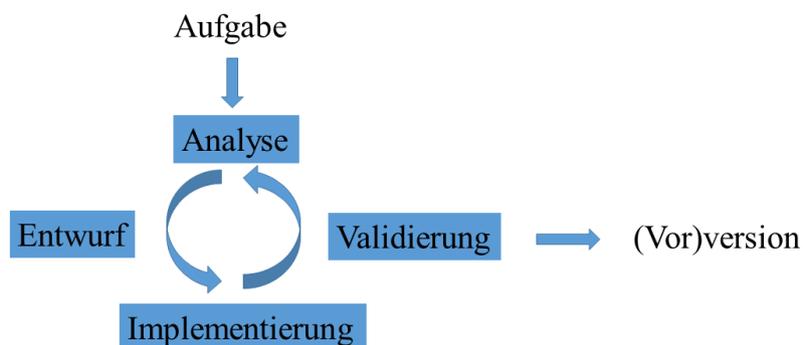


Abbildung 4: Iteratives Vorgehensmodell [17, S. 36]

Inkrementeller Softwareentwurf entwickelt die Software in Produktzyklen. Zuerst wird das Kernprodukt entwickelt und in weiterer Folge werden weitere Features hinzugefügt. Betriebssysteme werden so entwickelt, dass jedes Release weitere Funktionen liefert. Das angestrebte Ziel wird von Durchlauf zu Durchlauf besser erreicht. Die Anforderungsanalyse und die Konzeption erfolgen zu Beginn des Projekts und werden nicht mehr verändert.

Evolutionäre Entwicklungen führen eine Anforderungsanalyse und Konzeption bei jedem Durchgang neu durch. Damit kann man sich sehr flexibel Kundenwünschen anpassen. Objektorientiertes Programmieren erleichtert die ständige Überarbeitung der Software, da einzelne Basisklassen mit ihren Methoden unverändert in Gebrauch bleiben können. Wenn die Klienten nicht in der Lage sind, Anforderungen zu nennen, oder wenn völlig unklar ist, welchen Effekt der Rechnereinsatz hat, dann ist die evolutionäre Entwicklung das gebotene Vorgehensmodell [12, S. 172].

Die inkrementelle Entwicklung wird häufig mit der iterativen Entwicklung gleichgesetzt. Ludewig und Lichter sehen darin jedoch einen wesentlichen Unterschied. Iterative Entwicklung verbessert das System von Schritt zu Schritt in geplanten Durchgängen. Nur bereits festgelegte Komponenten werden hinzugefügt. Es erfolgt keine Analyse des Gesamtsystems. Die inkrementelle Entwicklung setzt die Ziele bei jedem Durchlauf weiter. Wird ein Programm erfolgreich eingeführt, verändern sich durch das Programm selbst die Bedürfnisse. Diese müssen häufig in einer neuen Iteration aufgearbeitet werden. [12, S 172-175]

#### 4.1.4. Verwendetes Vorgehensmodell

Im gegenständlichen Fall waren folgende Punkte vom Beginn an klar. Die Software soll

1. die Geodaten vollautomatisch übernehmen und auf ihre Richtigkeit kontrollieren,
2. verschiedene Schnittstellen durch Konfiguration unterstützen,
3. Kontakt mit den Planerstellern, Gemeinden und Fachdienststellen des Landes halten (automatische E-Mails),
4. im Fehlerfall automatisch korrigieren,
5. die Daten selbstständig administrieren.

Punkt 1, 2, 3 und 5 wurden zu Beginn entwickelt. Das war die Version 1.0 der Software. Danach wurde die geplante Erweiterung der automatischen Fehlerkorrektur realisiert, was zur Version 2.0 führte. Es liegt in diesem Maßstab eine iterative Entwicklung vor.

Welche Fehler Datenersteller hauptsächlich begehen, wie diese automatisch korrigierbar sind, war völlig unklar. Es erfolgte eine tief gehende Analyse, die zutage bringen sollte, womit der Geoinformatiker zu kämpfen hat. Auch war nicht klar, ob die Korrekturen in der notwendigen Qualität möglich sind und ob diese Daten von den Planern akzeptiert werden. Die automatischen Korrekturen wurden also evolutionär entwickelt. Es war völlig unklar, welcher Weg zum Erfolg führt.

Im Rahmen der Arbeit wurde erkannt, dass rechtliche Rahmenbedingungen zu setzen sind. Derzeit haben die Papierpläne Rechtsrelevanz. Geodaten, welche abgegeben werden, weichen manchmal von gestempelten rechtsgültigen Plänen ab. Das ist ein massiver Nachteil, da sich der Bürger nicht auf die Richtigkeit der dargestellten Situation verlassen kann. Web-Map-Services haben lediglich informativen Charakter und sollten nicht als Grundlage wirtschaftlicher Entscheidungen genommen werden. Um die digitale Raumplanung zu stärken, ist hier eine Lösung zu schaffen. Es kam also zu einer wesentlichen Erweiterung der Problemstellung, die ebenfalls zu behandeln ist. Neue erweiterte und geänderte Anforderungen sind Wesenszüge der evolutionären Entwicklung. Ein Projekt wird erstellt, man sieht das Ergebnis und verbessert es in einem weiteren Zyklus. Die Erweiterungen lagen hier nicht nur in der Software, sondern auch im administrativen Ablauf.

Evolutionäres Vorgehen dürfte ein Wesensmerkmal wissenschaftlicher Arbeiten sein, da Irrwege beschritten werden, sie zu korrigieren sind und mit mehreren Durchgängen ein brauchbares Produkt entsteht. Dabei kommt es naturgemäß zu einer strukturellen Korrosion. Das Überarbeiten des Softwarecodes führt zu einer „Zerfledderung“, welcher der Objektorientierung entgegenwirkt. Man soll sich daher auch Zeit nehmen, die Software nochmals zu überarbeiten, teile zur Gänze neu zu konzipieren, und nicht nur mit Stückwerk zu arbeiten.

Der Vorteil der Objektorientierung kommt dann zum Tragen, wenn elementare Klassen gebildet werden. Üblicherweise sind diese elementaren Klassen dadurch gekennzeichnet, dass sie nur aus wenigen Attributen bestehen. Aus diesen Klassen werden wiederum übergeordnete Klassen abgeleitet, welche die Objekte der elementaren Klasse verwenden. Das wird

so lange fortgesetzt, bis die Anforderung erfüllt ist. Kommt es im evolutionären Prozess zu einer neuen Entwicklung oder Abänderung, sind die elementaren Klassen der untersten Schichten meist nicht von einer Änderung betroffen und können übernommen werden.

Das iterative Vorgehen hat also zur Folge, dass schnell ein Endprodukt entsteht. Man kann ein Ergebnis vorweisen und hat das Ziel erreicht. Mit einem weiteren Durchlauf wurde die automatische Datenkorrektur entwickelt. Sie wird nun evolutionär den tatsächlichen Bedürfnissen angepasst. Die rechtlichen Erweiterungen sind ebenfalls von evolutionärem Charakter.

Teile des Systems waren vorab zu klären. So war z. B. nicht klar, ob es mit Python möglich ist, binäre Shape-Dateien zu verschicken. Daher wurde ein Prototyp erstellt. Prototypen können als Labormuster entwickelt werden, die wieder verworfen werden und nicht ins System einfließen, oder als Pilotsysteme, die sukzessive verbessert werden und ins System einfließen. Im ersten Fall kann der Prototyp nach dem Prinzip „Quick and Dirty“ entwickelt werden. Im zweiten Fall muss von Anfang an sauber entwickelt werden, da man an Systemteilen arbeitet. Wird ein technischer Aspekt abgeklärt, kann es auch notwendig sein, dass der eingeschlagene Weg gänzlich zu verwerfen ist. Deshalb sind Labormuster bei offenen Fragestellungen vorzuziehen. Liegt jedoch schon eine gut begründete Vermutung vor, kann auch eine Pilotstudie gewagt werden. Auf jeden Fall sollte vor der Pilotstudie der Entschluss fallen, ob ein Labormuster entsteht oder eine Pilotstudie. Wird nachträglich ein Labormuster zu einer Pilotstudie erklärt, fließt „Quick-and-dirty“-Code in das System ein, der meist nicht ausreichend getestet wurde.

#### 4.1.5. Vorgehensmodell versus Prozessmodell

Konkretisiert man das Vorgehensmodell, kommt man zum Prozessmodell. Das Vorgehensmodell liefert Hilfestellungen, welche Schritte nacheinander zu setzen sind. Damit ist jedoch noch nichts über die Form der Dokumente, der beteiligten Teams etc. gesagt. J. Grabowski zählt in seiner Vorlesung über Software-Engineering Wesensmerkmale eines Prozessmodells auf [18]:

- Organisation, Verantwortlichkeiten und Rollenverteilung;
- Struktur und Merkmale der Dokumente;

- einzusetzende Verfahren;
- auszuführende Schritte der Entwicklung, ihre Reihenfolge und ihre Abhängigkeiten (Vorgehensmodell);
- Projektphasen, Meilensteine und Prüfkriterien;
- Notationen und Sprachen;
- Werkzeuge.

Ludewig und Lichter weisen darauf hin, dass Prozessmodelle durch „Heilslehren, Glaubenskriege und Lagerdenken“ [12, S. 181] geprägt sind, was eine rationale Wertung schwermacht.

## 4.2. Prozessmodelle

### 4.2.1. Phasenmodell

Im Phasenmodell wird das Gesamtprojekt in einzelne Wegstücke geteilt und am Ende eines jeden Wegstückes steht ein sogenannter Meilenstein. Der Meilenstein muss gut definiert werden, und nur dann, wenn alle Bedingungen erfüllt sind, ist die Phase abgeschlossen. Die Definition des Meilensteins umfasst [12, S. 185]:

1. die Ergebnisse (Dokumente);
2. Prüfungen und ihre Kriterien ;
3. Wer die Prüfungen durchzuführen hat;
4. Person, die entscheidet, ob der Meilenstein erreicht wurde.

Ist eine Phase abgeschlossen, wird nicht mehr zu ihr zurückgekehrt. Damit ist das System äußerst starr und für wissenschaftliches Arbeiten schlecht geeignet.

Meilensteine haben jedoch auch große Vorteile. Verzögerungen werden bei jedem Meilenstein sichtbar. Immer wieder hört man in der Softwarebranche, dass „der Code nahezu fertig ist“, „die Architektur nahezu steht“. „Es sind 90 % des Weges gegangen.“ Wird von nahezu

fertigen Projektteilen gesprochen, sollte man hellhörig werden, denn die letzten 10 % brauchen wiederum die gleiche Zeitspanne. Viele neigen dazu, zuerst die einfachen 90 % zu entwickeln. Die komplexen 10 % der Entwicklung benötigen dann die gleiche Zeit.

Zeitschätzungen sind problematisch, da häufig unvorhersehbare Probleme, die fast immer eintreten, ausgeklammert werden. Es gibt jedoch nahezu kein Softwareprojekt, wo es nicht zu unerwarteten Problemen kommt, die zeitintensiv gelöst werden müssen. Beispiel für solche Probleme sind nicht einkalkulierte Netzwerkprobleme, mehrdeutige Metainformationen von Geodaten, die zu falscher Datenverwendung führen, und Ähnliches. Meilensteine reduzieren die Fehleinschätzung auf die Phase.

Arbeitet man immer mit den gleichen Personen zusammen, kennt man ihre Tendenz zur Fehleinschätzung und kann das in der Planung berücksichtigen. Das gilt natürlich im Besonderen für einen selbst. Auch wenn man alleine ohne Zeitdruck arbeitet, macht es Sinn, die einzelnen Arbeitsschritte im Voraus auch zeitlich zu planen und die benötigte Zeit der tatsächlich gebrauchten Zeit gegenüberzustellen. Damit verbessert man das eigene Gefühl für Arbeitsaufwendungen. Gelingt das nicht, erhält man zumindest einen Faktor, mit dem die geschätzte Zeit zu multiplizieren ist, um den tatsächlichen Aufwand zu erhalten.

Grundsätzlich sollte man zwischen Arbeitsstunden und Abgabeterminen unterscheiden. Abgabetermine sind häufig noch schwieriger einzuhalten als geplante Arbeitsstunden, denn während ein Projekt durchgeführt wird, kann es vorkommen, dass andere Projekte gestartet werden, an denen man mitarbeiten muss, was geplante Abgabetermine zusätzlich erschwert.

Die öffentliche Hand verfügt häufig über Referate, welche die Projektentwicklung hausintern durchführen. Werden Projekte ohne Beachtung der Auslastung zugeordnet, sind Abgabetermine nicht einzuhalten. Die Projekteigentümer sind darüber zu informieren und müssen sich einigen, welchem Projekt der Vorrang zu geben ist bzw. ob beide Projekte über eine entsprechend längere Zeit hinweg entwickelt werden. Zu einer solchen Einigung kommt es in der Regel nicht.

Gibt es keine hausinterne Verrechnung, fallen für den Projekteigentümer keine Kosten an und es werden auch Projekte initiiert, deren Aufwand in keinem Verhältnis zum Nutzen steht. Häufig vorhandene Kosten-Nutzen-Abschätzungen ohne echte Verrechnung sollten

hinterfragt werden, da sie oft Scheinzahlen enthalten. Ein sehr hilfreiches Mittel, um solche Projekt zu identifizieren, ist es, vom Auftraggeber eine Vorleistung in Form von benötigten Listen, Erfassungstätigkeiten etc. einzufordern. Wird diese Vorleistung nicht erbracht, kann das Projekt zurückgestellt werden, da es sicher nicht von hoher Bedeutung ist.<sup>10</sup>

#### 4.2.2. Unified-Software-Development-Prozess

Objektorientierung bietet in der Softwareentwicklung große Vorteile und wird seit den 1980 Jahren genutzt. Mit der Objektorientierung wurde ein eigenes Prozessmodell entwickelt. Es ist also das natürliche Projektmodell für objektorientierte Entwicklungen und wird Unified-Software-Development-Prozess oder kurz Unified-Prozess genannt. Dieses Modell wurde zuerst von den Autoren Ivar Jacobson, Grady Booch und James Rumbaugh in dem Buch „The Unified Software Development“ im Jahr 1999 vorgestellt [98]. Es handelt sich dabei um ein Phasenmodell mit einer Konzeptionsphase (Inception), einer Entwurfsphase (Elaboration), einer Konstruktionsphase (Construction) und einer Übergangsphase (Transition). Jede Phase besteht wiederum aus 5 Kern-Arbeitsabläufen (Core Workflows). Sie werden Anforderung (Requirements), Analyse (Analysis), Entwurf (Design), Implementierung (Implementation) und Test (Test) genannt. Abhängig von der Phase wird ein Kern-Arbeitsablauf intensiver oder weniger intensiv durchgeführt. Innerhalb einer Phase kann es zu iterativen Wiederholungen der Workflows kommen, wobei die Arbeitsergebnisse inkrementell erweitert werden. Üblicherweise wird das Modell durch eine Grafik wie Abbildung 9 dargestellt.

---

<sup>10</sup> Nicht jede Erkenntnis wurde aus dem in der Arbeit beschriebenen Projekt Flächenwidmungsplan gezogen. Sie sind jedoch von so essenzieller Bedeutung, dass sie hier nicht unerwähnt bleiben sollen.

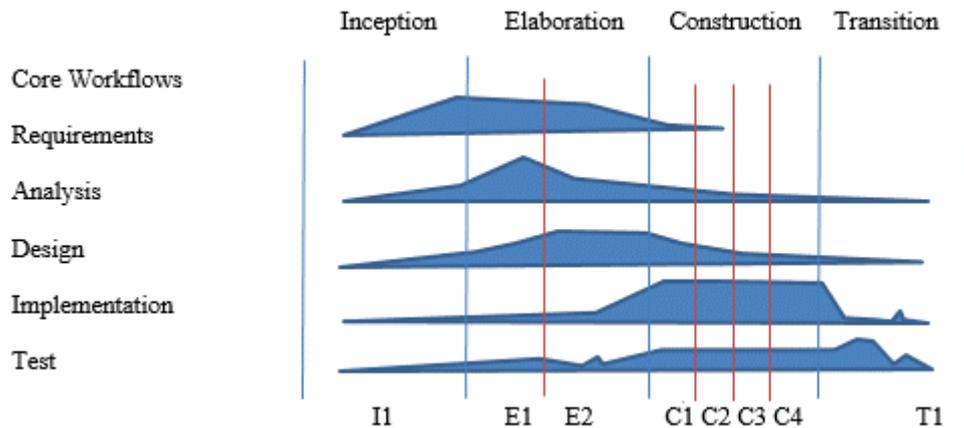


Abbildung 5: Unified-Software-Development-Prozess

Abbildung 9 zeigt horizontal die einzelnen Phasen. Vertikal werden die Kernaufgaben von oben nach unten durchgeführt, wobei die Dicke des Balkens die Intensität darstellen soll. Mit der Nummerierung C1, C2 etc. werden inkrementelle Iterationen angedeutet. Die Konstruktionsphase (Construction) wird in dieser Grafik viermal durchlaufen, wobei jeder Durchlauf die Funktionalität erweitert. Für ein tieferes Verständnis des Modells sind die Phasen genauer zu beschreiben.

So ist es Aufgabe der ersten Phase (Inception), zentrale Anwendungsfälle zu identifizieren bzw. festzulegen. Das Projekt ist abzugrenzen und die Voraussetzungen für den Projektfortgang sind zu treffen. Erste sehr grobe Use Cases werden geschrieben.

In der Phase der Elaboration erfolgt der eigentliche Entwurf. Fehlende Anforderungen werden noch gesucht und Entscheidungen im Bereich der Systemarchitektur sind zu treffen und niederzuschreiben. Use Cases werden stark verfeinert. Ein Prototyp zeigt die Funktionalität und Machbarkeit. Man versucht, Projektrisiken zu identifizieren.

Am Ende der Konstruktionsphase steht die Beta-Version zur Verfügung. Neben dem Aufbau von Datenbanken und dem Schreiben von Programmcode ist auch das Benutzerhandbuch zu vervollständigen.

In der vierten Phase wird das Produkt so lange verbessert, bis der Benutzer zufrieden ist. Dafür sind Rückmeldungen von Anwendern unabdingbar, die dann in das Produkt einfließen. Kann das Produkt vollinhaltlich genutzt werden, sind die Phase und der Prozess zu Ende.

Unter anderem bietet die Firma IBM/Rational die Software RUP (Rational Unified Process) an, welche konkret durch das Prozessmodell führt.

„Die konsequente und komplette Nutzung von RUP macht erst mit Teams über 10 Personen Sinn. Es sind über 30 Rollen für über 130 Aktivitäten vorgesehen und es werden über 100 verschiedene Artefakttypen (Arbeitsergebnistypen) vorgeschlagen, die erzeugt, dokumentiert und verwaltet werden müssen. Allerdings kann RUP sehr weitgehend reduziert und angepasst werden (Tailoring).“ [19]

#### 4.2.3. Agile Prozesse

Zweck des Software-Engineerings ist es, gute und hochwertige Software zu erstellen. Kommt es zu bürokratischen Prozessen, die für das vorliegende Projekt nicht angepasst sind, werden die geplanten Hilfestellungen zu schweren Steinen, die man im Rucksack mittragen muss. Viele Prozesse sind schwergewichtig und enthalten unnötige Komponenten, die nicht gebraucht werden.

2001 kam es in Utah (USA) zu einem Treffen von 17 Softwareentwicklern, welche Paradigmen eines leichtgewichtigen Prozessmodells formulierten. Diese Paradigmen wurden „Manifesto for Agile Software Development“ genannt und haben nachhaltig Einfluss auf die Softwarebranche genommen. Folgende Thesen wurden im Internet formuliert:

„We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.“ [20]

Das agile Manifest wird durch Prinzipien ergänzt, die hier ebenfalls wörtlich abgedruckt werden:

- „Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.“ [20]

Agile Prozesse versuchen also, dem Kundenwunsch möglichst zu entsprechen, was sich auch darin ausdrückt, dass die Entwickler die Bedürfnisse der Kunden ernstnehmen, auch wenn sie sich im Laufe des Prozesses ändern. Ein Forscher hat nicht immer mit Kunden zu tun. Nichtsdestotrotz kommt es bei Forschungsprojekten zu Änderungen des Plans. Oft wird ein anderer Weg eingeschlagen und der Forscher wirft seine bisherigen Konzepte über Bord, wenn er erkennt, dass es sinnvollere und wertvollere Ziele gibt bzw. dass andere Methoden erfolgversprechender sind. Dieser agile Ansatz der Forschung wurde früh erkannt. Schon 1867 wurde die Freiheit der Forschung als Artikel 17 in das Staatsgrundgesetz aufgenommen (siehe RGrBl. Nr. 142/1867).

Forschungstätigkeiten sind im Allgemeinen durch hohe Motivation gekennzeichnet und die Erfahrungen werden gerne bei Kaffeepausen ausgetauscht („face-to-face conversation“). Viele Forschungstätigkeiten sind Einzelarbeiten. Ansonsten entsprechen die oben formulierten Paradigmen und Prinzipien sehr dem universitären Arbeiten, welche geprägt ist von Freiheit statt von formalen Vorschriften, von gegenseitiger Hilfe und vor allem von viel Motivation.

Neue Erkenntnisse wirken oft wie Kundenwünsche, die während des Prozesses der Forschung auftreten. Sie lenken die Entwicklung in andere Bahnen und eröffnen oft neue Möglichkeiten oder auch Aufgaben.

An einem Beispiel sei das erläutert. Die abgegebenen Pläne bestehen aus der eigentlichen Planung und Layern, welche die Planung verständlicher machen (Ersichtlichmachungen). Für nahezu alle Ersichtlichmachungen gibt es entsprechende Ebenen im Land, die oft von einer nicht ausreichenden geometrischen Genauigkeit und fehlender Vollständigkeit sind. Im Rahmen dieser Dissertation war es auch geplant, Software zu entwickeln, welche die Geodaten des Landes mit den abgegebenen Plänen abgleicht. Ein Prototyp wurde entwickelt und hat die erwarteten Aufgaben erfüllt. Daneben wurden im Rahmen der Arbeit auch rechtliche Überlegungen angestellt, welche die Rechtsrelevanz der Geodaten gegenüber den gestempelten Papierplänen heben soll. Diese Überlegungen können nicht nur für die von Raumplanern über das Portal abzugebenden Pläne angewandt werden, sondern auch auf die Vergleichsebenen des Landes, die dann in rechtsrelevanter deklaratorischer<sup>11</sup> Qualität vorliegen (siehe Kapitel 3.7 Erweiterter Lösungsansatz des Teilbereiches Ersichtlichmachungen). Der Abgleich der Ebenen wird trivial, denn sie müssen identisch sein gegenüber den Ersichtlichmachungen der Planer. Geht man noch einen Schritt weiter, kann man

---

<sup>11</sup> Deklaratorische Eintragungen in ein Register repräsentieren einen Rechtsbestand, begründen ihn jedoch nicht. Wird erst durch die Eintragung das Recht begründet, spricht man von konstituierenden Eintragungen.

So kann ein Papierplan der die Rechtsmaterie konstituierende Akt sein. Werden diese Pläne für räumlich wirkende Verordnungen von einer Zentralstelle erstellt, kann diese Stelle die zur Erstellung der Pläne notwendigen Geodaten in ein deklaratorisches Register eintragen, das dann ab einen Stichtag vollständig und aktuell wäre und den Rechtsdaten entspräche. Die deklaratorischen Daten können über ein WebGIS der Öffentlichkeit präsentiert werden. Innerhalb der Landesverwaltung fallen viele dafür infrage kommende Ebenen an. Es bedarf lediglich einer Weisung des Landesamtdirektors, um diesen Zustand zu erreichen.

die Schnittstelle der abzugebenden Pläne dahin gehend ändern, dass die Ersichtlichmachungen nicht mehr vom Planer abgeben werden, sondern vom Land beigefügt werden.

Forschung ist also massiv kundenorientiert, sofern man das Wissen als Kunde ansieht.

Bekannte agile Prozesse sind Crystal, Scrum und Extreme Programming (XP), wobei Letzteres mit Abstand die höchste Aufmerksamkeit erreicht hat [12, S. 221].

#### 4.2.3.1. Extreme Programming (XP)

Ein wesentliches Element in XP ist der Kunde, der immer eingebunden ist. Die Entwicklung erfolgt iterativ und inkrementell. Zuerst wird das Kernprodukt gebaut und dann werden die Funktionen erweitert. Das Produkt wird in möglichst kurzen Zyklen erstellt. Das ermöglicht frühzeitige Akzeptanztests,<sup>12</sup> die zeigen, ob das Produkt den Wünschen entspricht. Jeder Zyklus gibt die Möglichkeit einer entsprechenden Überarbeitung.

Der Kunde wird in der Entwicklung benötigt und er muss viel Zeit für das Projekt erübrigen. Im gegenständlichen Fall war kein expliziter Kunde vorhanden, es wurden jedoch Gespräche mit verschiedenen Portalnutzern geführt.<sup>13</sup>

Bei einem linearen Phasenmodell wird der Bedarf zu Beginn erhoben. Dann erfolgt die Umsetzung und der Kunde wird erst wieder bei der Abnahme der Software eingebunden. Der Kunde muss seinen Bedarf sehr genau kennen und der Softwareingenieur muss alles exakt und richtig verstehen, was in der Praxis selten vorkommt. Agile Entwicklungen unter Einbindung des Kunden führen zu besserer Software.

---

<sup>12</sup> Akzeptanztests überprüfen die Zufriedenheit des Users mit der Lösung. Hier spielen die Oberfläche und die Praxisrelevanz eine entscheidende Rolle. Komponententests prüfen die Korrektheit der verwendeten Methoden. Der Computercode wird auf Richtigkeit überprüft, wobei es verschiedene Verfahren gibt.

<sup>13</sup> Ing. Markus Pfandler, Mag<sup>a</sup> Sigrun Rutrecht, Ing. Uwe Kreuzer, Birgit Wünscher, Mag Christoph Pfeifer, Christian Holzer, Dipl.-Ing. Knud Ulm, Dipl.-Ing. Weixelberger, Dipl.-Ing. Penzinger, Dipl.-Ing. Fuxjäger sei für wertvolle Beiträge gedankt. Leider ist diese Liste nicht vollständig.

Wird ein Softwareprodukt eingeführt, ändert sich das Verhalten der Nutzer. Ein Beispiel ist etwa das Mobiltelefon. Als eine Anwendung wurden von Mobilfunkbetreibern Kurznachrichten (SMS) implementiert. Dieser Dienst führte dazu, dass weniger Telefongespräche geführt und mehr Nachrichten verschickt wurden.

Ein lineares Phasenmodell berücksichtigt den meist schwer vorhersehbaren Änderungseffekt im Userverhalten im Allgemeinen nicht. Produktzyklen sind in der Lage, auf geändertes Userverhalten durch die Einführung von Softwareversionen zu reagieren.

Kurznachrichten müssen nun nicht mehr über eine kleine Tastatur vorgegeben werden, sondern können gesprochen werden. Software wandelt die Sprache in Text um.

XP formuliert eigene Werte. Das sind Kommunikation, Einfachheit, Feedback, Mut und gegenseitiger Respekt. Daraus ergeben sich Techniken, welche das Vorgehen eigentlich beschreiben:

- XP setzt auf „Pair Programming“. Zwei Entwickler teilen sich eine Tastatur, wobei jeder nur einige wenige Minuten codiert und dann die Eingabeeinheit seinem Partner überlässt. Darüber hinaus arbeitet man nur wenige Tage mit dem gleichen Partner zusammen. Diese Methode ist wohl die umstrittenste, da es scheint, dass der Personalbedarf verdoppelt wird. Diesem Argument werden die höhere Qualität, der Lerneffekt und die gegenseitige Motivation entgegeng gehalten. Im Rahmen dieser Dissertation war es leider nicht möglich „Pair Programming“ durchzuführen.
- „Test First Programming“ ist ein Verfahren, um Software zu testen, wobei zuerst der Code für die Tests geschrieben wird und danach die Methoden, welche getestet werden sollen. Das Verfahren wird auch als „Test-Driven-Development“ [21, S. 115] bezeichnet. Robert C. Martin schildert im Detail, wie er es von Kent Beck, einem der Gründerväter der agilen Bewegung, gelernt hat. Zuerst werden ganz kurze Tests geschrieben und die Methode wird nur so weit entwickelt, dass ein Kompilieren möglich ist. Dann wird wieder Testcode geschrieben und ein wenig Produktivcode. Test Driven Development zwingt den Entwickler, sich vorab exakt zu überlegen, wie die Methode funktionieren soll (Design). Die Tests helfen anderen Programmierern, zu verstehen, was die Methode kann (Dokumentation), und der Hauptvorteil liegt darin,

dass die Tests immer wieder durchgeführt werden können. Ändert man später irgendetwas im Programmcode, kann man alle Teile einfach wieder testen. Man braucht keine Angst vor unerwünschten Seiteneffekten haben.

- Incremental Design: „Das Design des Systems ist bei XP im Unterschied zu nicht-agilen Vorgehensmodellen keine Phase mehr, sondern wird begleitend zur Programmierung ausgeführt“ [22].
- Das System sollte so gestaltet sein, dass die Testläufe schnell und einfach durchgeführt werden können (10-Minuten-Build).
- Continuous Integration: Arbeiten mehrere am System, integrieren sie ihre Software mehrmals am Tag in das Gesamtsystem. Jeder weiß, was der andere gerade entwickelt hat. Man hat immer ein Produkt zur Verfügung, was dem Ziel von kurzen Produktzyklen entgegenkommt.
- Shared Code: Der Programmcode gehört allen und alle sind kollektiv für dessen Qualität verantwortlich. Das ist auch eine Folge des Pair-Codings mit wechselnden Teams. Der Punkt wird häufig diskutiert. Offensichtlich fühlen sich viele nicht wohl mit den Gedanken, niemanden für Fehlfunktionen verantwortlich zu machen. Andererseits geht die Verantwortung auf das Team über, welches sich selbst kontrolliert.
- Refactoring: Jede iterative inkrementelle Entwicklung hat zur Folge, dass der Code zerfleddert. Es entsteht ein Flickwerk, das von Zeit zu Zeit überarbeitet werden muss.
- Story Cards: Story Cards sind Karteikarten, wo in Schlagworten die Anforderungen niederschreiben werden. Sie werden in den einzelnen Iterationen abgearbeitet. Jede Story Card sollte in wenigen Tagen erledigt sein. Die in der Alltagssprache formulierte Softwareanforderung wird auch als User Story bezeichnet.
- Stand Up Meetings sind 15-minütige tägliche Treffen, in denen nicht diskutiert wird, sondern Aufgaben verteilt werden, und man tauscht sich über Probleme aus. Um nicht zeitlich auszufern, werden diese Meetings stehend abgehalten und auf zu spät kommende Personen wird nicht gewartet.
- Weekly Meeting sind Treffen, die meist am Montag stattfinden und die Wochenarbeit organisieren. Ein Rückblick erfolgt und die Storycards werden ausgeteilt.
- Informative Workspace: Der Arbeitsfortschritt soll für alle einfach erfassbar dargestellt werden, indem fertigen Story Cards auf eine Pinnwand angebracht werden.
- Sit Together: Um das Team zusammenzuschweißen, sollen die Leute in einem großen Raum zusammenarbeiten.

- KISS (keep it simple, stupid): Sicher schnell und zuverlässig werden nur einfache Probleme gelöst. Die einfachere Lösung ist die deutlich bessere Lösung!
- YAGNI (you ain't gonna need it): Problemlösungen auf Vorrat sind kontraproduktiv. Sie versperren den kreativen Weg, denn man wird verführt, die vorhandenen Lösungen zu verwenden. Es gibt jedoch auch Gegenbeispiele, wo Problemlösungen auf Vorrat sehr produktiv wirken. Geoinformationssysteme stellen verschiedene Problemlösungsalgorithmen bereit, die zur Lösung komplexer Aufgaben verwendet werden können. Beispiele für diese Problemlösungen sind die wohlbekannten Tools wie Intersect, Buffer, Clip etc. ArcGIS Desktop stellt etwa 700 verschiedene vorgefertigte Problemlösungen bereit. Die Erfolge von Geoinformationssystemen sind sicher unbestritten. Die Kreativität liegt in der Kombination der Werkzeuge.
- DRY (don't repeat yourself): Jede Funktion soll nur einmal entwickelt werden. Das ist sehr vorteilhaft, insbesondere bei Wartungsarbeiten und der Pflege des Programms.
- Whole-Team: Das Team soll alle benötigten Spezialisten enthalten. Ein Wir-Gefühl des Teams soll entstehen und Verantwortung für das gesamte Projekt wird übernommen. Es gibt keinen Außenstehenden, auf den die Schuld geschoben werden kann, wenn etwas nicht funktioniert. Stellt man sich vor, dass das Netzwerk von einem Dritten aufgebaut und betreut wird, würde sicherlich das langsame Netzwerk schuld sein, und nicht etwa ein Softwareentwurf, der extrem viel Leitungsverkehr verursacht.
- Energized Work: Überstunden sind nicht sinnvoll. Sie beuten Körper, Geist und Seele aus. Die Ursache vieler Überstunden liegt in einer falschen Zeitplanung. Zum Abgabetermin werden Überstunden gefordert. Um korrekte Zeitpläne zu erstellen, sollte man sich vor Augen halten, dass der Arbeitsaufwand in aller Regel unterschätzt wird. Ich gehe für mich persönlich von einem Faktor von 2 bis 3 aus, den ich für mich plane. Viele kennen ihren Faktor nicht bzw. rechnen nicht damit. Man kann ihn für die Kollegen aus der Erfahrung schätzen, falls man Zeitpläne zu erstellen hat.

Ludewig weist darauf hin, dass diese Punkte natürlich nicht unumstößlich sind. Was vernünftig erscheint, wird gemacht. So kann ein „Stand Up Meeting“ auch einmal sitzend durchgeführt werden und nur eine Person entwirft am Beginn ein Konzept [12, S. 225].

### 4.3. Bewertung und angewandter Prozess

„The secret of getting ahead is getting started“ ist ein Sinnspruch, welcher unter anderem auf Kleidungsstücken aufgedruckt ist und der auf viele Lebensbereiche angewandt werden kann. Auch die Entwicklung von Software muss begonnen werden, um vorwärtszukommen. Das Vorgehensmodell „Code and Fix“ wird häufig verhöhnt oder überhaupt nicht als Vorgehensmodell angesehen. Es hat jedoch einen großen Vorteil. Man beginnt mit der Arbeit. Im Laufe des Projektes wird man sehen, ob doch ein hochwertigeres Vorgehensmodell notwendig ist.

Die selektive Auswahl der Vorgehensmodelle und Prozessmodelle zeigt, welche für dieses Projekt als sinnvoll angesehen wurden. Meilensteine, die zentraler Bestandteil des Wasserfallmodells sind, wurden als notwendig und sinnvoll angesehen. Ludewig beurteilt Projekte, die ohne Meilensteine auskommen, grundsätzlich als gefährlich [12 S. 188]. Auch in diesem Projekt wurden Etappenziele definiert. Das führt zu einem schnelleren Vorwärtskommen. Die Analyse der häufigsten Fehler war ein solches Zwischenziel. Es wurde nicht mehr überarbeitet oder erweitert. Alle anderen Etappenziele wurden mehrfach überarbeitet. Die Ist-Analyse benötigt häufig keine Revision. Ist-Analysen sind nur dann zu ändern, wenn sich im Lauf des Projektes zeigt, dass neue Erhebungen notwendig sind oder sich die Situation selbst ändert. Das kann auch durch den Einsatz der entwickelten Software selbst verursacht werden. Ein Meilenstein benötigt eine definierte Abgabe, Prüfung und auch eine Person, die entscheidet, ob der Meilenstein erreicht wurde. Da an diesem Projekt nur eine Person gearbeitet hat, musste auch diese entscheiden, wann eine Phase abgeschlossen ist. Dementsprechend sollte man eher von Etappenzielen sprechen, da es grundsätzlich schwer ist, sich selbst zu kontrollieren.

Hauptsächlich wurde das Vorgehen von XP beeinflusst, wobei jedoch wesentliche Elemente wegfielen. „Pair Programming“ und „Sit Together“ im klassischen Sinn gab es nicht. Hilfestellungen seitens der Universität, Erfahrungsaustausch mit Kollegen, Literaturstudium und das Internet ersetzen diese Punkte.

Hohe Qualität des Codes war ein wesentliches Ziel und es wurde das Verfahren des „Test-Driven-Development“ angewandt, das sich als sehr vorteilhaft herausgestellt hat. Die Testroutinen stehen in der Datei, in der auch der Produktivcode abgelegt ist, und können so

einfach und jederzeit gestartet werden. Ein großer Vorteil war auch die Tatsache, dass Python eine Interpreter-Sprache ist. Komplexe Stellen wurden bewältigt, indem jede einzelne Zeile in einem eigenen Fenster ausprobiert wurde. Das Verhalten jeder Zeile kann so getestet und mit unterschiedlichen Parametern überprüft werden. Man kann auch Unterschiedliches ausprobieren. Entspricht der Code, wird er in das Programmierfenster kopiert.

Der inkrementelle Ansatz ist sicher auch stark von der Objektorientierung beeinflusst. Basisklassen werden entwickelt. Bei einer neuerlichen Überarbeitung können diese wiederverwendet werden oder die Klasse wird durch weitere Methoden bereichert. Das Vorhandensein von Testroutinen stellt sicher, dass die alten Codeteile nach wie vor funktionieren und nicht durch Seiteneffekte gestört werden. Objektorientierung trägt viel dazu bei, die Probleme in kleine Portionen einzuteilen. Jede Klasse erfüllt ihre kleinen Aufgaben, und so kann man dem Ziel KISS (keep it simple, stupid) leicht nachkommen. Schreibt man einen Code, so gibt es mehrere Wege, um die Aufgabe zu lösen. Es ist die einfachere und weniger komplexe Methode zu wählen. Da der Code immer sehr kurz ist, kann das meist ohne großen Aufwand erfolgen. Das Software-Engineering kennt verschiedene Methoden, die Komplexität von Software zu messen. McCabe fasste das Flussdiagramm eines Computerprogrammes als Graphen auf und formulierte die zyklomatische Zahl der Graphentheorie als zyklomatische Komplexität der Software. Im Kapitel 4.5 wird genauer darauf eingegangen.

Liest man Code, erkennt man schnell, was leicht zu verstehen ist, ohne Komplexität messen zu müssen. Eine gute Dokumentation ist entscheidend. Der beste Ort der Programmbeschreibung ist das Programm selbst, da die Dokumentation immer griffbereit und an der richtigen Stelle vorliegt. Die Dokumentation des Codes erfolgt jedoch nicht nur durch Kommentarzeilen, sondern auch wesentlich durch den Code selbst. Variablennamen, sprechende Methodenaufrufe und Klassennamen dokumentieren den Code. Objektorientiertes Programmieren ermöglicht sehr kompakte Schreibweisen. Im Nachhinein ist der Code schwer lesbar, wenn er auch am ersten Blick als elegant erscheint. An einem Beispiel in Python unter Verwendung der Bibliothek `arcpy` (ESRI Methoden) soll das erläutert werden.

```
lyr ...Layer
```

Schwer lesbar:

```
If int(arcpy.GetCount_management(lyr).getOutput(0))>0: ...
```

Gut lesbar:

```
AnzahlFeatureAlsObjekt = arcpy.GetCount_management(lyr)
EigentlicheAnzahl = AnzahlFeatureAlsObjekt.getOutput(0)
AnzahlAlsZahl = int(EigentlicheAnzahl)
If AnzahlAlsZahl > 0:
```

Kompakter Code gilt als elegant. Es geht jedoch Dokumentation verloren, die durch Variablenamen erfolgt.

UML (Unified Modeling Language) ist eine der verbreitetsten Modellierungssprachen und normiert Struktur- und Verhaltensdiagramme. Es wurde sehr konsequent ein Klassendiagramm mitgeführt.

„Jede Objektklasse wird in UML durch ein Kästchen beschrieben, das drei Unterteilungen aufweist, und zwar für den Namen der Objektklasse, die Attribute (inklusive der Geometrie- und Topologieattribute) und allfällige Operationen, die dieser Objektklasse zugeteilt sind.“ [23 S. 356]

Das Diagramm stellte sich als wesentlich für die Softwareentwicklung dar und bildete die zentrale Methode, um den Überblick zu bewahren. Es stellte sicher, dass nicht das Gleiche an mehreren Stellen codiert wurde. Leider wurden versteckte Methoden nicht im UML dokumentiert, wovon im Nachhinein abgeraten wird, denn es erhöht die Gefahr, diese Methoden mehrfach zu schreiben. DRY (dont't repeat yourself) wird gewahrt, wenn man den Überblick behält. Das Klassendiagramm wurde mit einem Tabellenkalkulationsprogramm geführt. Das Programm war also nicht für die Darstellung von UML-Diagrammen ausgelegt, weshalb sich die Führung des Diagrammes als mühselig herausstellte. In Nachhinein betrachtet sollten Entwickler das Diagramm entweder mit entsprechender Software zeichnen oder Papier, Bleistift und Radiergummi zur Hand nehmen. Es gibt Open-Source-Produkte, welche nicht nur das Diagramm zeichnen, sondern auch einen Rumpfcodes erstellen. Papier, Bleistift und Radiergummi haben den Vorteil, dass nicht genormte Anmerkungen

möglich sind. Das war natürlich auch im Tabellenkalkulationsprogramm möglich und sehr vorteilhaft.

Wird man durch die Klassenstruktur gezwungen, einen Code zu wiederholen, ist meist Refactoring notwendig. Die Struktur ist zu überarbeiten und die gleichartigen Teile sind zusammenzufassen. Basisobjektklassen sollten diese zentralen Aufgaben übernehmen und diese Fähigkeiten an übergeordnete Klassen vererben.

User Stories sind ein wesentliches Element der Steuerung des agilen Prozesses. In wenigen Sätzen wird auf einer Karteikarte eine Anforderung niedergeschrieben, die dann zu programmieren ist. In diesem Projekt wurde digital ein Merkheft geführt, in dem Anforderungen von Externen kurz dokumentiert wurden. Es wurden auch Gründe festgeschrieben, warum etwas nicht gemacht wurde, um die Argumente und Rechercheergebnisse wieder parat zu haben, falls danach gefragt wird. Im Allgemeinen waren die Vorgaben aber von den Analyseergebnissen der wesentlichen Fehler bestimmt. Story Cards wurden auch im Programmcode selbst abgelegt. Das war möglich, da alleine am Projekt gearbeitet wurde. Damit stehen die Story Cards immer zur Verfügung. Die räumliche Nähe zum Code erhöht dessen Lesbarkeit.

#### 4.4. Metrik

Mithilfe von Metriken versucht man, Software zu bewerten. Metriken ermöglichen Entwicklern, Softwarecodes untereinander in Bezug zu setzen. Aus der Anzahl an Zeilen, aus denen ein Programm besteht, kann gesagt werden, welches Programm länger ist. Für viele Metriken existieren Erfahrungswerte. Unterschreitet oder überschreitet der Wert ein gewisses Maß, kann mit Problemen in der Software gerechnet werden. Ein Beispiel ist der Wartbarkeitsindex, der für „pflegeleichte“ Programme über 85 liegen sollte. Wichtige Metriken sollen hier zentral vorgestellt werden, um bei den einzelnen Qualitätskriterien darauf verweisen zu können. Sie sind ein Schlüssel für gute Software. Darüber hinaus wird eine Metrik auch erweitert.

Die Erhebung der Maßzahlen ist genormt (ISO/IEC 15939 (2007) Systems and software engineering – Measurement process). Es existiert Software, welche die Metriken automatisch aus dem Quellcode ableitet. Ein Beispiel ist Radon.

Die Metriken wurden verwendet, um die Qualität der entwickelten Software zu heben bzw. zu bewerten. Damit steht dieser Abschnitt in einen engen Zusammenhang zum Kapitel 4.5 Qualität von Software und kann als Vorbereitung dazu angesehen werden.

#### 4.4.1. Anzahl an Linien – Lines of Code

Die bekannteste Maßzahl ist wohl die Anzahl der Lines of Code (LOC). Dabei werden die physikalischen Zeilen, Programmzeilen, Kommentarzeilen und Leerzeilen gezählt. Programmzeilen können auch Kommentare enthalten, weshalb die Anzahl an physikalischen Zeilen kleiner als die Summe an Kommentarzeilen, Leerzeilen und Programmzeilen sein kann.

```
anz = int(arcpy.GetCount_management(fc).getOutput(0)) #Anz..Zeilen von fc
```

Die Anzahl an Programmzeilen hängt stark vom Programmierstil des Entwicklers ab, weshalb diese Kenngröße nur bedingt brauchbar ist um den Umfang eines Programmes zu beschreiben. Zur Angabe von Größenordnungen ist diese Metrik jedoch hervorragend geeignet und sie ist Teil abgeleiteter Metriken. Sie sollte nie ganz außer Acht gelassen werden.

#### 4.4.2. MacCabe-Metrik – Cyclomatic complexity

McCabe geht davon aus, dass ein Programm kompliziert ist, wenn sein Flussdiagramm auf viele verschiedene Arten durchlaufen werden kann. Er führte im Jahr 1976 die zyklomatische Komplexität ein. Dabei wird das Flussdiagramm eines Programms als Graph aufgefasst und die zyklomatische Komplexität ist jene Zahl, welche angibt, auf wie viele linear unabhängige Arten der Graph durchschritten werden kann. Das kann mit der nachstehenden Formel errechnet werden.

$$Z(G) = e - n + p$$

Z(G)	zyklomatische Komplexität des Graphen; Anzahl der linear unabhängigen Pfade
e	Anzahl der Kanten(edges)
n	Anzahl der Knoten (nodes)
p	Anzahl der Außenverbindungen

Hat man einen Eingang und einen Ausgang, nimmt p den Wert 2 an [12, S. 315].

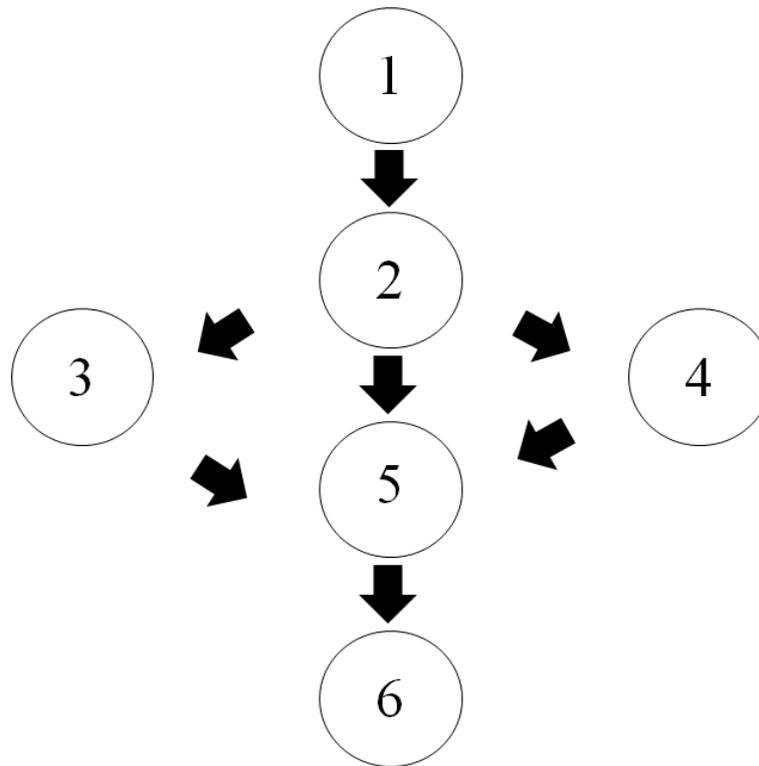


Abbildung 6: Zyklomatische Komplexität am Beispiel

Zyklomatische Komplexität = Anzahl der Kanten – Anzahl der Knoten + 2

$$3 = 7 - 6 + 2$$

Ein einfacher, alternativer Weg, die McCabe-Komplexität zu bestimmen besteht im Abzählen von Schlüsselwörtern wie if, while etc. und Addieren von 1. Man zählt die binären Verzweigungen plus 1. Im Internet wird das einfach erklärt:

„Cyclomatic Complexity = ( 1 + ifs + loops + cases ) where:

- ifs is the number of IF operators in the function,
- loops is the number of loops in the function,
- cases is the number of switch branches in the function (without default)“ [24]

Die McCabe-Komplexität lässt sich also durch Abzählen von Schlüsselwörtern des Quellcodes ermitteln.

Große Bedeutung hat die McCabe-Komplexität auch in Zusammenhang mit der Testtheorie. Da die zyklomatische Zahl die Anzahl der linear unabhängigen Schleifen angibt, entspricht sie der Anzahl an Tests, die notwendig sind, wenn jeder Zweig zumindest einmal durchlaufen werden soll.

Ludewig und Lichter weisen darauf hin, dass es sich eher um ein Volumenmaß als um ein Komplexitätsmaß handelt [12, SS. 313–316].

#### 4.4.3. Halstead-Metrik

Halstead entwickelte 1977 weitere Metriken, welche auf der Anzahl von Operanden und Operatoren beruht [25]. Variablen und Konstanten stellen die Operanden dar. Operanten sind also Symbole für Daten. Operatoren führen zu Aktionen, wie die Schlüsselworte if, while, for. Auch + - und Ähnliches werden zu den Operatoren gezählt, sowie Einrückungen, welche Code zusammenfassen.

Gezählt wird nun:<sup>14</sup>

– Gesamtanzahl der Operatoren	$N_1$
– Anzahl der unterschiedlichen Operatoren	$n_1$
– Gesamtanzahl der Operanden	$N_2$
– Anzahl der unterschiedlichen Operanden	$n_2$

Wobei gilt:

Halstead-Volumen  $V$

$$V = N \log^2(n)$$

wobei gilt

$$\begin{array}{ll} n = n_1 + n_2 & n \dots \text{Vokabulargröße} \\ N = N_1 + N_2 & N \dots \text{Programmlänge} \end{array}$$

<sup>14</sup> In Großbuchstaben wird die Gesamtzahl von Operatoren bzw. Operanden bezeichnet. Diese kann wesentlich höher als die Anzahl unterschiedlicher Operatoren bzw. Operanden sein, da jeder Operator bzw. Operand mehrfach im Quellcode vorkommen kann.

Schwierigkeitsgrad (Difficult Level) D, Fehlerneigung

$$D = n_1/2*(N_2/n_2)$$

Implementieraufwand E (Effort to implement)

$$E = V * D$$

Implementierzeit T (time to implement)

$$T = E / 18$$

wobei gilt

T in Sekunden

Anzahl an Fehlern B (Bugs)

$$B = E^{2/3}/3000$$

[26]

Die Implementierzeit und die Anzahl an Bugs sind individuell verschieden. Im Kapitel 4. Software-Engineering wurde bereits ausgeführt, dass die Zeiten für die Entwicklung des gleichen Problems bei verschiedenen Entwicklern im Rahmen einer Studie um den Faktor 2,7 schwankten. Das kann auch für die Fehlerhäufigkeit vermutet werden.

Die Anzahl der Fehler steigt exponentiell und kann kleingehalten werden, wenn der Implementieraufwand klein ist. Das ist dann der Fall, wenn das Programm kurz und einfach ist. Das Agile-Prinzip „keep it simple, stupid“ ist der Schlüssel zum fehlerreduzierten Programmieren. Der Schwierigkeitslevel sinkt, wenn mit wenigen Operatoren gearbeitet wird. Es ist mit einem kleinen Sprachumfang zu arbeiten. Die zweite Stellschraube ist das Verhältnis der Gesamtanzahl an Operanden zur Anzahl der unterschiedlichen Operanden, die im Idealfall 1 erreicht, im Allgemeinen jedoch größer ist. Muss eine Variable vielfach belegt werden, steigt die Komplexität.

#### 4.4.4. Maintainability-Index

Der Maintainability-Index misst die Wartbarkeit von Software und liefert Werte zwischen 0 und 100, wobei kleine Werte für schlechte Wartbarkeit stehen und große Werte für eine gute Maintainability.

„Der Wartbarkeitsindex wird auf der Basis der Zeilenmetriken, der zyklomatischen Zahl von McCabe und der Halstead-Metriken berechnet.“ [27, S. 43].

Folgender Formelapparat kommt zum Einsatz:

$$MI = MI_{woc} + MI_{cw}$$

$MI_{woc}$ ... Maintainability Index without commands

$MI_{cw}$ ... Maintainability Index commands wight

wobei

$$MI_{woc} = 171 - 5,2 * \ln(\text{aveV}) - 0,23 * \text{aveG} - 16,2 * \ln(\text{aveLOC})$$

$$MI_{cw} = 50 * \sin(\sqrt{2,4 * \text{perCM}})$$

$\text{aveV}$  ...mittleres Halstead-Volumen der einzelnen Module

$\text{aveG}$  ...mittlere zyklomatische Zahl

$\text{aveLOC}$  ...mittlere Anzahl an Codezeilen ohne Kommentaren

$\text{perCM}$  ...prozentuelles Verhältnis von Kommentarzeilen zur Codelänge

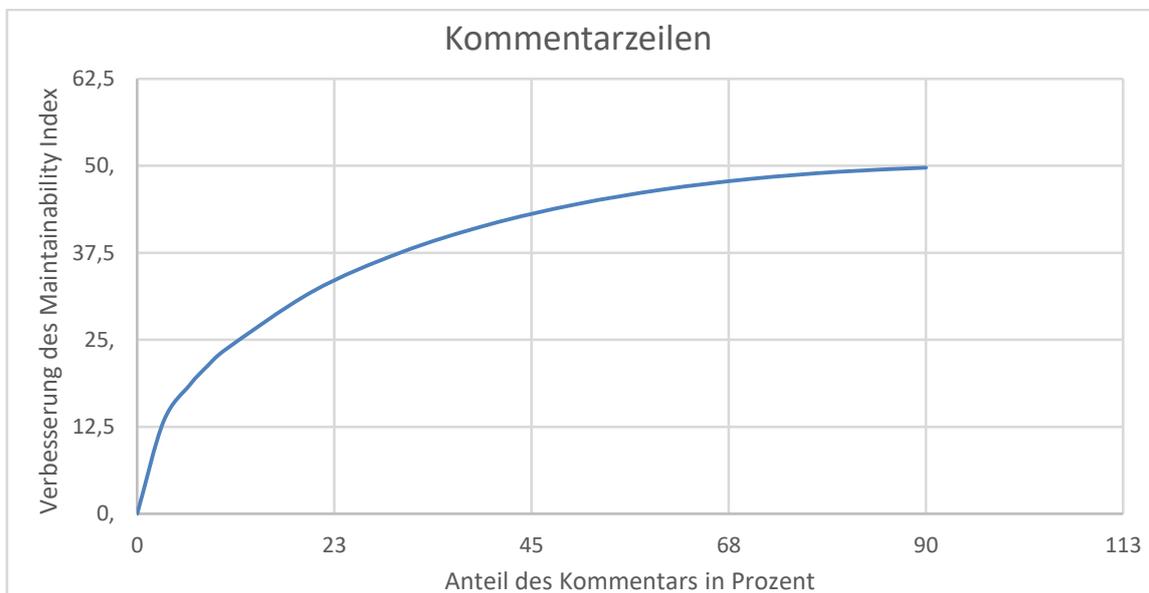


Abbildung 7: Wartbarkeitsindex und Kommentarzeilen

Abbildung 11 zeigt die Verbesserung der Wartbarkeit durch Kommentierung. In diesem Modell wird bis zu einem Drittel Kommentar gut belohnt. Danach wird der Einfluss eher gering.

Die Codelänge geht einerseits direkt mit dem Term  $-16,2 * \ln(\text{aveLOC})$  in den Wartbarkeitsindex ein und andererseits indirekt über das Halstead-Volumen. Beispielhaft wurde das proportionale Verhältnis zwischen Halstead-Volumen und LOC mit 16:1 ermittelt ( $V =$

16\*LOC).<sup>15</sup> Damit lässt sich nun ein Diagramm zeichnen, welches den Zusammenhang zwischen Codelänge und MI-Abnahme zeigt.

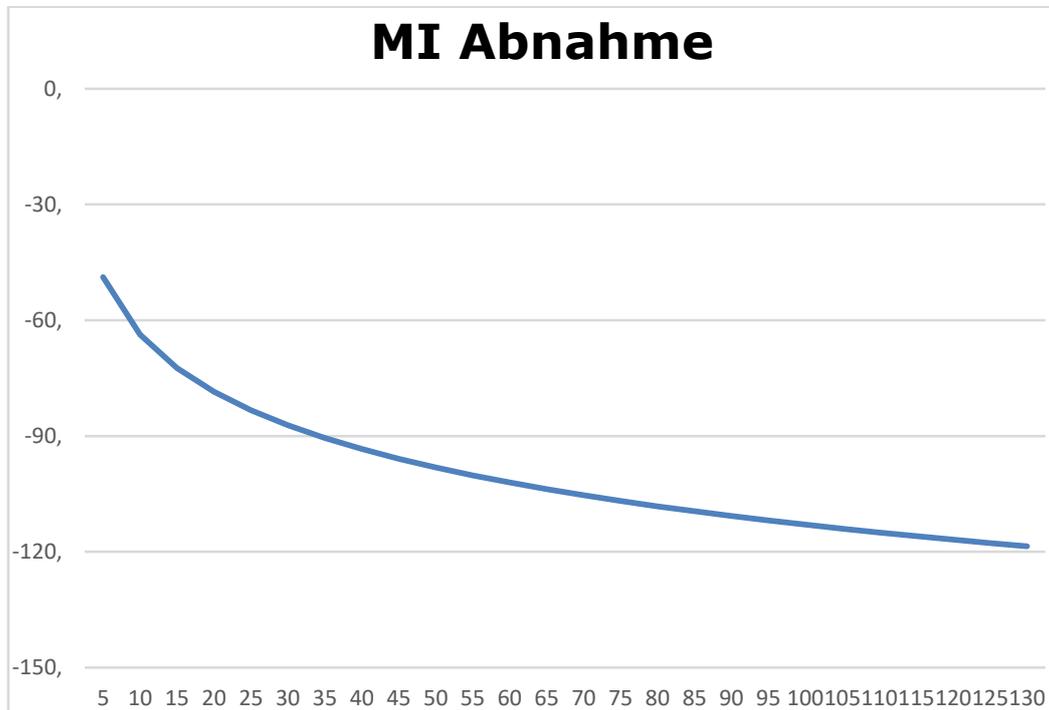


Abbildung 8: Wartbarkeit und Programmlänge

Es besteht ein hoher Einfluss zwischen der Programmlänge und der Wartbarkeit. Kurze Methoden und Programme sind essenziell. Die Wartbarkeit wird wesentlich durch die Konzeption der Klassen bestimmt. Kluge Klassendefinition kann zu kurzen Programmen führen. Es ist wichtig, jedes Programm in möglichst viele Einzelschritte zu gliedern, die in Module ablegt werden und idealerweise mehrfach genutzt werden.

Die zyklometrische Zahl geht mit einem linearen Term in das Wartbarkeitsmodell ein. Jedes zusätzliche if, while etc. erhöht den Kontrollfluss um 1 und reduziert die Wartbarkeit um  $-0.23$ . Die zyklometrische Zahl sollte laut Ludewig Lichter unter 7 [12] und nach X. Cullmann und K. Lambertz unter 15 [27] liegen. Sie beschreibt die Komplexität. Sehr komplexe, gerade noch akzeptable Programme reduzieren die Wartbarkeit also um den Faktor

<sup>15</sup> Eine durchschnittliche Methode wurde ausgewählt, wobei sich die Durchschnittlichkeit auf LOC (ohne Kommentarzeilen) und die zyklometrische Komplexität bezieht. Für diese Methoden wurde das Verhältnis bestimmt.

1.61 bzw. 3.45. Im Verhältnis zu fehlenden Kommentaren und zu langen Programmen wird der Einfluss der Komplexität auf die Wartbarkeit gering bewertet.

Bei einem evolutionären Vorgehen, wo die Software mit jedem Zyklus neu überarbeitet wird, kommt dem Maintainability-Index große Bedeutung zu. Microsoft empfiehlt Werte über 20. Unter 10 sollte er nicht fallen und dazwischen ist der Code kritisch zu betrachten [28].

#### 4.4.5. Objektorientierte Metriken

„Besonders nachteilig an dem Verfahren nach McCabe ist jedoch, dass diese nur die Architektur eines Programms berücksichtigt, dagegen aber die einzelnen Anweisungen und Ausdrücke in deren Komplexität wie beispielsweise Schachtelungstiefen von Schleifen oder den Einfluss langer Sequenzen nicht in die Wertung einbezieht.“

[29]

Die neuere Forschung entwickelte dementsprechend auch Metriken auf Objektbasis. Beispiele dafür sind die Kenngrößen RFC (Response for a Class) und WMC (Weighted Methodes for Class)

In der objektorientierten Programmierung werden Klassen gebildet, wobei unter einer Klasse die Definition eines Satzes von Variablen und Programmen, mit denen diese Variablen bearbeitet werden können, verstanden wird. Diese Programme werden Methoden genannt. Methoden können assoziiert sein. An einem Beispiel sei das veranschaulicht.

Die Variablen Punktnummer, Rechtswert und Hochwert bilden die Klasse PUNKT. In der Klasse Punkt kann die Methode „schreibePunktcoordinate“ programmiert werden, welche am Bildschirm die Punktnummer, die X-Koordinate und die Y-Koordinate in einer definierten Form ausgibt. Mit Hilfe der Klasse PUNKT lässt sich die Klasse STRECKE bilden. Eine Strecke wird aus einem Anfangs- und Endpunkt gebildet. Die Klasse STRECKE besteht also aus zwei Objekten der Klasse PUNKT. In der Klasse STRECKE sei die Methode „ermittleLänge“ abgelegt, welche mithilfe des pythagoräischen Lehrsatzes die Entfernung

von Anfangs- und Endpunkt ermittelt. Darüber hinaus steht in dieser Klasse aber auch die Methode „schreibePunktcoordinate“ zur Verfügung. Die Klassen sind assoziiert.

RFC gibt die Anzahl an ausführbaren Methoden an, wobei nicht nur die direkt aufrufbaren Methoden gezählt werden, sondern auch die über Assoziation erreichbaren Methoden. Man geht davon aus, dass ein Programm umso schwieriger zu verstehen ist, je mehr Methoden vorliegen.

WMC misst die McCabe-Komplexität einer Klasse und gilt als Maß für die Erweiterbarkeit einer Klasse.

#### 4.4.6. Maßzahlen der Zuverlässigkeit

Ludewig und Lichter schlagen als Maß für die Zuverlässigkeit die *Mean Time between Failure* (MTBF) vor:

MTBF = Betriebsdauer/Fehleranzahl.

Nicht jeder Fehler ist gleich bedeutend, sondern hängt wesentlich von seinem Schadenspotential ab. Dem entsprechend wird die Unzuverlässigkeit von Ludewig und Lichter als gewichtetes Mittel der Fehler pro Betriebsdauer berechnet, wobei die Schadenspotentiale als Gewichte herangezogen werden.

Unzuverlässigkeit

$$U = \frac{\sum k_i \cdot f_i}{t}$$

U	Unzuverlässigkeit
$k_i$	Kosten für den i-ten Fehler
$f_i$	Anzahl des Auftretens dieses Fehlers
t	Betriebsdauer

Die Zuverlässigkeit ergibt sich als der Kehrwert der Unzuverlässigkeit.

$$Z = \frac{1}{U}$$

U...	Unzuverlässigkeit
Z...	Zuverlässigkeit

Nun werden die Dimensionen eliminiert, indem die Zeit und die Kosten in Bezug zu fixen Werten gesetzt werden. Es wird als Zeit die tägliche Betriebsdauer eingeführt, indem die beobachtete Betriebsdauer in Stunden durch 8 Stunden für einen vollen Arbeitstag dividiert wird. Die mittlere tägliche Unzulässigkeit wird mit  $U'$  bezeichnet und in ein prozentuelles Verhältnis zu den täglichen Betriebskosten der Software gesetzt:

$$\bar{U} = \frac{U'}{B + U'}$$

- ... Unzuverlässigkeit pro Betriebstag in Prozent zu den Betriebskosten pro Tag
- ... Unzuverlässigkeit pro Tag
- ... Betriebskosten eines Tages

Ludewig und Lichter merken kritisch an, dass die Zuverlässigkeit steigt, wenn die Betriebskosten sich erhöhen. Arbeitet man etwa auf angemieteter Hardware und steigen die Preise dafür, ändert sich nichts an der Zuverlässigkeit der Software [12, S. 328].

Die Metrik hat jedoch auch andere Schwächen. Eine lange Betriebsdauer erhöht ebenfalls die Zuverlässigkeit. Arbeitet man auf einen langsamen Computer, wird die Zuverlässigkeit jedoch nicht erhöht.

Für die in dieser Arbeit geschriebene Software ist eine Metrik der Zuverlässigkeit, die von der Betriebsdauer abhängt, nicht einsetzbar. Dieses System ist dann zuverlässig, wenn die Prüfung und Verarbeitung von Plänen erfolgreich war, unabhängig von der Größe des Plans und damit von der Betriebsdauer. Es macht also Sinn, anstatt der Betriebsdauer die Anzahl der Jobs einzuführen.

Darüber hinaus sind für Anwender die Kosten bei einem Versagen im Allgemeinen konstant. Der Benutzer des Portals nimmt mit dem Portalbetreiber Kontakt auf, schildert das Problem, und der Fehler wird korrigiert. Fehler, welche die Aufgabenerfüllung nicht behindern, werden nicht störend wahrgenommen. Das sind etwa falsche Einträge in die interne Protokolldatei. Aber auch Rechtschreibfehler in den automatisch verschickten Mails, stören die Anwender kaum.

Damit wäre für diese Software ein ideales Zuverlässigkeitsmaß: „number of jobs between failure.“ In einer Formel ausgedrückt:

$$Z = \frac{J}{F}$$

Z... Zuverlässigkeit  
J... Anzahl an Jobs  
F... Anzahl an gravierenden Fehlern, die das gewünschte Endergebnis verhindern

Es liegt ein einfaches Maß für die Zuverlässigkeit vor, deren Eingangsgrößen leicht zu erfassen sind. Die meist geschätzte Größe Kosten tritt nicht auf. Den einzigen Spielraum bildet die Größe F. Welche Fehler als gravierend anzusehen sind, ist eine individuelle Entscheidung.

Empfohlen wird dieses Maß der Zuverlässigkeit für Software, die einen zu erreichenden Endzustand herstellt, der erfüllt wird oder auch nicht. Bei einem gravierenden Versagen entstehen immer die gleichen Kosten, da jeder Fehler zur Folge hat, dass die Aufgabe nicht erfüllt wird und die Erfüllung andersartig erfolgt. Es wird vorausgesetzt, dass die andersartige Erledigung immer gleichviel kostet.

Kritiker könnten einwerfen, dass es Sinn macht, mit den Fehlerkosten zu multiplizieren, da ein Unterschied besteht, ob die Software etwa die Uhrzeit am Computer anzeigt oder einen teuren Fertigungsprozess steuert. Funktioniert die Uhr nicht, ist sie unzuverlässig, und so ist es auch bei einer Maschinensteuerung. Aus dem Einsatzgebiet der Software folgt die Kostenabschätzung für die Unzuverlässigkeit. So wird man für einen teuren Fertigungsprozess eine höhere Zuverlässigkeit fordern als für eine möglicherweise nicht benötigte Uhr.

Natürlich muss die Software einige Zeit beobachtet werden und es muss zumindest ein Fehler auftreten, um diese Größe berechnen zu können.

Diese Norm unterscheidet sich kaum von der MTBF. Verwendet die MTBF die Betriebsdauer der Software, wird hier die Anzahl an Programmläufen zwischen zwei Fehlern herangezogen. Deshalb ist zu erwarten, dass damit die gleichen Anforderungen an die Metrik erfüllt sind. Nichtsdestotrotz ist das zu prüfen.

Die Zuverlässigkeit wurde erhalten, indem die Anzahl der Jobs durch die Anzahl der gravierenden Fehler dividiert wurde. Die Zuverlässigkeit beschreibt also die Anzahl der Jobs zwischen zwei Fehlern. Diese Metrik legt eine Reihenfolge fest (Ordinalskala). Eine Software, die hundert Jobs zwischen zwei Fehlern ausführt, ist schlechter als eine Software, die

1000 Jobs fehlerfrei schafft. Auf Ordinalskalen kann der Medianbildung sinnvoll angewandt werden. Darüber hinaus lässt sich der Abstand eines Intervalls sachlich begründen (Intervallskala). Misst man die Zuverlässigkeit zweier Programme, können die Messergebnisse miteinander durch Differenzbildung in Beziehung gesetzt werden. Man kann Aussagen treffen wie etwa jene: „Die Software A ist um 900 Jobs zuverlässiger als die Software B.“ Bei der Bildung eines arithmetischen Mittelwertes wird vorausgesetzt, dass gleichgroße Intervalle gleiche Mengen beinhalten. Das ist bei der Intervallskala und hier der Fall, weshalb auch eine Mittelwertbildung zulässig ist. Für die Ratio- oder auch Verhältnisskala wird ein absoluter Nullpunkt gebraucht. Es stellt sich also die Frage, wann Software absolut unzuverlässig ist. Dann soll ihr der Wert 0 zugewiesen werden.

Für Fehler werden nur ganze Zahlen verwendet, da es keine halben Fehler gibt. Um eine Division durch null zu vermeiden, muss zumindest ein Fehler vorliegen. Die Zuverlässigkeit wird 0, wenn das ohne einen einzigen Programmlauf erfolgt. Das ist unsinnig, da ein nicht exekutierte Programm keine Fehler macht. Das bedeutet im Umkehrschluss, dass jedes Programm eine gewisse Zuverlässigkeit aufweist.

Es liegt kein natürlicher Nullpunkt vor und damit auch keine Verhältnisskala. Mit einem gedanklichen Trick kann Abhilfe geschaffen werden.

Es wurde von gravierenden Fehlern ausgegangen, welche die Programmexekution beenden bzw. eine Nichterfüllung der Aufgabe zur Folge haben. Es ist irrelevant, ob ein oder mehrere Fehler zum Versagen führen. Eine Software hat also nur einen zählbaren schweren Fehler pro Job. Versagt eine Software immer, würde die Anzahl der Jobs jenen der zählbaren Fehler entsprechen. Da gilt:

$$Z = \frac{J}{F}$$

Z      Zuverlässigkeit

J      Anzahl an Jobs

F      Anzahl an gravierenden Fehlern, die das gewünschte Endergebnis verhindern

Z ist 1, wenn  $J = F$  gilt. Absolut unzuverlässige Software hat somit die Größe 1. Natürlich wäre es schön, wenn absolute Unzuverlässigkeit mit der Zahl 0 verbunden wäre. Wir subtrahieren 1 und erhalten:

$$Z' = \frac{J}{F} - 1$$

Die Zuverlässigkeit  $Z'$  stellt nun auch eine Verhältnisskala dar, wo geometrische Mittelwertbildungen möglich sind bzw. Aussagen wie „Die Software A ist doppelt so zuverlässig wie die Software B“.

#### 4.5. Qualität von Software

„Unter Softwarequalität versteht man die Gesamtheit der Merkmale und Merkmalswerte eines Softwareprodukts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“ [30, S. 257]

Softwarequalität sagt also aus, wie weit die meist zu Beginn niedergeschriebenen Anforderungen erfüllt wurden, und ergänzt diese Festlegungen noch um selbstverständliche Gegebenheiten. Um eine Beurteilung von Qualität vornehmen zu können, gibt es Normen. Ein Beispiel ist die Normenreihe ISO/IEC 250xx „Software-Engineering – Qualitätskriterien und Bewertung von Softwareprodukten“, mit deren Hilfe die Qualitäten verschiedener Softwareprodukte verglichen werden können. Dabei werden Qualitätsmerkmale (factors) mit ihren Untergliederungen in Teilmerkmale (criteria) unterschieden.

Ludewig und Lichter stellen ein einfacheres System zur Qualitätsbewertung vor, welches ursprünglich von M. Azuma der Waseda-Universität in Tokyo stammt. Dabei werden die Qualitätskriterien Robustheit, Korrektheit, Zuverlässigkeit, Wartbarkeit, Flexibilität, Gebrauchstauglichkeit, Effizienz und Sicherheit bewertet [12, SS. 307–308]. Die Bewertung erfolgt durch Gutachter, die Prozentwerte für das Erreichen der Kriterien vergeben. Das Verfahren wird von der *Nippon Electronic Company* (NEC) eingesetzt.

J. Jürjens stellt in seiner Vorlesung über Softwarekonstruktion an der TU Dortmund fest, dass „Verlässlichkeit / Korrektheit / Zuverlässigkeit / Sicherheit / Robustheit – Performanz – Benutzungsfreundlichkeit, Gebrauchstauglichkeit (Usability) – Wartbarkeit – Wiederverwendbarkeit – Portierbarkeit – Interoperabilität“ durch Software-Engineering sicherzustellende Eigenschaften sind [31]. Es handelt sich also um wesentliche Eigenschaften, die Software erfüllen soll, wobei Verlässlichkeit, Korrektheit, Zuverlässigkeit, Sicherheit und Robustheit Gemeinsamkeiten haben.

Will man guten Code schreiben, sollte man die Qualitätseigenschaften immer im Auge behalten. In weiterer Folge sollen diese Qualitätskriterien erläutert werden. Maßnahmen zur Qualitätssicherung werden bei den einzelnen Kriterien erläutert.

#### 4.5.1. Korrektheit

Korrektheit ist gegeben, wenn Software die spezifizierten Anforderungen erfüllt. Die Anforderungen liegen meist in Form eines Schriftstückes vor und die Software sollte sich so verhalten, wie es festgelegt wurde. Es besteht also eine direkte Beziehung zwischen der Software und der Anforderung. Das Ausmaß der Entsprechung wird Korrektheit genannt. Es ist nicht nur schwierig, korrekte Software zu programmieren, sondern auch herausfordernd, eine vollständige korrekte Dokumentation, die das Verhalten beschreibt, zu erstellen. Diese Eigenschaft gilt als die wichtigste Qualität, die Software erfüllen sollte. Ein Programm muss richtig rechnen und auf Anfragen die gewünschten Ergebnisse retournieren. Die Softwarebeschreibung wird von Anwendern nicht immer gelesen. Das kann zur Folge haben, dass Software als inkorrekt angesehen wird, obwohl sie entsprechend dieser Definition korrekt arbeitet.

Ein Fehler in der Software sollte nicht korrigiert werden, indem die Spezifikation umgeschrieben wird.

##### 4.5.1.1. Vermeidung von Komplexität

Das Auffinden und Beseitigen von Fehlern ist eine aufwendige Angelegenheit. Deshalb sollten Fehler von Beginn an vermieden werden. Menschen sind in der Lage einfache Probleme nahezu fehlerfrei zu bewältigen. Bei komplexeren Problemstellungen nimmt die Fehlerwahrscheinlichkeit stark zu.

Die agile Technik „KISS“ (keep it simple, stupid) ist also essenziell für Korrektheit. Grob gesprochen ist ein Programm dann einfach, wenn der Output eine möglichst direkte Konsequenz des Inputs darstellt. Hat man mehrere Möglichkeiten, eine Methode oder Funktion zu programmieren, lässt sich etwa mit der Halstead-Metrik Difficult Level sagen, welcher

Weg im Sinn des KISS sinnvoller und besser ist. Im Kapitel 4.4 werden einige Metriken vorgestellt, die sinnvoll verwendet werden können.

Betrachtet man seinen eigenen Code und sucht nach Schwachstellen, sollte man die Kommentarzeilen im Blickfeld haben. Haben Methoden lange Erläuterungen nötig, um sie später warten zu können, kann von komplizierten Strukturen ausgegangen werden. Im Sinn der Korrektheit sollten diese Teile neu überarbeitet werden. Diese Methode ist in der Literatur nicht zu finden. Das dürfte darin begründet liegen, dass Entwickler vermehrt auf die Dokumentation ihrer Routinen verzichten, wenn Dritte auf diese Weise Software bewerten. Es sei deshalb nochmals darauf hingewiesen, dass dieser Weg nur für die Eigenkontrolle sinnvoll ist.

X. Culmann und K. Lambertz geben an, dass zumindest 30 % aller Zeilen Kommentare sein sollten. Machen die Kommentarzeilen mehr als 75 % des Quellcodes aus, ist die Datei als Dokumentation anzusehen [27]. Gleicht der Code einer Dokumentation, sollten die im vorherigen Absatz angestellten Überlegungen zur Revision führen.

#### *4.5.1.1.1. Hilfsmittel zur Vermeidung der Komplexität*

Mit Metriken kann man versuchen, die Qualität der Software zu messen. Lines of Code (mit Kommentaren, ohne Kommentare, mit Leerzeilen, ohne Leerzeilen) sind die ältesten und beliebtesten Metriken. Man sollte diese Metrik nicht überstrapazieren, da sonst der Quellcode auf die Erfüllung einer Norm getrimmt wird und das eigentliche Ziel von guter Software aus den Augen verloren wird. Ein sehr langes Programm kann sinnfrei in viele Unterprogramme gespalten werden, indem der Code in Blöcken von ca. 30 Zeilen gegliedert wird. Gehen wir jedoch von ehrlichem Bemühen um guten Code aus, so ist die Anzahl an Zeilen sicherlich ein erster Hinweis auf die Qualität des Softwareentwurfs.

„Dateien, die länger als 400 Programmzeilen sind (10–100 Funktionen) sind meistens zu lang, um als Ganzes verstanden zu werden.“ [27].

Eine Funktion oder ein Modul sollte aus nicht mehr als 40 Zeilen bestehen und zumindest ein Drittel sollten Kommentarzeilen sein, wie bereits im vorherigen Kapitel festgestellt

wurde. Leerzeilen erhöhen die Lesegeschwindigkeit des Codes und betonen die Strukturierung, die jedoch schon durch Einrückungen gegeben ist. Es ist sehr praktisch, die Funktion oder das Modul auf einer Bildschirmseite zu sehen, was zentraler scheint, als ein Prozentsatz an Leerzeilen. Manche Entwickler stellen daher ihre Bildschirme im Hochformat auf, was natürlich betriebssystemseitig berücksichtigt werden muss.

Gehen wir davon aus, dass in einer Datei eine Klasse mit ihren Methoden und Funktionen abgelegt wird und im Mittel eine Methode 30 Zeilen benötigt, führt die Forderung nach Dateilängen von weniger als 400 Programmzeilen zu ca. 13 Methoden pro Klasse. Diese Überlegung bezieht sich auf LOC inklusive Kommentaren. 13 Methoden pro Klasse wurden im gegenständlichen Projekt nie überschritten. Wichtig ist, dass das UML-Diagramm bei Abänderungen des Planes sofort adaptiert wird, um den Überblick über die Dateien (Klassen) und Methoden zu behalten.

Liegt eine hohe Anzahl von LOCs vor, sollte der Softwareentwurf neu überarbeitet werden. Kleinere Einheiten sind auch leichter zu testen, die Wahrscheinlichkeit der Wiederverwertbarkeit an einer anderen Stelle steigt, was ebenfalls ein Qualitätskriterium darstellt.

Bei den bisherigen Überlegungen handelt sich um reine Mengenangaben. Die Komplexität durch verschachtelte Strukturen bleibt unberücksichtigt. Sie kann mit der McCabe-Metrik gemessen werden.

Eine profunde Einschätzung der McCabe-Komplexität und des Risikos eines Fehlers im Programm liefert Wave [24].

Tabelle 3: zyklomatische Komplexität und Fehlerrisiko [24]

<b>„Cyclomatic Complexity</b>	<b>Risk Evaluation ...</b>
1–10	A simple module without much risk
11–20	A more complex module with moderate risk
21–50	A complex module of high risk
51 and greater	An untestable program of very high risk “

Noch wurde die Objektstruktur nicht berücksichtigt. Dafür liegen eigene Metriken vor. RFC misst die Anzahl an Methoden und assoziierten Methoden, wie im Kapitel 4.4.5 aufgeführt wurde.

Die McCabe-Komplexität auf Methodenebene und die RFC-Methode auf Objektebene bilden eine gute Basis, um abzuschätzen, ob das Prinzip „keep it simple, stupid“ eingehalten wird. Bei agilen Prozessen wird iterativ und inkrementell vorgegangen. Die Software zerfleddert und es erfolgt ein Refactoring. Das bedeutet, dass die Software auf Basis des alten Codes überarbeitet wird. Diese Kennzahlen können für das alte und neue Programm ermittelt werden und geben Aufschluss über den Erfolg des Refactoring.

Die Literaturangaben für akzeptable Werte sind unterschiedlich und liegen für die McCabe-Komplexität zwischen 7 und 15. McCabe empfiehlt:

„When the complexity exceeded 10 they had to either recognize and modularize subfunctions or redo the software.” [32]

Steitzer geht von fehlerhafter Software ab einem RFC > 40 aus und weist jedoch gleichzeitig auf die Problematik hin, dass mit dieser Grenze ca. zwei Drittel aller Beanstandungen unbegründet sind [33, S. 46].

Natürlich lässt sich auch mit dem Schwierigkeitsgrad (*difficult level*) D nach M. H. Halstead arbeiten, der die Fehleranfälligkeit und damit die Korrektheit beschreibt [25]. Die Komplexität der Objektstruktur bleibt unberücksichtigt, da nur die Codelänge und der ausgenutzte Sprachumfang eingehen.

#### 4.5.1.1.2. Prinzipielle Ansätze zur Komplexitätsbewältigung

Komplexität und ihre Reduktion sind zentrale Punkte der Softwareentwicklung. Zu Beginn der EDV waren die Programme kurz und einfach. Dann wurden auch komplexere Fragestellungen mit Software gelöst und damit stieg die Komplexität. Das ist gleichbedeutend mit einer höheren Fehlerrate. Strategien zur Komplexitätsbewältigung (Fehlervermeidung) wurden entwickelt.

Die strukturierte Programmierung der 1970 Jahre war eine Antwort auf diese neuen Schwierigkeiten. Der in dieser Arbeit schon mehrfach zitierte Dijkstra war Wegbereiter dieser Bewegung, welche sich hauptsächlich gegen die GOTO-Anweisung wandte, mit deren Hilfe immer wieder aus sequenziell durchgeführten Programmen gesprungen wurde. Häufig war der Code aufgrund der vielen Sprünge nicht mehr lesbar und wurde deshalb als „Spaghetticode“ bezeichnet. Jede Nudel bildet einen Pfad. Es liegt eine extrem hohe McCabe-Komplexität vor. Dem stellte diese Bewegung die Aufsplitterung des Codes in kürzere Prozeduren entgegen. Insgesamt gab und gibt es drei Konstruktionselemente:

- 1) Sequenzen (Abfolge von nacheinander ausgeführten Programmen);
- 2) Verzweigungen (if then else, switch);
- 3) Iteratoren (while, for ...).

Wurde ein Codeteil mehrfach verwendet, sprang man also nicht mehr mit einer GOTO-Anweisung an diese Stelle, sondern ein Unterprogramm oder eine Funktion mit wohl definierter Aufgabe wurde aufgerufen. Globale Variablen gelten im gesamten Programm und lokale Variablen stehen im Namensraum des Unterprogramms bereit. All diese Prinzipien haben nichts von ihrer Bedeutung verloren und sind nach wie vor wesentliche Bestandteile, um Software überschaubar und einfach zu halten.

In den 1980 Jahren wurden die ersten objektorientierten Systeme entwickelt. Sie waren die Antwort auf einen Komplexitätsschub, der durch grafische Oberflächen gegeben ist. Bis heute bilden sie den Stand der Technik und auch diese Software wurde nach objektorientierten Grundsätzen entwickelt.

Attribute werden sinnhaft zusammengefasst und jedes Bündel an Attributen verfügt über verschiedene Methoden, die auf diese Attribute angewandt werden können. Diese Einheit wird Objekt bezeichnet. Mehrere gleichartige Objekte bilden eine Klasse. Ein Objekt wird als Instanz einer Klasse angesehen. Klassen können Objekte anderer Klassen verwenden. Das wurde im Kapitel 4.4.5 am Beispiel der Klasse PUNKT und STRECKE erläutert. Daten und Programme für diese Daten – sogenannte Methoden – bilden also eine Einheit, was die wesentliche Neuerung ausmacht. Erhält eine Klasse ein Objekt einer anderen Klasse, stehen dort nicht nur die Variableninhalte des Objektes zur Verfügung, sondern auch all die Methoden dieses Objektes. Für das im Kapitel 4.4.5 gegebene Beispiel mit den Klassen

PUNKT und STRECKE bedeutet es, dass die Klasse STRECKE auch die Methode `schreibePunkt` der Klasse PUNKT kennt, da sie auf dieser Klasse aufbaut. Sie hat dessen Eigenschaften geerbt.

Objektorientierte Programme, wie das hier entwickelte, bestehen aus vielen Klassen mit vielen Methoden, die miteinander kommunizieren. Die durch das ROC gemessene Komplexität auf Klassenebene bringt zum Ausdruck, dass es schwierig ist, sich all diese Methoden zu merken und anzuwenden.

Fast immer wird nicht nur mit eigenen Methoden gearbeitet, sondern auch mit Methoden und Objekten, die über Bibliotheken importiert werden. In diesem speziellen Fall wurde hauptsächlich auf Methoden der Firma ESRI Inc. zurückgegriffen, sie wurden als Basis der Arbeit verwendet. Dieser größte Hersteller von Basissoftware im GIS-Bereich bietet mit dem Produkt ArcGIS Desktop eine grafische Oberfläche an, mit der Karten gezeichnet und analysiert werden können. Integraler Bestandteil der Oberfläche ist ein Modelbuilder, mit dem die Verarbeitungsschritte von Geodaten grafisch aneinandergesetzt werden können. Einfache Programme werden so geschrieben, ohne sich über den exakten Methodenaufruf Gedanken machen zu müssen (Case tool). Das grafisch erstellte Programm lässt sich als Pythonroutine exportieren und nachbehandeln. Das gestattet eine verhältnismäßig schnelle Programmierung von Einzelteilen. Die Methoden stehen mit Masken und erläuternden Hilfetexten (Toolbox) bereit.

Der Modelbuilder, mit seiner Möglichkeit Source-code zu exportieren, stellen eine wesentliche Reduktion der Komplexität dar und führt zu einer Kapazitätssteigerung beim Codieren. Kurze Sequenzen ohne Verzweigungen lassen sich so rasch erledigen. Der exportierte Pythoncode ist nachzubehandeln und in dem eigentlichen Programmcode zu integrieren.

Es ist von großem Vorteil, wenn man die verschiedenen Methoden von ESRI kennt bzw. vor der Programmierung einer Fragestellung nach einer entsprechenden Methode sucht. Für nahezu alle Fragestellungen im GI-Bereich wurden Methoden von ESRI entwickelt. Leider sieht man immer wieder mühevoll geschriebene Programme, die vom Basissystem ebenfalls bereitgestellt werden. Darüber hinaus ist die Wahrscheinlichkeit, dass die bereitgestellte Methode korrekter abläuft, höher als eine Nachprogrammierung. Die Methoden des Basissystems wurden schon von vielen tausenden Usern benutzt, welche Fehlverhalten reklamieren.

Hat man ein Basissystem im Einsatz, sollte es weitestgehend genutzt werden, damit sich die Investition lohnt. Das Land Steiermark verfügt über ausreichend Lizenzen, sodass niemanden durch diese Software eine Lizenz weggenommen wird. Das gilt auch für die Lizenzstufe „advanced“ mit dem höchsten Funktionalitätsumfang.

Neue Ansätze können ebenfalls die Komplexität reduzieren. Einer dieser Ansätze sind Agenten. Im Folgenden soll das Grundprinzip erläutert werden.

Jedes Individuum eines Ameisenstaates hat nur eingeschränkte Möglichkeiten. Der Ameisenstaat ist jedoch in der Lage, komplexe Aufgaben zu bewältigen. Das Individuum stellt einen Agenten dar, der mit anderen Agenten Botschaften austauscht und interagiert. Trotz bescheidener Fähigkeiten eines Individuums ergeben sich in der Summe Lösungen für komplexe Fragestellungen. In Anlehnung an die Biologie wird von Schwarmintelligenz gesprochen.

Die Wirkung von Agenten lässt sich gut an dem Beispiel zeigen, wie Ameisen den kürzesten Weg zur Nahrungsquelle finden. Jede Ameise – also jeder Agent – sondert während des Gehens Duftstoffe ab und ist auch in der Lage, die abgesonderten Düfte anderer Ameisen wahrzunehmen. Nun starten eine Menge Ameisen in alle Richtungen auf der Suche nach Nahrung. Hat die Ameise eine Nahrungsquelle gefunden kehrt sie entlang ihrer Duftspur ins Nest zurück. Während des Rückwegs sondert sie weitere Duftstoffe ab, sodass die Duftspur nun doppelt so stark riecht. Die Ameise, die den kürzesten Weg gefunden hat, wird zuerst wieder im Nest sein. Der nun doppelt so starke Duft veranlasst viele Ameisen, diesen Weg zu gehen, denn Ameisen bevorzugen stark duftende Wege. So entsteht eine Ameisenstraße zur nächsten Nahrungsquelle.

Bietet man zuerst nur eine lange Strecke an, so werden die Ameisen auf dieser Straße laufen. Eröffnet man danach eine kurze Strecke, wird sie nicht gefunden. Digitale Nachbildungen dieses Systems können das Problem meistern, indem der Duftstoff verdunstet. Es wird ein neuer Weg gesucht und der Algorithmus konvergiert nicht zu schlechten Lösungen, selbst wenn diese verspätet angeboten werden [34].

„As such, a key notion is that simple behavioral rules generate complex behavior. This principle, known as K.I.S.S (‘Keep it simple, stupid’) is extensively adopted in the modelling community“ [35].

Agentenbasierte Modelle bieten die Möglichkeit, Komplexität in der Programmierung zu reduzieren. Es werden wenige Verhaltensregeln programmiert und damit sinkt die Wahrscheinlichkeit von Fehlern. Da das gewünschte Verhalten indirekt kodiert wird, ist nicht sichergestellt, dass das System in jedem Fall wie gewünscht reagiert. Die Problematik der korrekten Programmierung wird ersetzt mit dem Problem, die richtigen Regeln zu finden.

Nichtsdestotrotz ergibt sich mit Agenten eine Vielzahl neuer Möglichkeiten, die ohne diesen Ansatz kaum modellierbar wären. So kann der Autoverkehr in Stoßzeiten simuliert werden und der Verkehrsfluss bei Unfällen beobachtet werden.

Agentenbasierte Modelle lassen sich mit der Agent Analyst Extension von ArcGIS modellieren. Dabei wird der Open-Source-Toolkit Repast verwendet.<sup>16</sup>

#### 4.5.2. Robustheit

Jan Jürjens definiert Robustheit als Toleranz gegenüber nicht spezifizierten Bedingungen bzw. Rahmenbedingungen [31]. Software soll ein sinnvolles Verhalten in Ausnahmesituationen an den Tag legen.

Wird eine Problemsituation vorhergesehen, findet sie im Allgemeinen Niederschlag in der Programmierung und wird als Korrektheit bezeichnet, was im Kapitel 4.5.1 behandelt wurde. Bei Robustheit geht es um alles, was nicht bedacht ist.

Im Allgemeinen wird der Begriff jedoch nicht so streng ausgelegt. Wir sprechen hier von Robustheit bei sinnvollem Verhalten auf unübliche Bedingungen. Beispiele sind das Wegbrechen von externem Speicherplatz, die Nichterreichbarkeit von Lizenzservern etc.

Bei der Analyse bereits abgegebener Pläne wurden Abstürze festgestellt und Ursachen gesucht. Es ergab sich eine Liste, die universellen Charakter hat, und man sollte sich bei jeder größeren Software fragen, ob es sinnvoll ist, diese Punkte zu berücksichtigen:

1. Verhalten bei Programmierfehlern;

---

<sup>16</sup> Die entwickelte Software ist ohne Agenten ausgekommen. Das Konzept wurde vorgestellt, um einen vollständigen Blick auf die Bewältigung von Komplexität zu gewähren.

2. Verhalten bei einem Neustart, wenn zuvor ein Absturz erfolgte;
3. ein notwendiger Server wird abgeschaltet;
4. die Datenbank wird abgeschaltet;
5. notwendige Lizenzen stehen nicht zur Verfügung.

Ad1) Python bietet mit der Funktion

```
try:
    Befehl
except:
    Befehl
```

ein machtvolles Werkzeug, um Ausnahmesituationen zu behandeln. Es wird der Funktionszweig `try` abgearbeitet. Kommt es dabei zu einer Ausnahmesituation, die einen Programmabbruch zur Folge hätte, wird anstatt abubrechen der Zweig `except` durchgeführt. Stellt man das Hauptprogramm und kritische Passagen unter die `try-except`-Funktion, kann auf den Fehler reagiert werden.

Als Ausnahmebehandlung bietet sich ein Eintrag in ein Protokollfile an. Um Softwarefehler schnell zu finden, sollten auch die Systemmeldungen der verwendeten Basissoftware<sup>17</sup> mitgespeichert werden. So erhält man Auskunft über die Problemstellung, die sich nicht immer durch einen Neustart mit der gleichen Eingabe reproduzieren lässt. Die entwickelte Software versendet automatisch E-Mails. Diese Funktionalität kann auch für Ausnahmesituationen genutzt werden, was für die nächsten Erweiterungen angedacht ist.

Ad2) Vor dem Verlassen einer Methode werden alle in der Methode angelegten Hilfsdateien, temporär benötigte Featureklassen etc. gelöscht. Kommt es zu einen unerwarteten Absturz innerhalb dieser Routine, etwa durch einen Stromausfall oder einen Programmierfehler, bleiben diese Hilfsgrößen bestehen und blockieren einen neuerlichen Durchlauf. Es ist also essenziell, am Beginn der Methode ebenfalls zu kontrollieren, ob diese temporären Elemente existieren, und sie gegebenenfalls zu löschen.

---

<sup>17</sup> arcpy-Bibliothek von ESRI

Diese Vorgehensweise ist sinnvoll, wenn die Software nur hintereinander gestartet wird. Die im Rahmen dieser Arbeit realisierte Software wird alle 10 Minuten gestartet und verarbeitet die eingelangten Pläne. Die Verarbeitung großer Pläne kann über 10 Minuten dauern und ein zweiter Prüf- und Verarbeitungsjob wird gestartet. Die Software muss also gleichzeitig an mehreren Problemen arbeiten können. Das tritt beim Upload von Plänen extrem selten auf. Bei Internetanwendungen ist das eine sehr häufige Anforderung. Im Rahmen eines anderen Projektes wurde ein Programm entwickelt, wo in einem WebMap-Service ein Zentralpunkt angegeben wird und das Bevölkerungsprofil innerhalb eines „Service Areas“ des Straßennetzes ermittelt wird. Diese Funktion wird gleichzeitig von vielen Beamten genutzt, und es ist sicherzustellen, dass Hilfsgeodatenätze sich gegenseitig nicht blockieren. Vor dem Anlegen von einer temporären Featureclass etc. ist also nachzusehen, ob diese schon existiert, und es ist gegebenenfalls eine Featureclass mit einem anderen Namen anzulegen. Es bietet sich an, die notwendigen Hilfsgrößen mit fortlaufenden Nummern hochzuzählen.

Man kann die Multiuserfähigkeit auch herstellen, wenn der Prozess zu Beginn einen neuen Arbeitsbereich anlegt und konsequent nur in diesem Bereich gearbeitet wird. Existiert bereits ein solcher Arbeitsbereich, wird mit einer Versionsnummer hochgezählt. Damit kann das Problem an einer einzigen Stelle gebündelt gelöst werden.

Multiuserfähigkeit macht Software generell robuster, da Programmreste eines alten Durchgangs die Exekution bei einem neuerlichen Aufruf nicht stören. Legt man Wert auf Zuverlässigkeit, wird man die Software multiuserfähig entwickeln, auch wenn diese Eigenschaft für den Standardbetrieb nicht benötigt wird.

Ad 3) Die Analyse der alten Software hat ergeben, dass es in 1 % der Programmexekutionen zu Abstürzen gekommen ist. Ursache für dieses extrem hohe Ausfallrisiko sind einerseits Programmierfehler und zum anderen die verteilte Softwarearchitektur, die seit geraumer Zeit, einem Spargedanken folgend, modern ist. Jeder Server erledigt eine Aufgabe für viele verschiedene Anwendungen. Fällt ein Server aus, kann die Software nicht mehr ihren Aufgaben nachkommen und es kommt zum Absturz des Gesamtsystems. Im Anhang 4.1 werden die beteiligten Server und ihre Aufgaben aufgezählt.

Die Ursachen für den Ausfall von Servern liegen einerseits in Netzwerkproblemen und andererseits in unangekündigten Wartungsarbeiten. Rechner werden gebootet mit dem Hintergedanken, dass derzeit wahrscheinlich niemand den Server benutzt oder in Kürze benutzen wird. Es liegt die Vermutung nahe, dass von der Systemverwaltung Systemarbeiten als prioritär angesehen werden und das Terminieren anderer Anwendungen dafür in Kauf genommen wird.<sup>18</sup> Der User beurteilt jedoch immer, die sich ihm zeigende Anwendung, so dass dieses Vorgehen als extrem kontraproduktiv anzusehen ist.

Die unter ad 1 beschriebene Möglichkeit, Programmabstürze aufgrund von Programmierfehlern zu kanalisieren, greift auch bei Serverabschaltungen.

Ad 4) Steht eine Datenbank plötzlich nicht mehr zur Verfügung, hat die Software darauf zu reagieren. Die Datenbank exekutiert in diesem Zusammenhang ein Rollback und der alte Zustand tritt wieder ein. Die Software, welche die Datenbank anspricht, determiniert sinnvoll mithilfe des „Try-except“-Konstrukts. Steht die Datenbank wieder bereit, wird der gesamte Vorgang wiederholt. Probleme dieser Art sind nicht aufgetreten.

Ad 5) Für die Durchführung des Programms wird die Lizenzierungsstufe „advanced“ der Basissoftware ArcGIS-Desktop benötigt. Diese Lizenzen stehen im Amt der Stmk LREG im Floating-Betrieb zur Verfügung. Das bedeutet, dass bis zu 10 unterschiedliche Rechner (exakt Prozessoren) gleichzeitig die Software nutzen dürfen. Rufen mehr Prozessoren ArcGIS mit der Lizenzierungsstufe „advanced“ auf, können die ArcGIS-spezifischen Funktionen nicht ausgeführt werden. Die Software prüft, ob das Auschecken der Lizenz erfolgreich war. Ist das nicht der Fall, wird 10 Minuten gewartet und ein neuerlicher Versuch erfolgt. Wenn es aus betriebswirtschaftlichen Gründen nicht möglich ist einer Anwendung eine permanente Lizenz bereitzustellen, dann ist diese Vorgehensweise zielführend.

Es macht Sinn, zu protokollieren, wie häufig es zu Problemen dieser Art kommt. Man hat dadurch die Möglichkeit, auf sicheres Datenmaterial aufzubauen, mit dem man abschätzen kann, ob eine fixe Lizenz für diese Anwendung sinnvoll ist.

---

<sup>18</sup> Das trifft für Server im Bereich des GIS-Steiermark und damit meinem Einflussbereich zu. Server der allgemeinen EDV stehen mit hoher Zuverlässigkeit bereit. Im GIS – Bereich werden diese Regeln der allgemeinen EDV weniger streng beachtet als in allgemeinen EDV Bereichen.

Im gegenständlichen Fall hat sich das Lizenzproblem als nicht relevant herausgestellt, da nahezu immer ausreichend Lizenzen bereitstehen. Ist das einmal nicht der Fall, erfolgen die Datenkontrolle, die Korrektur und die Einbringung etwas später. Bei persönlichen Gesprächen mit Anwendern wurde das nie als Problem genannt.

Besteht nicht die Möglichkeit, Programmteile, die begehrte Lizenzen benötigen, in betriebsschwache Zeiten zu verlegen, ist abzuschätzen, ob es wirtschaftlich Sinn macht, die Funktionalität nachzuprogrammieren bzw. auf freie Lizenzen<sup>19</sup> zurückzugreifen.

#### 4.5.3. Zuverlässigkeit

Es besteht eine Nahebeziehung zwischen Korrektheit und Zuverlässigkeit. Jan Jürjens definiert Zuverlässigkeit als die dauerhafte Einsetzbarkeit einer Software, wobei sich der Anwender bei einer hohen Zuverlässigkeit erlauben kann, von der Software abhängig zu sein [31, S. 21]. Erfüllt Software die Spezifikation, wird von Korrektheit gesprochen. Ist eine Software korrekt, zeigt vernünftiges Verhalten jenseits der Spezifikation und kann diese auch dauerhaft und immer wieder erbringen, ist sie auch zuverlässig.

Benutzer lesen nur selten Spezifikationen. Zuverlässige Software erfüllt erwartete Wesensmerkmale, auch wenn diese nicht schriftlich dokumentiert sind. Die Spezifikation sollte sich auch niemals von erwarteten Wesensmerkmalen unterscheiden, da die Benutzbarkeit (usability) einer Software darunter stark leidet.

Zuverlässigkeit steht in einem direkten Zusammenhang mit dem Schadenspotenzial einer Software. Ein Programm, welches die Bremsen eines Fahrzeuges steuert, wird zuverlässiger arbeiten müssen und den Kundenerwartungen auch jenseits von Spezifikationen in einem höheren Grad entsprechen müssen als etwa ein Textverarbeitungsprogramm, wo nach alternativen Möglichkeiten gesucht werden kann, wenn etwa das Zentrieren einer Textstelle nicht funktioniert.

Hohe Zuverlässigkeit kann erreicht werden, indem zuerst versucht wird, Korrektheit herzustellen. Danach ist die Software umfangreichen Black-Tests zu unterziehen. Unter Black-

---

<sup>19</sup> Z. B. Qantum GIS [www.qgis.org](http://www.qgis.org)

Tests werden Tests des Gesamtsystems verstanden. In unserem Fall werden ganze Flächenwidmungspläne hochgeladen und getestet. Aber nicht nur Flächenwidmungspläne, sondern auch andere nicht spezifizierte Pläne und Musikdateien werden der Prüf- und Verwaltungsroutine unterzogen. Es macht Sinn, diese Prüffälle in eine Routine zu packen, sodass diese standardisierte Prüfung des Gesamtsystems auch bei kleineren Änderungen durchgeführt werden kann. Prüfungen des Gesamtsystems werden idealerweise nicht vom Entwickler durchgeführt, da der Entwickler die Schwächen des Systems mit seinen Tests tendenziell umschiffert. Das Verhalten jenseits der Dokumentation wird vom Entwickler intuitiv als verständlich angesehen, da er es sonst nicht so programmiert hätte. Ein Fremder sieht das möglicherweise ganz anders. Darüber hinaus tendieren Entwickler dazu, ihr Programm zu verteidigen, und sind gegenüber Schwächen uneinsichtig.

Arbeitet man alleine an einem Softwareprojekt, sollte man bewusst die Rolle wechseln, was viel Selbstdisziplin benötigt. Für jede gefundene Schwachstelle sollte man sich eine Belohnung versprechen. Man sollte sich die Ergebnisse ansehen und überlegen, ob man das auch anders sehen und interpretieren könnte.<sup>20</sup>

Natürlich kann man auch versuchen, die Zuverlässigkeit zu messen. Der Entwickler wird dann versuchen, im Rahmen der bereitstehenden Ressourcen, wie verfügbare Entwicklungszeit, Testumgebung etc., diese Maßzahl gut zu erfüllen. Bei Hardware verwendet man häufig die Mean Time between Failure (MTBF). Ludewig und Lichter schlagen dieses Maß auch für Software vor [12 SS. 326-328]. Es wurde im Kapitel 4.4.6 Maßzahlen der Zuverlässigkeit bereits besprochen.

Die MTBF kann erst nach Inbetriebnahme der Software bestimmt werden und es müssen mehrere Fehler auftreten, um eine valide Aussage treffen zu können. Der Fehler wird nicht sofort ausgebessert. In Fall dieser Software ist das nicht gegeben, da jedes Fehlverhalten unverzüglich bereinigt wurde. Ein Maß für die Zuverlässigkeit kann also nicht eruiert werden. Da jedoch ein starker Zusammenhang mit der Korrektheit gegeben ist, werden weite Teile der Zuverlässigkeit durch die Korrektheit und ihre Maße abgedeckt.

---

<sup>20</sup> Bei der Entwicklung dieser Software war es notwendig, die Rollen zu wechseln, da kein eigener Tester bereitstand.

Das im Rahmen der Dissertation entwickelte Programm ist nun bereits über ein Jahr im Einsatz und es wurden fünf Ausfälle registriert, wobei zwei Ausfälle auf das Konto einer irrtümlichen Abschaltung der Software gehen.

#### 4.5.4. Effizienz

##### 4.5.4.1. Allgemeines

Unter Effizienz wird der sparsame Umgang mit Ressourcen verstanden, wobei das Leistungsniveau der Software in einem möglichst günstigen Verhältnis zu den eingesetzten Betriebsmitteln stehen soll. Das sind zum einen das Zeitverhalten und zum anderen das Verbrauchsverhalten.

Das Zeitverhalten ist als gut zu bewerten, wenn ein hoher Durchsatz von der Software erreicht wird. Darunter wird verstanden, wie viele Jobs in einer Zeiteinheit bewältigt werden können. Das Zeitverhalten kann aber auch an der Antwortzeit bzw. Berechnungszeit gemessen werden. Antwort- und Berechnungszeiten können stark differieren. Unter Berechnungszeit (calculation period) kann die benötigte Zeit vom Start der Software mit ihren Übergabeparametern bis zur Bereitstellung der Ergebnisse angesehen werden. Als Antwortzeit (response time) wird in der Regel die Zeitspanne zwischen Absenden der Anfrage und Erhalt der Antwort verstanden. Software mit einer kurzen Berechnungszeit kann trotzdem eine hohe Antwortzeit aufweisen, denn die Antwortzeiten werden wesentlich durch Signallaufzeiten in den Übertragungsmedien und den Internetworking-Komponenten wie Switches und Router bestimmt [36].

Neben dem Zeitverhalten umfasst das Verbrauchsverhalten auch den notwendigen Speicherplatz auf Festplatten und in Datenbanken sowie den benötigten Hauptspeicher und die beanspruchte elektrische Energie. Der Stromverbrauch ist essenziell, wenn Handy-Apps, welche die Batterie als Energiequelle nutzen, in Einsatz kommen. In diesem Projekt spielte das keine Rolle.

Eine genaue Definition des Verbrauchsverhaltens kann unter ISO/IEC 9126 nachgelesen werden.

Komplexe Algorithmen sind im Einsatz, um große Datenmengen performant zu bearbeiten. Setzt man diese Programme auch für kleine Datenmengen ein, können die Programme kontraproduktiv sein und langsamer arbeiten, als eine einfache Problemlösungsstrategie. Aufwendige Software bietet daher verschiedene Vorgehensweisen an und wählt in Abhängigkeit der vorliegenden Datenmenge den optimalen Weg. Möchte man aus wirtschaftlichen Gründen nicht mehrere Lösungen programmieren, sollte man die häufigsten Datenmengen im Blick haben und entsprechende Algorithmen entwickeln.

Ingenieure bauten und bauen immer leistungsfähigere Hardware, sodass Effizienz oft als sekundär betrachtet wurde. Läuft Software auf einem lokalen PC und liegen Effizienzprobleme vor, so ist eine bessere Hardware meist die billigste Methode zur Beseitigung der Probleme.

Häufig läuft Software jedoch auf vielen Computern, was zu einer Änderung der wirtschaftlichen Betrachtung führt. Moderne Softwarearchitektur bedient sich auch angemieteter Rechenzeiten und Speicherplätze in der Cloud, Server haben viele Anfragen gleichzeitig zu erledigen und Mobiltelefone sollen komplexe Aufgaben erledigen, was zu knappen Ressourcen führt. Effizienz ist daher wieder ein aktuelles Thema.

Anwendungen der Geoinformatik waren immer in Hinblick auf ihren Ressourcenbedarf problematisch, sodass in dieser Branche Effizienz immer wichtig war. Steigende Kapazitäten der Hardware werden von neuen Datenerfassungsmethoden gefressen, die zu größeren Datenmengen führen. Moderne Hardware kann heutzutage problemlos Analysen mit allen 1,6 Mio. Grundstücken der Steiermark in wenigen Minuten erledigen, was vor 10 Jahren noch mehrere Stunden gedauert hat. Für viele Anwendungen werden heutzutage jedoch auch Laserscandaten herangezogen, die leistungsfähige Computer an den Rand ihrer Kapazitäten bringen. Messstationen erfassen Luftgütewerte etc. permanent, was zu großen Datenmengen führt.

#### 4.5.4.2. Strategie

Eine Methode, Effizienz zu erreichen, besteht darin, Redundanzen zu vermeiden. Dadurch werden die Algorithmen häufig komplexer und schwieriger zu lesen. Wiederverwendbarer, verständlicher und wartbarer Code schöpft nicht alle Mittel der Effizienzsteigerung aus. Da

ein komplexes Programm in der Regel mehr Fehler aufweist, führt extreme Effizienzsteigerung zur Reduktion der Korrektheit und Zuverlässigkeit. Aus diesen beiden Gründen und auch aus wirtschaftlichen Überlegungen sollte ein Entwickler die einzelnen Komponenten auf ihre Effizienz prüfen. Nur häufig benutzte oder auffällig langsame Module sollten effizienzsteigernden Maßnahmen unterzogen werden.

#### 4.5.4.3. Organisatorische Maßnahmen

Die Laufzeit zu steigern, ist aufwendig und fehleranfällig. Man kann auch die Ablauforganisation dahin gehend verändern, dass Laufzeiten nicht ins Gewicht fallen. Organisatorische Maßnahmen haben ein hohes Potenzial zur Steigerung der Nutzerzufriedenheit. Das im Rahmen der Arbeit geschriebene Programm

1. übernimmt Geodaten,
2. prüft Geodaten,
3. repariert Geodaten,
4. verwaltet Geodaten,
5. speichert Geodaten.

Die in der Produktionsdatenbank abgelegte Geoinformation wird über Nacht in eine Publikationsdatenbank repliziert, welche durch einen WebMap-Service als Kartendienst im Internet bereitgestellt wird. Der Raumplaner, der die Daten über ein Internetprotal abgibt, möchte möglichst schnell wissen, ob die Daten korrekt vorliegen. Punkt 1 benötigt nur Bruchteile einer Sekunde und auch die Prüfung und Reparatur der Geodaten ist schnell. Ein echtes Laufzeitproblem besteht beim Speichern der Geodaten, was viele Minuten in Anspruch nehmen kann. Es ist ein Fall aufgetreten, bei dem der Vorgang auch eine halbe Stunde benötigt hat. Der Benutzer des Systems kann unmöglich so lange warten.

Eine Analyse der Zugriffe auf das Abgabeportal hat gezeigt, dass es im Beobachtungszeitraum von einem halben Jahr niemals vor 7:00 Uhr oder nach 19:00 Uhr benutzt wurde. Nachdem Punkt 3 erledigt wurde, stehen alle für den Raumplaner notwendige Informationen bereit. Das sind eine Fehlerliste und ein korrigierter Datenvorschlag. Es macht Sinn, den Benutzer danach über das Ergebnis durch eine automatische E-Mail zu informieren und nicht mit dem Antwortschreiben bis zum Programmende zu warten. Der Raumplaner soll

zum ehest möglichen Zeitpunkt die für ihn relevante Information erhalten. Um Prozessorleistung und Hauptspeicher zu sparen, wenn die Maschine von Planern genutzt wird, erfolgt das Verwalten und Speichern in den Stunden vor Mitternacht. Der eigentlich langwierige Teil wird in der Nacht durchgeführt, sodass eine performante Lösung für den Benutzer übrig bleibt.

Die Software ist so gestaltet, dass mit ihr verschiedene Planarten geprüft werden können. Die Beschreibung der Pläne erfolgt in einer relationalen Datenbank, indem miteinander in Bezug stehende Tabellen ausgefüllt werden. Diese Art der Planbeschreibung ist komfortabel. Insbesondere die beiden Tabellen mit den jeweils rund 1000 Records für unterschiedliches Speichern und Löschen beim Einbringen von Plänen in die unternehmensweite Geodatenbank konnten aus einigen Basiszeilen durch entsprechende Tabellen- und Textmanipulationen erstellt werden.

#### 4.5.4.5. Schlüsselstellen in der Software

Das Programm verwendet objektorientierte Strukturen, weshalb der Code leichter zu lesen ist und weniger komplex wurde. Durch den Einsatz objektorientierter Konzepte stiegen die Korrektheit und die Zuverlässigkeit des Codes. Um einen Plan prüfen zu können, ist es notwendig, die relationalen Tabellen der Planbeschreibung zu lesen und daraus Objekte zu bauen, welche die Sollstruktur beschreiben. Der Aufbau dieser Objekte hat sich als zeitintensiv herausgestellt. Bis zu 3 Minuten waren notwendig, um die Planbeschreibung aus der Datenbank zu extrahieren und die notwendigen Objekte aufzubauen. Das stellt eine nicht akzeptable Laufzeit dar. Eine Möglichkeit wäre gewesen, mit einer objektorientierten Datenbank zu arbeiten, die jedoch als langsam gilt. Im Amt der Steiermärkischen Landesregierung stehen darüber hinaus keine Lizenzen für solche Datenbanken zur Verfügung. Zusätzliche Lizenzen und neue Software waren möglichst zu vermeiden, um die Komplexität und die Kosten zu minimieren. Um trotzdem einen schnellen Aufbau der Objekte zu gewährleisten, wurden die Möglichkeiten von Python zur persistenten Speicherung von Objekten in Dateien genutzt. Nach dem Editieren der planbeschreibenden Tabellen werden in einem separaten Programm die Objekte aufgebaut und aus dem Hauptspeicher in Dateien kopiert. Dieser Vorgang wird als „pickle“ (einpökeln) bezeichnet, wodurch eine Assoziation mit dem Haltbarmachen von Fleisch hergestellt werden soll. Da dieser Vorgang im Rahmen der Programmwartung erfolgt, ist die Laufzeit für den Aufbau der Objekte aus

den Tabellen und für das Einpökeln irrelevant. In Pickle-Dateien bereitgestellte Objekte können von Python extrem schnell gelesen werden, weshalb im Prüfungsfall immer mit den Pickle-Dateien gearbeitet wird. Gegenüber dem Lesen aus der Datenbank ist der Objektaufbau aus den Pickle-Dateien um den Faktor 1000 schneller.

Der Pickle-Mechanismus stellt eine „serialization“ dar. Dabei werden die Objekte in eine sequenzielle Abfolge zerlegt und binär in eine Datei gespeichert. Der Mechanismus wurde entwickelt, um Objekte zwischen verschiedenen Computern oder Programmen austauschen zu können. Er beinhaltet jedoch nicht die Methoden der Objekte. Die lesende Software benötigt die vollständige Klassendefinition des gelesenen Objekts.

Der aufgezeigte Weg stellt eine Brücke zwischen relationalen Datenbanken und Objektorientierung dar. Er hat sich als erfolgreich erwiesen, wenn wenige sehr große Objekte benötigt werden. In der relationalen Datenbank kann die Objektbeschreibung leicht gewartet werden. Der Aufbau der Pickle-Dateien ist rechenintensiv, was jedoch im Rahmen der Programmwartung nicht von Bedeutung ist. Die Objekte können sehr schnell aus diesen Dateien gelesen werden, da ein sequenzieller Datenstring verarbeitet wird.

Werden viele unterschiedliche Objekte benötigt, scheint der vorgeschlagene Weg nicht zielführend zu sein, da eine große Menge an Dateien zu verwalten wäre, was die Komplexität ansteigen lässt. Objektorientierten Datenbanken ist eindeutig der Vorzug zu geben. Ein möglicher Zwischenschritt wäre es, die Pickle-Datei in eine Datenbank zu speichern. Man legt dafür eine Tabelle mit einem „*long binary field*“ an, welches den Datenstrom aufnehmen kann. Die Tabelle kann beliebige andere Spalten enthalten, die ein schnelles Auffinden und Verwalten der Objekte ermöglicht.

Darüber hinaus hat das Verfahren den Nachteil, dass Methoden nicht übertragen werden. Es wird so zwar ein schnelles Lesen garantiert, aber auch vorausgesetzt, dass das lesende Programm und das die Pickle-Dateien schreibende Programm aus einem Guss erstellt wurden. Ändert man Methoden im schreibenden Programm, sind die Arbeiten im lesenden Programm nachzuziehen. Im gegenständlichen Fall liegt eine einzige Software vor, die in Abhängigkeit von Environmentvariablen die Pickle-Dateien schreibt bzw. sie im Rahmen von Prüf- und Verwaltungsverfahren liest. Programmänderungen sind somit nicht doppelt zu berücksichtigen, weshalb diese Schwäche irrelevant für das Projekt ist.

#### 4.5.4.6. Nutzung des Arbeitsspeichers

Neben der Laufzeit ist auch der Umgang mit dem Hauptspeicher wesentlich für die Effizienz einer Software. Das in dieser Dissertation entwickelte Prüf- und Reparaturprogramm läuft auf einem Server, der auch für andere ebenfalls essenzielle Aufgaben benötigt wird. Das sind die Geodatenbereitstellung für Kunden des Landes, Analysen des WebMap-Services des Landes und vieles mehr.

GI-Systeme stellen immer Werkzeuge zur Verfügung, mit denen Geodatenätze in Abhängigkeit von vorgegebenen Parametern abgeändert und kombiniert werden. Das Ergebnis ist meist wiederum ein Geodatenatz, der aufs Neue mit Werkzeugen bearbeitet wird. Man hat üblicherweise eine Menge verschiedener Geodatenätze, welche Zwischenergebnisse in einer Kette von Geooperationen sind. Die Laufzeit kann um ein Vielfaches reduziert werden, wenn diese Zwischenergebnisse im Hauptspeicher gehalten werden. ESRI bietet diese Möglichkeit mit dem `in_memory`-Workspace an. Der Zugriff ist um vieles schneller als auf Geodaten, die auf Platten liegen. Der Hauptspeicher ist im Gegensatz zur Festplatte deutlich kleiner und die dort zwischenzeitlich gehaltenen Daten müssen ausreichend Speicherplatz finden. Es muss im Arbeitsspeicher noch Platz für das Programm selbst und eventuell andere parallel laufende Programme vorhanden sein. Der extrem schnelle `in_memory`-Workspace sollte also nur für kleine Datenmengen verwendet werden. Nicht benötigte Zwischenergebnisse sind ehestmöglich zu löschen. Beim Entwurf der Software ist abzuschätzen, wie groß die Geodatenätze maximal werden können und ob ein entsprechender Hauptspeicher vorliegt. Geht Hauptspeicher aus, lagert das Betriebssystem gerade nicht benötigte Teile des Hauptspeichers auf die Festplatte aus, die bei Bedarf wieder geholt werden (Swapping). Kommt die Maschine in das Swappen, geht der Laufzeitvorteil verloren. Die Prozedur wird um vieles langsamer gegenüber einer herkömmlichen Speicherung auf der Platte. Der Swapper kann nicht oder nur schwer vorhersehen, welche Teile des Hauptspeichers zukünftig gebraucht werden, was zur Folge hat, dass immer wieder ausgelagert und nachgeladen werden muss. Ein zielgerechtes Laden des benötigten Geodatenatzes ist wesentlich effizienter. Darüber hinaus leidet die Stabilität der Maschine durch das Swapping.

Ein gut überlegter Einsatz des `in_memory`-Workspace kann eine Performancesteigerung auf Kosten der Ressource Hauptspeicher bringen.

#### 4.5.4.7. Spatial Reference and Geoprocessing

Der Raumbezug wird innerhalb von ESRI durch eine Kartenprojektion (z. B. Gauß-Krüger-Meridianstreifen 34), ein geodätisches Datum (z. B. Fundamentalpunkt Hermannskogel, Besselipsoid), eine Maßeinheit (z. B. Meter) und zumindest einen XY-Bereich und eine Auflösung festgelegt (siehe [37]).

Die Koordinaten werden als Ganzzahlen gespeichert. Hier gibt es die Low-Precision mit einem Datenumfang von 4 Byte, welche es ermöglicht, ca.  $2.147 \cdot 10^9$  unterschiedliche Zahlen abzuspeichern, und die High-Precision<sup>21</sup>, welche mit  $9.007 \cdot 10^{15}$  unterschiedlichen Zahlenwerten arbeitet. Eine vorliegende Koordinate wird in den Zahlenraum von 4 Byte bzw. 8 Byte umgerechnet, indem eine Nullpunktverschiebung erfolgt und durch die Auflösung (resolution) dividiert wird. Nachkommastellen werden gerundet. Die Wahl des Nullpunktes garantiert, dass keine negativen Zahlenwerte auftreten.

“Persisted coordinate = Round( (map coordinate – minimum domain extent) / resolution)” [38]

Eine Auflösung von 0.01 entspricht bei Einheiten von Metern dem Zentimeter und 0.001 dem Millimeter. Die Low-Precision erlaubt es, die gesamte Erde mit einem Umfang von 40.000 km mit einer Auflösung von ca. 2 cm abzuspeichern.

„These spatial reference properties can have significant impact on the performance and results generated by a geoprocessing tool.“ [37]

Werden mehrere Geodatensätze mit einem Werkzeug verarbeitet, wie es etwa die Standardoperation Intersect vorsieht, müssen die Geodatensätze in ein gemeinsames räumliches Bezugssystem transformiert werden. ESRI verwendet hier den räumlichen Bezug des Ergebnisdatensatzes, falls in ein Feature-Dataset gespeichert wird. Das ergibt sich aus der Tatsache, dass ein Feature-Dataset nur Geodaten eines einzigen zuvor festgelegten Bezugssystems aufnehmen kann. Gibt das Ziel keinen räumlichen Bezug vor, wird der räumliche Bezug des ersten bearbeiteten Geodatensatzes verwendet.

---

<sup>21</sup> Seit ArcGIS 9.2

Die Toleranz, ebenfalls ein Element des Raumbezuges, ist strikt von der Auflösung zu trennen. Darunter wird jene Distanz verstanden, in der zwei benachbarte Koordinaten als lagegleich angesehen werden. Es handelt sich also um eine Entfernung, in welche die Koordinate in x-Richtung oder in y-Richtung bei Clusterbildungen verschoben werden kann. Die Toleranz muss so groß wie die Auflösung oder größer sein. ESRI empfiehlt eine Toleranz, die zumindest 10-mal größer als die Auflösung ist [39].

Meist arbeitet man in einem Koordinatensystem mit einem fixen Datum. Die räumlichen Auflösungen und Domains der einzelnen Layer sind jedoch unterschiedlich, was zu zeitaufwendigen Umrechnungen führt. Arbeitet man in einem einzigen Feature-Dataset, wo nur Geodatenätze mit gleichen räumlichen Bezug gespeichert werden, erfolgt nur während des Importes eine Überführung in das Zielsystem.

Der zeitaufwendigste Schritt der entwickelten Software ist die Überführung der kontrollierten und aufbereiteten Geodaten, welche in einer Filegeodatabase (fGDB) liegen, in den unternehmensweiten ArcSDE. Hier ist es von besonderer Bedeutung, dass die räumlichen Bezugssysteme übereinstimmen, was eine wesentliche Beschleunigung bringt.

#### 4.5.4.8. Vermeidung unnötig großer Datensätze

Wird die Toleranz zu klein gewählt, werden als gleich gedachte Linien nicht mehr als solche erkannt, da kleinste Ungenauigkeiten zum Tragen kommen. Es kommt bei Verschnitten etc. zu unnötigen Sliver-Polygonen bzw. Löchern.

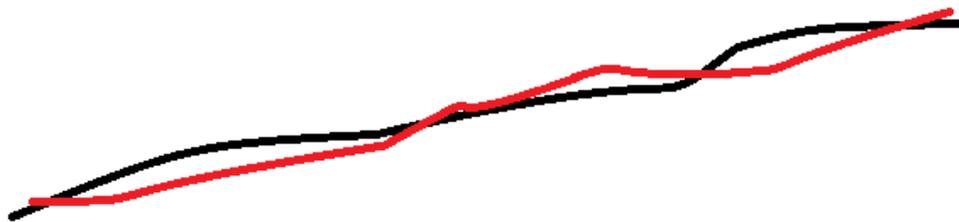


Abbildung 9: Entstehung von Sliver-Polygonen

Die Geodatenätze werden durch Sliver-Polygone groß, unhandlich und langsam. Darüber hinaus verfälschen die kleinen Teilflächen die Ergebnisse und verursachen kartografische Probleme. Zusätzliche Generalisierungen sind notwendig.

#### 4.5.4.9. Korrekte Geometrie

Laufzeitprobleme können aus einer unsauberen Geometrie resultieren. Deshalb wollen wir die Geometrie näher betrachten. Die Feature Class unterscheidet sich von der Object Class durch das Vorhandensein einer Geometriespalte des Typs Geometry. In den einzelnen Zeilen dieser Spalte ist die Gestalt abgelegt. Folgende Geometrietypen wollen wir betrachten, da sie in diesem Projekt Verwendung fanden:

1. Point
2. Multipoint
3. Line, Polyline
4. Polygon

Ad1)

ID	SHAPE	NAME
1	x=1,y=2	Waldeshöhe
2	x=3,y=4	Tiefpunkt

Jeder Punkt wird separat durch seine Attribute beschrieben.

Ad2)

ID	SHAPE	NAME
1	x=1,y=2   x=3,y=5   x=7,y=2	Waldeshöhen
2	x=3,y=4	Tiefpunkte

Das Feature mit dem Namen „Waldeshöhen“ besteht aus 3 unterschiedlichen Punkten. Ändert man den Attributwert „Waldeshöhen“, gilt der geänderte Wert für alle drei Standpunkte.

Ad3)

Das Geometriefeld kann einen Linienzug (path) enthalten, mehrere zusammenhängende Linienzüge oder gänzlich getrennte Linienzüge.

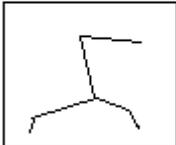
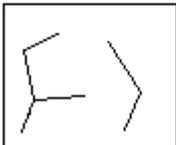
<b>ID</b>	<b>Shape</b>	<b>Attribut A</b>	<b>Attribut B</b>
<b>1</b>		<b>abc</b>	<b>def</b>
<b>2</b>		<b>ghi</b>	<b>jkl</b>
<b>3</b>		<b>mno</b>	<b>opr</b>

Abbildung 10: Das Geometrieelement für Linien

Ad4)

Das Feature besteht aus einer Teilfläche bzw. aus mehreren Teilflächen. Eine Fläche wird durch einen geschlossenen Linienzug gebildet. Dieser geschlossene Linienzug wird als „Ring“ bezeichnet.

Um eine Fläche mit einem Loch zu speichern, werden ein innerer Ring und ein äußerer Ring angelegt. Ringe dürfen sich nicht gegenseitig überlagern (not self intersecting).

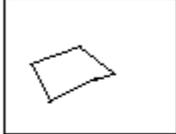
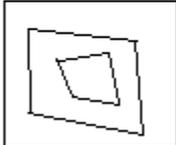
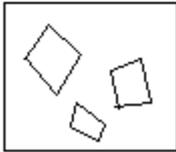
<b>ID</b>	<b>Shape</b>	<b>Attribut A</b>	<b>Attribut B</b>
<b>1</b>		<b>abc</b>	<b>def</b>
<b>2</b>		<b>ghi</b>	<b>jkl</b>
<b>3</b>		<b>mno</b>	<b>opr</b>

Abbildung 11: Das Geometrieelement für Flächen

Der Path (das Segment einer Polyline) bzw. der Ring (der Grundbestandteil einer Fläche) wird als eine Abfolge von Koordinaten gespeichert. Zwischen diesen Stützpunkten (Vertex) sind im Allgemeinen geradlinige Verbindungen festgelegt. Es können jedoch auch Kreisbögen (circular arcs), Bogenelemente von Ellipsen (elliptical arcs) und Bezier-Kurven (Bezier curves) vorliegen [40].

Verschiedene Polygone (Feature) können sich gegenseitig überlagern. Innerhalb eines Polygons dürfen sich die durch Ringe gebildeten Teilflächen jedoch nicht überlagern. Um das zu reparieren, bietet ESRI das Werkzeug „Repair Geometry“ an, mit dem alle Überlagerungen innerhalb einer Fläche beseitigt werden. Self Intersections sind häufig zu beseitigen. Es werden jedoch auch noch andere Reparaturen durchgeführt, welche die Verarbeitungsgeschwindigkeit dieser Geodatensätze erhöhen.

- „Null-Geometrie: Der Datensatz wird aus der Feature-Class gelöscht. Sollen Datensätze mit Null-Geometrie behalten werden, deaktivieren Sie im Dialogfeld des Werkzeugs die Option Features mit Null-Geometrie löschen, oder setzen Sie bei der Skripterstellung den Parameter "delete\_null" auf "KEEP\_NULL"
- Falsche Ringanordnung: Die Geometrie wird zwecks einer richtigen Ringanordnung aktualisiert.

- Falsche Segmentausrichtung: Die Geometrie wird zwecks einer richtigen Segmentausrichtung aktualisiert.
- Schnittpunkte mit sich selbst: Polygonflächen, die sich überschneiden, werden zusammengeführt.
- Offene Ringe: Offene Ringe werden durch Verbindung ihrer Endpunkte geschlossen.
- Leere Teile: Teile ohne Wert oder mit NULL-Wert werden gelöscht.
- Doppelter Stützpunkt: Einer der Stützpunkte wird gelöscht.
- Nicht übereinstimmende Attribute: Die Z- oder M-Koordinate wird zwecks einer Übereinstimmung aktualisiert.
- Diskontinuierliche Abschnitte: Aus einem diskontinuierlichen Abschnitt werden mehrere Abschnitte gebildet.
- Leere Z-Werte: Der Z-Wert wird auf 0 festgelegt.“ [41]

Jeder gelieferte Geodatensatz wird, nach der Kontrolle des grundsätzlichen Aufbaus,<sup>22</sup> dieser Operation unterzogen, was nicht nur die Geschwindigkeit erhöht, sondern auch die Robustheit der Anwendung steigert.

Geodatensätze können doppelte idente Features enthalten. Mit dem Werkzeug „Delete Identical“ kann man auf einfachem Weg Redundanzen beseitigen. Dabei ist wiederum eine Toleranz anzugeben, ab der Koordinaten als ident angesehen werden. In der Schnittstelle des BEV kann nachgelesen werden:

„Die Koordinaten von Grenzpunkten (Punkte der KDB-GP) werden wie bisher in Metern mit 2 Nachkommastellen (cm-Genauigkeit) angegeben. Linienstützpunkte ( ‚VERTEX‘ einer ‚POLYLINE‘) werden, wenn sie koordinatenmäßig mit Punkten der KDB-GP ident (substituiert) sind, ebenfalls mit 2 Nachkommastellen angegeben. Bei allen übrigen rein graphischen Linienstützpunkten werden 3 Nachkommastellen (mm-Genauigkeit) angegeben.“ [42]

---

<sup>22</sup> Grundsätzlicher Aufbau: Die Anzahl der Features wird gezählt. Diese Methode versagt, wenn die Anzahl der Geometrielemente nicht mit der Anzahl der Attribute übereinstimmt. Das ist häufig der Fall, wenn die Editiersitzung nicht ordentlich geschlossen wird. Darüber hinaus kontrolliert die Methode, ob alle notwendigen Dateien, welche einen Geodatensatz des Formates Shape ausmachen, vorhanden sind ( \*.shp...Geometrie, \*.dbf...Attribute \*.shx...Verknüpfung Geometrie Attribute).

Der Millimeter ist eine gute Wahl für alle Anwendungen, die auf Basis der Grundstücksgrenzen arbeiten, denn die grafischen Zwischenpunkte sind mit Millimeterstellen abgelegt. Der FWP basiert auf dem Kataster.

Features aus automatischen Prozessen können pixelförmige Grenzlinien aufweisen, wenn Rasteranalysen erfolgten und die Ergebnisse danach in Vektordatensätze überführt wurden. Diese Geodaten sollten mit den Generalisierungswerkzeugen der Kartografie/Toolbox vereinfacht werden.<sup>23</sup> Das Werkzeug „Simplify Polygon“ ersetzt langgestreckte Polylines durch Geradenstücke, wobei der maximale Abstand der resultierenden Geraden zu den alten Vertices angegeben werden kann. Will man Treppmuster entfernen, ist die ursprüngliche 1,5-fache Pixelgröße eine gute Wahl. Da es kaum Anwendungen gibt, die exakter als der Kataster sind, kann man mit „Point-remove“-Distanzen von 1–2 Millimetern nichts falsch machen. Im Rahmen der Übernahme von Flächenwidmungsplänen und örtlichen Entwicklungskonzepten dürfen diese Operationen nicht durchgeführt werden. Sie verändern den Datensatz des Planers. Teilweise werden Raumplanern Grundlagendaten vom Land bereitgestellt, die diese treppenförmigen Abgrenzungen enthalten. Diese werden dann wiederum als Teile von Flächenwidmungsplänen rückübermittelt. Es wäre sinnvoll, die Geodatensätze schon im Land zu generalisieren. Die Übernahme würde mit geringeren Datenmengen und damit schneller laufen. Auch der Planer hätte mit diesen schlankeren Daten Vorteile bei der Verarbeitung. Es muss jedoch angemerkt werden, dass diese Effekte bei gemeindeweisen Plänen eher gering ausfallen. Der größte Vorteil dürfte wohl im Land selbst liegen, wo steiermarkweite Daten verarbeitet werden. Diese Daten erhält das Land wiederum von Zivilingenieuren und sie werden im Amt nicht verändert, da die rechtliche Verantwortlichkeit des Plan-Erstellers abgeschwächt würde und man ihn nicht aus der Verantwortung für seine Festlegung entlassen möchte.

Besteht ein Feature aus zu vielen Vertices, kann das ebenfalls in Abhängigkeit von der zur Verfügung stehenden Hardware zu Laufzeitproblemen führen. Die Anzahl der Vertices pro Feature kann mithilfe des Werkzeuges „Dice“ reduziert werden, wobei die maximale Anzahl an Zwischenpunkten als Parameter angegeben ist. Wird diese Anzahl überschritten, wird das Feature geteilt und die Attribute werden dupliziert. Von dieser Methode wurde Abstand genommen, da ein Aufspalten von Flächen zu nicht gewünschten Grenzlinien führt

---

<sup>23</sup> Werkzeugbox des Produkts ArcGIS-Desktop der Firma ESRI. Andere Firmen bieten Ähnliches an.

und der Laufzeiteffekt bei Flächenwidmungsplänen kaum ins Gewicht fällt. Landesintern ist das zu empfehlen.

#### 4.5.4.10. Web-Service

FWPs und ÖEPs werden im Internet mithilfe eines WebMap-Services visualisiert. Für den Nutzer ist auch die Effizienz dieses Dienstes entscheidend, weshalb er integraler Bestandteil der Arbeit ist. Das WebMap-Services des Landes erzeugte im Jahr 2014 insgesamt 39.604 Mio. Karten für das Internet und 3.557 Mio. Kartenbilder für das Intranet, wobei im Internet 6 % der Kartenbilder FWP und OEP betrafen und im Intranet 1%. Diese hohen Lasten bedingen, dass auf die Geodaten schnell zugegriffen werden muss. Dem Index der Geodaten, der Datenbank, dem Rendering sowie der Hardwarekonfiguration kommen damit viel Bedeutung zu. Darüber hinaus soll nicht unerwähnt bleiben, dass Laufzeitprobleme bei anderen Geodatenätzen auftraten, die von dieser Abhandlung ebenfalls profitieren sollen und Basis für diese Überlegungen sind.

##### 4.5.4.10.1. *Index der Geodaten*

Grundsätzlich ist zwischen Indices auf Geometriedaten und Attributdaten zu unterscheiden. Attributive Indices können für Spalten erstellt werden, wobei jeder Eintrag sich unterscheidet, aber auch für Spalten mit mehrfach identen Einträgen. Ersteres sind meist Look-up-Tabellen, wo über einen Schlüssel, Detailinformationen zu einem Eintrag festgehalten sind. Diese Schlüsselspalte ist in der Look-up-Tabelle eindeutig. Da bei der Erstellung eines Attributiven-Index der Standard auf „nicht eindeutig“ festgelegt ist, wird auch für Schlüsselfelder gerne damit gearbeitet. Dieser Fehler im Datenbankentwurf macht sich nur durch langsamere Zugriffszeiten bemerkbar, weshalb er nicht bemerkt wird. Große Erfolge im Laufzeitverhalten sind durch Bereinigung dieses Umstandes zu erzielen. Man sollte eindeutige attributive Indexe verwenden, wo immer möglich. Natürlich macht ein Index nur Sinn, wenn diese Spalte auch in Abfragen selektiv benötigt wird und eine gewisse minimale Diversität vorliegt. Besondere Vorsicht ist bei Schlüsseln geboten, wo der attributive Schlüssel aus mehreren Spalten besteht. Einen zweispaltigen Index stellt man sich am besten durch Analogie zu einem Telefonbuch vor, wo nach Nachname und in weiterer Folge nach Vorname sortiert wurde. Sucht man nach Mayer Hans wird die Telefonnummer schnell

gefunden, da zuerst nach Mayer gesucht werden kann und dann nach Hans. Braucht man jedoch alle Hans, kann der Index nicht genutzt werden, sondern es wird ein Fullscan der Datenbank durchgeführt. Jede einzelne Zeile wird dabei mit der Where-Klausel der Abfrage verglichen, was zeitaufwendig ist. Der Index kann jedoch genutzt werden, wenn alle Mayer gesucht werden. Die erste Spalte ist das primäre Suchkriterium, gefolgt vom sekundären Suchkriterium usw. [43].

Ein schnelles Rendern erfordert, dass alle Objekte innerhalb des aktuellen Bildschirmausschnittes in kurzer Zeit gefunden werden. Auch bei nahezu allen anderen Geooperationen ist eine Datenreduktion auf einen Ausschnitt eine notwendige Vorverarbeitung um die große Datenmenge in hinreichender Zeit zu prozessieren. Die Wissenschaft bietet verschiedene Modelle an, und einige sind auch in kommerziellen Produkten umgesetzt. Im Amt der Steiermärkischen Landesregierung wird der ArcSDE verwendet, um Geodaten zu speichern und zu verwalten, und dessen Technologie zur Indizierung der Daten kommt somit zur Anwendung. Der WebMap-Service ist ein Client, dem die Daten bereitgestellt werden.

Der Index wird im ArcSDE durch das Anlegen von regelmäßigen Adresszellen erstellt. Die Adresszellen kann man sich wie regelmäßige quadratische Kacheln vorstellen. Es wird zu jeder Adresszelle gespeichert, welche Objekte innerhalb der Zelle liegen. Fallen Objekte in mehrere Zellen, werden sie in all diesen Zellen referenziert [23, S. 346].

Eine räumliche Abfrage, wie etwa die Darstellung aller Objekte des gerade aktuellen Bildschirmausschnittes, ermittelt zuerst die benötigten Adresszellen und verarbeitet dann die referenzierten Einzelobjekte der betroffenen Zellen. Um nicht zu viele unnötige Objekte abzuarbeiten, macht es Sinn, die Adresszellen kleinzuhalten. Die Adresszellen schmiegen sich besser an den tatsächlich benötigten Bereich an. Sie sollten jedoch nicht zu klein sein, da sonst viele Objekte in mehreren Zellen liegen und in jeder Zelle referenziert werden. Eine räumliche Abfrage, die mehrere benachbarte Zellen beinhaltet, verarbeitet ein über die Zellen hinausragendes Objekt mehrfach, was einen unnötigen Rechenaufwand darstellt.

Sind die Objekte in Bezug auf ihre Größe sehr heterogen, sollten die großen Objekte in einem eigenen Adressgitter verwaltet werden. ESRI bietet drei Adressgitter an, die Verwendung finden können. Meist reicht ein einziges Gitter aus. Es gibt auch Werkzeuge, die für einen vorzugebenden Geodatensätze gute Gittergrößen ermitteln. Verfeinerungen können

erfolgreich getroffen werden, da diese Parameter auf Hardwareannahmen beruhen. Die Gittergrößen sollten sich um ein Vielfaches unterscheiden.

Folgende Vorgehensweisen für optimale Performance haben sich als sinnvoll herausgestellt:

- Ein guter Standardwert für einen räumlichen Index ist die dreifache mittlere Ausdehnung der Objekte. Ca. 80 % der Features sollten in eine Zelle fallen. Auf SDE-Layer-Ebenen (einer Kommandosprache für den SDE) kann der Prozentsatz ermittelt werden [44].
- Die Verwendung eines zweiten oder eventuell dritten größeren Adressgitters ist meist kontraproduktiv, da jede Ebene durchsucht werden muss. Nur bei wirklich großen unterschiedlichen Ausdehnungen macht das Sinn [44]. Beispiel im Amt der Steiermärkischen Landesregierung ist etwa eine sehr detailreiche Waldkartierung mit durchschnittlichen Größen unter einem halben Hektar. Die zusammenhängenden Wege bildeten eine einzige Fläche und deren Bounding Rectangle maß knapp 70 km \* 50 km.

Neben Adressfeldern gibt es noch andere Technologien zur Erstellung räumlicher Indices. Eine Möglichkeit ist der R-Tree. Objekte werden durch eine Rectangle-Bounding-Box beschrieben. Sie bilden die unterste Ebene. Nebeneinander liegende Objekte werden mit einer weiteren eigenen Bounding-Box zusammengefasst. Das lässt sich weiter wiederholen und es entsteht eine hierarchische Struktur, die in einem Baum abgebildet wird [23, SS. 293–295]. Der ArcGIS-Server, auf dem unsere Web-Services beruhen, ist in der Lage, nicht nur den ArcSDE als Quelle zu verwenden, sondern neben vielen anderen auch Oracle ST\_Geometry einzubinden. Hier wird der R-Tree zur Indizierung verwendet und damit bleiben Gittereinstellungen wirkungslos. Das bedeutet aber auch, dass hier keine Fehler beim Festlegen von Gittergrößen gemacht werden können.

ArcSDE als Aufsatz von Oracle wird hinfällig und man kann Geodaten direkt in der Datenbank speichern, was vermutlich performanter ist. Neben Oracle bieten auch die anderen Datenbankanbieter Geometriefelder an, die eingesetzt werden können.

SQL-Server verwendet für die Speicherung von räumlicher Information den Typ Microsoft SQL-Server Geometry bei planaren Daten und den Typ Microsoft SQL-Server Geography

bei Daten auf Basis von geografischer Länge und Breite. Im GIS-Steiermark ist eine Lizenz für Microsoft SQL-Server vorhanden, sodass der Schritt einer Datenüberführung in diese Geometrietypen leicht gegangen werden kann.

Da Oracle von der allgemeinen EDV als Standard definiert ist, kann auch mit Oracle ST\_Geometry gearbeitet werden. Entsprechende Tests sind geplant. Diese Datenbank würde dann auch von der allgemeinen EDV gewartet werden, was im GI-Bereich zu einer dringend benötigten Aufgabenreduktion führen würde.

#### 4.5.4.10.2. Datenbank

Nachdem die Geodaten in Hinblick auf ihre Effizienz optimiert wurden, ist die Datenbank zu betrachten. Auch hier gibt es Stellschrauben, die davon abhängen, auf welcher Plattform der ArcSDE betreiben wird. Für das Amt der Steiermärkischen Landesregierung ist das, wie bereits erwähnt, Microsoft SQL-Server. Genauer zu den nachstehenden Möglichkeiten kann unter [45] nachgelesen werden.

Microsoft SQL-Server tendiert dazu, den ganzen Hauptspeicher zu beanspruchen. Für andere Anwendungen bleibt mitunter zu wenig Platz, was besonders für den Aufsatz ArcSDE (gsrvr.exe) gilt. Deshalb sollte der SQL-Server in seinem maximalen Hauptspeichergebrauch begrenzt werden. Der tatsächliche Memory-Gebrauch des SQL-Servers kann mit dem Enterprise Manager betrachtet werden.

Die Statistiktabelle der Geodatenansätze sind aktuell zu halten, damit die internen Optimierungen korrekt erfolgen können. Das ist besonders wichtig, wenn häufige Datenänderungen erfolgen und im Multiversion Modus gearbeitet wird. Im gegenständlichen Fall ermöglicht ein entsprechender Datenbankentwurf, jede Änderung rückgängig zu machen. Mehrfache schreibende Zugriffe auf eine Gemeinde werden schon bei der Abgabe blockiert, weshalb die Datenbankfunktionalität der Versionierung nicht benötigt wird. Versionierung wurde ausgeschaltet, was zu einer Performance-Steigerung führt, da Delta-Tabellen nicht berücksichtigt werden müssen. Nichtsdestotrotz ist es notwendig, von Zeit zu Zeit einen „fullscan on all tables“ durchzuführen.<sup>24</sup>

---

<sup>24</sup> `sdetable -o update_dbms_stats -t <table_name> -m "with fullscan" -i <instance> -D <db>`

Durch ständiges Ändern der Geodatensätze werden die Indexdateien in ihrer physikalischen Anordnung auf der Platte zerrissen. Zum Lesen des Index sind dadurch mehrere Bewegungen des Lesekopfes der Platte notwendig, was in der Summe zu Geschwindigkeitsverlusten führt. Die Platte fragmentiert. Deshalb sollte der Index auf Ebene des ArcSDE reorganisiert werden.<sup>25</sup>

Microsoft SQL-Server schreibt ein Transaktionsprotokoll, das in seiner Größe immer wieder durch Backups beschränkt werden sollte, um nicht zu viele Ressourcen zu verbrauchen. Das kann mit Microsoft SQL-Studio erfolgen [46]. Speichert man das Protokoll auf eine eigene Platte, fragmentiert es nicht die Datendateien. Das gilt auch für die Auslagerungs- und Temporärdateien.

Indexe werden vom SQL-Server in Pages abgelegt, die auf einmal gelesen werden. Sie sollten groß sein (FillFactor), damit ein Lesevorgang die gesamte benötigte Information enthält. Werden sie jedoch zu groß, schleppt man einen nicht benötigten Overhead mit. Auf jedem Fall kann eine Änderung des FillFactor nur wirken, wenn danach der Index auf Ebene des SQL-Servers neu aufgebaut wird [47].

#### 4.5.4.10.3. *MXD*

ArcGIS-Desktop speichert die Darstellungen, Maßstabsbegrenzungen und Filterungen in binären MXD-Dateien. Diese Dateien können auch für den Webdienst des ArcGIS-Servers als Basis verwendet werden. Man braucht für die Erstellung eines Kartendienstes keine eigene Umgebung mehr beherrschen, wie es beim ArcIMS notwendig war,<sup>26</sup> und hat darüber hinaus noch alle Möglichkeiten der grafischen Gestaltung, die ArcMap-Desktop bietet. Eine schnelle MXD-Datei ist notwendig für ein schnelles Web-Service auf Basis von ArcGIS-Server.

---

<sup>25</sup> `sdetable -o rebuild_index -t <tablename> -x all -D <db>`

<sup>26</sup> Die Internetdarstellung im ArcIMS wurde durch Map-and-View-Configuration-Dateien gesteuert. Sie heißen \*.axl und sind wie XML-Dateien aufgebaut. Axl steht für ArcXML.

#### 4.5.4.10.4. *Daten*

Jede Rechenoperation kostet Zeit. Deshalb ist es im Hinblick auf die Verarbeitungsgeschwindigkeit sinnvoll, die Daten so vorzuhalten, wie sie benötigt werden, auch wenn das eine redundante Datenhaltung zur Folge hat. Wenn Geodaten mit Tabellen verknüpft werden, um Abfragemöglichkeiten auf diese Informationen zu bieten, dann kostet diese Operation Rechenzeit. Es macht Sinn, diese Daten persistent an die Tabelle zu hängen. Die Verarbeitungsgeschwindigkeit wird mit Speicherplatz und einer Erhöhung der Komplexität erkaufte.

Liegen Geodaten in verschiedenen Koordinatensystemen vor, werden sie „on the fly“ in ein gemeinsames System übergeführt. Das ist sehr komfortabel, aber auch entsprechend rechenintensiv.

Liegt eine Tabelle mit Koordinaten bzw. mit Kilometerwerten und der entsprechenden Route vor, können diese Daten mit ArcMap-Desktop problemlos als Event-Layer visualisiert werden. Wird der Eventlayer in einer Vorprozedur in eine FeatureClass übergeführt, kann diese schneller gerendert werden. Eventlayer sind langsamer als entsprechende Featurelayer.

#### 4.5.4.10.5. *Darstellung*

Komplexe Darstellungen sind zeitintensiv. Eine einfache Symbolik kann hingegen performant erstellt werden. Im Speziellen stellt sich die Frage danach, welche Symbole welches Zeitverhalten an den Tag legen, was durch Probieren ermittelt werden kann. Simple Marker, Simple Line und Simple Polygon Symbols können am schnellsten aufgebaut werden, wobei die „ESRI\_Optimized Styles“ um ca. 60 % schneller sind als die schon schnellen „ESRI Styles“ [48].

Wird eine komplexere Punktsymbolik benötigt, empfiehlt es sich, diese als True-Type-Font zu entwickeln. Das ist aufwendig, jedoch deutlich performanter, als die Symbole durch Überlagerung von Simple Marker als Multi-Layer-Symbol aufzubauen. Das gilt nicht nur für Punkte, sondern auch für Linien, die durch eine Abfolge von Punkten visualisiert werden, und auch für flächenfüllende Punktmuster.

ArcMap bietet auch die Möglichkeit, Rastergrafiken als Punktsymbole einzubinden, wobei das Format EMF schneller als BMP, PNG, GIF oder JPG dargestellt wird [48]. Diese Art der Punktdarstellung ist die langsamste Symbolik. Dem steht die Einfachheit der Generierung einer komplexen Darstellung gegenüber.

Simple Lines sind deutlich schneller als Cartographic Lines, Hash Lines oder Marker Lines.

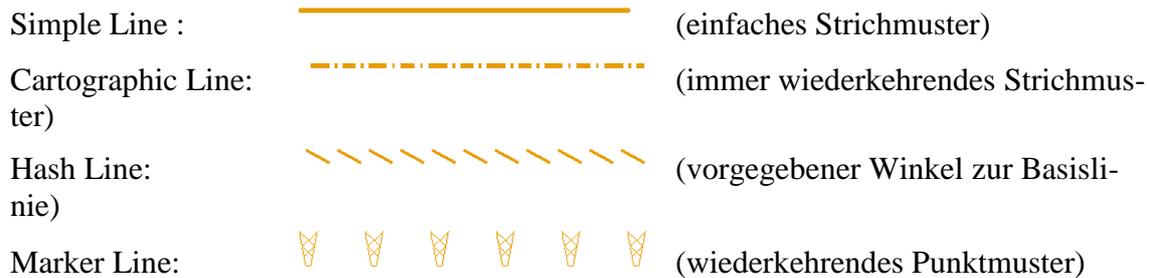


Abbildung 12: Verschiedene Typen an Linien [49]

Auch Linien lassen sich aus Basislinien kombinieren, was grundsätzlich zu Geschwindigkeitseinbußen führt. Die Kombination von zwei Cartographic Lines oder zwei Hash Lines ist schneller als jene von einer Cartographic Line mit einer Hash Line. Im ersten Fall (zwei Cartographic Lines oder zwei Hash Lines) erfolgt nur ein Durchlauf für das Rendern, in dem beide Liniensymbole gezeichnet werden. Im zweiten Fall (Cartographic Line mit Hash Line) sind zwei Durchläufe notwendig [48].

Stellt man große Mengen an Linien dar, ist zu erkennen, dass der Bildschirmaufbau einer dickeren Linie langsamer ist als jener von dünnen Linien.

Auch für Flächen gilt, dass der „ESRI\_Optimized“-Style der schnellste ist. Randlinien sind tunlichst zu vermeiden. Werden sie benötigt, gilt, was einige Zeilen zuvor über Linien gesagt wurde. Multilayer sind langsamer als ein einziger Layer. Werden etwa eine Basisfarbe und eine Punktmuster bzw. Linienmuster benötigt, kommt der Verwendung von schnellen Punkten oder Linien besondere Bedeutung zu.

Freistellungen und Maskierungen erhöhen die Lesbarkeit bei kleine Elementen oder geringen Kontrasten, sind jedoch immer ein Geschwindigkeitsproblem.

#### 4.5.4.10.5.1. Text

Um eine stumme Karte zu vermeiden, ist Text unumgänglich. Die Textbeschriftung erfolgt meist aus einem oder mehreren Attributen des Features. Mit Formatierungs-Tags besteht die Möglichkeit, den Inhalt von verschiedenen Spalten unterschiedlich darzustellen. Ein Spalteninhalt wird fett geschrieben und die anderen Spalteninhalte werden in einer neuen Zeile und kursiv ausgegeben.

Tabelle 4: Beschriftungsbeispiel Widmung

Widmung	Min	Max
WA	0,1	0,4

**WA**  
*0,1 – 0,4*

Darüber hinaus besteht innerhalb eines Geodatensatzes die Möglichkeit, Beschriftungsklassen zu bilden und die Beschriftungen entsprechend der Klasse unterschiedlich zu gestalten. Kleine Flächen können etwa nur mit der Widmung (z. B. WA) beschriftet werden und für große Fläche wird die minimale und maximale Bebauungsdichte hinzugefügt. Natürlich muss nicht mit Format-Tags gearbeitet werden, sondern jeder Klasse kann eine Standardbeschriftung zugewiesen werden, wobei verschiedene Schriftarten, Kursivschriften, Fettschriften, Schriftgrößen und Farben vorgesehen sind. Textfreistellungen (Halo), Masken oder Schatten vervollkommen die Möglichkeiten.

Haben sehr viele Beschriftungen zu erfolgen, werden gerne kleine Schriftgrößen verwendet, die nur mehr gut lesbar sind, wenn sie freigestellt werden. Die Schrift wird dabei mit einem Rand in Kontrastfarbe umgeben, sodass sich der Buchstabe gut vom Hintergrund abhebt. Je weniger Serifen vorkommen umso besser ist kleines Schriftgut zu lesen.

Im Allgemeinen ist die Menge an Text, welche in einer Karte unterzubringen ist, das gravierendste Problem. Der FWP und der OEP stellen keine Ausnahme dar. Es gilt, zu vermeiden, dass sich Beschriftungen überlagern oder wesentliche Features verdecken. Die Standard Label Engine von ESRI stellt dafür Verdrängungsalgorithmen zur Verfügung, welche nach verschiedenen Positionen sucht. So kann beispielsweise ein Punkt vorzüglich links

oben im Abstand von 0,3 cm beschriftet werden. Kommt es jedoch zu einem Konflikt mit einem anderen Element, erfolgt die Beschriftung recht unter dem Punkt in möglicherweise einem größeren Abstand. Welche Beschriftung weichen muss, wird durch Gewichte gesteuert. Auch Featureklassen können gewichtet werden, um deren Objekte nicht hinter Schrift verschwinden zu lassen. Neben der Standard Label Engine von ESRI ist die Maplex Label Engine integriert, welche noch ausgefeilter vorgeht. Sie ist langsamer.

Die Annotation ist ein eigener Datentyp. Diese complex feature classes entstehen vorwiegend durch Überführung von dynamischen Beschriftungen in starren Text, wobei jedem Textelement der Einsetzpunkt der Beschriftung, die Ausrichtung des Textes, die Schriftart und vieles mehr beigelegt werden. Es entsteht eine starre Beschriftung, die dann meist von Hand überarbeitet wird, um ein kartografisch schönes Schriftgut zu haben, das persistent in einer Datenbank vorliegt.

Je aufwendiger die Beschriftung selbst bzw. das Auffinden des Ortes der Beschriftung ist, desto langsamer erfolgt das Setzen des Textes. Kommt es aufgrund des Platzmangels zu vielen Verdrängungen, muss die neue Position erst aufwendig ermittelt werden. Besonders schwerwiegend schlägt die Vermeidung von Überlagerungen mit Features zu Buche. Format-Tags, verschiedene Beschriftungsklassen, welche durch komplexe SQL-Abfragen gebildet werden, kosten Rechenzeit.

Erlaubt man in jeder Featureklasse eine einzige Art der Beschriftung, was nur durch Duldung von sich überlagernden Beschriftungen möglich ist, bzw. man akzeptiert eine Unterdrückung der Beschriftung, falls nicht ausreichend Platz an der einen Position bereitsteht, hat man eine sehr performante Beschriftung vorliegen, was auf Kosten einer schönen Kartografie geht.

In einer Vorprozedur kann eine Annotation erstellt werden, welche Schriftgut enthält, das von der Featureklasse gelöst ist. Ist das gesamte Schriftgut einer Karte in einer einzigen Annotation gesammelt, kann das Labeling schnell sein. Diese Vorprozedur ist im Allgemeinen nicht zeitkritisch und es ist möglich, komplexe Beschriftungsregeln anzuwenden. Bei jedem geänderten Feature muss die Annotation jedoch neu erstellt werden. Das schöne Kartenbild und die performante Darstellung werden durch Redundanzen erkaufte. Man hat ein komplexeres Datenmodell. Darüber hinaus wirkt die Annotation äußerst kontraproduktiv, wenn neben einer Annotation noch andere dynamische Beschriftungen erfolgen. Die starre

Annotation nimmt viel Platz ein. Auch das Arbeiten mit mehreren Annotationen ist problematisch.

Textfreistellungen sind rechenintensiv. Schatten in Kontrastfarben wirken ähnlich und sind schneller. Die verwendete Schriftart ist kaum bedeutend. Trotzdem sollten für kleine Beschriftungen einfache Schriften (z. B. Arial) verwendet werden, da sie einfacher zu lesen sind. Viele gute Tipps sind in ESRI Help enthalten [50].

#### 4.5.4.10.4. *Gecachter Kartenservice*

Im Fall des FWP's und des OEP's wurden hinreichend schnelle Lösungen durch die in Kapitel 4.5.4. beschriebenen Maßnahmen gefunden. Es ist jedoch nicht auszuschließen, dass die Anforderungen an die Kartografie weiter steigen. In diesem Fall müsste mit einem gecachten Kartenservice gearbeitet werden. Dabei wird ein regelmäßiges Raster an Kartenbildern vorab gerendert und bei Bedarf wird auf die entsprechenden Bilder zugegriffen. Mehrere Maßstabsebenen werden unterstützt, wobei für jeden Maßstabsbereich eigens aufgelöste Bilder anfallen. Es ist nicht notwendig, die Bilder vor ihrer ersten Verwendung zu erstellen. „On the fly caching“ legt die Rasterbilder erst an, wenn sie das erste Mal angefordert werden, und speichert dann die Bilder. Derjenige User, welcher ein Gebiet zum ersten Mal aufruft, hat einen verhältnismäßig langsamen Service vorliegen. Für nachfolgende Abfragen des Gebietes stehen die Bilder dann bereit, was zu schnellen Antworten führt. Konzentriert sich das Userverhalten auf wenige Hot-Spots, kommen fast alle in den Vorteil eines performanten Dienstes.

Im gegenständlichen Fall des FWP's und OEP's erfolgt die Veröffentlichung erst am Tag nach der Lieferung der Geodaten. Es ist ausreichend Zeit vorhanden, einen gecachten Service für alle sich ändernden Gemeinden zu rechnen. Kein Anwender wird durch längere Wartezeit vergrämt. Komplexe Signaturen, welche Hilfeebenen benötigen, sind ebenfalls denkbar, da sie in einer Vorprozedur erstellt werden können. Das würde vor allen dicke Randlinien betreffen, welche als Buffer gerechnet und flächig dargestellt einen schöneren Verlauf bei spitzen Ecken liefern [99].

Im Allgemeinen liegen viele verschiedene Layer vor, welche unterschiedlich komplexe Darstellungen aufweisen. MXDPERFSTAT ist eine kostenfreie Software von Andrew Sakowicz, welche langsame Symbolik, schlecht getunte Datenbanken, „Projection on the fly“ etc. aufzeigt. Sie kann unter folgender Adresse heruntergeladen werden: <http://resources.arcgis.com/en/help/main/10.2/index.html#//00s600000014000000>

#### 4.5.4.10.5. *Hardwarekonfiguration – Lastverteilung*

Ein Mittel der Performancesteigerung ist die Lastverteilung. Mehrere Server arbeiten parallel an einem Problem. Die allgemeine Informatik verwendet Lastverteilung gerne bei großen Mengen an HTTP-Anfragen, die schnell beantwortet werden wollen. Ein Load-Balancer (Frontend-Server) erhält die Anfragen und teilt sie den bereitstehenden Rechnern entsprechend ihrer Auslastung zu. Werden auch die Daten redundant auf jedem Rechner vorgehalten, verkraftet das System den Ausfall eines Computers.<sup>27</sup> Es wird nicht nur die Performance gesteigert, sondern auch die Ausfallsicherheit. So können Server-Farmen entstehen, die nicht nur schnell sind, sondern auch hochverfügbar (high availability).

Der Load-Balancer, oder besser gesagt seine Internetadresse, erhält alle Anfragen. Das eintreffende Datenpaket ist an seinem scheinbaren Ziel angelangt. Nun wird vom Load-Balancer die physikalische Adresse (MAC-Adresse) ausgetauscht. Es erhält die Adresse eines wenig belasteten Rechners des Verbundes und das Datenpaket wird über einen Switch weitergereicht. Der neue Zielrechner verarbeitet das Datenpaket und antwortet direkt der Ausgangsadresse. Damit laufen die Antworten nicht über den Load-Balancer, womit er entlastet wird. Dieses Verfahren wird Flat based SLB genannt.

Das Domain-Name-System (DNS) übersetzt die für Menschen halbwegs gut merkbaren URL-Adressen (z. B. [gis.steiermark.at](http://gis.steiermark.at)) in für Maschinen verarbeitbare IP-Adressen (etwa 192.958.234.267). Werden einem Hostnamen mehrere IP-Adressen hinterlegt, die wechsel-

---

<sup>27</sup> Nicht jedes System erkennt den Ausfall eines Rechners und ist damit imstande, die Ausfallsicherheit zu erhöhen. Es gibt auch Systeme, welche mehrere CPUs bedienen, aber auf eine Datenbasis zugreifen, die natürlich ausfallsicherer als ein Raid-System (Redundant Array of Independent Disks) ausgebaut sein kann.

seitig als Ergebnis von Anfragen zurückgeliefert werden, kann das System als Load-Balancer genutzt werden. Das System wird DNS Round Robin genannt. „Das ist die einfachste Möglichkeit der Lasterteilung“ [51]. Damit ist die Lösung kostengünstig.

Das Management eines solchen Systems ist aufwendig und große Serverfarmen haben auch entsprechenden Energie- und Platzbedarf. Entscheidende Faktoren für den Betrieb eines Load-Balancers sind die Anzahl an Clients, ihre Anfrageraten, die zu übertragenden Datenmengen und der Aufwand, die Daten bereitzustellen.

Der FWP und der OEP werden über das Web-GIS des Landes publiziert. Der typische Client verschiebt das Kartenbild vielfach, was zu einer hohen Anfragerate führt. Darüber hinaus ist der Rechenaufwand zur Ermittlung eines Kartenbildes beträchtlich. Lediglich die zu übertragende Datenmenge ist im Vergleich zu Videosevernen etc. gering. Zwei parallel laufende Maschinen sind in der Lage, die Aufgaben befriedigend zu erfüllen, wobei das Server-Load-Balancing über DNS erfolgt, da das System billig ist und die zu retournierenden Daten nicht allzu groß sind. Fällt eine IP-Adresse aus, werden alle Anfragen von der verbleibenden Adresse übernommen, womit die Ausfallsicherheit erhöht wurde.

Aus der Sicht des Nutzers soll das Ergebnis in Echtzeit am Bildschirm erscheinen. Baut man ein solches System und hat nur beschränkte finanzielle Mittel zur Verfügung, wird man Abstriche eingehen müssen. Die Meinungen, welche Antwortzeiten noch vertretbar sind, gehen weit auseinander. Es gibt jedoch eine Minimalanforderung, auf die man sich berufen kann. Laut INSPIRE gilt:

„Die Antwortzeit für das Senden eines ersten Ergebnisses auf eine Suchdienstanfrage beträgt in einer normalen Situation höchstens 3 Sekunden.

Für ein Bild mit 470 Kilobyte (z. B.  $800 \times 600$  Pixel mit einer Farbtiefe von 8 Bit) beträgt die Antwortzeit für das Senden eines ersten Ergebnisses auf eine „Get Map“-Anfrage an einen Darstellungsdienst in einer normalen Situation höchstens 5 Sekunden.

Mit einer normalen Situation ist ein Zeitraum ohne Spitzenbelastung gemeint. Eine normale Situation ist 90 % der Zeit gegeben.“ [52]

Eine entsprechende Überwachung des Systems hat ergeben, dass die bereitgestellte Hardware hinreichend ist.

Viele Rechenzentren bedienen sich extrem leistungsstarker Computer, auf denen mehrere virtuelle Rechner laufen. Auch im Amt der Steiermärkischen Landesregierung wird von dieser Technologie Gebrauch gemacht, da es billiger ist, einige große Rechner zu betreiben als viele kleine Maschinen.

Der Vorteil virtueller Rechner besteht darin, dass man ihnen Ressourcen zuordnen kann. Ein Administrator ist in der Lage, jedem virtuellen Server auf dem tatsächlichen Bedarf hin, innerhalb der zur Verfügung stehenden Gesamtkapazitäten, anzupassen. Das kann auch temporär erfolgen. So kann bei einer Wahl dem virtuellen Server der „Abteilung für Gemeinden und Wahlen“ mehr Leistung zugeordnet werden, die den Rest des Jahres nicht benötigt wird und anderen Abteilungsservern zugeordnet ist. Ein Cluster mit Lastverteilung ist hierfür geeignet. Die benötigten Leistungsparameter müssen nur entsprechend zugeordnet werden. Das bedingt natürlich, dass ein Großrechner grundsätzlich in der Lage ist, diese Last und weitere Lasten zu stemmen. Kurzzeitige und vorhersehbare Spitzen sind so gut in den Griff zu bekommen.

Der Betrieb von virtuellen Servern erhöht das Risiko für das Gesamtsystem. Der Rechner sollte über eine zweite Stromversorgung verfügen, im Sinne eines redundanten Netztes. Damit kann ein Stecker kaputt gehen und das zweite Netzteil übernimmt seine Funktion. Das gilt auch für andere Hardwarekomponenten wie Netzwerkkarten etc. Virtuelle Maschinen laufen im Hauptspeicher, weshalb man sich etwa durch „Memory Mirroring“ (permanente Spiegelung des Hauptspeichers und gegebenenfalls Zugriff auf die Kopie) gegen den Ausfall von Hauptspeichermodulen absichern kann. Der Rechner kann in einem Cluster mit einer zweiten Maschine betrieben werden, wobei ein Rechner die Aufgaben des anderen übernehmen kann, was die Verfügbarkeit erhöht. Entscheidend ist natürlich, wie schnell die zweite Maschine den Dienst der ersten Maschine ersetzen kann.

Ein Virus kann einfacher alle virtuellen Server in Mitleidenschaft ziehen. Er braucht sich nicht mehr über ein Netz verteilen.

#### 4.5.5. Sicherheit

##### 4.5.5.1. Problembeschreibung und Begriffsbestimmung

Sicherheit als Qualitätseigenschaft von Software beschreibt die Fähigkeit von Systemen, unberechtigte Zugriffe auf Programme und Daten zu verhindern. Unberechtigte Zugriffe können sowohl vorsätzlich als auch zufällig erfolgen (vgl. ISO/IEC 9126).

Sicherheit gliedert sich in

- a. Vertraulichkeit (Confidentiality),
- b. Integrität (Integrity),
- c. Verfügbarkeit (Availability).

Für diese Untergliederung wird gerne die Abkürzung CIA verwendet. Diese Schlagworte stehen für

- den Schutz der Information vor unbefugter Kenntnisnahme und Benutzung(C),
- die Richtigkeit und Vollständigkeit der Information(I),
- sowie die jederzeitige Verfügbarkeit der Information(A) [53].

Die Vertraulichkeit der Information wird nicht nur auf den Speichermedien bedroht, sondern kann auch beim Datentransfer und der Datenverarbeitung Gefahren ausgesetzt sein. Die Richtigkeit und die Vollständigkeit der Information (Integrität) können durch illegale Manipulation oder unbewusste Bearbeitungen verloren gehen, wenn etwa Systemverwalter Reparaturen durchführen und Teilbereiche vergessen. Die Verfügbarkeit ist dann gegeben, wenn die Daten autorisierten Nutzern zum vereinbarten Zeitpunkt in der genormten Form auch tatsächlich bereitstehen.

Eine Information ist authentisch, wenn ihre Echtheit zweifelsfrei feststeht (Authenticity). Die Forderung der Integrität führt dazu, dass Änderungen an Daten nachvollziehbar sein müssen. Man muss wissen, wer was geändert hat (Zurechenbarkeit – accountability) und die Person soll die Manipulation nicht abstreiten können (non repudiation).

„Dieser Schutzbedarf ist nicht neu. Denken wir an die Siegelung von Dokumenten zum Zweck des Nachweises ihrer Herkunft zurück. Heute ist das die digitale Signatur“ [54]

Genau in die andere Richtung verweist die Anonymität. Es gibt EDV-Lösungen, bei denen hoher Aufwand getrieben werden muss, um die Person, welche das System nutzt, geheim zu halten. Ein Beispiel sind etwa „Whistleblowing-Systeme“, wo Mitarbeiter von Unternehmen Missstände melden können, die sie aufgrund ihrer hierarchischen Einbindung und aufgrund von Angst vor Repressionen Vorgesetzten nicht zur Kenntnis bringen können (vgl. Whistleblowing im Arbeitsrecht [55]).

#### 4.5.5.2. Symmetrische und asymmetrische Kryptografie

Um Nachrichten vor unberechtigter Nutzung zu schützen, werden diese verschlüsselt. Bei Symmetrischen kryptografischen Verfahren verfügen sowohl der Sender als auch der Empfänger einer Nachricht über die exakt gleichen Schlüssel, mit denen sie die Nachricht sowohl verschlüsseln als auch entschlüsseln können. Dieses Verfahren ist sinnvoll, wenn der Schlüssel über einen sicheren Kanal ausgetauscht werden kann.

Asymmetrische Kryptografie arbeitet mit zwei unterschiedlichen Schlüsseln. Der eine Schlüssel wird als öffentlicher Schlüssel (public key) bezeichnet und der andere Schlüssel als privater Schlüssel (private key). Mit dem öffentlichen Schlüssel können Nachrichten nur verschlüsselt, aber nicht entschlüsselt werden. Möchte man von jemandem eine geheime Nachricht erhalten, übermittelt man ihm den öffentlichen Schlüssel. Der Empfänger verschlüsselt seine Information und überträgt sie. Fängt jemand den öffentlichen Schlüssel und die geheime Nachricht ab, sind beide Informationen für den Dieb wertlos, da mit dem öffentlichen Schlüssel die Nachricht nicht entschlüsselt werden kann. Das ist nur mit dem privaten Schlüssel möglich, der nie über das Internet übertragen wurde.

Eine gute Einführung in die Kryptografie findet man im Buch „Sieben Wunder der Informatik“ [56, SS. 217–246].

Das Verfahren ist auch geeignet, eine elektronische Unterschrift zu leisten. Nachstehendes soll den Vorgang verdeutlichen.

Verschlüsseln und Entschlüsseln sind langwierige Rechenoperationen. Deshalb bildet man den Hashwert eines Dokuments. Unter Hash wird eine kurze Zeichenfolge verstanden, die aus Dokumenten mithilfe eines schnellen Rechenalgorithmus erstellt werden kann. Diese Zeichenfolge ist wesentlich kürzer als das Dokument selbst und, auch nur minimal unterschiedliche Dokumente führen zu verschiedenen Hashwerten. Der Hashwert wird mit dem privaten Schlüssel verschlüsselt und dem Dokument beigelegt. Dieser Code hat unabhängig von der Größe des Dokuments eine fixe Länge (z. B. 160Bit) und bildet die elektronische Unterschrift. Der Empfänger des Dokumentes lädt sich den öffentlichen Schlüssel von einer Zertifizierungsstelle herunter und decodiert den beigelegten Hashwert. Nun errechnet der Empfänger aus dem erhaltenen Dokument abzüglich der elektronischen Unterschrift erneut den Hashwert und vergleicht den decodierten Hashwert mit dem gerechneten Hashwert (siehe [100][101]). Sind diese Hashwerte gleich, wurde das Dokument nicht verändert. Man kann Dateien elektronisch unterschreiben, und jedermann, der dieses File erhält, kann mit dem öffentlichen Schlüssel überprüfen, ob das Dokument unverändert ist. Die öffentlichen Schlüssel stehen auf der Homepage einer Zertifizierungsstelle zum Download bereit. Darüber hinaus weiß man, dass der Unterzeichner über den privaten Schlüssel verfügt. Zertifizierungsstellen geben diese privaten Schlüssel aus und listen auf, wer welchen Schlüssel erhalten hat. Die Zertifizierungsstelle garantiert die Zuordnung zu Personen, welche ihren Schlüssel – das Zertifikat – nicht weitergeben sollten. Das System von öffentlichen und privaten Schlüsseln erfüllt die Forderung der Vertraulichkeit.

#### 4.5.5.3. Sicherheit und Recht

##### 4.5.5.3.1. Gesetzlicher Rahmen

Möchte man rechtsrelevante Informationen anbieten, kommt der Sicherheit eine zentrale Rolle zu. Rechtsrelevanz wird nur gewährleistet werden können, wenn die elektronischen Daten die tatsächlichen Verhältnisse widerspiegeln und nicht fälschlicherweise oder durch

bewusste Manipulation geändert wurden (Vertraulichkeit). Bis dato erfolgte noch keine bewusste Falschinformation.<sup>28</sup> Irrtümliche Übermittlungen von alten Planentwürfen, fehlerhafte irrtümliche Attribuierungen kommen vor. Nichtsdestotrotz kann nicht davon ausgegangen werden, dass keine böshaften Manipulationsversuche erfolgen werden. Derzeit haben lediglich die gestempelten Papierpläne Rechtskraft, wobei das Original bei der Gemeinde aufliegt und ein Gleichstück im Amt der Steiermärkischen Landesregierung anzufinden ist. Bei abweichenden digitalen Geodaten kann der tatsächliche Rechtsstand einfach erhoben werden.

Sicherheit kostet viel Aufwand und Mühe, wobei im Amt der Steiermärkischen Landesregierung bereits eine weit fortgeschrittene Sicherheitsarchitektur besteht. So können PDF-Dateien mit Amtssignaturen versehen werden. Die Amtssignatur ist in E-GovG § 18 und § 19 geregelt. (BGBl. I Nr. 10/2004). Dem folgend haben amtssignierte Dateien die gleiche Beweiskraft wie öffentliche Urkunden. Die Amtssignatur beinhaltet eine Bildmarke, die optisch überprüft werden kann. Hat man Zweifel an der Echtheit des Dokumentes kann die PDF-Datei unter der Adresse [https://egov.stmk.gv.at/zza\\_as/](https://egov.stmk.gv.at/zza_as/) hochgeladen werden und die Anwendung kontrolliert, ob es sich tatsächlich um ein Schriftstück des Amtes der Stmk LREG handelt. Die Frage der Echtheit von Landesdokumenten kann jederzeit beantwortet werden. Es existiert Software, welche in der Lage ist, PDF-Dokumente zu signieren.<sup>29</sup> Damit kann das Land Karten und Pläne unterschreiben.

Verlockend erscheint es, Geodaten elektronisch zu unterschreiben. Geoinformatiker könnten in ihrer Welt verbleiben und wären sich sicher, mit korrekten Daten zu arbeiten. Wie elektronische Unterschriften auszusehen haben, ist im Signaturgesetz geregelt und wird in der Signaturverordnung 2008 – SigV 2008 weiter konkretisiert. Nachfolgender Auszug bezieht sich auf das Gesetz:

#### § 19

---

<sup>28</sup> Seit dem Jahr 1993 arbeite ich im GIS-Steiermark und bin unter anderem für die technische Abwicklung der Übernahme von Flächenwidmungsplänen zuständig. Dieser für mich überschaubare Zeitraum beträgt 23 Jahre.

<sup>29</sup> Die Software und das Konzept zur Signierung von Software wurde von der Abteilung 1 des Amtes der Stmk LREG konzipiert und entwickelt. Es steht nun für alle Abteilungen zur Nutzung bereit und ist somit nicht Teil der Dissertation. Es wird lediglich benutzt.

(2) Die Amtssignatur dient der erleichterten Erkennbarkeit der Herkunft eines Dokuments von einem Auftraggeber des öffentlichen Bereichs. Sie darf daher ausschließlich von diesen unter den näheren Bedingungen des Abs.3 bei der elektronischen Unterzeichnung und bei der Ausfertigung der von ihnen erzeugten Dokumente verwendet werden.

(3) Die Amtssignatur ist im Dokument durch eine Bildmarke, die der Auftraggeber des öffentlichen Bereichs im Internet als die seine gesichert veröffentlicht hat, sowie durch einen Hinweis im Dokument, dass dieses amtssigniert wurde, darzustellen. Die Informationen zur Prüfung der elektronischen Signatur sind vom Auftraggeber des öffentlichen Bereichs bereitzustellen.

(Stammfassung BGBl. I Nr. 10/2004, geändert BGBl. I Nr. 7/2008, BGBl. I Nr. 125/2009, BGBl. I Nr. 111/2010, BGBl. I Nr. 83/2013)

Im Absatz (2) formuliert der Gesetzgeber allgemeine Ziele. Die erleichterte Erkennbarkeit als ein amtliches Dokument wird angestrebt. Im Absatz (3) des Paragraphen wird das konkretisiert, indem eine Bildmarke gefordert ist. Damit wird auf elektronischem Weg das alte und bewährte Konzept des Siegels bzw. des Rundstempels weitergeführt. Der Bürger kann auf einen Blick erkennen, dass es sich um ein öffentliches Dokument handelt. Das Konzept der elektronischen Unterschrift benötigt aus technischer Sicht keine Bildmarke, was die Frage aufwirft, warum der Gesetzgeber einen elektronischen Stempel fordert.

Der Bürger sollte möglichst einfach feststellen können, dass es sich um ein amtliches Dokument handelt. Spezielles Wissen über Sicherheit in der EDV soll dafür nicht notwendig sein. Mit der Bildmarke kann das auf einem Blick erfolgen.

In Absatz (2) wird die Bildmarke vor Fälschung geschützt. Der rechtliche Schutz vor Missbrauch wird durch eine Strafbestimmung verstärkt. E-GovG § 22 sieht Verwaltungsstrafen bis 20.000 Euro vor, wenn nicht durch ein anderes Gesetz höhere Strafen zu verhängen sind.

PDF-Dateien können mit Amtssignaturen versehen werden. Geoinformationssysteme können Karten und Pläne in diesem Format erstellen, womit die Ergebnisse rechtliche Bedeutung erlangen können. Das wird im E-GovG § 20 geregelt.

„Ein auf Papier ausgedrucktes elektronisches Dokument einer Behörde hat die Beweiskraft einer öffentlichen Urkunde (§ 292 der Zivilprozessordnung – ZPO, RGBl. Nr.113/1895), wenn das elektronische Dokument mit einer Amtssignatur versehen wurde.“

(Stammfassung BGBl. I Nr. 10/2004, geändert BGBl. I Nr. 7/2008, BGBl. I Nr. 125/2009, BGBl. I Nr. 111/2010, BGBl. I Nr. 83/2013)

#### 4.5.5.3.2. *Rechtsrelevante Geodaten*

Viele Geoinformatiker wünschen sich, dass Geodaten selbst rechtsrelevant werden. Im Bereich des FWP's und des OEP's müssten nur die Geodaten ausgetauscht werden. Abweichungen zum Papierplan wären irrelevant, da den Geodaten der Vorzug gegeben wird. Die Register der Geoinformatiker wären per definitionem richtig. Das würde ihre Bedeutung und damit automatisch ihre Qualität heben.

Jedoch sollte eine Verwaltung davon Abstand nehmen, Geodaten diese Bedeutung zu geben, da nur eine kleine Minderheit mit Geodaten umzugehen versteht. Nur wenige besitzen entsprechende Programme oder können diese bedienen.

Der Gemeinderat beschließt und genehmigt den in Papierform vorliegenden FWP und OEP. Diese Pläne existieren im Allgemeinen auch als PDF-Dokumente, welche wie Papierpläne betrachtet werden können, ohne Spezialwissen zu benötigen. Diese PDF-Dokumente können mit einer Amtssignatur versehen werden.

Die Firma ESRI hat im Jahr 1998 den Aufbau des Geodatenformats Shape mit dem Dokument „ESRI Shapefile Technical Description“ [58] offengelegt. Dieses Format hat sich zum Quasistandard entwickelt und wird von vielen Herstellern von Basissoftware im Bereich GIS unterstützt. Ursprünglich bestand ein Geodatensatz aus lediglich drei Dateien, wobei eine Datei die Geometriedaten (.shp) enthält, eine weitere die Attributdaten (.dbf) und eine dritte Datei setzte die Geometrie mit den Attributen in Beziehung (.shx). Neben dieser Basis entstanden mit der Zeit weitere Dateien mit unterschiedlichen Aufgaben. Ein Beispiel ist die prj-Datei, welche die Projektionsbeschreibung enthält und unabdingbar für die Funktionalität der „projection on the fly“ ist. Denkbar ist eine weitere zusätzliche Datei, welche für jede Basisdatei (.shp, .dbf, .shx) die verschlüsselten Hashwerte enthält. Damit wäre der

Geodatensatz unterschrieben. Die formalen Kriterien einer Amtssignatur sind jedoch nicht gegeben, da keine Bildmarke vorliegt und nur mit Spezialprogrammen die Information gelesen werden kann.

#### *4.5.5.3.3. Eingangsforderung der Sicherheit*

Die elektronische Unterschrift garantiert die Integrität (integrity) der Daten. Jede Veränderung der Daten kann einfach erkannt werden. Es ist gewährleistet, dass man keine gefälschte Version erhält, da der Absender geprüft werden kann. Fasst man den Begriff der Identität weiter, so beinhaltet er auch die Richtigkeit der Daten. Das ist ein derzeit existierendes Problem. Es kann nicht gesagt werden, ob die elektronischen Daten mit den rechtsgültigen Papierplänen übereinstimmen und ob alle kleinen Änderungsverfahren digital vorliegen. Dieses Schlüsselproblem ist mit Softwaretechnik nicht lösbar. Vielmehr muss an organisatorischen und gesetzlichen Maßnahmen gearbeitet werden, um dieses Ziel zu erreichen. Diese Maßnahmen sind im Kapitel 3.6 Lösungsansatz FWP angeführt.

Vertraulichkeit (confidentiality) kann durch die elektronische Unterschrift nicht garantiert werden, da der eigentliche Inhalt des Dokuments im Klartext übertragen wird. Asymmetrische Kryptografie kann auch das leisten, wie im Eingangsbeispiel von Kapitel 4.5.5.2 gezeigt wurde. Der FWP und der OEP legen auf hohe Transparenz und Bürgerbeteiligung wert. Vertraulichkeit wird nicht benötigt.

Die asymmetrische Kryptografie setzt voraus, dass der private Schlüssel nicht in falsche Hände kommt. Absolute Sicherheit kann also nicht garantiert werden. Asymmetrische Kryptografie sorgt dafür, dass erhebliche kriminelle Energie für unerlaubte Manipulationen notwendig wird. Im Kapitel 3.6 wurde ein Verfahrensablauf für FWPs und OEPs vorgeschlagen, welches mit digitalen Signaturen arbeitet. Der Plan, der signiert als PDF-Datei vorliegt, wird ausgedruckt und wie üblich in Papierform rechtlich beschlossen. Elektronische Manipulationen fallen bei der Diskussion im Gemeinderat und Raumordnungsbeirat auf. Problemgebiete werden immer genau untersucht und kriminelle Manipulationen sind hauptsächlich dort zu erwarten.

#### 4.5.6. Verfügbarkeit

Die Verfügbarkeit ist dann nicht gegeben, wenn das gewünschte Ergebnis nicht geliefert werden kann. Verfügbarkeit wird als Prozentzahl angegeben, wobei gilt:

Verfügbarkeit = tatsächliche Servicezeit / geplante Servicezeit.

Die einzelnen Qualitätskriterien korrelieren sehr stark miteinander, was sich insbesondere bei der Verfügbarkeit zeigt. Eine hohe Verfügbarkeit wird unter anderem durch gute Wartbarkeit erreicht, da bei Softwarefehlern schnelle Reparaturen möglich sind. Die Korrektheit beschreibt nahezu Ähnliches, denn wenn alles korrekt funktioniert, ist es auch verfügbar. Korrektheit fokussiert jedoch auf Softwaretechnik und weniger auf Netzwerk und Ähnliches.

Verfügbarkeit kann unter mangelnder Geschäftskultur leiden. Software im laufenden Betrieb zu installieren oder zu updaten, ist ein Problem für die Verfügbarkeit aller Anwendungen, welche noch auf dem Rechner laufen, da die Maschine häufig neu zu booten ist.

Die Systemverwaltung fokussiert sich auf die Basis, welche sie bereitstellt und sieht laufende Anwendungen als sekundär an. Deshalb werden bei Systemarbeiten Risiken für laufende Anwendungen eingegangen. Für den User, der im Mittelpunkt stehen sollte, ist es jedoch nur langfristig wichtig, mit aktuellen Basiskomponenten zu arbeiten. Die Anwendung ist für ihn zentral. Bei komplexen Systemen ist es sehr schwierig, vorherzusehen, wie sich Systemänderungen auswirken. Deshalb ist es unabdingbar, regelmäßige Servicefenster einzurichten. Der User ist gewohnt, dass zu diesen Zeitpunkten der Dienst nicht bereitsteht, und kann mit organisatorischen Maßnahmen das Zeitfenster überbrücken. Ideale Servicefenster sind betriebsschwache Zeiten der Software.

Betrachten wir das Gesamtsystem, sollten wir in Hinblick auf die Verfügbarkeit zwischen den Upload der FWPs und die Darstellung im Internet für die breite Öffentlichkeit.

Für den Upload gilt: Der Planungsprozess, die Diskussion und gegebenenfalls die Revision des FWPs sind aufwendig und dauern zumindest ein halbes Jahr. Im Allgemeinen wird mit noch längeren Zeiträumen zu rechnen sein. Angesichts dieser Zeithorizonte ist ein kurzzeitiger Ausfall etwa des Portals zum Upload der Geodaten unproblematisch. Werden aus den

Geodaten Pläne erstellt (siehe Kapitel 3.6), welche von politischen Gremien (Gemeinderat, AROB<sup>30</sup>) zu beschließen sind, muss die Funktionalität zuverlässig bereitstehen, da der eine oder andere Raumplaner den Internetservice im letzten Augenblick nutzen wird. Führt man eine probabilistische Sicherheitsanalyse durch (Was kann versagen? Wie wahrscheinlich ist es? Was sind die Auswirkungen?), wird man bei der Frage nach den Auswirkungen von fehlender Verfügbarkeit im schlimmsten Fall mit einer Verschiebung einer Gemeinderats-sitzung bzw. Ausschusssitzung rechnen müssen. Es treten Verluste in der Größenordnung von 5–10 Arbeitsstunden auf. Der indirekte Schaden aufgrund des schlechten Images ist als wesentlich gravierender einzuschätzen und kann in der Frühphase des Systems das Ende dieser Vorgehensweise bedeuten.

Für die Internetdarstellung gilt: Der Verfügbarkeit der Daten kommt eine hohe Bedeutung zu. Nach den Orthofotos und dem Grundstückskataster ist die Flächenwidmung der drittbeliebteste Web-Service des Landes. Im Jahr 2014 waren es 6915 längere Arbeitssitzungen,<sup>31</sup> welche die Datenbasis Flächenwidmung verwendeten. Widmungsinformation kann bequem über das Internet abgefragt werden und der Bürger muss nicht in das Gemeindeamt fahren oder dort anrufen, um zu erfahren, wie das Grundstück gewidmet ist. Eine große Gefahr besteht, da der derzeitig dargestellte Sachverhalt von der tatsächlichen rechtlichen Situation abweicht. Nur im Gemeindeamt bzw. der Abteilung 13 liegen rechtsgültige analoge Pläne vor und die digitalen Daten weichen teilweise davon ab. Aus Telefonaten mit Usern, die Spezielles wissen wollten, geht hervor, dass das Service von folgenden Gruppen genutzt wird:

- Bürger, welche einen Baugrund suchen;
- Immobilienmakler;
- Bankangestellte, welche Kredite besichern lassen wollen;
- Rechtsanwälte mit diversen Fragestellungen.

Jeder Ausfall verursacht zusätzliche Arbeit bei den Betroffenen. Die derzeitige digitale Vorinformation entfällt und zumindest ein Telefonat mit dem Gemeindeamt ist notwendig. Würden die digitalen Daten qualitativ soweit verbessert werden, dass sie mit rechtsgültigen

---

<sup>30</sup> AROB Ausschuss für Raumordnung und Baurecht des Landtages

<sup>31</sup> Arbeitssession über 2 Minuten

Plänen sicher übereinstimmen, würden viele Behördenwege gänzlich entfallen und die Rechtssicherheit würde steigen.

Der digitalen Verfügbarkeit des Web-Services kommt eine hohe Bedeutung zu. Der volkswirtschaftliche Schaden ist auf jeden Fall hoch einzuschätzen. Abschätzungen sind immer schwierig und beruhen auf vielen Annahmen und Handlungsszenarien.

Fällt der Dienst nur kurzzeitig aus, tritt kein nennenswerter Schaden auf, da diese Information nur in den seltensten Fällen ad hoc benötigt wird. Steht der Dienst lange nicht bereit (1–2 Tage), besorgt sich der User die Information auf anderem Weg. Er wird im Gemeindeamt anrufen und die Widmung fernmündlich erfragen, was einen zusätzlichen Beamten sicherlich 10 Minuten bindet, da er die Information ausheben muss. Weitere 2 Minuten sind anzusetzen, weil nicht davon ausgegangen werden kann, dass man sofort mit dem zuständigen Beamten spricht und man erst weitergeleitet werden muss. Eine weitere Minute kann angesetzt werden, da der Bedienstete aus seiner bisherigen Tätigkeit herausgerissen wurde und nach Auskunftserteilung wieder in die alte Arbeit hineinfinden muss. Eine telefonische Anfrage, die sowohl den Beamten als auch den Bürger bindet, verursacht einen Aufwand von ca. 26 Minuten. Ein User benötigt pro Sitzung viele Kartenbilder des digitalen FWPs. Insgesamt kam es im Jahr 2014 zu 6915, längeren Sessions in denen FWPs abgefragt wurden. Ist der komplizierte Weg der telefonischen Anfrage zu gehen, wird sicherlich nur die Hälfte abgefragt. Gehen wir von 70 Euro Schaden pro Stunde aus, ergibt sich der Schaden für einen Tag Ausfall in Höhe von:

$$70 \text{ €} * 26 / 60 * 6915 / 365 = 574 \text{ €}.$$

Die Verfügbarkeit des Web-Services ist quantitativ geschätzt, wobei die anfallenden Ersatzarbeiten in ihrem Ausmaß geschätzt wurden. Deshalb handelt es sich um eine Größenordnung, und nicht um eine exakte Größe.

Der Ausfall des Service für wenige Tage im Jahr ist vertretbar, wenn auch nicht wünschenswert. Die Effizienz wurde im Kapitel 4.5.4 ausführlich behandelt, da mangelnde Ressourcen zu Systemabstürzen führen und jeder Absturz zum Verlust von Kunden führt.

Die Verfügbarkeit kann auch gehoben werden, wenn die Monopolstellung des Landes mit dieser Information aufgeweicht wird. Das Land selbst könnte den Gemeinden Dateien bereitstellen, mit denen sie den FWP bzw. den OEP im Netz darstellen können. Schon in den 90er Jahren wurden vom Land FWPs nach dem Konzept der Klickable Maps angeboten [60]. Diese HTML-Dateien und Bilddateien lassen sich nach entsprechender Programmierung automatisch erstellen und könnten den Gemeinden überlassen werden. Die einfache Struktur ermöglicht es jeder Gemeinde, den Plan in ihre Website einzubinden.

Was hier beispielhaft verbal erfolgte, lässt sich auch schematisieren und wird in der Literatur als Risikodiagramm [61] bezeichnet. Es findet in vielen Branchen Anwendung, wie in der Textilindustrie, in der Medizin oder bei der Softwareentwicklung. Mögliche Schadensfälle werden finanziell bewertet und die Auftretenswahrscheinlichkeit wird geschätzt. Die Auftretenswahrscheinlichkeit wird auf der Abszisse aufgetragen und das zugehörige Schadensausmaß als Ordinatenwert.

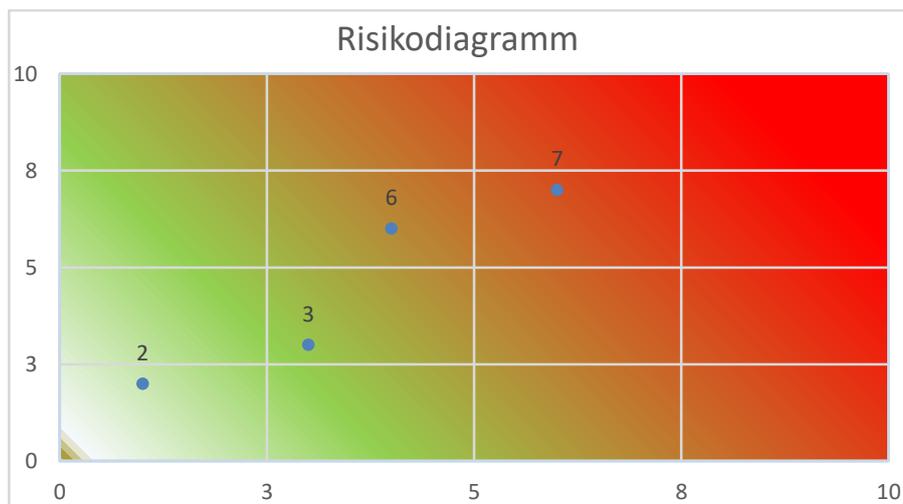


Abbildung 13: Risikodiagramm

Tabelle 5: Risikoanalyse

Schaden	P	S
A	3	3
B	1	2
C	4	6
D	6	7

P... Auftretenswahrscheinlichkeit mit Werten von 1-10, wobei 1 für sehr gering und 10 für sehr hoch steht.

S ... Schadensausmaß mit Werten von 1-10, wobei 1 für sehr gering und 10 für sehr hoch steht.

Die Größen werden geschätzt. Je weiter man in den roten Bereich kommt, desto dringender ist ein Handeln erforderlich. Der Vorteil der Methode liegt darin, dass man sich mit Risiken beschäftigt, was häufig unterlassen wird. Viele Projekte ignorieren mögliche Probleme.

Es wird versucht, das Ausmaß des Schadens in Kombination mit der Auftretenswahrscheinlichkeit zu bewerten, was eine genauere Abschätzung des Problems bringt und zu einem besseren Mitteleinsatz führt.

Es liegt in der Natur der Sache, dass man nicht alle Risiken erkennen wird. Man kann jedoch die Problemfelder vieler Projekte zusammentragen und so eine Checkliste aufstellen. Auch im Internet sind solche Listen zu finden.

Das Risikodiagramm kennt viele Spielarten. So können die Rollen der X- und Y-Achse vertauscht sein und unterschiedliche Skalen eingeführt werden. Ludewig und Lichter arbeitet etwa mit geschätzten Schäden in Euro und Auftretenswahrscheinlichkeiten in Promille, die logarithmisch abgetragen werden [12, SS. 132 -135].

Dem aufmerksamen Leser ist sicher die Nahebeziehung zur Zuverlässigkeit (reliability) aufgefallen, wobei diese Eigenschaft eine höhere Affinität zur Korrektheit aufweist und die Verfügbarkeit auch durch überlastete Netzwerke etc. beeinträchtigt wird.

#### 4.5.7. Wartbarkeit

Zur Definition der Wartbarkeit zitiert H. Balzert die Norm ISO/IEC 9126-1:

„Fähigkeit des Softwareprodukts änderungsfähig zu sein. Änderungen können Korrekturen, Verbesserungen oder Anpassungen der Software an Änderungen der Umgebung, der Anforderungen und der funktionalen Spezifikationen einschließen.“  
[62, S. 116]

Wartbarkeit ist also die Eigenschaft, welche beschreibt, wie einfach ein System verändert werden kann. Im Kapitel 4.5.6 wurde die Verfügbarkeit besprochen. Ist ein System gut wartbar, wird weniger Zeit für die Reparatur bei Fehlern benötigt und die Verfügbarkeit erhöht sich.

H. Balzert erläutert, dass die Wartbarkeit von der Analysierbarkeit, der Änderbarkeit, der Stabilität und der Testbarkeit beeinflusst wird [62, SS. 116–117].

Die Analysierbarkeit eines Quellcodes wird stark von der Lesbarkeit beeinflusst. Sprechende Variablen, Kommentarzeilen und geringe Komplexität verbessern die Lesbarkeit des Programms. Programme führen Abstraktionsebenen ein und versuchen so, komplexe Sachverhalte in beherrschbare Einheiten zu überführen. Diese abstrakten Gebilde sind ebenfalls wesentlich für die Analysierbarkeit des Codes. Sie sollten möglichst bekannte Einheiten abbilden, die mit dem Realleben gut in Verbindung gebracht werden können. Eine Klasse kann eine solche Abstraktionsebene sein. In diesem Programm wurden Verwaltungsläufe, wie sie im Amtsbetrieb üblich sind, als solche Einheiten gewählt. Das sind etwa die Abgabe und der Kontrollakt, aber auch andere verständliche Einheiten wie die Reparaturwerkstatt. Jede Funktion wurde im Quellcode beschrieben, wobei die Beschreibung der Funktion entsprechend der Python Syntax unter dreifachen Doppelhochkomma gestellt wurde (z. B. `"""Text"""`). Das bewirkt, dass dieser Text mithilfe der Standardfunktion `__doc__` aufgerufen werden kann. Input, Output und Vorbedingungen werden aufgelistet.

Wurde ein Programm verstanden, kann es abgeändert werden. Es ist schwierig, ein Gesamtsystem zu überblicken.

„Der Begriff Modularisierung beschreibt die Möglichkeit, ein System aus Teilsystemen aufzubauen, die wiederum als Einzelkomponenten (Modulen) entfernt, hinzugefügt oder durch andere ersetzt werden können.“ [63, S. 11]

Hier spielt das Geheimnisprinzip eine wichtige Rolle. Jedes Modul ist nur durch eine definierte Schnittstelle erreichbar. Innerhalb des Moduls wird mit lokalen Variablen gearbeitet, die von außen nicht sichtbar sind. Datenkapselung und Modularisierung sind wesentliche Elemente der Objektorientierung. In Python können lokale Speicherräume gebaut werden, indem die Variablennamen mit `__` beginnen. In analoger Weise können Funktionen und

Methoden einer Klasse unsichtbar bleiben. Globale Variable sind möglichst zu vermeiden oder müssen unverwechselbar sein.

Die Stabilität beschreibt die Fähigkeit eines Produkts, unerwartete Wirkungen von Software zu vermeiden [30, S. 116]. Um dieses Ziel zu erreichen, wurden Testroutinen für die einzelnen Funktionen der Klassen im Quellcode abgelegt. Diese Routinen stehen auskommentiert am Ende des exekutiven Quellcodes. Im Wartungsfall werden die Testprogramme aktiv geschaltet und können durchlaufen werden. Eine umfangreiche Testung im Wartungsfall ist dadurch gegeben, was neben der Stabilität auch die Testbarkeit erhöht.

Die Stabilität wird erhöht, wenn gleich nach dem Funktionsaufruf die Übergabeparameter auf Vorbedingungen geprüft werden (Assertions).

Python sieht dafür den Befehl „assert“ vor, dem die Bedingung folgt, welche vorzuliegen hat. Wird die Bedingung nicht eingehalten, generiert Python einen AssertionError und der Programmlauf wird abgebrochen [64, SS. 507–508]. Die entwickelte Software arbeitet jedoch mit einem eigenen Error-Handling, das diese Funktionalität inkludiert. Regelverstöße werden in einem Systemprotokoll festgehalten. Darüber hinaus soll eine gut interpretierbare Fehlermeldung mit dem Hinweis, dass an dem Problem gearbeitet wird, den Nutzer zufriedenstellen. In Zukunft wird das Error-Handling noch um eine Mail-Komponente erweitert, damit der Systemverantwortliche in Echtzeit über Probleme informiert wird und nicht erst das Systemprotokoll lesen muss. Werden die Fehler in einer Datenbank geschrieben, sind Statistiken einfach möglich.

Python besitzt die Möglichkeit einer dynamischen Typzuweisung von Variablen. Es kann passieren, dass die Variablen einen unerwarteten Typ aufweisen, weshalb Typprüfungen anzuraten sind. Das leistet die Funktion `type`.

In analoger Weise können auch die Ergebnisse der Methoden und Funktionen überprüft werden, bevor sie an das aufrufende Programm übergeben werden. Man spricht dann von den Nachbedingungen. Die Korrektheit der Methode wird geprüft und ein Fehler nicht verschleppt.

Im Fall von Reparaturarbeiten an Funktionen, die vielfach aufgerufen werden, kommt es vor, dass Programmierer die Funktion unter einem anderen Namen kopieren (z. B. `xxx_2`)

und dort die Änderungen durchführen. Die veränderte Funktion wird nur an der dringend benötigten Stelle aufgerufen. Man versucht so, unerwünschte Seiteneffekte zu vermeiden. Damit verbaut man sich die Möglichkeit, noch unentdeckte Fehler zu beseitigen. Der Programmcode zerflodert und es wird ein Refactoring notwendig.

Schreibt man Computerprogramme, steht man immer wieder vor ähnlichen Fragestellungen, für die im Idealfall bekannte Entwurfsmuster verwendet werden. Es erhöhen sich die Lesbarkeit und die Wartbarkeit des Codes. Darüber hinaus kann davon ausgegangen werden, dass die Entwurfsmuster eine hohe Korrektheit aufweisen. Mithilfe dieser Entwurfsmuster ist es auch möglich, sehr zeitökonomisch zu arbeiten. Um für GI-Projekte des Landes über eine ausreichende Anzahl an Entwurfsmustern zu verfügen, wendete sich das Amt der Stmk LREG an das Universitätszentrum Rottenmann, wo im Rahmen einer wissenschaftlichen Arbeit vorhandene Entwurfsmuster in AML-Quellcodes identifiziert wurden. Diese Entwurfsmuster wurden nach Python unter Verwendung der ESRI-Geoprocessing-Bibliothek portiert [65]. Die Musterlösungen stehen auf einem zentralen Server und allen Entwickler können diese Codeschnipsel verwenden und durch weitere Entwurfsmuster ergänzen. Für gängige Probleme, welche keinen GI-Konnex haben, lassen sich Entwurfsmuster im Internet finden, und die Firma ESRI bietet auf ihrer Homepage<sup>32</sup> ebenfalls Musterlösungen für den GI-Bereich an.

Wartungsfreundliche Software zu entwickeln, stellt einen Mehraufwand dar. Je länger ein Produkt im Einsatz steht, desto wichtiger ist eine wartungsfreundliche Software. Diese Qualitätseigenschaft kommt auch zum Tragen, wenn die Softwarebetreuer wechseln. In der Literatur finden sich viele Arbeiten, die versuchen, den Wartungsaufwand abzuschätzen. Ludewig und Lichter tragen einige dieser Arbeiten zusammen. Sie sprechen, resultierend aus ihrer Literaturrecherche, von Aufwänden zwischen 48 % und 67 % des Gesamtaufwandes [12, S. 569]. Weiter stellen die beiden Autoren fest, dass Anpassungen und Erweiterungen den Löwenanteil der Wartungskosten ausmachen, gefolgt von Korrekturen [12, S. 570].

Wenn diese Zahlen auch mit vielen Unsicherheiten behaftet sind – wie wartungsfreundlich wurde programmiert, wie zukunftsweisend war der erste Programmentwurf – kann davon ausgegangen werden, dass der Wartung eine zentrale Bedeutung zukommt. Wartungsfreundliche Software ist unumgänglich. Es liegt nahe, Entwicklung und Wartung in eine

---

<sup>32</sup> <http://www.esri.com/>

Hand zu legen, da an sonst die Gefahr besteht, dass die Entwicklung den Mehraufwand für einen wartungsfreundlichen Entwurf scheut.

Der Maintainability-Index misst die Wartbarkeit von Software. X. Cullmann und K. Lambertz geben folgende Bewertung ab:

Tabelle 1: Wertung des Maintainability-Index [27]

0 – 64	schlechte Wartbarkeit
65–85	mäßige Wartbarkeit
86–100	gute Wartbarkeit

Man kann seinen Code messen und gegebenenfalls überarbeiten, um eine bessere Wartbarkeit entsprechend diesem Index zu erreichen. Diese Vorgehensweise geht von der Annahme aus, dass der Index auch tatsächlich das misst, was er vorgibt. Man kann darüber natürlich trefflich diskutieren, die sich ableitenden Maßnahmen scheinen jedoch auf jedem Fall sinnvoll. Entsprechend der Formel für den Index, welcher im Kapitel 4.4.4 vorgestellt wurde, kann gute Wartbarkeit durch ausreichende Kommentierung erreicht werden, wobei zumindest ein Drittel des Quellcodes aus Kommentaren bestehen sollte. Kurze Methoden führen zu guter Wartbarkeit. Man sollte versuchen Methoden unter 20–30 Zeilen zu schreiben. Das Halstead-Volumen und damit der Maintainability-Index bleiben gering, wenn einerseits die mittlere Programmlänge kurz gehalten wird und andererseits das Vokabular gering bleibt. Ein beliebter Fehler ist das Einführen von unnötigen Variablen, welche das Lesen eines Programmcodes komplex macht. Gleiche Dinge sollten durchgehend gleich bezeichnet werden. Man sollte nur den notwendigen Sprachumfang verwenden. Es macht keinen Sinn, gewissen Funktionalitäten einmal aus einer Bibliothek zu holen und sich dann einer anderen Konstruktion zu bedienen. Der Sprachumfang sollte nicht zur Gänze ausgereizt werden.

#### 4.5.8. Flexibilität

Unter Flexibilität wird die Anpassungsfähigkeit von Software verstanden, wobei zwischen interner Flexibilität, externer Flexibilität und Runtime-Flexibilität unterschieden wird. Ers-

tere beschreibt die Fähigkeit von Code, durch geringe Anpassungen neue Probleme zu lösen, ohne ein Redesign notwendig zu machen. Die externe Flexibilität gibt an, wie weit Objektklassen und Methoden sich auf viele unterschiedliche Aufgaben anwenden lassen. Runtime-Flexibilität liegt vor, wenn hohe Anpassungsfähigkeit an verschiedene Problemstellungen möglich ist. Die Anpassung erfolgt ohne Manipulation von Code, etwa durch Konfigurationsdateien oder mit Environmentvariablen [66].

Nahezu jede Software von Bedeutung hat die Eigenschaft, dass sie das Verhalten der User ändert. Die Software leistet etwas, was zu Änderungen der Arbeitsabläufe und auch der Computernutzung führt. Diese Eigenschaft von Software wurde am Beispiel von Kurznachrichten von Mobiltelefonen im Kapitel 4.2.3 beschrieben. Daraus ergibt sich, dass Software aus sich selbst heraus veraltet und an neue Gegebenheiten angepasst werden muss. Hohe Flexibilität vermindert die Anpassungsarbeit, da im Allgemeinen nichts grundsätzlich Neues gemacht werden muss, sondern die alten Bausteine verwendet werden können.

Die automatische Abarbeitung von Geodaten hat zur Folge, dass der Planersteller nur sehr kurzzeitig auf die Ergebnisse der Prüfroutine bzw. die automatischen Korrekturen warten muss, was wiederum bewirkt, dass einige Planersteller dazu übergehen, nicht den gesamten Plan abzugeben, sondern jede Ebene nach ihrer Bearbeitung hochladen und auf Korrektheit prüfen. Die abgegebenen Pläne dürfen nicht über das Gemeindegebiet hinausreichen. Ersichtlichmachungen der Flächenwidmungspläne, wie etwa Richtfunkstrecken oder Überflutungsflächen, reichen über die Gemeindegrenze hinweg und werden gerne mit der automatischen Korrektur der Software bereinigt. Der Planer hat den Vorteil, die richtigen Gemeindegrenzen zu verwenden.

Innerhalb der Entwurfsphase hat man den größten Einfluss auf die Flexibilität der Software. Externe Flexibilität wird in ein Projekt getragen, indem die Klassen und Methoden so allgemein gewählt werden, dass sie in anderen Projekten Verwendung finden können. Das gelingt am einfachsten, wenn man sich an gültige Konzepte und Vorstellungen hält. Hat man bei der Analyse der Problemstellung alle verwandten Fragestellungen mitberücksichtigt, können sie auch in der Codierung bearbeitet werden, und die Runtime-Flexibilität steigt. Ein guter Weg die (Runtime-)Flexibilität auf das sinnvolle und benötigte Maß zu heben, besteht darin, alle möglichen Nutzer und ihre verschiedenen Arbeitsweisen zu erfassen. Environmentvariablen bieten sich für die Realisierung an, die in userspezifischen Da-

teilen vorgehalten werden. Für die interne Flexibilität gilt alles, was bereit über die Wartbarkeit von Software geschrieben wurde, da sich die interne Flexibilität kaum bis gar nicht von der Wartbarkeit unterscheidet.

Wesentlich für die Anpassungsfähigkeit ist, dass alle Programmeinheiten möglichst klein sind. Nur kleine Einheiten können entweder exakt oder mit leichten Abänderungen in andere Systeme integriert werden. Kleine Einheiten erhöhen das allgemeine Verständnis der Software und damit die interne Flexibilität und Wartbarkeit.

Möchte man Klassen und Methoden wiederverwenden, ist es notwendig, den Überblick zu bewahren. Hier hat es sich als besonders effizient herausgestellt, alle Klassen und Methoden in ein UML-Diagramm einzutragen. Man hat einen schnellen Überblick über alle vorhandenen Bausteine und erkennt auch Abhängigkeiten.

Die Tools von ESRI, welche als Bibliothek in Python eingebunden werden können, sind Methoden extremer externer Flexibilität. Kaum eine Fragestellung im GI-Bereich lässt sich nicht durch Kombination einiger Tools lösen. Die Nutzung dieser Werkzeuge machte es erst möglich, ein Produkt dieser Größenordnung mit vertretbarem Aufwand zu realisieren. Auch andere Firmen bieten großartige ähnliche Bibliotheken an.

Die entwickelnden Klassen zum Schreiben und Verwalten von Protokolldateien für das Handling von Ausnahmesituationen und das Gestalten, Versenden und Protokollieren von Mails haben das größte Potenzial, in andere Programme integriert zu werden. Um das Potenzial der externen Flexibilität zu nutzen, ist es sinnvoll, die Klassen auf einem zentralen Speicherbereich abzuleben, sodass sie von allen Entwicklern des Landes eingesehen und benutzt werden können. Mit der Sammlung von Entwurfsmustern wurde bereits Ähnliches realisiert. Diese Sammlung arbeitet jedoch mit einsatzfähigen Klassen, die eine gewisse Grundfunktionalität zur Gänze abdecken. Die Entwurfsmuster erfüllen noch keine Aufgabe und stellen kleine Bausteine für eine Programmierung dar.

Ein hohes Augenmerk wurde auf die Runtime-Flexibilität gelegt. Eine beliebige Anzahl von verschiedenen Planarten kann über das System geprüft, verwaltet und korrigiert werden. Der Planaufbau wird durch Tabellen einer relationalen Datenbank beschrieben, und neue Planarten können durch Erweiterung der Tabelleninhalte angelegt werden.

Die Beschreibung neuer Planarten kann durch Wartung der relationalen Ebenen erfolgen. Die Wartungsarbeiten könnten auch im Internet erfolgen, was noch zu programmieren wäre. Python sieht dafür das CGI-Modul vor, mit dem diese Masken zu entwickeln wären. Es könnte dann jedermann eigene Prüfroutinen und Korrekturroutinen entwickeln. Eine strukturierte Verwaltung der Daten ist denkbar. Jeder Internetuser könnte analog zum FWP und OEP eine Datensammlung konfigurieren und diese sukzessive befüllen.

Zusätzlich gibt es Tabellen mit den Regeln zur Prüfung der Daten, wobei die attributive Prüfung hauptsächlich über Muster arbeitet. Es wurde mit der Bibliothek der „Regulären Ausdrücke“ von Python gearbeitet. Ein Beispiel soll Vorteile der Mustererkennung zeigen. Die Planzeichenverordnung sieht für die Festlegung von Gebieten mit Bebauungsplänen vor, dass sie mit dem Buchstaben B, gefolgt von einer fortlaufenden Nummer, kodiert werden (B1, B2 usw.). Es wäre aufwendig, die Prüftabelle mit allen real vorkommenden Werten aufzufüllen. Wesentlich einfacher ist es, festzulegen, dass der Buchstabe B gefolgt von einer Ganzzahl vorzugeben ist, was in der Syntax der regulären Ausdrücke folgendermaßen geschrieben wird: `\AB+d\Z`. Darüber hinaus wird auch die Vorgabe von Listen, aber auch von den Werten selbst unterstützt. Das Arbeiten mit Tabellen hat sich vor allem beim Festlegen der Regeln für die Speicherung als sehr vorteilhaft herausgestellt, da es sehr aufwendig gewesen wäre, diese mehr als 1000 Regeln vorzugeben. Es wurde nur wenige Regeln eingetippt, und mithilfe von Tabellenoperationen in ArcGIS wurden daraus die weiteren notwendigen Records abgeleitet. Eine Beschreibung der Prüfmöglichkeiten und der möglichen Regeln zur Verwaltung der Daten erfolgt im Kapitel 7.4.

Es wurde mit einer Vielzahl an Environmentvariablen gearbeitet, die eine flexible Anpassung an die Situation ermöglichen. Die Software kann zum Testen der Daten oder zum Verwalten bzw. für beide Arbeiten verwendet werden. Das Mailing kann generell ausgeschaltet werden oder für gewisse Gruppen aktiviert werden. Hier soll keine vollständige Liste der Möglichkeiten geboten werden, sondern nur aufgezeigt werden, wie Flexibilität erreicht werden kann.

Namen von Datenbanken, unveränderte Teile von Pfadnamen und Ähnliches werden in aller Regel am Beginn eines Programmes festgelegt, indem die Werte globalen Variablen zugewiesen werden. Im Wartungsfall braucht nur diese Zuweisung bearbeitet werden. Da der Quellcode bearbeitet wird, fällt das schon in den Bereich der internen Flexibilität. Es hat

sich im Rahmen des Projektes gezeigt, dass es praktisch ist, möglichst viel mit Environmentvariablen zu erledigen und diese in Dateien abzulegen. Man kann damit unterschiedlichste Umgebungen schaffen, die immer wieder gebraucht werden. Diese Erkenntnis wurde in einem parallel laufenden kleineren Projekt<sup>33</sup> sofort umgesetzt, wo sämtliche Festlegungen über eine Steuerdatei erfolgen. Jeder User kann sich eine eigene Umgebung schaffen. Wesentlich war, dass in den vorgefertigten Steuerdateien auch Erläuterungstexte zulässig und ausreichend vorhanden waren. Die Nutzer passen ihre Software sehr weitreichend verschiedenen Situationen an.

Zur Nutzung der internen Flexibilität wurden einzelne Stücke des Quellcodes an einer zentralen Stelle abgelegt, auf die alle Entwickler Zugriff haben. Sie können den Quellcode lesen, aber auch verbessern, falls sie Schwächen oder Fehler entdecken. Lesender Zugriff besteht für jeden Landesbediensteten, wobei dieser wahrscheinlich von den wenigsten genutzt wird. Es soll jedoch das Prinzip der größtmöglichen Offenheit gewahrt werden.

Im Laufe des Betriebs hat sich herausgestellt, dass neben der eigentlichen Prüf- und Verwaltungssoftware noch zusätzliche Programme notwendig sind, welche Datenreparaturarbeiten und statistische Analysen erledigen. Insgesamt können 17 Wartungsaufgaben standardisiert abgearbeitet werden. Zum größten Teil wurden diese Programme aus den vorliegenden Klassen und ihren Methoden abgeleitet. Die Programmierung der Wartungssoftware benötigte etwa 16 Stunden, was eine Folge der hohen externen und der internen Flexibilität ist.

Tabelle 7: Wartungsprogramme

Nr.	Erläuterung
1	Objekte für eine Fallart (FWP/OKE ...) aufbauen
2	nicht geprüfte Fälle auflisten
3	Fälle der letzten sieben Tage auflisten
4	nicht verwaltete Fälle auflisten

<sup>33</sup> Eine beliebige Anzahl an vorzugebenden Gemeinden wird in einer vorgegebenen Papierblattgröße geplottet. Das Programm erstellt eine Übersichtskarte einen Blattschnitt für jede Gemeinde und erzeugt die Detailblätter. Das zu verwendende Kartendesign, Karteninhalte sowie die Speicherplätze für die Karten werden über userspezifische Konfigurationsdateien festgelegt.

5	Uploads einer Gemeinde in eine Textdatei überführen
6	Prüf- und Verwaltungskennung rücksetzen
7	für eine Upload_id die Abgabeparameter anzeigen
8	SysArbeit (Workspace) ausleeren – aufräumen
9	Lieferung nicht OK setzen
10	Steuertabellen überprüfen (referenzielle Integrität usw.)
11	Verwaltung durchführen für eine vorzugebende Upload-ID
12	prüfen und verwalten aller ausstehender Fälle (main)
13	löschen einer Abgabe aus dem SDE und rücksetzen des Prüfergebnisses auf LieferungNichtOK (mit Upload-ID)
14	Statusfeld = leer und Feld geprüft = inAbarbeitung (nicht erkannter Absturz der Software)
15	Oracle-Steuertabelle für ALLE Gemeinden in eine Textdatei überführen
16	löschen durch Setzen der Kennung loe_verw
17	verwalten mit mehreren Upload-IDs 12345,17345,17569
18	Auflage zu rechtsgültigem Plan umsetzen

#### 4.5.9. Gebrauchstauglichkeit

Die Norm DIN EN ISO 9241 definiert Gebrauchstauglichkeit als

„Ausmaß, in dem ein Produkt durch bestimmte Benutzer/-innen in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“.

Die Gebrauchstauglichkeit zielt also auf die Mensch-Maschine-Schnittstelle ab. Benutzbarkeit, Benutzerfreundlichkeit, Softwareergonomie und „easy to use“ sind Schlagworte, die in die gleiche Richtung weisen. Die Normenreihe zählt die Qualitätskriterien Aufgabenangemessenheit (a), Selbstbeschreibungsfähigkeit (b), Lernförderlichkeit (c), Steuerbarkeit (d), Erwartungskonformität (e), Individualisierbarkeit (f) und Fehlertoleranz (g) auf (EN ISO 9241-110).

Ad a) Unter der Aufgabenangemessenheit versteht man eine angepasste Funktionalität, welche die benötigte Aufgabe mit einem Minimum an Angaben leistet. Die wenigen einzugebenden Parameter bei der Bedienung der Software lassen das Produkt in dieser Hinsicht ideal erscheinen. Bei der Abgabe des FWP's wird der Anwender gezwungen, die Baulandbilanz anzugeben. Die Baulandbilanz beschreibt die Flächensummen von definierten Widmungskategorien, welche jeweils für bebaute Flächen und unbebaute Flächen zu ermitteln sind. Die Baulandbilanz ist ein wesentlicher Indikator für Flächenwidmungspläne und wird von jedem Raumplaner ermittelt. Natürlich können diese Werte auch aus den abgegebenen Geodaten ermittelt werden, und damit ist das Qualitätskriterium der Aufgabenangemessenheit nicht bestmöglich erfüllt. Die Redundanz ist für Kontrollzwecke eingeführt worden. Die vorgegebene Baulandbilanz muss mit der aus den Geodaten errechneten Baulandbilanz übereinstimmen. Im Planungsprozess entstehen viele Varianten des FWP's, die sich durch unterschiedliche Baulandflächen unterscheiden. Dabei kann es sehr leicht passieren, dass eine falsche Variante übermittelt wird, was mithilfe des Abgleichs der Baulandbilanz erkannt wird.

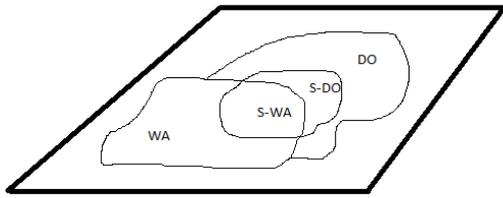
In aller Regel meint man, dass die Aufgabenangepasstheit erfüllt ist, wenn keine Mehrfachangaben notwendig sind. In diesem Produkt wird jedoch wesentlich weiter gegangen. Die abgegebenen Geodaten können ja auch als eine Art Eingabe verstanden werden. Die Software erkennt nicht nur die fehlerhaften Daten, sondern führt auch viele lästige Reparaturen selbstständig durch, wie das Abschneiden von Überständen über die Gemeindegrenze, das Löschen überflüssiger Spalten oder die Korrekturen von fehlerhaften Beschreibungen. Bei persönlichen Gesprächen mit Nutzern des Systems wurde die Sinnhaftigkeit dieser Korrekturen hinterfragt, wobei das Abschneiden der Überstände als die erfreulichste automatische Korrektur eingestuft wurde.

Ad b) Die Selbstbeschreibungsfähigkeit ist gegeben, wenn das System genutzt werden kann, ohne dass der Anwender Handbücher studieren muss. Für das entwickelte Produkt sind keine Einschulungen notwendig. Die Eingabemaske, welche von der allgemeinen EDV programmiert wurde, erfüllt diese Kriterien. Es existiert für Anwender keine Beschreibung der Oberfläche und es kam in dem nun schon mehrjährigen Betrieb der Oberfläche zu keiner einzigen Rückfrage im Zusammenhang mit der Oberfläche.

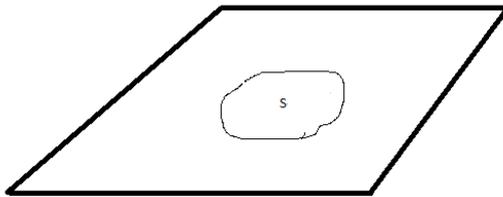
Wesentlich komplexer als die Bedienung der Eingabemaske ist es, die Schnittstelle der Geodaten zu verstehen. Im weiteren Sinn kann auch der Entwurf der Schnittstelle als Usability-Kriterium verstanden werden.

Die Verordnung der Schnittstelle 2007 hat in vielen Bereichen Klarheit geschaffen. Wurden in der Schnittstelle 2003 nur der Aufbau der Geodaten und die zu verwendenden Planzeichen beschrieben, kamen mit der Schnittstelle 2007 auch Verweise auf die Gesetzesstellen. Damit ist klar, welche Rechtsmaterie wo abgelegt wird. Bei der Schnittstelle 2003 war diese Zuordnung nicht schriftlich niedergelegt, was in wenigen Einzelfällen zu Interpretationsproblemen führte. Guidelines wurden eingeführt, was zu einem einfacheren Verstehen der Schnittstelle führt. Hier einige Beispiele:

- Alle Planobjekte, welche nicht vom Gemeinderat beschlossen werden und verpflichtend im FWP ersichtlich zu machen sind, werden in Ebenen abgelegt, welche mit der Bezeichnung `ersl_...` beginnen. Das sind die sogenannten Ersichtlichmachungen.
- Der Geometriotyp ist am Ende des Namens der Ebene codiert. (...`_f` für flächenhafte Objekte, ...`_l` für linienhafte Objekte, ...`_p` für punktförmige Objekte).
- In allen Ebenen des FWPs kommt die Spalte WIDMUNG vor, welche das Objekt klassifiziert, und ein Eintrag ZSW (Zusatzwidmung) beinhaltet quantitative Ausprägungen bzw. eine weitere Untergliederung. Lediglich eine Ebene benötigt zwei Zusatzwidmungen und die Hauptebene der Nutzungen enthält weitere Spezifikationen (maximale und minimale Bebauungsdichte, maximale Bauhöhe).
- Die Hauptkategorien (WA...Wohnen allgemein, WR...Wohnen rein, LF...Land und Forstwirtschaft usw.) werden in Großbuchstaben geschrieben. Sie beschreiben die Gemeinde vollständig und eindeutig. Alle Beschränkungen der Hauptkategorien sind in einer eigenen Ebene abgelegt, was eine wesentliche Änderung gegenüber der Schnittstelle 2003 darstellt, wo Beschränkungen durch zusätzliche Widmungskategorien beschrieben wurden.



Schnittstelle 2003:  
Sanierungsgebiete werden speziell kategorisiert.



Schnittstelle 2007: Für die Einschränkung Sanierungsgebiet ist eine eigene Ebene vorgesehen.

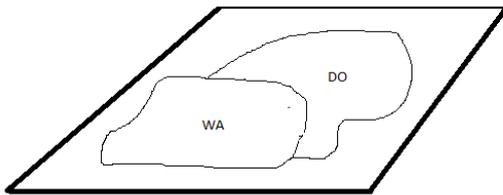


Abbildung 14: Schnittstelle 2003–2007

Neben diesen Verbesserungen wurden viele weitere Vereinheitlichungen eingeführt, welche nicht weiter ausgeführt werden sollen, da die Entwicklung der Schnittstelle federführend durch M. Weissenbeck erfolgte und ich lediglich beratend für GI-Fragen zur Seite stand.

Ad c) Die Maske des Uploads besteht aus wenigen Eingabefeldern und ihre Bedienung ist daher trivial. Die Felder werden von jedem Fachmann ohne Probleme verstanden. Wesentlich komplexer ist die Schnittstelle, in der die Daten zu liefern sind. Alle Vereinheitlichungen welche im Block b Selbstbeschreibungsfähigkeit aufgelistet sind, erhöhen die Lernförderlichkeit wesentlich.

Ad d) Die Eingabemaske wurde entsprechend des Landesstandards entwickelt. Die Reihenfolge der Eingabe ist beliebig, Felder können bis zum Abschließen immer wieder geändert werden. Da nur eine einzige Maske existiert, sind die Aufwendungen zur Erfüllung dieses Qualitätskriteriums nicht allzu hoch.

Die Prüfung gegenüber diesem Qualitätskriterium hat ergeben, dass der Bedarf besteht, abgegebene Pläne durch eine neuerliche Übermittlung zu überschreiben. Es erscheint sinnvoll, hier in einem weiteren Release die Software zu erweitern. Der abgegebene Plan wird im Intranet visualisiert, wobei lediglich der Planer seine Daten ansehen kann. Er hat die Möglichkeit einer optischen Überprüfung, bevor die breite Öffentlichkeit die Daten sieht, und kann den Plan freigeben oder verwerfen.

Ad e) Der Plan wird hochgeladen und dieser Vorgang ist mit einem Minimum an Metadaten zu ergänzen (Art des Planes, Gemeindenummer, Verfahrensfall, Auflageplan oder rechtsgültiger Plan, Test oder Übergabe, Baulandbilanz). Der Planer verfügt über diese Basisinformationen und es ist ihm bewusst, dass diese Eckpfeiler für die Verwaltung von vielen Plänen unumgänglich sind. Seit dem Betrieb des Portals wurden diese Vorgaben noch von keinem Planer infrage gestellt. Mit dem Abschicken des Planes wird ein Popup geöffnet, welches den Planer über eine E-Mail informiert. Er erhält eine Bestätigung der Abgabe, in welcher auf die nun folgende Prüfung hingewiesen wird. Ein weiteres E-Mail beinhaltet das Prüfprotokoll. Liegen keine Fehler vor, wird der Planer aufgeklärt, dass die Daten am darauffolgenden Tag im Internet veröffentlicht werden. Andernfalls erhält er die vom Reparaturprogramm ausgebesserten Geodaten zur Überprüfung und Neueinreichung.

Ad f) Individualisierbarkeit meint, dass ein System an die Bedürfnisse und das Wissen des Nutzers angepasst wird. Da die Maske für ausgesprochene Spezialisten der Raumplanung entwickelt wurde, sind inhaltliche Anpassungen nicht notwendig. Erleichterte Bedienungen durch Änderungen der Schriftgröße können über den Internetbrowser, in dem die Maske läuft, erfolgen.

In den vorherigen Menüpunkten – insbesondere der Selbstbeschreibungsfähigkeit – wurde der Plan selbst auch als eine Art Eingabe aufgefasst. Da der Sinn des Projektes darin besteht, Vorgaben in Plandaten zu prüfen und Regelverstöße zu korrigieren, versucht dieses Programm, die Individualisierbarkeit der Plandaten möglichst zu unterbinden. Bis auf den planungsautonomen Teil besteht keine Individualisierbarkeit der Geodaten. Geringfügige Ausnahmen werden im Kapitel 7.4.2.4 angeführt.

Ad g) Ein System ist fehlertolerant, wenn es Irrtümer in der Eingabe erkennt und korrigiert. Das ist im hohen Maß für die abgegebenen Geodaten der Fall, wo Verstöße gegen die Norm repariert werden. Die einzelnen Maßnahmen werden im Kapitel 7.4.3 beschrieben.

Werden Fehler in den abgegebenen Metadaten diagnostiziert – fehlerhafte Planart, Gemeinenummer, Verfahrensfallnummer –, erfolgt derzeit eine entsprechende Mitteilung per E-Mail. Die Vorgabe wird als falsch eingestuft und die Geodaten werden nicht übernommen. Die Kontrollen werden soweit wie möglich durchgeführt, damit ein möglichst vollständiger Fehlerbericht entsteht.

#### 4.5.10. Softwaretest

##### 4.5.10.1. Allgemeines

Softwaretests werden nicht nur zur Überprüfung der Korrektheit herangezogen, sondern können auch die Qualitätskriterien Robustheit, Performance, Verlässlichkeit, Sicherheit und Benutzerfreundlichkeit und Gebrauchstauglichkeit prüfen. Es handelt sich also um eine universelle Methode für die Erstellung brauchbarer Software.

Alle Softwaretests sollten – zumindest in einer idealen Welt – abgeschottet erfolgen, dass ein Versagen zu keinen negativen Konsequenzen führt. Diese trivial anmutende Forderung wird leider nicht immer erfüllt. Meist sind es ökonomische Gründe, Faulheit oder Selbstüberschätzung, welche zu (kleinen) Änderungen an Produktionssystemen führen. Es gibt aber auch Konstellationen, wo keine andere Möglichkeit besteht, als in der Produktionsumgebung zu testen. Auf jeden Fall sollte immer eine Risikoabschätzung erfolgen. Dabei sollte Murphys Gesetz im Hinterkopf mitschwingen, das lautet: „Whatever can go wrong will go wrong.“

Traurige Berühmtheit erlangte die Simulation eines totalen Stromausfalls im Atomkraftwerk von Tschernobyl, wobei geprüft werden sollte, ob der Reaktor sich dann noch selbst kühlen kann. Durch die Verstrickung mehrerer unglücklicher Zufälle geriet der Reaktor außer Kontrolle und führte zu einer verheerenden Atomkatastrophe [67].

Das Testen der Software ist für die Korrektheit von wesentlicher Bedeutung. Umfangreiche Tests garantieren hohe Qualität. Ein vollständiger Test müsste alle möglichen Eingaben mit allen Möglichkeiten der Software kombinieren, was nur bei sehr kleinen Programmen mit beschränkten Eingabemöglichkeiten erfüllbar ist. Es wird deshalb nach Möglichkeiten gesucht, mit möglichst wenigen Tests möglichst viel Fehlverhalten aufzudecken. Es gilt der

von Edsger W. Dijkstra geprägte Satz „Program testing can be used to show the presence of bugs, but never show their absence!“ [68].

#### 4.5.10.2. Äquivalenzklassen von Tests

Beim Test von Software versucht man mit möglichst wenigen Tests, möglichst viele Fehler aufzudecken, um den Aufwand zu minimieren. Jeder Test besteht aus einer Eingangsinformation, welche eine bekannte Ausgangsinformation liefern soll. Es gilt nun, die Eingangsinformationen so zu variieren, dass alle Fehler aufgezeigt werden, jedoch keine unnötigen Tests erfolgen, die den gleichen Fehler reproduzieren.

„Zwei Testfälle  $f_1$  und  $f_2$  sind im Hinblick auf den Erfolg des Tests stark äquivalent, wenn sie austauschbar sind, weil beide geeignet oder beide ungeeignet sind, einen bestimmten Fehler aufzuzeigen.“ [12, S. 501]

Die Analyse der Eingabebereiche ermöglicht es, Wertebereiche festzulegen. Am Beispiel der Verfahrensfallnummer soll das gezeigt werden. Die Verfahrensfallnummer besteht aus der Stammnummer und der Änderungsnummer, wobei die Stammnummer Werte von 1 bis 9 und die Änderungsnummer Werte von 00 bis 99 annehmen kann. Wir bilden für den Test die Äquivalenzklassen:

$[-\infty,0]$  ...negative Äquivalenzklasse der Stammnummer       $[-\infty,0]$  ...negative Äquivalenzklasse der Änderungsnummer

$[1,9]$  ....positive Äquivalenzklasse der Stammnummer       $[1,99]$  ...positive Äquivalenzklasse der Änderungsnummer

Negative Äquivalenzklassen beschreiben unzulässige Eingabewerte, die vom Programm abzufangen sind.

Ein Standardfall der Äquivalenzklasse sollte getestet werden. Auch die Grenzwerte sind zu testen, da hier häufig Fehler auftreten. Wird ein Stapel verarbeitet, ist zu kontrollieren, ob das für ein beliebiges Element, für das erste Element und für das letzte Element funktioniert. Die Äquivalenzklassenmethode wurde zur Grenzwertanalyse erweitert. Im Netz finden sich

hervorragende Unterlagen, welche die Äquivalenzklassenmethode und die Grenzwertanalyse erläutern [69].

Im Fall von Geoanalysen kann die Bildung der Äquivalenzklassen schwieriger sein. Betrachten wir die Programmunit, welche prüft, ob für jede (Siedlungs-)Fläche auch ein (Siedlungsschwer-)Punkt vorhanden ist. Wir können beispielsweise eine Prüfung mit drei Polygonen und drei Siedlungsschwerpunkten durchführen, wobei alle Punkte innerhalb der Fläche liegen. Führen wir einen weiteren Test durch mit vier Polygonen und vier Siedlungsschwerpunkten, werden wir keine zusätzlichen Fehler aufdecken. Die Tests sind äquivalent. Wesentlich sinnvoller wäre eine Prüfung:



Eine Fläche und kein Punkt liegen vor.



Ein Punkt aber keine Fläche ist vorhanden.



Ein Punkt und eine Fläche existieren, wobei der Punkt in der Fläche liegt.



Ein Punkt und eine Fläche sind vorhanden, wobei der Punkt außerhalb Fläche liegt.

Abbildung 15: Räumliche Äquivalenzklasse mit keinem oder einem Objekt

Damit werden der Grenzwert des Fehlens eines Objektes und die gegenseitige Lage von Punkt und Fläche beim Vorhandensein jeweils einer Ausprägung getestet. Es fehlen noch die Tests von mehreren – zumindest zwei – Objekten jedes Typs. Die Anzahl der Prüffälle steigt.

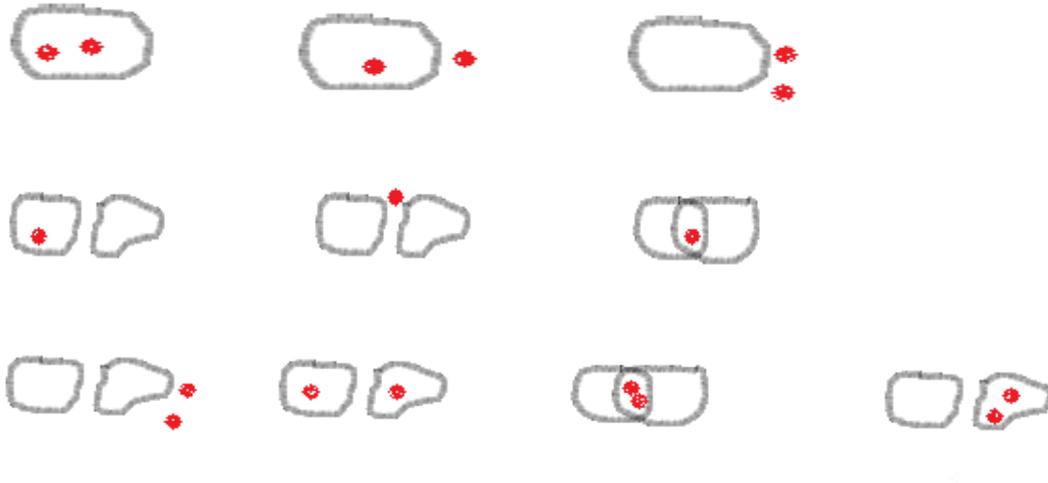


Abbildung 16: Räumliche Äquivalenzklasse mit einem oder mehreren Objekten

Es sind die möglichen Lagen zueinander für ein und kein Objekt sowie für ein und zumindest zwei Objekte zu testen. Testet die Prüfung lediglich ein Geoverarbeitungswerkzeug, welches im gegenständlichen Fall von der Firma ESRI stammt und weltweit im Einsatz ist, wird man kaum Fehler aufdecken können. Diese Zuverlässigkeit ist natürlich auch bei vielen anderen professionellen Softwarekomponenten gegeben, wie etwa die Oracle- Datenbank und vieles mehr. Ihre Verarbeitung als Ganzes hat jedoch ebenfalls entsprechend überprüft zu werden.

Liest man Dateien ein oder bearbeitet Schleifen, treten Probleme auf bei fehlendem Inhalt, wenn lediglich ein Wert vorliegt sowie am Schleifenende. Es macht also Sinn, auf Null, Eins und Mehrere zu testen. Hat man einen String als Eingabemöglichkeit, sollte man außer mit der Länge Null und der maximal zulässigen Zeichenlänge Tests mit Sonderzeichen, mit Leerzeichen mitten im String sowie mit String bestehend aus einem Leerzeichen durchführen.

In Python muss der Datentyp nicht explizit definiert werden und logische Abfragen zwischen Variablen von verschiedenen Typen sind möglich. Diese große Freiheit ist mit schwer zu findenden Fehlern verbunden. Entsprechende Tests sind anzuraten.

Werden Methoden von Objekten getestet, ist es sinnvoll, auch entsprechende Tests mit Childs dieser Objekte durchzuführen.

Prüft man eine Softwarekomponente lediglich gegenüber der Spezifikation, greift man zu kurz, da Software auch sinnvoll bei unzulässigen Eingaben reagieren sollte. Die Robustheit zu testen, ist umso wichtiger, je größer das Projekt ist, da bei der Wiederverwendung die Eingabebedingungen dem Entwickler möglicherweise nicht zur Gänze bewusst sind. Lässt man die Software kurz vor Beendigung abstürzen – wir simulieren einen Stromausfall – muss ein zweiter Durchlauf möglich sein. Viele Programmunits legen Hilfsdateien, Zwischengeodatensätze und Ähnliches an. Ein Absturz kann zur Folge haben, dass die Aufräumarbeiten noch nicht beendet sind. Um die Unit sicher zu bewältigen, ist am Beginn zu kontrollieren, ob Reste alter Exekutionen existieren, welche zu löschen sind. Die Environmentvariable `overwriteoutput` setzt den Geoprocessor von ESRI in die Lage, alte Ergebnisse zu überschreiben. Fehlender Speicherplatz, der Ausfall von Netzlaufwerken, Verbrauch aller Lizenzen des Lizenzservers sind weitere zu testende Problemfelder.

#### 4.5.10.3. Methodik

Erfolgt die Festlegung der Testfälle in Kenntnis des Programmcodes, spricht man von Glassbox-Tests oder auch Whitebox-Tests. Man kann mit Überdeckungsmaßen arbeiten. Ludewig und Lichter unterscheiden zwischen der Anweisungsüberdeckung, der Zweigüberdeckung und der Termüberdeckung. Ersteres ist gegeben, wenn jede Anweisung zumindest einmal ausgeführt wurde. Die Zweigüberdeckung ist ein strengeres Maß und setzt voraus, dass jeder Zweig durchlaufen wurde. Bei jeder Verzweigung muss der True- und False-Weg gegangen werden. Die Anzahl der Wege entspricht der zyklomatischen Komplexität (siehe Kapitel 4.4.2) Hat man einen Zweig ohne Code,<sup>34</sup> muss auch dieser getestet werden. Die Termüberdeckung ist gegeben, wenn bei komplexen Bedingungen (Term1 and Term2 or Term3) jeder Term aus seine Wirkung getestet wurde [12, SS. 514–532].

Es existieren Programme, welche den Grad der Überdeckung anzeigen (pycharm).

Von Blackbox-Tests spricht man dann, wenn der Tester den Code nicht kennt.

„Dies ist die wichtigste Form des Tests. Alle anderen Tests ergänzen den Black – Box Test, sie können ihn nicht ersetzen.“ [12, S. 504]

---

<sup>34</sup> Das ist der Befehl `pass` in python

Dabei sollten die Funktionsüberdeckung, die Eingabeüberdeckung und die Ausgabeüberdeckung geprüft werden und Leistungsgrenzen und Mengengrenzen ausgelotet sowie Fehlersituationen hergeführt werden [12, SS. 504–513].

Neben den klassischen Verfahren der Whitebox- und Blackbox-Tests etablierte sich mit den objektorientierten programmieren das Test-Driven-Development (TDD). Robert C. Martin schildert eindrucksvoll, wie er dieses Verfahren von Kent lernte [21, SS. 115–116].

1. Zuerst wird ein Test programmiert. Der eigentliche Code für diesen Test ist noch nicht vorhanden.
2. Der Programmcode wird so weit entwickelt, dass er den Test und alle alten Tests besteht.
3. Es wird ein Refactoring durchgeführt.

Ein Durchlauf dieser drei Schritte sollte nur wenige Minuten dauern und solange wiederholt werden, bis der Produktivcode fertig ist. An einem Beispiel soll das erklärt werden. Ein Objekt soll die Gemeindenummer und die Verfahrensfallnummer speichern. Es muss sichergestellt werden, dass die Attribute Textfelder speichern.

Tabelle 8: Beispiel TDD

<pre># Test Instanz anlegen gemnr1,vfnr1 = '60307','0600' mInfo = Meta(gemnr1,vfnr1)</pre>	<p>Man beginnt mit einem Test, der die Instanz anlegt. Starten man das Skript, wird eine Fehlermitteilung das Fehlen der Klasse melden</p>
<pre>class Meta(object): def __init__(self,gemnr,vfnr): self.Gemnr, self.Vfnr = gemnr, vfnr # Test 1 Instanz anlegen gemnr1,vfnr1 = '60307','0600' mInfo = Meta(gemnr1,vfnr1)</pre>	<p>Wir definieren die Klasse und starten den Test, der nun bestanden wird.</p>
	<p>Refactoring ist derzeit nicht notwendig.</p>

<pre>class Meta(object):     def __init__(self,gemnr,vfnr):         self.Gemnr, self.Vfnr = gemnr, vfnr     # Test 1 Instanz anlegen     gemnr1,vfnr1 = '60307','0600'     mInfo = Meta(gemnr1,vfnr1)     # Test 2 Zahlenwerte zu Text     gemnr2,vfnr2 = 60308,600     mInfo2 = Meta(gemnr2,vfnr2)     if mInfo2.Gemnr == '60308': print "OK"     else: print "Fehler Test 2"</pre>	<p>Man schreibt nun einen weiteren Test, der Zahlenwerte für die Objektattribute übergibt. Sie müssen bei der Initialisierung zu Textattributen umgewandelt werden, um den Test zu bestehen. Startet man die Tests, tritt der Fehler auf.</p>
<pre>import string class Meta(object):     def __init__(self,gemnr,vfnr):         self.Gemnr, self.Vfnr = str(gemnr),         str(vfnr)         .....         .....         usw.</pre>	<p>Man erweitert den Produktivcode.</p>

TDD führt alle Tests immer wieder durch. Bei der Entwicklung der Testcodes werden Fehler im Produktivcode entdeckt und umgekehrt. Fehler werden frühzeitig erkannt, und da immer alle Tests durchlaufen werden, werden durch neuen Code keine Fehler in alte Code-teile hineingetragen. Die Tests selbst dokumentieren die Funktionalität der Software.

M. Martin recherchiert die Defektrate bei Anwendung von TDD:

„Das ist kein isolierter Effekt. Es gab mehrere Berichte und Studien, die eine signifikante Defektreduzierung beschreiben. Ein Unternehmen nach dem anderen – von IBM über Microsoft bis zu Sabre und Symantec – und ein Team nach dem anderen hat Defektreduktionen um den Faktor 2, 5 und sogar 10 berichtet.“ [21, S. 119]

Es existieren Oberflächen, welche die Verwaltung der Tests übernehmen und den Grad der Überdeckung anzeigen können (z. B. PyCharm). Noch nicht entwickelte Komponenten werden durch Platzhalter ersetzt (stub) und Testtreiber (test driver) verwalten die Eingabe (siehe auch IEEE Std 610.12(1990)).

Die Integration von Software in eine Architektur ist damit nicht getestet. Auch die Usability ist nicht überprüft. TDD fordert vom Entwickler viel Disziplin und TDD alleine garantiert lediglich, dass Tests und Programm zusammenpassen.

Testphasen werden gerne gekürzt und häufig als Zeitpuffer zwischen dem Fertigstellungstermin und dem Auslieferungstermin angesehen. TDD testet direkt während der Entwicklung, und diese Zeit kann deshalb schwer gekürzt werden.

#### 4.5.10.4. Vorgehensweise

ArcGIS-Desktop besitzt mit dem Modelbuilder die Möglichkeit, Geoverarbeitungsschritte grafisch aneinanderzufügen. Das geht mittels Drag-and-drop sehr rasch und man braucht sich nicht um die Syntax zu kümmern. Die grafischen Zusammenstellungen der Modellketten lassen sich in Python-Skripte exportieren, welche nachbehandelt und dann in den Code eingefügt wurden.<sup>35</sup> Die Entwicklung ging also in diesen Fällen nicht atomar vor, indem jeder einzelne Befehl getestet wurde, wie es im TDD üblich ist. Die Tests wurden auf ganze Blöcke beschränkt. Die Korrektheit dieser Blöcke wurde durch Tests mit Äquivalenzklassen und Grenzwerten geprüft (leerer Geodatensatz, Geodatensatz mit einem Element, Geodatensatz mit mehreren Elementen). Dabei ging es nicht so sehr um den Test der ESRI-spezifischen Komponenten, sondern um die eigenen Codeteile, welche um die arcpy-Methoden geschrieben wurden. TDD kam immer dann zu Einsatz, wenn keine geografischen Modelle entwickelt werden, wie etwa das Mailing-System und das Protokollsystem. Hier hat sich der Einsatz von TDD teilweise auf kritische Funktionen beschränkt. Je kleiner die Einheit war, desto einfacher war der Einsatz von TDD zu bewerkstelligen. Auch wurden Erweiterungen und Änderungen separat in Form von TDD entwickelt und dann in den Monolithen eingefügt. Mit dieser Vorgehensweise konnten besonders komplexe Teile schrittweise gelöst werden. Um die Termüberdeckung bei kritischen Stellen zu garantieren, sind Entscheidungsmatrizen eine sinnvolle Hilfestellung.

Im Anhang – Fehlerbericht 4.2 Fehler der Einzelpläne wird darauf hingewiesen, dass bereits 331 Planabgaben von FWPs existieren, auf der die Analyse der Fehler fußt. Diese Echtdateien, welche die Gesamtheit der Abgaben von FWPs eines Zeitraums von 2,6 Jahren darstellen, können zu Testzwecken herangezogen werden. Die Gebrauchstauglichkeit wird damit getestet, was wesentlich wichtiger ist als das Auffinden von Fehlern in Codeteilen, welche in der Realität nicht erreicht werden. Dabei handelt es sich um klassische Blackbox-

---

<sup>35</sup> Iteratoren und Bedingungen, welche im Modelbuilder auch realisierbar sind, können jedoch nicht nach Python exportiert werden und sind von Hand zu kodieren. Auch ist der Code in Bezug auf Variablen nachzubehandeln.

Tests oder auch Akzeptanztests, denn sie zeigen, ob das System brauchbar ist. Um das Verhalten unter Last zu testen, wurden bis zu zehn Pläne gleichzeitig zur Bearbeitung abgeschickt.

## 5. Qualität der abzugebenden Geodaten

Ein fixer Bestandteil in der Arbeit des GI-Ingenieurs ist die Überprüfung der Geodaten auf die Tauglichkeit für die gestellte Aufgabe. Andernfalls setzt man sich der Gefahr aus, trotz hervorragender Arbeit schlechte bzw. falsche Ergebnisse zu liefern. In der Informatik wird auch vom GIGO-Prinzip gesprochen, es meint „garbage in garbage out“. Diese Aufgabe muss durch die in der Arbeit präsentierte automatische Vorgehensweise abgedeckt sein. Dazu ist es notwendig, die einzelnen Einflussfaktoren zu betrachten und ihre Wirkung zu beurteilen.

Das entspricht der Prüfung gegenüber dem Qualitätsmodell nach ISO 19113, wo folgende Qualitätskriterien aufgelistet werden:

- Vollständigkeit,
- logische Konsistenz,
- Positionsgenauigkeit,
- Zeitgenauigkeit,
- thematische Genauigkeit,
- Zweck,
- Verwendung,
- Herkunft.

N. Bartelme erklärt diese Begriffe im Detail [23, SS. 244–247].

### 5.1. Vollständigkeit

Die Vollständigkeit legt fest, inwieweit ein Geodatensatz eine Ausprägung der realen Welt oder auch von Rechtssituationen widerspiegelt. Objekte können fehlen oder es sind zu viele Objekte vorhanden (siehe N. Bartelme [23, S. 245]).

Die Prüfung auf fehlende Features erfolgt durch einen Abgleich mit anderen Geodatenbanken. Dabei werden die Datenbestände des Landes – das sind vor allem die Datensätze des

Projektes ROKAT<sup>36</sup> – mit den FWP verglichen. Die im Land vorliegenden Daten sind jedoch häufig von einer geringeren Dimension. Für viele Themen liegen im Projekt ROKAT Punkte anstatt Flächen vor. Diese Landesdaten haben eine ausreichende Genauigkeit für den Zielmaßstab 1:25.000 und können damit nicht direkt für Vergleiche mit Elementen des FWPs herangezogen werden, der im Maßstab 1:5000 vorliegt. Eine ungefähre Überprüfung ist trotzdem möglich.

Das Objekt mit der höheren Dimension wird um einen Betrag gebuffert, welcher sich aus den unterschiedlichen Genauigkeiten ergibt. Liegt das Element mit der geringeren Dimension nun innerhalb des Buffers, werden Lagerichtigkeit und Vollständigkeit angenommen [70].

Fehlende Objekte werden gut erkannt. Das Verfahren zeigt grobe Positionsfehler an. Die Landesdaten werden damit bestmöglich für Kontrollzwecke ausgewertet. Trotzdem ist die entsprechende Softwareunit derzeit nicht im Einsatz, da sie im Verfahrensablauf kontraproduktiv wirkt. Es werden Fehler in verordneten Plänen aufgezeigt.

Die oben beschriebene Aufgabe würde sich auch mit einem Fuzzy-Ansatz lösen lassen. Dafür werden für das gelieferte Objekt A und für das Vergleichsobjekt B Fuzzy-Sets  $F_A$  und  $F_B$  gerechnet. Das erfolgt, indem zuerst ein feinmaschiges Raster erstellt wird, welches für jede Rasterzelle die Entfernung zum Objekt speichert. Auf jede Zelle dieser beiden Rasterdatensätze wird eine FuzzyMembership-Funktion angewandt, die der Zelle in Abhängigkeit von der Entfernung zum Objekt Werte zuordnet. Zellen innerhalb des Objekts wird der Wert 1 zugeordnet und weit entfernten Objekten 0. Im Übergangsbereich werden in Abhängigkeit von der Distanz Werte zwischen 0 und 1 errechnet, wobei häufig ein linearer Ansatz gewählt wird. Der Wert 1 steht also für „innerhalb des Objektes“ und der Wert 0 für „außerhalb des Objektes“. Wir können 1 als „ja“ und 0 als „nein“ auffassen. Dazwischen gibt es alle Schattierungen von wahrscheinlich ja (0,99) bis sehr unwahrscheinlich (0,01). Man spricht auch von unscharfen Mengen. Im Fall des Vergleiches zweier Objekte gleicher Dimension, können wir die unscharfen Bereiche als Lageunsicherheit interpretieren. Sinnvollerweise wird man dann als Fuzzy-Membership-Funktion die Glockenkurve nach Gauß verwenden. Die beiden Fuzzy Sets  $F_A$  und  $F_B$  werden miteinander in Beziehung gesetzt und es werden so Widersprüche aufgezeigt. Weist eine Zelle einen sehr hohen Wert im Raster  $F_A$

---

<sup>36</sup> ROKAT Raumordnungskataster

auf und die lagegleiche Zelle im Raster  $F_B$  einen sehr niedrigen Wert oder umgekehrt, kann von einem Lagefehler der Objekte zueinander ausgegangen werden. Bei großen Objekten kann sogar gesagt werden, in welchem Bereich des Objektes der Fehler vorliegt.<sup>37</sup>

Liegen Objekte unterschiedlicher Dimension vor, ist die Situation wesentlich schwieriger. An einem fiktiven Beispiel soll der Fall abgehandelt werden. Im FWP sind Fußballplätze als Rechtecke zu liefern und im Land gibt es einen Punktdatensatz, der die Lage der Plätze zeigt. Wir erstellen mit einer ausgewählten Fuzzy-Membership-Funktion zwei Fuzzy-Sets  $F_A$  und  $F_B$  und färben diese nach gleichartigen Fuzzy-Werten ein. Da die Fuzzy-Membership-Funktionen auf der Entfernung zum Objekt beruhen, zeigt der gelieferte Geodaten-satz ein Rasterbild  $F_A$  konzentrischer Rechtecke, die mit größerer Entfernung in Kreise übergehen. Das Landesraster  $F_B$  besteht sofort aus konzentrischen Kreisen. Ein Vergleich der Raster ist nun schwieriger, da die unterschiedlichen Geometrien andere Bilder bedingen, auch dann, wenn die gelieferten Daten und die Vergleichsdatensätze korrekt sind. Führt man die Rechtecke jedoch in ihre Zentralpunkte über, kann wie im vorherigen Fall vorgegangen werden, da nun wieder Objekte gleicher Dimension vorliegen.

Der reine Vektoransatz ist vom Verständnis einfacher als der Rasteransatz, was Vorteile bei der Begründung bringt. Muss man Rechenschaft darüber ablegen, warum ein Objekt als falsch eingestuft wurde, lässt sich der Umstand mit dem Satz „Das gelieferte Objekt ist außerhalb der Toleranz zum Kontrollobjekt im Amt der Steiermärkischen Landesregierung“ erklären. Der Fuzzy-Ansatz benötigt mehr Erläuterungen.

Die Vektormethode benötigt deutlich weniger Speicherplatz als die pixelbasierte Vorgehensweise. Beide Kontrollmethoden sind schnell, insbesondere dann, wenn die Zwischenergebnisse im Hauptspeicher gehalten werden (in\_memory-Workspace).

Es kann bei beiden Methoden nicht gesagt werden, ob der Datensatz des Planers fehlerbehaftet ist oder der Layer im Land eine Unkorrektheit aufweist. Genauer weiß man nur, wenn garantiert werden kann, dass die Geodaten des Landes den Rechtsstand widerspiegeln.

---

<sup>37</sup>  $(F_A \wedge \text{not } F_B) \vee (F_B \wedge \text{not } F_A)$

Neben dem Vergleich von Objekten unterschiedlicher Datenbestände gibt es auch gegenseitige Lagebedingungen innerhalb eines Plans. Z. B. muss der sogenannte Siedlungsschwerpunkt in einem Siedlungsgebiet liegen. Entsprechende Überprüfungen wurden koordiniert. Die Abgrenzung zu anderen Qualitätskriterien ist nicht immer eindeutig. Im nachstehenden Kapitel 5.2 wird weiter darauf eingegangen.

## 5.2. Logische Konsistenz

Die logische Konsistenz beschreibt, inwieweit eine vorgegebene Datenstruktur eingehalten wird. Dabei geht es um Domains von Attributen, Formatregeln, topologische Bedingungen und hierarchische Beziehungen [23, S. 245–246].

Die Abgrenzung der Qualitätskriterien zueinander ist nicht immer eindeutig festlegbar. Im OEP werden Siedlungsgebiete, die als Flächen festgelegt sind, mit einem Siedlungsschwerpunkt versehen. Innerhalb eines jeden Polygons hat also ein Punkt zu liegen. Das Fehlen des Punktes ist einerseits eine Verletzung der Vollständigkeit und andererseits ein Verstoß gegen die topologische Regel, dass ein Punkt innerhalb der Fläche liegen muss. Im FWP muss die Nutzungsebene das gesamte Gemeindegebiet ohne Lücken und Überlagerungen beschreiben. Wurde ein Grundstück nicht gewidmet, liegt eine Lücke vor und wir haben einen Verstoß gegen die logische Konsistenz und die Vollständigkeit vorliegen.

Lücken entstehen in der Regel durch ungenaue Datenerfassung. Ganze Grundstücke werden selten vergessen. Die topologische Regel, als Teil der logischen Konsistenz, hat auch Überschneidungen zum Qualitätskriterium der Positionsgenauigkeit. Wir können die Überlagerungen und Lücken der Nutzungsebene auch als fehlerhafte innere Genauigkeit der Geometrie ansehen.

Für die automatische Abarbeitung von Geodatenätzen ist die logische Konsistenz unabdingbar. Namen von Ebenen und Spalten sowie Attributdomains müssen eingehalten werden. Nur wenige Widersprüche zur festgelegten Norm sind ohne Bedeutung. Wurde z. B. ein Textfeld zu lange definiert, wird das nicht als Fehler angesehen. Die Prüfungen mit den Attributdomains zeigen, ob die Inhalte korrekt sind. Sind die Domains eingehalten, ist die definierte Feldlänge egal. Beim Einbringen der Geodaten in die Datenbank des Landes wer-

den die gelieferten Ebenen in Layer mit korrekten Spaltendefinitionen kopiert, wobei unterschiedliche Längenformate kein Problem sind.<sup>38</sup> Analoges gilt für lange und kurze Ganzzahlen<sup>39</sup> sowie lange und kurze Kommazahlen.<sup>40</sup>

Die Prüfung der logischen Konsistenz wird im Kapitel 7.4.2.4 beschrieben. Diese Tests überdecken alle Festlegungen der Schnittstelle.

Die Widmungsebene des FWP's nutz\_f muss die gesamte Gemeinde ausfüllen. Es wird geprüft, ob die Gesamtfläche der Gemeinde und die Summe der Widmungsflächen übereinstimmen. Hier ist noch eine Erweiterung der Prüfung denkbar, indem eine Topologieprüfung durchgeführt wird. Das hätte den Vorteil, dass Planern der Ort der Überlagerung bzw. des Loches bekannt gegeben werden kann. Die Basissoftware bietet Werkzeuge zur Prüfung der Topologie an, wofür die teure Lizenzstufe „advanced“ benötigt wird. Alle anderen Prüfungen finden mit der Lizenzstufe „basic“ das Auslangen. Derzeit benötigt nur der Programmteil der Verwaltung die Lizenzierungsstufe „advanced“.

Es wird geprüft, ob alle Objekte innerhalb der Gemeindegrenze liegen, wofür die Gemeindeabgrenzungen bekannt sein müssen. Diese ergibt sich aus dem Envelope aller Grundstücke der Gemeinde. Über die innere Positionsgenauigkeit der Grundstücke gibt die Schnittstellenbeschreibung des BEVs Auskunft:

„Linienstützpunkte (,VERTEX‘ einer ,POLYLINE‘) werden, wenn sie koordinatenmäßig mit Punkten ident (substituiert) sind, ebenfalls mit 2 Nachkommastellen angegeben. Bei allen übrigen rein graphischen Linienstützpunkten werden 3 Nachkommastellen (mm-Genauigkeit) angegeben.“ [71, S. 9]

Arbeitet man bei der Ermittlung der Gemeindegrenzen der Steiermark auf Basis von Grundstückskatasterdaten mit einer Toleranz von einem Millimeter, innerhalb der benachbarte Koordinaten zusammengezogen werden, kommt es lediglich zu drei topologischen Fehlern. Ein Fehler kann sofort bereinigt werden, weil er eine Insel inmitten einer Gemeinde darstellt. Die beiden verbleibenden Sliver (Loch-Länge 64,4 m, an dickster Stelle

---

<sup>38</sup> Append mit der Option NOTEST

<sup>39</sup> Short Integer, Long Integer,

<sup>40</sup> Float, Double

1,11 m/Überlagerungs-Länge 133,7 m. an dickster Stelle 0,62 m) benötigen eine Abklärung durch das BEV und beziehen sich auf die Stichtagsdaten des Katasters 1. Januar 2016.

Da laut Planzeichenverordnung 2007 für die Darstellung von Gemeindegrenzen eine 0,5 mm dicke Linie verwendet wird (siehe LGBl. Nr. 12/2008 Stück 3), was bei dem Maßstab von 1:5000 eine Überdeckung in der Natur von 2,5 Meter ergibt, können die Sliver auch durch einen automatischen Algorithmus (`arcpy.eliminate_management`) beseitigt werden.<sup>41</sup> Zieht man in Betracht, dass die Steiermark aus 1,6 Mio. Grundstücken besteht, ist die DKM als topologisch richtig einzustufen. Es gibt kaum Datensammlungen dieser hohen Qualität.

Da die Planer vom Land die Gemeindeabgrenzungen erhalten, kann es zu keinen Widersprüchen kommen.

Bei der DKM, welche aus positionsgenauen Grenzkatastergrundstücken (12 %)<sup>42</sup> und weniger zuverlässigen Grundsteuerkatastergrundstücken (88 %) besteht, werden vor allem die Grundsteuerkatastergrenzen immer wieder Grenzberichtigungen unterzogen. Handelt es sich bei der Grundstücksgrenze gleichzeitig um die Abgrenzung zweier Gemeinden (Bezirke, Bundesländer), wird auch diese Verwaltungsgrenze mit verändert. Darüber hinaus tauschen Gemeinden ihr Zuständigkeitsgebiet untereinander ab. Im Bereich der Grenzberichtigungen kommt es zu Verschiebungen, die in aller Regel unter 10 m sind, und Gemeinden tauschen im Allgemeinen nur wenige Grundstücke untereinander ab. Die Gemeindegrenze ist relativ stabil.

Im Rahmen der Gemeindestrukturreform wurden jedoch viele Gemeinden vereinigt. Einige wurden geteilt und die Teile wurden mit anderen Gemeinden zusammengefasst. Von den

---

<sup>41</sup> Darüber hinaus enthält die DKM des BEVs auch eine Ebene aller Linien, wobei Grundstücksgrenzen, die gleichzeitig Gemeindeabgrenzungen darstellen, speziell gekennzeichnet sind. Die Gemeinden lassen sich also einerseits durch Zusammenfassung der Grundstücke als auch durch Extraktion der Gemeindegrenzen und anschließende Flächenbildung bestimmen. Leider weist diese Klassifikation einige wenige Fehler auf, weshalb von diesem Weg Abstand genommen wurde.

<sup>42</sup> Die Prozentzahl bezieht sich auf dem Grundstückskataster der Steiermark, Stichtagsdaten 1. Januar 2016

542 Gemeinden blieben am Ende des Prozesses 287 Kommunen übrig. Da die Datenlieferung zukünftig für diese stark differierenden Gebiete erfolgt, kommt es zu Problemen im Updatefall.

Ein gemeinsamer Datenbestand des FWPs und OEPs wird aufgebaut, indem die Objekte des FWPs und OEPs mit ihrer Gemeindenummer versehen werden und in steiermarkweite Ebenen kopiert werden. Im Fall eines Updates werden die Objekte mit dieser Gemeindenummer gelöscht und durch die neuen Objekte ersetzt. Das Verfahren beruht auf fixen Gemeindegrenzen, was nicht real ist.

Gemeinden, die von der Gemeindestrukturreform betroffen sind, hören auf zu existieren. Für die aus den alten Gemeinden gebildeten Kommunen werden neue Gemeindenummern vergeben. Die neuen Gemeinden werden von einem Kommissär verwaltet, der die Geschäfte übernimmt, bis ein neuer Gemeinderat gewählt wurde. Der Kommissär schreibt den alten FWP fort. Diese Vorgehensweise wurde im digitalen Bereich nachgebildet. Zusammengelegte Gemeinden erhalten die Gemeindenummer der neuen Gemeinde. Einige (vier) alte FWPs bzw. OEPs waren zu teilen, und den Teilbereichen ist die jeweilig neue Gemeindenummer zuzuordnen. Das Update kann nun wiederum mit der Gemeindenummer erfolgen.

Der kritische Leser wird einwerfen, dass man mit entsprechenden Gemeindeflächen die alten Daten löschen könnte und dann die neuen Objekte in das bestehende Loch einfügt. Die Arbeit der Umschreibung von Gemeindenummern in den alten Plänen wäre obsolet. Da die Gemeindeabgrenzung jedoch veränderlich ist, kann es bei diesem Verfahren passieren, dass alte Planteile erhalten bleiben oder dass von Nachbargemeinden Objektteile gelöscht werden. Historische Gemeindedaten in der notwendigen Präzision sind nicht vorhanden.

Natürlich kann es bei diesem Verfahren dazu kommen, dass sich benachbarte Gemeinden im Randbereich überlagern oder Lücken auftun. Mithilfe eines SQL-Statements kann die benötigte Gemeinde aus der steiermarkweiten Sicht extrahiert werden. Der Inselplan entspricht den derzeitigen Verhältnissen. Der Gesamtplan zeigt die rechtliche Situation mit den topologischen Problemen, die jedoch die rechtliche Realität widerspiegeln. Sind die Ände-

rungen zu gravierend, kann ein Änderungsverfahren durchgeführt werden. Die Änderungsdaten werden im Web-Dienst über den Stammdaten gezeichnet, damit die Informationen des Stammpfandes sowie der Änderungen abgefragt werden können.

Mit dem präsentierten Verfahren, wird die rechtliche Praxis digital abgebildet. Aufwendige Änderungen des FWP's und OEP's aufgrund von Gemeindegrenzänderungen sind nicht notwendig.

Im Web-Service des Landes kann der beliebte steiermarkweite Datenbestand angeboten werden, wobei es im Randbereich der Gemeinden zu Löchern oder Überlagerungen kommen kann, entsprechend der Rechtssituation. Beim FWP geht es darum, räumliche Rechte und Pflichten abzubilden. Widersprüche sind selten und bedürfen einer amtswegigen oder richterlichen Bereinigung, die nach vorgegebenen Abläufen erfolgt, wo alle Beteiligten ihre Rechte wahren können. Bedürfnisse der Geoinformation haben zurückzutreten. Die Topologieprüfung kennt deswegen auch die Möglichkeit, Ausnahmen auf Ebene der Objekte zu definieren.

### 5.3. Positionsgenauigkeit

Die notwendige Positionsgenauigkeit wird durch den Zweck festgelegt. Aufgabe des FWP's ist es das Gemeindegebiet in Bauland, Freiland und Verkehrsflächen aufzuteilen (vergl. § 26 StROG, LGBl. Nr. 49/2010 Stück 21). Mit dieser Kategorisierung werden räumliche Rechte und Pflichten festgelegt. Die Baulandkategorien sind mit der Berechtigung verknüpft, dort Gebäude zu errichten (vergl. § 30 StROG LGBl. Nr. 49/2010 Stück 21). In den Freilandkategorien ist das nur in Ausnahmefällen möglich (vergl. §33 StROG LGBl. Nr. 49/2010 Stück 21). Der Plan muss zeigen, welche Grundstücke oder Grundstücksteile mit welchen Berechtigungen versehen sind. Die absolute Lagerichtigkeit ist dabei nebensächlich. Es entsteht Rechtssicherheit, denn der Bürger weiß, was auf seinem Grundstück zulässig ist. Dementsprechend erfolgt die Planung auf Basis des Grundstückskatasters. Die Katasterdaten, auf welchen die Planungen erfolgten, sind Teil des FWP's und dürfen nicht durch aktuellere Unterlagen ersetzt werden. In diesen aktuelleren Unterlagen könnten Verschwenkungen von ganzen Rieden korrigiert worden sein, wodurch der notwendige Zusammenhang zwischen Grundstück und Widmung verloren gehen würde. Die höhere Positionsgenauigkeit anderer Katasterdaten würde zu einer wesentlichen Verschlechterung des

Gesamtdatenbestandes führen. Mit der Integration der Katasterdaten in das Datenpaket FWP wird der rechtsgültige Papierplan digital nachgebildet.

Wesentlich ist nicht die absolute Positionsgenauigkeit, sondern die relative Positionsgenauigkeit zum Kataster. Da in der Regel ganze Grundstücke gewidmet werden und Planer mit automatischem Fangen arbeiten, ist eine hinreichende relative Genauigkeit gegeben. Es ist zulässig, Teile von Grundstücken zu widmen, was deutlich seltener erfolgt. Elektronisch lässt sich die relative Positionsgenauigkeit der Festlegungen zum Kataster kontrollieren, indem alle Grundstücksgrenzen um einen kleinen Betrag (0,5 m) gebuffert werden. Aus den Baulandabgrenzungen, die das wesentliche Element des FWPs sind, werden die Vertices extrahiert. Diese Punkte müssen dann entweder auf einer Grenzlinie liegen oder außerhalb des Bufferbereichs. Das lässt sich mit Standardoperationen durchführen. Für Vertices innerhalb des Buffers, die jedoch nicht auf der Grenzlinien liegen, wird angenommen, dass auf das automatische Fangen vergessen wurde.

Erfolgte die Flächenerfassung ohne „Autocomplete“-Funktionalität, wo der angrenzende Teil eines Polygons zu einem Nachbarpolygon von Computer ermittelt wird, und ohne automatischen Fangen, liegen Überlagerungen und Löcher vor, welche mit einer Topologieprüfung ermittelt werden können. Die anzuwendenden Regeln lauten „must not have gaps“ und „must not overlap“.

Automatische Bereinigungen sind möglich. Dabei werden die Vertices innerhalb einer Toleranz zusammengezogen (Standardwerkzeug integrate). Wird dem Kataster gegenüber den Widmungen ein höheres Gewicht zugeordnet, werden die Vertices der Widmung an die Vertices des Katasters angepasst.

Das Standardwerkzeug integrate führt Koordinaten unterschiedlicher Ebenen innerhalb kleiner Distanzen zusammen (siehe auch [102]). Grundstücksgrenzen werden aus Grenzpunkten gebildet, wobei Grenzpunkte klare Eckpunkte des Grenzverlaufes sein können oder auch Zwischenpunkte innerhalb von nahezu geraden Grenzverläufen. Diese Zwischenpunkte werden Läufer genannt. Wendet man das Werkzeug integrate an um grob erfasste Widmungsflächen an den Kataster anzupassen, versagt die Methode bei Läufern, da in der Nähe von Läufern in aller Regel keine Zwischenpunkte bei der ungefähren Erfassung der Widmungsflächen gesetzt wurden.

Der OEP wird auf Basis des Orthofotos im Maßstab 1:10000 dargestellt. Dieser Planungsmaßstab vereinfacht Genauigkeitsfragen. Der FWP berücksichtigt den OEP, wobei beide Pläne meist von einem Raumplaner erstellt werden. Um Interpretationsprobleme zu verhindern, erfolgt die Erfassung des OEPs meist auch auf Basis des Katasters. Es gilt das zuvor Geschriebene.

#### 5.4. Zeitgenauigkeit

Alle Termine im Zusammenhang mit Raumplanungsverfahren werden in der Anwendung Terminevidenz der Abteilung 13 geführt, wobei auf alle benötigten Informationen dieses EDV-Projektes Lesezugriff besteht. Planabgaben benötigen die Vorgabe des Verfahrensfalls. Über die Verfahrensfallnummer kann mit der Terminevidenz das Datum der Kundmachung des Plans ermittelt werden. Ab diesem Zeitpunkt tritt der Plan in Rechtskraft und gilt so lange, bis ein neuer Plan kundgemacht wird.

Leider werden nicht immer alle Pläne digital abgegeben, weshalb der im Web-GIS gezeigte Plan nicht aktuell ist.

In nicht periodischen Abständen erfolgt ein Abgleich der Verfahrensfallnummern gelieferter Geodaten mit der Terminevidenz, welche die Verfahrensfallnummern der aktuellen Planversionen der Gemeinden für den FWP und den OEP enthält. Aktualisierungen werden eingefordert. Dabei kann es vorkommen, dass bis zu 5 % der Pläne beanstandet werden.

Grundsätzlich wird zwischen Stammplänen und Änderungen unterschieden, wobei Stammpläne das gesamte Gemeindegebiet beschreiben und Änderungen nur Teilbereiche. Änderungen heben nur diese Teilbereiche des Gemeindegebietes auf. Die digitale Abgabe der Änderungen ist softwaretechnisch realisiert. Dabei werden die Änderungsdaten wie eine elektronische Briefmarke über den Stammplan geklebt. Nur eine verschwindend kleine Anzahl an Änderungen wurde so abgegeben, da diese Abgabe erst im Oktober 2016 verordnet wurde.

Um den Bürgern trotzdem die Änderungen zugänglich zu machen, wurde eine Anwendung entwickelt, mit der ein Beamter die Gebiete der Änderungen erfasst. Diese Änderungsgebiete werden mit dem Textdokument der Änderung verlinkt. Im Web-Service erscheinen

diese Bereiche mit einer dicken lila Randlinie und das die Änderung begründende Schriftstück kann mit einem Klick eingesehen werden. Derzeit sind etwa 800 Änderungen erfasst. Seit etwa Dezember 2015 kam es aus personellen Gründen nur zu wenigen Neuerfassungen von Änderungen.

Aktualität kann nur sichergestellt werden, wenn der Ablauf des Genehmigungsverfahrens dahin gehend abgeändert wird, dass ohne digitale Abgabe korrekter Geodaten des FWPs oder OEPs nicht in Rechtskraft treten. Vorschläge zur Verfahrensänderung sind im Kapitel 3.6 beschrieben und das Kapitel 7.3 beschreibt die softwaretechnische Umsetzung dieser Vorschläge. Eine neue Planzeichenverordnung ist mit 1. Oktober in Kraft getreten (siehe LGBI. Nr. 80/2016), in der digitale Änderungen verpflichtend vorgesehen sind. Diese Verordnung kann als Zwischenschritt zum digitalen Gleichstück eines rechtsgültigen Planes gesehen werden.

### 5.5. Thematische Genauigkeit

Die Richtigkeit der Geodaten beschränkt sich im derzeitigen Verfahren auf die Frage, inwieweit die Geodaten mit dem rechtsrelevanten Papierplan übereinstimmen. Es kann keine Garantie für die Richtigkeit der Geodaten abgegeben werden. Dieses Problem lässt sich nur mit Änderungen des Verfahrensablaufes in den Griff bekommen. Vorgeschlagen wird, dass der Plan nicht mehr vom Raumplaner erstellt wird. Dieser übermittelt die Geodaten dem Land und aus diesen Geodaten wird der rechtsgültige Plan erstellt. Damit ist sichergestellt, dass der Rechtsplan nicht von den Geodaten des Landes abweicht. Eine genaue Beschreibung dieses Verfahrens erfolgt im Kapitel 3.6 und das Kapitel 7.3 beschreibt die Auswirkungen der Verfahrensänderung auf die Software.

Manche Geoinformatiker wünschen sich, dass den Geodaten selbst die Rechtsgültigkeit zugeordnet wird. Damit wäre die thematische Richtigkeit per Gesetz gegeben. Im Kapitel 4.5.5.3 wurde ausgeführt, dass die elektronische Unterschrift amtlicher Dokumente auch eine Bildmarke benötigt. Für die Bildmarke besteht keine technische Notwendigkeit. Der Leser des amtlichen Dokumentes soll optisch erkennen können, dass es sich um ein offizielles Dokument handelt. Die Lesbarkeit eines digitalen Dokumentes für die breite Masse

ist dem Gesetzgeber wichtig. Deshalb wird vermutet, dass Geodaten, die nur mit komplexen Programmen visualisiert werden können, keine Rechtsgültigkeit erlangen werden. Diese Einschränkung scheint sinnvoll zu sein.<sup>43</sup>

Im Kapitel 7.4.3.7.2 wird ein selbstlernender Algorithmus vorgestellt. Er bessert Verstöße gegenüber Attributdomains in der Weise aus, wie es andere Planer zuvor getan haben. Diese Korrekturen sind naturgemäß nicht immer richtig, was zur Folge hat, dass die thematische Genauigkeit auch verschlechtert werden kann. Dem kann entgegengewirkt werden, indem der Korrekturalgorithmus nur lernt, was mehrfach wiederholt vorgegeben wird. Schwellwerte sind notwendig. Kommt es häufiger zu differierenden Ausbesserungen, müssen gelernte Korrekturvorschläge auch wieder vergessen werden. In Bezug auf thematische Richtigkeit ist eine extrem schnelle Auffassungsgabe kontraproduktiv und Vergessen eine Notwendigkeit.

Im Kapitel 3.7 wurden Amtsabläufe vorgeschlagen, welche zu Qualitätsverbesserungen von Ersichtlichmachungen führen. Da diese Daten auch Planungsgrundlagen sind, steigt damit die Qualität des FWPs und OEPs

## 5.6. Zweck

Der Zweck des OEPs und FWPs besteht in seiner rechtlichen Wirkung, welche im steirischen Raumordnungsgesetz 2010 geregelt ist. Novellen passen Zweck und Wirkung an. Eine Zusammenschau ist auf der Internetseite „Steirisches Raumordnungsgesetz“ zu finden [72].

Durch Novellierungen kann es zu Verschiebungen der rechtlichen Wirkung kommen. So wurde die Festlegung Dorfgebiet um den Ausschluss zusätzlicher Bauformen erweitert. Es kommt zu einer semantischen Transformation, wobei das Versteinerungsprinzip gilt, wel-

---

<sup>43</sup> Technisch wäre es ein Leichtes, für die Shapedateien eines Geodatensatzes den Hashcode zu ermitteln und ihn samt Bildmarke in einer zusätzlichen Datei abzulegen. Die Datei wäre die elektronische Unterschrift der Geodaten.

ches besagt, dass die zum Zeitpunkt der Kundmachung gültige Rechtsvorschrift anzuwenden ist. Eine Verschränkung des Zwecks mit der Aktualität ist gegeben. Eine eingehende Beschreibung dieses Effektes findet man im Kapitel 7.4.4.

## 5.7. Verwendung

Der FWP und im geringeren Ausmaß der OEP werden hauptsächlich als Auskunftssystem verwendet. Dabei geht es nicht nur um die Widmung der Grundstücke, sondern auch um Ersichtlichmachungen, welche einerseits die Festlegungen begründen und andererseits Aufschluss über die Wohnqualität geben. So sind die roten Gefahrenzonen, wie sie von der Wildbach- und Lawinverbauung festgelegt werden, ein Ausschlussgrund für Bautätigkeiten, und zeigen auf, warum gewisse Gebiete als Freiland festgelegt sind. Geruchskreise um Tierhaltebetriebe weisen darauf hin, dass es zu Belästigungen aufgrund von Gestank kommt.

Neben der Aufgabe als Auskunftssystem über rechtliche Festlegungen werden der FWP und der OEP als Datenquelle für andere Datenbanken verwendet. Beispiele sind Skilifte und Infrastruktureinrichtungen wie Kindergärten und Schulen, die im FWP verzeichnet sind und aus dieser Datenquelle extrahiert werden.

Viele Analysen zielen auf das Bauland ab, welches Teil des FWPs ist. So wird untersucht, ob die Haltestellen des öffentlichen Verkehrs die Baulandflächen ausreichend versorgen, Feuerwehren in vorgegebenen Zeiten auf Basis des Straßennetzes die Flächen erschließen und Ähnliches mehr.

Die Verwendung von Daten, in welcher Form auch immer, führt zu Verbesserungen. Jede Abfrage des FWPs im digitalen Atlas stellt eine Kontrolle dar. Widersprüche zum Rechtsstand fallen auf. Darüber hinaus sind Raumplaner und Gemeinden sich bewusst, dass heutzutage ein Großteil der Information digital konsumiert wird. Deshalb machen sich in vielen Kommunen Gemeindebedienstete und Raumplaner die Mühe, die digital abgegebenen Pläne noch einmal auf ihre inhaltliche Richtigkeit zu überprüfen. Wir erkennen das an Mehrfachabgaben des gleichen Planes. Widersprüche der alten Abgabe gegenüber dem rechtsgültigen Papierplan werden korrigiert (siehe Anhang – Fehleranalyse 4.2).

Die Gemeinden Graz, Wien etc. bieten ihre FWP's in einem Kartendienst an, wobei sie zuvor in einem Haftungsausschluss auf mögliche thematische Ungenauigkeiten hinweisen. Die Daten sind jedoch fehlerfrei. Der Internetservice des Amtes der Stmk LREG weist ebenfalls auf thematische sowie zeitliche Fehler hin, die bei einigen Gemeinden tatsächlich vorliegen. Positive Erfahrungen untergraben die Bedeutung des steirischen Warnhinweises insbesondere bei Usern aus Wien, wie bei telefonischen Beratungsgesprächen festgestellt werden musste. Aus dem Dilemma kann man sich befreien, indem man versucht, möglichst schnell ähnliche Qualität zu erreichen.

Das Bauland, als wichtigster Auszug des FWP's, findet Eingang in viele Analysen, die im Amt der Steiermärkischen Landesregierung in der Regel das gesamte Landesgebiet betreffen. Es handelt sich dabei um Inselpläne, welche elektronisch zusammengefügt wurden. Grundlage der einzelnen Pläne war der zum Planungszeitpunkt aktuelle Grundstückskataster dieser Gemeinde. Da der FWP nur alle 10 Jahre erneuert werden muss, haben wir Unterschiede bis zu 10 Jahren in der Planungsgrundlage, was zu Spannungen im steiermarkweiten Geodatenatz führt. Aufgrund der hohen Qualität der DKM, die es schon seit mehr als 10 Jahren gibt, sind diese Einflüsse nicht gravierend.

## 5.8. Herkunft

Es muss sichergestellt sein, dass die Personen, welche Geodaten übermitteln, dazu berechtigt sind. Im derzeitigen Verfahrensablauf, wo den Geodaten noch nicht die entscheidende Rolle zukommt, erfolgt das mit der Vorgabe eines Usernamens und eines Passwortes. Bis dato kam es zu keinen Problemen in Bezug auf die Herkunft der Daten. Nur berechtigte Personen haben mit dem Portal kommuniziert.

Nichtsdestotrotz wird mehr Sicherheit benötigt, wenn die Rolle der Geodaten sich ändert und Rechtsfolgen mit den Geodaten verbunden werden. Wenn Daten über das Internet ausgetauscht werden, dann muss die Herkunft der Daten zweifelsfrei bekannt sein und es muss sichergestellt sein, dass die Daten nicht manipuliert wurden. Das ist ein Problem, mit dem sich die allgemeine EDV sehr intensiv beschäftigt. Es wurde von der Kryptografie ein Verfahren mithilfe von Signaturen entwickelt, welches bei den softwaretechnischen Qualitätsaspekten beleuchtet wurde (siehe Kapitel 4.5.5). Entsprechende Softwarekomponenten sind im Amt der Stmk LREG vorhanden und können auch in dieses Projekt integriert werden.

Gemeinden, die ihrerseits Schreiben noch nicht rechtsgültig digital fertigen können, müssten aufrüsten, um Pläne digital signiert abzuliefern. Dabei handelt es sich um eine Investition, welche von Gemeinden vielfach genutzt werden könnte.

## 6. Qualitätsbegleitende Maßnahmen und Gegebenheiten

Diese Dissertation erfolgte, wie nahezu alle Projekte in der Softwareentwicklung und Geo-datenverarbeitung, eingebettet in ein Umfeld aus Kollegen und Nutzern. Dem Projekt wurde durch die Arbeit anderer Speicherplatz bereitgestellt, die Sicherung wurde von den Mitarbeitern der zentralen EDV des Landes übernommen, die Lizenzserver für die Basissoftware<sup>44</sup> wurde von Kollegen aufgesetzt und gewartet, das Netzwerk wurde von anderen geplant und vieles mehr. Für diese perfekt funktionierenden Leistungen sei gedankt. Den Bürgern wurde ein System bereitgestellt, welches Auskunft über die Widmung erteilt und für Raumplaner wurde eine Plattform geschaffen, mit der sie ihre Produkte kontrollieren und präsentieren können. Um dies zu erreichen, ist ein hohes Maß an Professionalität notwendig. Neben dem klassischen handwerklichen Können gehören dazu Verlässlichkeit und die Fähigkeit, im Team und mit dem Team zu agieren.

### 6.1. Passiv aggressives Verhalten, Kritikfähigkeit

Sieht man ein Problem auf ein Projekt zukommen, kann man auf zwei Arten darauf reagieren. Zum einen ist es möglich, Gegenmaßnahmen einzuleiten, auch wenn die Problemstellung nicht in den eigenen Bereich fällt. Gute Kontakte und freundschaftliche Verhältnisse machen es leicht, Dinge anzusprechen und zu einem guten Ende zu bringen. Zum anderen kann man auch Vermerke und Gesprächsnotizen verfassen, mit dem Ziel für negative Folgen nicht verantwortlich gemacht werden zu können. Nur im zugewiesenen Zuständigkeitsbereich wird man aktiv. Das einzige Interesse scheint darin zu liegen, möglichst unbeschadet das Problem zu überstehen, und es gegebenenfalls für einen beruflichen Aufstieg zu nutzen. Diese aggressive Passivität lähmt Projekte und führt dazu, dass wertvolle Zeit in unnötige Absicherung verschwendet, und nicht für die Problembeseitigung aufgewandt wird. Ein gutes Klima ist notwendig, und Kritik muss angenommen werden können. Das setzt insbesondere voraus, dass man Kritik als fachlich fundierte Stellungnahme abgibt und dass diese Stellungnahme nicht dazu dient, das Gegenüber bloßzustellen und herabzuwürdigen (aktive Kritikfähigkeit). Langfristig wird das berufliche Umfeld nur dann Kritik annehmen, wenn man selbst auch bereit ist, Kritik als wohlgemeinten Verbesserungsvorschlag

---

<sup>44</sup> ArcGIS-Desktop, Microsoft – SQL Server, Oracle, ArcGIS Server, SDE

anzunehmen (passive Kritikfähigkeit). Kritik ist ein wertvoller Schatz, der es ermöglicht, über den eigenen Tellerrand zu blicken.

Kritik kann nur dann von anderen oder einem selbst angenommen werden, wenn sie aufzeigt, mit welcher Norm sie in Konflikt steht und in welcher Quantität dieser Widerspruch besteht. Eine kurze Risikoanalyse ist sinnvoll, bevor man ein Problem kritisiert. Natürlich lässt sich auch die Normativität hinterfragen. Ist es wirklich notwendig, einen Dienst 24 Stunden am Tag bereitzustellen und damit große Probleme mit Wartungstätigkeiten zu verursachen, oder reicht auch ein 12-Stunden-Service aus?

## 6.2. Selbstverpflichtung

In der arbeitsteiligen Gesellschaft sind wir darauf angewiesen, uns auf die Ergebnisse der Arbeit anderer zu verlassen. Man geht Selbstverpflichtungen ein und baut auf Commitments anderer auf. Wird eine Zusage nicht eingehalten, kann es zu einer Kettenreaktion an Unzuverlässigkeit kommen, da alle aufbauenden Verpflichtungen nicht eingehalten werden können.

Deshalb ist es von essenzieller Bedeutung, vor jedem Commitment zu prüfen, ob dieses auch tatsächlich eingehalten werden kann. Es obliegt demjenigen, der ein Commitment eingeht, die reale Umgebung einzuschätzen und nicht von einer idealen Welt auszugehen. Ein bekanntes Beispiel soll das klarmachen.

Jemand hat eine Verabredung am Montag um 8:00 Uhr im Zentrum von Graz und kommt eine halbe Stunde zu spät. Als Begründung wird ein Verkehrsstau angegeben. Sein Gegenüber wird – wie es die Höflichkeit gebietet – das Argument wohlwollend entgegennehmen und sich denken, dass die Straßen in Graz am Morgen immer verstopft sind. Es handelt sich um eine Banalität, die jeder, der in Graz wohnt, kennen müsste. Verkehrsstaus sind im Zeitmanagement einzuplanen.

In GI-Projekten verhält es sich gleich. Geht man ein Commitment ein, ist die Umgebung zu überprüfen. Neben den softwarespezifischen Voraussetzungen ist abzuklären, ob man ausreichend Zeit für eine Aufgabe zur Verfügung hat. Verpflichtet man sich auf eine Lösung in 10 Tagen, so kann das Gegenüber erwarten, dass man in 10 Tagen das Ergebnis abgeliefert.

Man hat abzuschätzen, wie viele sonstigen Aufgaben in diesem Zeitraum vorkommen werden und in die Kalkulation mit einzubeziehen. Um ein Gefühl für den tatsächlichen Zeitaufwand zu bekommen, ist es sinnvoll, Aufgaben in ihrem Zeitaufwand zu schätzen und gegenüber der Realität abzugleichen. Meine Schätzungen sind in der Regel viel zu optimistisch. Ich benötige etwa doppelt so lange, wie ich annehme, zu brauchen. Hat ein Projekt mittlere Priorität, wird es durch andere wichtigere Tätigkeiten wiederum um den Faktor 2–3 in die Länge gezogen. Ein Commitment sollte nur eingegangen werden, nachdem man sich einige Zeit zurückgezogen hat, versucht hat, alle Umgebungsgrößen abzuschätzen, und den Zeitbedarf mit seinen eigenen Faktoren multipliziert hat.

Kennt man seine eigenen Faktoren, ist darauf Rücksicht zu nehmen, wenn von anderen Leistung eingefordert wird. Ich unterschätze die Arbeit, die ich von anderen einfordere. Darüber hinaus sind hohe individuelle Unterschiede häufig. Auch ist es höchst unprofessionell, Kollegen zu einem Commitment zu überreden. Man kann davon ausgehen, dass es nicht eingehalten wird.

Commitments sollten eingehalten werden. Sie mit Ritualen zu versehen und aus ihnen mehr zu machen, als sie eigentlich sind, ist nicht zweckdienlich. Schon vor 2000 Jahren wurde das knapp und trefflich vom Evangelisten Matthäus festgehalten.

„Ihr habt gehört, dass zu den Alten gesagt worden ist: Du sollst keinen Meineid schwören, und: Du sollst halten, was du dem Herrn geschworen hast. Ich aber sage euch: Schwört überhaupt nicht, weder beim Himmel, denn er ist Gottes Thron, noch bei der Erde, denn sie ist der Schemel für seine Füße, noch bei Jerusalem, denn es ist die Stadt des großen Königs. Auch bei deinem Haupt sollst du nicht schwören; denn du kannst kein einziges Haar weiß oder schwarz machen. Euer Ja sei ein Ja, euer Nein ein Nein; alles andere stammt vom Bösen.“ (Mt 5.33 – 5.37)

Wie man tatsächliche Selbstverpflichtung in der Sprache erkennt, beschreibt eindrucksvoll M. Martin in seinem Buch *Clean Coder* [21, SS. 87–95].

### 6.3. Fertig

Spricht man mit Kollegen über zu erledigende Aufgaben, hört man häufig den Begriff fast fertig. Leider zeigt die Erfahrung, dass zwischen fast fertig und ganz fertig noch erhebliche Aufwendungen notwendig sind.

In vielen Lebensbereichen herrscht eine Ungleichverteilung zwischen dem Mitteleinsatz und dem Ertrag. Das Prinzip wird auch Pareto-Prinzip genannt und meint im Speziellen, dass mit 20 % des Einsatzes 80 % des Effektes erzielt werden. Der Volkswirt Wilfredo Pareto untersuchte den italienischen Grundbesitz und fand heraus, dass 20 % der Familien 80 % des Landes besaßen. In vielen anderen Bereichen wurden ähnliche Zahlenwerte ermittelt [73, S. 146].

Wir können davon ausgehen, dass weite Teile der GI-Arbeit mit wenig Mühe zu erledigen sind. Sie machen Freude und man hat mit wenig Mitteleinsatz viel getan. Am Schluss der Arbeit werden übrig gebliebene Tätigkeiten erledigt, die wie Kleinigkeiten aussehen. Intuitiv wissen wir über ihren großen Aufwand im Vergleich zum Erfolg und scheuen die Erledigung, weshalb sie am Schluss durchgeführt wird. Dem Pareto-Prinzip folgend wird nach 80 % der erbrachten Leistung gerne von fertig gesprochen. Die tatsächliche Fertigstellung wird noch ca. 5-mal so lange dauern. Diese gewagte Feststellung bedarf sicher wissenschaftlicher Studien, die eher in den Bereich der Psychologie als der Geoinformatik fallen. Erfahrene Kollegen sind sich des Pareto-Effektes bewusst, und gehen entsprechend vorsichtig mit dem Begriff fertig um.

Ludewig und Lichter führen dazu aus, dass sich nur in einer idealen Welt der Fertigstellungsgrad folgenderweise ergibt [12, SS. 139–140]:

Fertigstellungsgrad = Ist-Aufwand / geschätzten Gesamtaufwand.

Diese Annahme ist jedoch nicht zielführend, da der Gesamtaufwand nicht ausreichend genau geschätzt werden kann. Sie schlagen zum Zeitpunkt der Prognosestellung eine neue Aufwandsschätzung vor.

Fertigstellungsgrad = Ist-Aufwand / (Ist-Aufwand + Restaufwand).

Die Vorgehensweise erscheint sehr professionell, da man sich mit dem Problem der Aufwandsschätzung des verbleibenden Restes noch einmal detailliert auseinandersetzen muss.

## 7. Die entwickelte Software

### 7.1. Allgemeines

Die für die Analyse der Abgaben notwendigen Pläne wurden mit einer alten Software eingelesen, welche ich konzipiert habe und die von einer Firma codiert wurden. Die alte Software war in der Lage, Daten zu prüfen. Es stellte sich heraus, dass alles neu gemacht werden muss, will man in Richtung Rechtsgültigkeit gehen. Nun habe ich im Rahmen dieser Dissertation, von der Organisation beginnend bis zur Codierung, ein neues Programm entwickelt, welches die Daten prüft, repariert, verwaltet und notwendige Datenextrakte zieht. Diese Software soll in diesem Kapitel beschrieben werden.

Ausgenommen möglicher Wartungsarbeiten, läuft das System im Amt der Stmk LREG selbstständig. Es ist kein Bedienpersonal für die Übernahme der Geodaten notwendig, da alle Arbeitsschritte digital abgebildet sind und alle notwendigen Entscheidungen vom Rechner getroffen werden können.

Die Analyseergebnisse der alten Abgaben, wie sie im Kapitel 2 beschrieben werden, sind für die Entwicklung von Programmen herangezogen worden, welche automatische Fehlerkorrekturen leisten. Um die Menge an verschiedenen Attributfehlern zu beherrschen, wurde ein selbstlernendes System entwickelt. Ansonsten erfolgen die Korrekturen im Rahmen der Kontrollen mit maßgeschneiderten Algorithmen. Die Beschreibung der komplexen Datenverwaltung erfolgt in zwei Tabellen mit jeweils ca. 1000 Einträgen. Änderungen sind aufgrund der datenbankbasierenden Beschreibungen der Planinhalte und ihrer Verwaltung leicht durchzuführen.

Um ein Projekt dieser Größe und der geforderten Qualität entwickeln zu können, war es notwendig, Strategien zur Beherrschung der Komplexität zu entwickeln. Vorgehensmodelle wurden bewertet und Handlungsstrategien entwickelt. Qualitätskriterien von Software und Geodaten, mit denen in der Regel Bewertungen durchgeführt werden, wurden verwendet, um gute Software zu schreiben.

Die Software wurde so konzipiert, dass verschiedene Planarten mit ihr abgearbeitet werden können. Die Beschreibung der Pläne, der Prüfungen und der Datenverwaltung erfolgt in

Datenbanken und kann auf beliebige Themen ausgeweitet werden. Neben der Abwicklung des OEPs und des FWP's werden derzeit Beleuchtungskörper von Gemeinden über die Schnittstelle abgearbeitet. Das war die erste Erweiterung.

Alle Entwicklungen erfolgten im Hinblick auf den Wunsch, den Geodaten die notwendige Qualität zu verleihen, damit sie als rechtsrelevante Gleichstücke gelten können. Dazu bedarf es aber auch rechtlicher Schritte, die aufgezeigt wurden. Derzeit erlangte gerade die Planzeichenverordnung 2016 Rechtskraft (siehe LGBL. Nr. 80/2016), welche einen Zwischenschritt darstellt. Um Geodaten zu Gleichstücken der rechtsgültigen Pläne zu machen, ist eine Novellierung des Raumordnungsgesetzes notwendig, was aus Sicht der Verwaltung schwieriger ist als die Änderung einer Verordnung.

## 7.2. Use-Case-Diagramm

Um einen Überblick über die entwickelte Software zu erhalten, eignet sich ein Use-Case-Diagramm. Die einzelnen Aufgaben der Software und die Interaktion des Users werden skizziert.

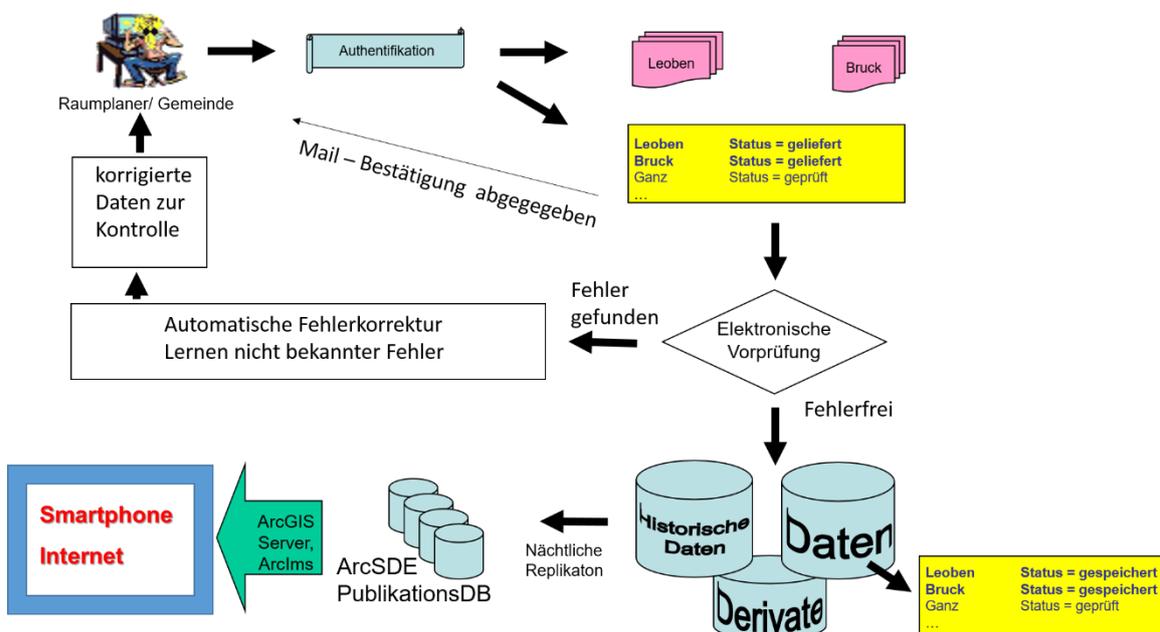


Abbildung 17: Use-Case-Diagramm der Datenübernahme

Gemeindemitarbeiter oder von ihnen beauftragte Raumplaner authentifizieren sich und übermitteln ein gezipptes Datenfile. Zuvor werden vom Abgabeformular Metadaten abgefragt, wie die betroffene Gemeinde, der Verfahrensfall und natürlich auch die Art des Planes. Der Erhalt der Daten wird durch eine automatische E-Mail bestätigt. Die Originaldaten werden archiviert und danach entpackt, wobei die vorgegebenen Metadaten die Speicherstruktur bestimmen. Die Metadaten des Übergabeformulars werden in einer Oracle-Tabelle gespeichert. Diese Tabelle, welche in Abbildung 21 als gelbes Rechteck erscheint, ist für das Projekt essenziell, denn sie speichert neben den Metadaten auch den Verarbeitungszustand der Lieferung. Der hier beschriebene Vorgang benötigt kaum Rechnerressourcen, sodass viele Abgaben parallel verarbeitet werden können.

Die elektronische Vorprüfung, Korrektur und Datenverwaltung der abgegebenen Files, welche nach der Abgabe folgen, sind wesentlich zeitaufwendiger. Ein automatischer Task fragt in kurzen periodischen Abständen die Oracle-Tabelle ab und startet die Prüfroutine, falls neue Daten eingelangt sind. Die Prüfroutine ist so ausgelegt, dass auch mehrere Prüfungen parallel abgearbeitet werden können. Kleine Änderungen einer Gemeinde können Revisionen einer anderen ganzen Gemeinde überholen. Auf die einzelnen Prüfungen wurde im Rahmen der Fehleranalyse detailliert eingegangen. Derzeit überprüft die Software die Struktur und den Aufbau der Geodaten. Das sind z. B. die Namen der Ebenen, die zugeordneten Geometrietypen, der Spaltenaufbau und die Attributinhalt.

Die Software sieht auch inhaltliche Kontrollen vor, wobei vorhanden Ebenen des Landes mit den abgegebenen Geodaten abgeglichen werden. Dabei können Ebenen verschiedener Maßstäbe und Dimensionen miteinander in Beziehung gesetzt werden. Das Verfahren eignet sich vor allem, um Ersichtlichmachungen zu verifizieren. Aber auch eine Kontrolle des Baulandes gegenüber dem Regionalplan ist auf diese Weise möglich. Diese Fähigkeiten der Software sind derzeit nicht aktiv. Im Verfahren werden nämlich die Geodaten erst nach der Kundmachung übergeben. Der Plan ist damit schon rechtsgültig. Das Aufzeigen inhaltlicher Fehler ist nicht sinnvoll, da in diesem Stadium keine Planänderungen mehr möglich sind. Werden zukünftig digitale Daten zur Auflage abgegeben, wo der Plan auf seine inhaltliche Richtigkeit geprüft wird, kann diese Kontrolle aktiviert werden. Die Abgabe von Auflagen ist in der Software und im Datenmodell fertig entwickelt.

Bei Regelverstößen versucht die Software, die Daten zu korrigieren. Das hohe Potenzial automatischer Korrekturen wurde bei der Analyse abgegebener Pläne dargelegt. Die automatische Korrektur besteht aus zwei unterschiedlichen Strängen. Zum einen gibt es statische Korrekturen, welche nach vorgegebenen Algorithmen ablaufen. Zum anderen gibt es ein lernendes System. Beim ersten Auftreten des Fehlers kann dieser noch nicht korrigiert werden. Das System fordert eine Korrektur ein und analysiert die ausgebesserten Daten. Es merkt sich, wie der Raumplaner die Daten verändert hat. Diese Information kann für automatische Korrekturen herangezogen werden. Es ist wichtig, anzumerken, dass vom System bearbeitete Daten nicht weiter verwendet werden. Sie werden lediglich durch eine automatische E-Mail dem Raumplaner zugestellt, welcher die automatischen Kontrollen auf ihre Richtigkeit hin überprüft. Der Plan soll nicht automatisch von einem Computerprogramm abgeändert werden. Es ist wichtig, dass der Planer lediglich unterstützt wird und persönlich über den Dateninhalt entscheidet. Viele Korrekturarbeiten werden damit vereinfacht. Aus persönlichen Gesprächen mit Planerinnen ging hervor, dass das Abschneiden von Überständen über die Gemeindegrenze die am meisten geschätzte Korrektur ist.

Abschluss jeder Prüfung ist immer eine E-Mail, welches das System an den Datenübermittler schickt. Diese E-Mail enthält das Prüfprotokoll. Bei einer fehlerfreien Datenübergabe geht die Mail auch an die offizielle Gemeindeadresse. Im Allgemeinen erteilt die Gemeinde Raumplanern den Auftrag, die Raumordnungspläne zu erstellen und auch die notwendigen digitalen Abgaben beim Land durchzuführen. Die Gemeinde weiß nun, dass die Datenabgabe erfolgreich war. Derzeit ist das meist der letzte Punkt des Verfahrens und es steht nichts mehr der Begleichung der Rechnung im Weg. Das positive Prüfergebnis wird in der Projekttabelle der Abgaben vermerkt.

Als nächster Schritt werden die Geodaten aufbereitet. Jedes Feature wird mit einer eindeutigen Herkunftskennung – der Upload-ID – und einigen weiteren Details versehen. Aus den Flächenwidmungsplänen werden Daten extrahiert, welche andere Ebenen updaten. Das erfolgt z. B. für die Ebene Bauland. Sie ist sehr beliebt, da die Information des FWPs auf wesentliche Grundelemente reduziert vorliegt. Darüber hinaus werden die abgegebenen Daten in eine historische Datenbank eingepflegt. Für diese Arbeiten sind wenige Grundschritte notwendig. Es bedarf jedoch einer großen Steuertabelle.

Jede Nacht wird die Produktionsdatenbank in die Publikationsdatenbank überführt. Auf diese Datenbank greift der ArcGIS-Server zu und stellt die Raumordnungspläne als Web-

Service bereit. Es gibt eine Intranetlösung und eine Internetlösung. Auf Erstere haben auch die Gemeinden Zugriff. Die beiden für PC-Bildschirme optimierten Lösungen werden durch Apps ergänzt, welche die Flächenwidmungspläne auf iOS und Android zeigen [74].

### 7.3. Erweiterungen in der Software zur Steigerung der Rechtsrelevanz

Im Kapitel 3 wurden Schritte aufgezeigt, wie man zu Geodaten kommt, welche die Rechts-situation widerspiegeln. Im Fokus stand das Verfahren, welches abgeändert werden muss. Die Gemeinde erstellt die Geodaten, aus denen vom Land die benötigten Pläne abgeleitet werden. Die Pläne macht das Land, und nicht mehr die Gemeinde, kund. Derzeit werden diese Änderungen amtsintern diskutiert und es wurde eine Arbeitsgruppe eingerichtet. Bis zur notwendigen Änderung des Raumordnungsgesetzes wird es noch einige Zeit dauern. Deshalb wurden diese Änderungen noch nicht codiert. Es gibt insbesondere im Bereich der Schnittstellenanpassung noch Unsicherheiten. Hier werden nun die notwendigen Software-erweiterungen aufgezeigt.

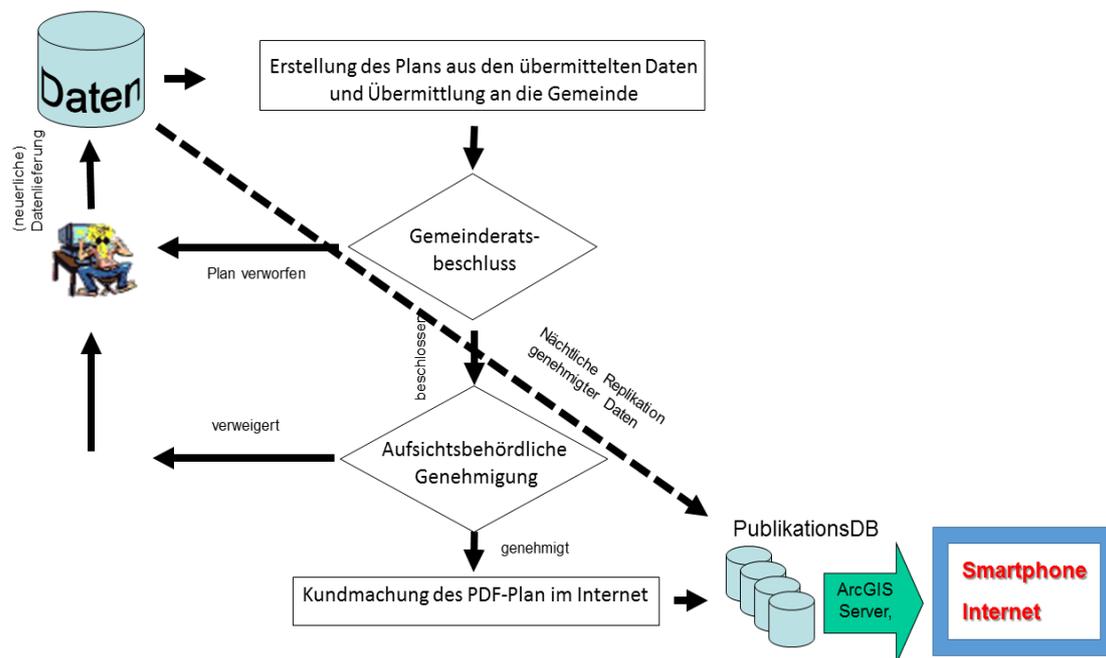


Abbildung 18: Softwareerweiterungen für rechtsrelevante Gleichstücke

Das erweiterte Verfahren setzt bei positiv vorgeprüften Plänen an, welche in die interne Datenbank des Landes eingefügt wurden. Aus den abgegebenen Daten wird ein Plan in

Form einer signierten PDF-Datei erstellt, auf welchen auch die eindeutige Nummer des Uploads aufgedruckt wird, um Verwechslungen vorzubeugen. Der Plan wird durch eine automatische E-Mail an die Gemeinde versandt, wo er dem Gemeinderat zum Beschluss vorgelegt wird. Wird der Plan verworfen, hat der Raumplaner die Einwendungen einzuarbeiten. Wird er hingegen beschlossen, wird das Land mit einer signierten E-Mail oder per Post über den erfolgten Beschluss informiert. Das Schreiben muss auch die Upload-ID enthalten. Der Plan liegt schon beim Land auf und braucht nicht übermittelt werden. Es kommt zum aufsichtsbehördlichen Genehmigungsverfahren. Wird die Genehmigung verweigert, wird die Gemeinde davon in Kenntnis gesetzt und der Prozess kann erneut beginnen. Andernfalls wird das PDF vom Land elektronisch kundgemacht. Das löst auch eine Replikation der Geodaten mit der entsprechenden Upload-ID in die Publikationsdatenbank des Landes aus. Auf diese Datenbank greifen die Internetdienste des Amtes zu und somit steht korrekte Geoinformation zur Verfügung. Einzelne Ebenen können ein- bzw. ausgeblendet werden, automatische Suchen sind möglich und natürlich ist auch der Maßstab im Web-Service frei wählbar. Das Lesen des Planes wird um vieles einfacher.

Die Extraktion des Baulandes und andere Aufbereitungen dürfen natürlich auch nur mit den beschlossenen Daten erfolgen.

## 7.4. Wesentliche Komponenten

Die im Kapitel 7.1 bis 7.3 beschriebene Software wird nun genauer beschrieben.

### 7.4.1. Hochladen

Im Rahmen der E-Government-Initiative wurde vom Amt der Stmk LREG Abteilung 1 das STERZ-Projekt initiiert (Steirisches Rechteverwaltungs- und Zutrittssystem). Es übernimmt die Aufgaben der Authentifizierung von Personen, die außerhalb der Landesverwaltung auf spezielle Applikationen des Landes zugreifen. Hier gibt es verschiedenen Sicherheitsklassen. Die Sicherheitsklasse 1, die derzeit für die Anmeldung verwendet wird, verlangt eine Authentifizierung mit Username und Kennwort bzw. Bürgerkarte. Die Sicherheitsklasse 2 ist eine Identifizierung über ein clientseitiges Zertifikat. Die Sicherheitsstufe 3 ist die höchste Sicherheitsstufe, wobei die Identifizierung durch eine Smartcard erfolgt. Sie ist für Landesbedienstete mit einem Zertifikat auf dem Dienstaussweis vorgesehen.

Die abgegebenen Pläne sind derzeit für den Rechtsvorgang nicht wesentlich. Auch kann der alte Zustand jederzeit wieder hergestellt werden, sodass eine böartige Manipulation wenig Schaden anrichtet. Jedes hochgeladene Feature wird mit der ID des Ladevorgangs versehen und Daten fehlerhafter oder manipulierter Lieferungen können gelöscht werden. Die Authentifizierungsstufe 1 scheint angebracht. Bei Bedarf kann das System jederzeit auf eine höherstehende Sicherheitsstufe umgestellt werden. Das wird dann sinnvoll sein, wenn die Geodaten im Rechtswerdungsprozess eine Rolle spielen.

Darüber hinaus muss festgehalten werden, dass Gemeinden und Planer sehr verantwortungsbewusst mit den Zugangsdaten umgehen. Zugangsdaten werden nur erteilt, wenn die Person dem Amt bekannt ist bzw. sie sich entsprechend ausweisen kann. Im Rahmen der Anmeldung gibt der Antragsteller seine E-Mail-Adresse bekannt, welche für die Kommunikation mit dem Datenüberbringer verwendet wird.

Das Interface, welches die Geodaten übernimmt, Metadaten erfasst und strukturiert auf einer Platte speichert, wurde von der Abteilung 1 nach einem entsprechenden Konzept von mir codiert. Alle anderen Komponenten, wie die Prüfung der Geodaten, die Reparatur der Geodaten mit den lernenden Komponenten sowie die regelbasierte Geodatenverwaltung, wurden von mir nicht nur konzipiert, sondern auch programmiert. Die von der Abteilung 1 entwickelte Internetapplikation wurde in Java geschrieben und läuft auf Basis von Tomcat.

#### 7.4.2. Prüfen, Reparieren Verwalten

##### 7.4.2.1. Allgemeines

Wurden Pläne am ROKAT-Portal abgegeben, starten die Prüfung, die Reparatur und die Verwaltung der Geodaten. Die Entwicklung erfolgte in der Sprache Python. Dabei handelt es sich um eine objektorientierte Skriptsprache, mit welcher die Softwaretools der Firma ESRI durch die Bibliothek `arcpy` angesprochen werden können. Die entwickelte Software benötigt die höchste Lizenzierungsstufe „advanced“. Diese teure Lizenz wird sowohl für einige Werkzeuge der Datenreparatur benötigt als auch für die Datenverwaltung. Die Prüfung der Geodaten kommt zum größten Teil mit der billigeren Lizenzierungsstufe „basic“ aus. Leider kann während des Programmlaufes nicht zwischen den Lizenzierungsstufen gewechselt werden.

#### 7.4.2.2. Aufruf, Environment-Variablen

Das Verarbeitungsprogramm wird in kurzen periodischen Abständen von einem automatischen Task gestartet. Zu Beginn kontrolliert die Software, ob offene Lieferungen vorliegen. Ist das nicht der Fall, wird die Verarbeitung terminiert.

Der aufmerksame Leser wird sich fragen, warum die Verarbeitung nicht vom Abgabeinterface gestartet wird. Python wird nicht von der allgemeinen EDV des Landes – der Abteilung 1 – unterstützt und ist nur im Referat Statistik und Geoinformation in Verwendung. Die allgemeine EDV fürchtet, eine Vielzahl an verschiedenen Produkten unterstützen zu müssen, und schottet sich notwendigerweise ab. Für die Geodatenverarbeitung auf Basis von ESRI ist Python ein unabdingbares Werkzeug. Von Standpunkt des Software-Engineering ist dieses Vorgehen sinnvoll und verständlich.

Wird die Verarbeitungssoftware ohne Parameter gestartet, poppt eine Oberfläche auf, über die Wartungsarbeiten durchgeführt werden können. Die Praxis hat gezeigt, dass neben der reinen Prüfung und Verwaltung der Geodaten noch andere Tätigkeiten notwendig sind. Um alle Schritte nachvollziehbar zu halten, ist es wichtig, diese Wartungstätigkeiten programmgestützt durchzuführen. Die Entwicklung der derzeit vorhandenen 18 verschiedenen Verwaltungstätigkeiten war sehr schnell und problemlos möglich. Hier zeigen sich die Vorteile der objektorientierten Entwicklung und der konsequente Versuch, Software flexibel zu codieren.

Ein Beispiel für diesen Verwaltungsroutinen ist ein Programm zum Löschen von Plänen aus den Datenbanken des Landes. Es ist bereits vorgekommen, dass irrtümlich falsche Planversionen hochgeladen wurden und die Daten gelöscht werden mussten. Auch muss immer wieder Auskunft geleistet werden, wann Daten hochgeladen wurden und warum sie nicht akzeptiert wurden. Die vom Computer erstellten Mails werden archiviert und können jederzeit hervorgeholt werden. Ausstehende Prüfungen und fehlende Verwaltungen – meist Kennzeichen fehlender Korrektheit der Software – können aufgelistet werden. Eine vollständige Liste der Wartungsprogramme ist im Kapitel 4.5.8 einzusehen.

Die entwickelte Software kann auch mit einem Parameter aufgerufen werden, der festlegt

1. ob lediglich geprüft und repariert wird,

2. ob lediglich positiv geprüfte Pläne verwaltet werden,
3. ob geprüft und repariert wird und anschließend bei einem positiven Prüfergebnis die Verwaltung erfolgt.

Die Datenaufbereitung (Punkt 2) benötigt knappe Ressourcen. Diese Gliederung ermöglicht es, Prüfung bzw. Reparatur (Punkt 1) von der amtsinternen Aufbereitung zu trennen. Der Planer merkt von dieser Trennung nichts, da der Plan immer erst am nächsten Tag im Internet zu sehen ist, nachdem die Produktionsdatenbank auf die Publikationsdatenbank repliziert wurde. Die Trennung von Prüfung (inkl. Reparatur) und Verwaltung (inkl. Datenextraktion) erlauben es dem Amt, die rechenintensiven Programmteile in die Nacht zu verlegen.

Darüber hinaus existieren mehr Environmentvariablen, welche prinzipielles Verhalten steuern. Die Software kann im Testbetrieb oder Echtbetrieb laufen, das automatische Mailing kann ausgeschaltet und eingeschaltet werden. Die automatische Datenkorrektur ist wegschaltbar. Die Gemeinden können generell aus dem automatischen Mailverkehr herausgenommen werden und die Bildschirmausgabe ist zu und wegschaltbar. Alle systembeschreibenden Festlegungen sind natürlich parametrisiert, wie der Container für die versandten Mails, die Speicherstruktur und vieles mehr.

#### 7.4.2.3. Objektorientiert

Um eine möglichst hohe Flexibilität und Wartbarkeit zu erreichen, wurde die Software objektorientiert entwickelt. Die einzelnen Klassen wurden penibel in einem UML-Diagramm dokumentiert. Diese grafische Darstellung der Klassen, ihrer Methoden und der Vererbung selbiger war essenziell für die Arbeit. Man verliert sich nicht in der Masse an Funktionalitäten. Zur grafischen Darstellung wurde Excel herangezogen. Das Tabellenkalkulationsprogramm ist nicht dafür gemacht und im Rückblick sollte man doch auf Programme zurückgreifen, welche die Erstellung von UML-Diagrammen besser unterstützen. Der Markt bietet Werkzeuge, welche nicht nur die grafische Darstellung erleichtern, sondern in der Lage sind, einen Rumpfcodes zu erzeugen. Code und Diagramm interagieren. Diese Entwicklungsumgebungen sparen sicher eine Menge Zeit, und Projekte dieser Größenordnung sollten mit ausgefeilten IDEs entwickelt werden. Ein Beispiel für eine solche IDE ist PyCharm, welche im Buch *Mastering PyCharm* beschrieben wird [75].

Alle Systembeschreibungen liegen in einer zentralen Datenbank. Das sind Tabellen, welche den Aufbau der abzugebenden Geodaten beschreiben, welche die Prüfungen enthalten, die Regeln der Geodatenverwaltung sowie die Komponenten des Lernens.

Das relationale Datenschema beschreibt den Aufbau beliebiger Planarten. Das System lässt sich einfach um eine neue Art von Plänen erweitern, in dem die Tabellen entsprechend gewartet werden. Derzeit können FWPs, OEPs, Auflagen (Planentwürfe) von FWPs und OEPs, kleine Änderungen der beiden Planarten sowie Beleuchtungskörper<sup>45</sup> mit dem Tool abgearbeitet werden.

Beim erstmaligen Aufbau der planbeschreibenden und verwaltenden Elemente können ähnliche Strukturen durch Duplikationen und nachträglicher Bearbeitung erzeugt werden. Dazu werden die entsprechenden Records in eine eigene Hilfstabelle kopiert, und die Änderungen erfolgen in kleinen Programmen, die im Calculator-Tool von ArcGIS-Desktop exekutiert werden können.

Die objektorientierte Entwicklung bedingt, dass die strukturbeschreibenden Elemente in Objektform benötigt werden. Diese Objekte sind aus der Datenbank zu bauen, was zeitintensiv ist. Auf dem Produktionsrechner benötigte der Aufbau der Objekte bis zu zwei Minuten, was eine untragbare Verzögerung darstellt. Natürlich kann man mit Programm- und Datenbankoptimierungen, wie attributiven Indices, Verbesserungen erzielen. Um den Aufbau der Objekte in den Bereich weniger Millisekunden zu bewerkstelligen, war jedoch ein völlig anderer Weg einzuschlagen. Python besitzt die Möglichkeit, Objekte in sogenannte Pickle-Dateien persistent abzulegen. Das englische Wort „to pickle“ steht für „einlegen“, „einpökeln“. Eingelegte Gurken oder eingepökelt Fleisch ist über lange Zeit haltbar. Genauso kann man Objekte in Dateien konservieren. Das Lesen der Pickle-Dateien geht extrem schnell.

---

<sup>45</sup> Das Projekt Beleuchtungskörper erfasst die Straßenlaternen mit den eingebauten Birnen sowie Wartungs- und Garantiefinformationen. Die Birnen sind teuer und haben lange Garanzzeiten. Primäre Aufgabe des Projektes ist es, Garantiesprüche im Schadensfall zu erkennen und Wartungsarbeiten zu erleichtern. Das Projekt wurde injiziert, weil das Land eher in der Lage ist Langzeitspeicherungen durchzuführen als kleine Gemeinden, die über keine adäquate Infrastruktur verfügen. Von Seiten des Landes besteht kein Interesse an diesen Daten, so dass das System als Gemeindeservice anzusehen ist.

Führt man eine neue Planart ein oder stehen Wartungsarbeiten an, so sind diese Tätigkeiten nicht zeitkritisch. Nach solchen Arbeiten werden alle benötigten Objekte mithilfe des Programms aus den Tabellen erzeugt. Es spielt keine Rolle, wenn dieser Prozess mehrere Minuten dauert. Die Objekte werden in Pickle-Dateien abgelegt. Sind Abgaben von Plänen zu bearbeiten, was zeitkritisch ist, erzeugt das Programm nicht mehr die benötigten Objekte aus der Datenbank, sondern liest sie von den vorbereiteten Pickle-Dateien ein und versieht sie mit den Methoden der entsprechenden Klasse. Der Vorgang benötigt keine Millisekunde auf dem vorgesehenen Rechner.

Pickle-Dateien ermöglichen es, die Vorteile relationaler Datenhaltung mit objektorientierter Entwicklung zu verknüpfen und lindern damit das „object-relational impedance mismatch“. Es ist nicht notwendig, mit objektorientierten Datenbanken oder einer objektrelationalen Datenbank zu arbeiten. Dadurch wäre zusätzliche Komplexität in das System gekommen, was wiederum zu erhöhter Fehleranfälligkeit führt. Die Korrektheit des Systems würde leiden. „Keep it simple, stupid“ ist ein Grundsatz der agilen Entwicklung.

#### 7.4.2.4. Prüfungen gegenüber der Schnittstellenverordnung

Geodaten unterscheiden strukturelle, geometrische (metrische und topologische) und thematische Eigenschaften [23, S. 26]. Dementsprechend gliedert sich auch die Prüfung in diese Aspekte.

Zuerst wird der Ordner mit den abgegebenen Daten auf die Existenz von Geodaten untersucht. Es ist durchaus im Bereich des Möglichen, dass irrtümlich Musikdateien oder Ähnliches abgegeben werden, welche die Stabilität des Systems gefährden könnten. Lediglich Geodaten des Formates Shape und definierte Informationsfiles sind zugelassen. Andere Dateien werden gelöscht.

Einige Methoden der arcpy-Bibliothek führen zu einem Absturz, sofern nicht die Geometriedatei (shp), die Sachdatendatei (dbf) und ihre Verknüpfungsdatei (shx) des Formates Shape vorhanden sind bzw. die Anzahlen von Geometrieelementen und Attributelementen nicht übereinstimmen. Zweiteres ist häufig der Fall, wenn die Attribute mit Excel nachbearbeitet werden oder die Editiersitzung nicht korrekt geschlossen wurde. Am Beginn wird das System auch vor diesen Fehlern geschützt.

Das Geometrieelement muss gewisse Regeln erfüllen. Besteht ein Polygon aus mehreren Teilflächen (Rings) dürfen sich diese z. B. nicht überlagern. Flächen müssen eine Ausdehnung besitzen und vieles mehr. Diese Regeln lassen sich mit einer einzigen Methode prüfen und gegebenenfalls werden die Geodaten repariert, was unproblematisch ist, da die vom Erfasser vorgegebene Gestalt nicht verändert wird. Auch damit wird die Stabilität aller nachfolgenden Schritte erhöht.

Die zuvor eingelesenen Objekte werden nun für die strukturelle Prüfung herangezogen. Das Vorhandensein von Pflichtebenen wird geprüft und definierte Ebenennamen, Geometrietypen sowie Spaltendefinitionen werden gecheckt. Übermittelte Ebenen müssen einen Inhalt aufweisen und das Koordinatensystem muss stimmen. Kein Element darf über die Gemeindegrenze bzw. das Bearbeitungsgebiet reichen.

Topologische Kriterien werden geprüft. Die Ebene nutzt das gesamte Gemeindegebiet eindeutig zu klassifizieren. Wurden Ebenen auf Basis von Verschnitten gewonnen, können Sliver-Polygone vorliegen. Mindestgrößen und Mindestlängen von Objekten müssen gegeben sein.

Wie in der Analyse bereits angeführt, erfolgten über 10.000 attributive Fehler in den untersuchten Spalten der Pläne. Es handelte sich um Verstöße gegenüber Domains. Die zulässigen Einträge werden in einer Tabelle vorgegeben. Dabei bestehen die Möglichkeiten,

- die Werte direkt vorzugeben,
- eine Liste von möglichen Werten festzulegen,
- ein Muster vorzugeben.

Die Vorgaben können sich auf eine oder mehrere Spalten erstrecken. Vorgaben über mehrere Spalten sind im Sinne von Folgebedingungen zu verstehen. Wenn in der Spalte A der Eintrag b vorliegt, muss in der Spalte C entweder d oder e stehen. Sanierungsgebiete werden mit dem Eintrag S in die Spalte Widmung der Ebene `beschr_f` festgelegt. Der konkrete Sanierungsbedarf wird in der Spalte ZSW kodiert wobei die Einträge (LM) für Lärm, (HW) für Hochwasser (AW) für Abwasser und (LU) für Luft zulässig sind. Als Folgebedingungen sind wiederum direkte Werte, Listen oder Muster zulässig.

Für die Mustererkennung wurden die „Regular Expression Operations“ von Python verwendet.

„Reguläre Ausdrücke sind Zeichenketten, die zur Analyse von Texten verwendet werden. Durch einen regulären Ausdruck *re* wird eine Menge von Strings definiert, die zu *re* passen.“ [64, S. 338]

Diese regulären Ausdrücke werden, gegenüber den abgegebenen Einträgen geprüft. Wird der Eintrag im Geodatenatz mit dem regulären Ausdruck als gültig anerkannt, stimmt die Vorgabe. Aufschließungsgebiete werden in der Ebene *beschr\_f* mit dem Eintrag *A* in der Spalte *Widmung* codiert und haben in der Spalte *ZSW* durch fortlaufende Nummern entsprechend der zeitlichen Reihenfolge codiert zu werden. Es sind also in der Spalte *ZSW* Eintragungen wie (1), (2) ... möglich. Alle real möglichen Werte vorzugeben, wäre mühevoll. Wesentlich einfacher lässt sich das über einen regulären Ausdruck beschreiben.

Regulärer Ausdruck:  $\backslash A[(d)]Z$

Wobei sich der Ausdruck aus folgenden syntaktischen Elementen zusammensetzt:

- $\backslash A$  ...Beginn der Zeichenkette
- $Z$  ...Ende der Zeichenkette
- $[(d)]$  ...Definition einer Menge von Zeichen, damit lassen sich auch Steuerzeichen von *Re* vorschreiben
- *d* ...ein beliebige Ziffer; das ist gleichbedeutend mit der Vorgabe  $[0-9]$

Alle zulässigen Einzelwerte, Listen und Muster werden in Steuertabellen verwaltet und daraus werden die Objekte abgeleitet, mit denen dann die Plandaten verglichen werden.

Dem Kenner der Schnittstelle ist möglicherweise aufgefallen, dass die zeitliche Reihenfolge von Aufschließungsgebieten noch weiter gegliedert werden kann. Einträge in der Form (1a), (1b) ... sind möglich, was durch ein weiteres Muster realisiert wird. Der entsprechende reguläre Ausdruck lautet:  $\backslash A[(d[a-z])]Z$

Werden FWP abgegeben, sind von den Datenlieferanten auch die Baulandbilanzen vorzugeben. Eine entsprechende Maske öffnet sich bei der Datenabgabe und ist auszufüllen. Dabei sind 13 Flächensummen, jeweils von bebauten und unbebauten Flächen auszuweisen.

Mit einigen Verschnitten und Geoanalysen wird die Baulandbilanz auch aus den abgegebenen Geodaten hergeleitet und mit den vorgegebenen Zahlen verglichen. Übersteigt die Abweichung eine definierte Schranke, welche als Prozentwert zur Gesamtgröße definiert ist, wird ein Fehler angezeigt. Diese Toleranz ist notwendig, da die im Einsatz befindlichen Programme der Planer anders runden bzw. mit unterschiedlicher Genauigkeit arbeiten.

Die Ebene `nutz_f` des FWP's hat das gesamte Gemeindegebiet lückenlos auszufüllen, wobei keine Überlagerungen unzulässig sind. Die Baulandbilanz, welche sich hauptsächlich aus der Ebenen `nutz_f` und `beschr_f` ableitet, kontrolliert in einem gewissen Maß auch die Richtigkeit von `nutz_f`. Die geforderte Topologie wird dadurch nicht hinreichend geprüft. Löcher und Überlagerungen werden erkannt, da die Gesamtfläche von `nutz_f` der Gemeindefläche gegenübergestellt wird. Dieser ressourcenschonende Check ist für alle praktisch aufgetretenen Fälle hinreichend. Theoretisch ist es möglich, dass fehlerhafte Überlagerungen genauso groß wie die Löcher sind. Dann würde die Kontrolle versagen. Alternativ zum Abgleich gegenüber der Gesamtfläche ist eine Topologieprüfung möglich. Man erspart sich durch die gewählte Vorgehensweise ein Überführen der Daten in ein Feature-Dataset, wo Topologieprüfungen möglich sind, den Aufbau der Topologie, die Regelvorgaben und die Validierung. Das alles sind rechenintensive Prozesse. Da jedoch durch die Topologieprüfung der Fehler lokalisiert werden kann, wird dieser Schritt in einem der nächsten Releases gegangen werden. Dem Planer soll dieses zusätzliche Service angeboten werden. Das Land verfügt über die notwendige Lizenzierung „advanced“ von ArcGIS. Nicht aller Planer haben diese teurere Softwarevariante bzw. Alternativprogramme, und dadurch Schwierigkeiten, die Fehler zu finden.

Im Bereich des OEPs ergibt sich die Notwendigkeit, dass zu allen Siedlungsschwerpunkten auch ein entsprechendes Gebiet vorliegt und umgekehrt. Wenn ein Punkt eingetragen wurde, hat ein räumlich überlagerndes Gebiet vorzuliegen. Diese Kontrolle wurde, wie alle Tests, allgemein formuliert, und kann auf beliebige Ebenen erweitert werden. Auch dafür reicht die günstige Lizenz „basic“. Zuerst wird die Anzahl der Flächen und Punkte überprüft. Stimmt sie überein, werden alle Flächen selektiert, welche sich mit einem Punkt überlagern. Nun müssen alle Flächen ausgewählt worden sein. Andernfalls liegt ein Lagefehler eines Punktes oder einer Fläche vor. Dieses Vorgehen setzt voraus, dass sich Flächen nicht überlagern dürfen, was hier der Fall ist.

#### 7.4.2.5. Vollständigkeit, Lagerichtigkeit

Neben den vom Gemeinderat beschlossenen Festlegungen enthalten der FWP und der OEP auch Ersichtlichmachungen. Das sind z. B. die Gefahrenzonen der Wildbach- und Lawinerverbauung, Überflutungsflächen des Landes, Geruchskreise, Naturschutzgebiete und vieles mehr. Um die Vollständigkeit und Richtigkeit dieser Ebenen zu überprüfen, wurde im Rahmen der Lehrveranstaltung Projektstudium von M. Brucher und M. Moser ein Pilot entwickelt, der auf einem Vektoransatz beruht und diese Fragestellung löst.<sup>46</sup> Die dabei gewonnenen Erkenntnisse wurden weitestgehend berücksichtigt und flossen in diese Arbeit ein [70]. Die Vektormethode, aber auch ein alternativer Rasteransatz, der sich der Fuzzy Logic bedient, wird im Kapitel 5.1 beschrieben.

Es ist anzumerken, dass die Prüfung nicht nur Fehler in den abgegebenen Plänen detektiert, sondern dass der Fehler auch in den Landesdaten vorliegen kann. Nicht immer sind die Daten des Landes aktuell und richtig.

Diese Methode kann nicht nur für Ersichtlichmachungen herangezogen werden, sondern auch zum automatischen Abgleich des sogenannten Gegenstromprinzips.

Das Gegenstromprinzip ist ein Grundsatz der Raumplanung, die auf verschiedenen Planungsebenen erfolgt. Die oberste Ebene ist das europäische Raumordnungskonzept. Darunter gibt es ein nationales österreichisches Raumordnungskonzept, einen Landesentwicklungsplan, einen regionalen Entwicklungsplan, das örtlichen Entwicklungskonzept, den Flächenwidmungsplan und den Bebauungsplan der Gemeinde. Der Einzelraum soll sich in die Ordnung des übergeordneten Gesamttraumes einfügen und bei der Planung des Gesamttraumes müssen die Gegebenheiten und Bedürfnisse des Einzelraumes berücksichtigt werden. Das bedeutet konkret, dass im Rahmen des Auflageverfahrens der regionale Entwicklungsplan gegenüber den OEP und dem FWP geprüft wird. Übergeordnete Zonen müssen untergeordnete Ausweisungen abdecken.

Die Komponente wurde nur kurzzeitig für einen einzigen Abgleich im Produktionssystem eingesetzt und hat sich als prinzipiell funktionstüchtig erwiesen. Da die Abgabe der Geodaten nach der Kundmachung des Planes in der Gemeinde erfolgt, womit er Rechtskraft

---

<sup>46</sup> Die Arbeit wurde von mir als Lehrbeauftragten betreut.

erlangt, ist diese Analyse unsinnig, da der Plan nicht mehr geändert werden kann. Nach einer entsprechenden Verfahrensänderung kann der Abgleich jedoch wertvolle Hilfestellungen bieten.

Ansätze auf Basis von Fuzzy Logic sind für diese Problematik denkbar, führen aber zu langen Rechenzeiten und wurden deshalb verworfen.

#### 7.4.3. Fehlerkorrektur

Nur fehlerfreie Lieferungen werden in den Datenbestand des Landes übernommen. Es wird jedoch versucht, Fehler automatisch zu korrigieren, um den Datenlieferanten zu unterstützen. Er erhält diese korrigierten Geodatenätze als Anhang zu einer Mail zugesandt und überprüft die Korrekturen auf Richtigkeit. Danach werden die Geodaten neuerlich übermittelt.

Eine Integration der vom Programm geänderten Datenbestände in die Register des Amts der Stmk LREG ohne vorherige Prüfung durch den Planer ist undenkbar, da der Planer für die abgegebenen Daten haftet. Diese Haftung und wesentliche Werte der Daten würden damit verloren gehen. Auch wäre es sehr problematisch, würde man von einem Computerprogramm Daten abändern lassen, welche den Wert eines Grundstücks erheblich ändern können. Bauland kann je nach Lage und Qualität Preise von 100 Euro pro Quadratmeter und darüber erreichen. Nicht bestockte Waldflächen sind schon für 20 Cent pro Quadratmeter und darunter zu erhalten.

Derzeit erfolgte noch kein Verfahren im Bereich der steirischen Flächenwidmung, wo die Geodaten eine (zentrale) Rolle spielten. Man kann jedoch davon ausgehen, dass sich das ändert.

Dieser Service<sup>47</sup> wurde eingerichtet, um die Prüfung bei schweren Fehlern nicht abbrechen zu müssen. Die reparierten Geodaten werden zur nächsten Detailprüfung weitergereicht

---

<sup>47</sup> Der Service heißt Hödlbert, nach dem Kaktus meiner jüngsten Tochter.

und der Planer erhält eine vollständige Liste der begangenen Verstöße, sofern die Reparaturen erfolgreich waren. Hat man bereinigte Geodaten vorliegen, wäre es schade, sie einfach wegzuwerfen.

Die Fehlerkorrektur kann über eine Environmentvariable ausgeschaltet werden. Das hat zur Folge, dass die korrigierten Daten nicht versandt werden. Die Korrekturen werden trotzdem durchgeführt, da in der Mehrzahl der Fehler sonst ein Abbruch der Kontrolle erfolgen müsste.

#### 7.4.3.1. Geometriereparatur, Elimination von Leerzeichen

Die Geometriespalte, im Allgemeinen mit Shape bezeichnet, muss gewissen Konventionen erfüllen. Diese Konsistenzbedingungen werden für alle Geodatsätze zu Beginn hergestellt (siehe Kapitel 4.5.4.9.). Werden die Bedingungen nicht eingehalten, können die Geanalysen mit diesen Datensätzen trotzdem durchgeführt werden. Verstöße gegen die Konsistenzbedingungen haben negative Auswirkungen auf die Laufzeit.

Auch sind führende und nachfolgende Leerzeichen in allen Textattributen nicht zulässig und werden automatisch gelöscht. Diese Korrekturen sind sehr hilfreich, da sie vom Daten-ersteller nur schwer zu erkennen sind.

#### 7.4.3.2. Sliver-Polygone

Bei Verschnitten von Ebenen können viele kleine Polygone entstehen, die meist sehr dünn und langgestreckt sind. Sie werden als Sliver bezeichnet. Beim FWP und OEP sind sie häufig unter einem Millimeter dick und 10–20 m lang. Diese Flächen im Editiermodus zu bearbeiten, ist mühevoll. Die automatische Reparatur selektiert diese Flächen und schlägt sie dem Nachbarn mit der längsten Grenze zu. Das erfolgt für alle Flächen kleiner als  $0,3 \text{ m}^2$ . Linienstücke unter 10 cm werden der Linie zugeordnet, an der sie anschließen. Stehen sie frei im Raum, werden sie gelöscht.

Diese Objekte sind durchweg unter der grafischen Auflösung des Zielmaßstabes von 1:5.000 groß. Ein Kreis von  $0,3 \text{ m}^2$  erscheint im Zielmaßstab von 1:5.000 mit einem Durchmesser von 0,1 mm. Linien unter 10 cm wären am Plan nur mehr 0,02 mm lang und

somit nicht erkennbar. Diese Grenzen stellen ein sehr vorsichtiges Auswahlkriterium dar und garantieren, dass es nicht zu ungewollten Zusammenfassungen kommt.

Für den OEP gilt Analoges.

Slivers werden entsprechend der ESRI-Knowledgebase definiert als:

„A small, narrow polygon feature, that appears along the borders of polygons following the overlay of two or more geographic datasets.“ [76]

Slivers sind also langgezogene dünne Flächen. Der Flächeninhalt ist klein und der Umfang für eine so kleine Fläche relativ groß. Daraus ergibt sich neben der Fläche ein zusätzliches Kriterium, Sliver-Polygone zu identifizieren. Diese Methode soll genauer untersucht werden.

Ein Seil wurde an den beiden Enden zusammengeknotet. Berührt sich das Seil an jedem Punkt, schließt es keine Fläche ein. Legt man das Seil zu einem Kreis aus, erhält man das flächengrößte Gebilde, für diesen Umfang. Die tatsächliche Polygonfläche (F), mit einem bekannten Umfang (U), wird in ein Verhältnis zur größtmöglichen Kreisfläche für diesen Umfang gesetzt.

Wir bestimmen den Radius des größtmöglichen Kreises aus dem gegebenen Umfang des zu untersuchenden Polygons

$$U = 2r\pi \rightarrow r = \frac{U}{2\pi}$$

und errechnen daraus die größtmögliche Fläche.

$$K = r^2\pi \rightarrow \left(\frac{U}{2\pi}\right)^2 \pi = \frac{U^2}{4\pi}$$

Danach dividieren wir die Polygonfläche durch diese Kreisfläche, um den neuen Formparameter zu erhalten.<sup>48</sup>

---

<sup>48</sup> Es handelt sich dabei um eine wohlbekannte Formel der Geoinformatik. Trotz intensiver Literaturrecherche bin ich jedoch auf keine entsprechende Literaturstelle gestoßen.

$$P = \frac{F}{K} = \frac{F}{\frac{U^2}{4\pi}} = \frac{4\pi F}{U^2}$$

Die Polygonmetrik kann Werte zwischen 1 (Kreis) und 0 (Linie) annehmen. P wird in der Literatur als Thinness-Faktor bezeichnet, und A. Tereshenkov geht davon aus, dass man Werte von 0–0,3 als Sliver identifizieren kann [77].

Wir wollen die Formel an einem Beispiel verifizieren. Der Faktor  $4\pi$  kann weggelassen werden, wenn das Ziel darin besteht, Sliver zu erkennen. Man muss lediglich einen anderen Grenzwert für die Detektion einführen ( $< 0.3/(4\pi)=0.024$ ).

$$P^* = \frac{F}{U^2}$$

Modellieren wir nun einen Sliver als Rechteck. Es wird durch seine Länge und Breite beschrieben, wobei die Breite im Fall eines Slivers sehr klein ist und als  $d$  bezeichnet wird, und die Länge durch den Ausdruck  $A*d$  festgelegt wird. Sie ist also ein  $A$ -faches von  $d$ , wobei  $A$  sehr groß sein kann. Die Fläche des Rechtecks beträgt  $F = Ad * d = Ad^2$  und die Länge des Umfangs  $U = Ad + Ad + d + d = 2d(A + 1)$ . Wir setzen diese Größen in den zuvor gefundenen Formparameter ein.

$$P^* = \frac{F}{U^2} = \frac{Ad^2}{(2d(A + 1))^2} = \frac{Ad^2}{4d^2(A + 1)^2} = \frac{A}{4(A + 1)^2}$$

Ähnlichen Rechtecken wird der gleiche Formparameter zugewiesen. Der Formfaktor hat eine Plausibilitätsprüfung bestanden.

Gehen wir vom Grenzwert 0,024 für den Faktor  $P^*$  aus, den A. Tereshenkov indirekt nennt [77], so folgt durch Einsetzen in die obige Gleichung, dass Rechtecke als Sliver angesehen werden, sobald eine Seite um das Achtfache länger als die andere Seite ist. Diese quadratische Gleichung liefert natürlich zwei Lösungen, wobei eine Lösung das Verhältnis der langen Seite zur kurzen Seite beschreibt und die andere Lösung das umgekehrte Seitenverhältnis.

Eine automatische Bereinigung ist möglich, indem die Sliver jener Fläche zugeordnet werden, mit der sie die längste Grenze teilen (ESRI Geoprocessing Function: Generate). Das sollte jedoch nur für sehr kleine Flächen erfolgen, um Fehlkorrekturen zu vermeiden. Flächen mit wenigen Quadratdezimetern können so bereinigt werden. Ein 20 m langes und 2 cm dickes Rechteck ist mit hoher Wahrscheinlichkeit auch ein Sliver. Alles, was einen kleineren Flächen/(Umfang\*Umfang)-Wert als 0,00025 aufweist und zusätzlich kleiner als 100 Quadratmeter ist, kann als Sliver eingestuft werden. In der Literatur findet man auch deutlich weniger strenge Grenzen, wie etwa bei A. Tereshenkov, der als Grenze für Sliver den Thinness Faktor<sup>49</sup> von 0,3 angibt [77].

#### 7.4.3.3. Koordinatensystem

Werden Geodatensätze ohne Koordinatensystembeschreibung abgegeben, werden diese ergänzt. Eine Lagekontrolle der Koordinatenwerte stellt sicher, dass keine falsche Zuordnung erfolgt. Geodaten mit einer falschen Koordinatenbeschreibung werden korrigiert. Auch hier ist eine Lagekontrolle der Koordinatenwerte notwendig, um die Korrektur abzusichern.

Geodaten in einem unzulässigen Koordinatensystem werden in das benötigte System umgerechnet. Das setzt voraus, dass zu dem Geodatensatz das System gespeichert wurde, in dem die Koordinatenwerte vorliegen. Diese Information wird beim Shape-Format in der Prj-Datei abgelegt. Transformationen sind nicht vorgesehen, da hauptsächlich Verwechslungen innerhalb des MGI-Referenzmodells vorkommen. Die Systeme GKM31, GKM34, BMNM31 und BMNM34 sind in der Steiermark von dieser Problematik betroffen.

Eine mögliche Erweiterung besteht darin, auch Geodaten zu behandeln, von denen nicht bekannt ist, in welchem System sie vorliegen. Das Koordinatensystem muss bestimmt werden. Dazu ist es notwendig, dass die Gemeinden in allen für die Steiermark relevanten Koordinatensystemen vorliegen. Diese Geodaten werden mit der Gemeindenummer auf die entsprechende Gemeinde hin gefiltert. Die Gemeindeumrisse liegen dann in den verschiedenen Koordinatensystemen in einem Geodatensatz vor und es muss lediglich kontrolliert

---

<sup>49</sup> Äquivalente Größe

T...Thinness

A...Area

P...Perimeter

werden, mit welcher Gemeindedarstellung der Plan zur Deckung kommt (select by location).

Eine Erweiterung der Systeme auf UTM33N, das im Amt der Stmk LREG favorisiert wird, sowie ETRS LAEA, das unter anderem von der Statistik Austria verwendet wird, wäre in diesem Fall sinnvoll. Sehr selten stößt man auch auf das Koordinatensystem MGI-M15, wobei es sich um ein projektbezogenes System handelt, welches für den Bau des Koralm-tunnels von einigen Zivilingenieuren eingeführt wurde. Das System beruht auf dem Referenzmodell MGI. Es wird die Gauß-Krüger-Projektion verwendet und der Hauptmeridian verläuft quer über die Koralm, exakt 15° östlich von Greenwich. Verzerrungen werden dadurch im Baubereich des Tunnels gering gehalten. Die Erweiterungen auf diese Systeme würden es erlauben, die Softwarekomponente auch bei anderen Fragenstellungen im Amt der Landesregierung einzusetzen.

#### 7.4.3.4. Überstände beseitigen

Insbesondere bei den Ersichtlichmachungen kommt es vor, dass Features über die Gemeindegrenze reichen. Diese Überstände werden abgeschnitten, was, wie aus mündlichen Rückmeldungen von Planern hervorgeht, besonders hilfreich ist. Der Entwicklungsaufwand für die erste Version dieser Software war verhältnismäßig gering. Features mit Ausdehnungen weit unter einem Millimeter führten zu Problemen, die nun abgefangen sind. So kann das Werkzeug auch verwendet werden, ohne vorher die Methode zur Beseitigung von Sliver-Polygonen zu exekutieren.

Überstände sind problematisch, da die Gemeindedaten zu steiermarkweiten Featureklassen zusammengefasst werden.

#### 7.4.3.5. Multipart to singlepart

Ein falscher Geometrietyp schient nicht reparierbar zu sein. Die Analyse hat gezeigt, dass bei 21 falschen Ebenen 17-mal Multipart-Strukturen anstelle von Singlepart-Geometrien geliefert wurden. Eine Überführung der Multipart-Geometrie in die Singlepart-Geometrie ist mit einfachen Mitteln möglich.

Eine mögliche Erweiterung kann darin bestehen, geschlossene Linienzüge in Flächen zu überführen. Es stellt sich jedoch die Frage danach, wo die Sachdaten zu finden sind. Es ist üblich, diese in Form eines Bezugspunktes innerhalb der Fläche zu führen. Über den Namen dieser Ebenen wissen wir jedoch nichts. Sind die Sachdaten an der Linie angebunden, könnten sie von dort mitübernommen werden.

#### 7.4.3.6. Umhüllende Erzeugen

Während der Erstellung der Dissertation haben sich verschiedene Notwendigkeiten zur Erweiterung ergeben. Eine der wesentlichsten Erweiterungen war, dass neben den FWPs und OEPs auch Änderungen dieser Daten übernommen werden. Im Prinzip haben sie die gleiche Datenstruktur wie die Stammpäne, beschreiben jedoch nicht das gesamte Gemeindegebiet, sondern nur einen kleinen Teilbereich. Diese Änderungen werden im WebMap-Service über die Stammpäne gezeichnet und können damit auch weggeschaltet werden. Die ursprüngliche Widmung ist ersichtlich. Diese Änderungen werden auch als „Briefmarken“ bezeichnet, da sie elektronisch über den Plan geklebt werden. Der Planer hat, das ist die einzige Ausnahme gegenüber Stammpänen, die Begrenzung des Änderungsgebietes vorzugeben.

Im Fall des FWPs beschreibt die Ebene `nutz_f` das gesamte Gebiet ohne Lücken und Überlagerungen. Fehlt die Umhüllende, kann sie aus der Ebene `nutz_f` abgeleitet werden (`dissolve`). Die Features der restlichen Ebenen müssen sich alle innerhalb dieser Ebene befinden.

Der OEP hat keine Ebene, aus der die Umhüllende abgeleitet werden kann.

#### 7.4.3.7. Semantische Korrekturen

Neben der Geometrie besteht ein Geodatsatz auch aus inhaltlichen Komponenten. Die Objekte werden einerseits über den Namen der Ebene und die Spaltennamen und besonders durch den Spalteninhalt beschreiben [23].

#### 7.4.3.7.1. Korrektur der Ebenenbezeichnungen sowie der Spaltennamen

Zur Korrektur der Ebenenbezeichnungen und Spaltennamen wurde eine Rechtschreibprüfung (Spellchecker) entwickelt. Wesentlich für den Spellchecker ist die Liste der gültigen Worte, welche aus den planbeschreibenden Objekten gewonnen wird. Im Gegensatz zu Rechtschreibprüfungen, wie sie in Schreibprogrammen vorkommen, ist die Liste kurz. Das System kennt viele unterschiedliche Korrekturmöglichkeiten, welche nacheinander auf den fehlerhaften Begriff angewandt werden. Nach jeder Korrektur erfolgt ein Abgleich mit der Liste gültiger Worte. War die Korrektur erfolgreich, wird die Suche abgebrochen. Korrigiert werden Buchstabenstürze, Fehler durch Groß- und Kleinschreibung, Leerzeichen am Beginn des Textes, am Ende des Textes oder mitten im Text, Sonderzeichen, verursacht durch eine andere Codepage, und Widersprüche, hervorgerufen durch `_` oder `-`.

Die Levenshtein-Distanz, ein Standardverfahren in der Rechtschreibprüfung, beschreibt, wie viele Operationen notwendig sind, um eine Zeichenkette in eine andere zu überführen. Deshalb wird sie auch Editierdistanz bezeichnet. Gleiche Wörter haben die Distanz 0. Jedes Einfügen oder Löschen erhöht die Distanz um den Wert 1, Austauschen erhöht die Distanz um den Wert 2. Die Computerlinguistik kennt effiziente Algorithmen, um diese Distanz zu ermitteln. Schuster stellt die Levenshtein-Distanz vor und erläutert sie [78]. Die Universalität der Methode ist beeindruckend und Vieles hätte dafür gesprochen, sie auch in diesem Programm anzuwenden. Im gegenständlich entwickelten Programm werden nur Änderungen durchgeführt, welche wesentlich geringere Manipulationen darstellen, wie einen Buchstaben zu ergänzen, wegzulassen oder gar auszutauschen, denn die bearbeiteten Spaltennamen und Ebenenbezeichnungen sind dafür zu kurz. Groß- und Kleinschreibung, Buchstabenstürze, Leerzeichen, `_` und `-` werden abgeändert. Auch wenn es sich z. B. bei einem Fehler in der Groß- und Kleinschreibung um das Austauschen von Buchstaben ( $A \rightarrow a$ ) handelt, ist eine Distanz von 2 groß und muss unterschiedlich gesehen werden zu einem völligen Wechsel des Buchstabens ( $A \rightarrow b$ ). Im Fall der Datenprüfung soll die automatische Korrektur nur sehr eingeschränkt arbeiten. Der Cupertino-Effekt – fehlerhafte Korrekturen bei kontextloser Rechtschreibprüfung – muss vermieden werden, und nur offensichtliche Fehler dürfen korrigiert werden [103].

Eine Erweiterung ist möglich, indem Ebenenbezeichnungen und Spaltennamen nicht singular betrachtet, sondern gemeinsam analysiert werden. Einige Spalten kommen nur bei gewissen Ebenen vor. Das schränkt die möglichen Ebenennamen ein. Für die Ebenen sind

jedoch unterschiedliche Attributwerte vorgesehen. Eine Inhaltsprüfung der Attribute wird angeschlossen. Diese liegt als ausprogrammierte Klassen vor und kann damit einfach realisiert werden. Die Ebene mit dem unbekanntem Namen wird der Inhaltsüberprüfung für alle möglichen Ebenen in einer Schleife unterzogen. Liegt Fehlerfreiheit der Attribute für nur eine Ebene vor, ist der korrekte Ebenenbezeichnung gefunden.

#### 7.4.3.7.2. *Korrektur der Attributinhalt*

Aus der Analyse geht hervor, dass bei 331 Abgaben von FWPs, 10.707 inhaltliche Fehler begangen wurden. Diese lassen sich in 20 unterschiedliche Gruppen einteilen, wobei für 16 Kategorien Programme zur Korrektur denkbar sind. Die direkte Umsetzung der Routinen stellt einen beträchtlichen Aufwand dar, will man sich nicht nur auf die wesentlichen Korrekturen beschränken. Alternativ dazu wurde ein selbstlernender Algorithmus entwickelt.

Werden Objekte mit inhaltlichen Fehlern bei der Prüfung entdeckt, schreibt die Software das fehlerhafte Feature in einen eigenen Zwischenspeicher. Dem Planer wird eine Liste der begangenen Fehler in Form einer automatischen E-Mail übermittelt. Nun wartet das System auf eine ausbessernde Lieferung. Nach dem Eintreffen des korrigierten Planes erfolgt die Prüfung erneut. Wurde eine fehlerfreie Lieferung diagnostiziert, wird das Lernen gestartet. Mithilfe der Geometrie kann das falsche Objekt dem korrigierten Objekt zugeordnet werden und man erkennt, wie das Attribut ausgebessert wurde. Neben dem falschen Wert werden der ausgebesserte Wert, der Ebenenname und die Planart in eine Tabelle persistent abgelegt. Es entsteht sukzessive ein Verzeichnis der begangenen Attributverstöße und ihrer Korrekturen. Diese Tabelle (Brain) wird für Ausbesserungsarbeiten bei zukünftigen Lieferungen herangezogen. Mit der Zeit können immer mehr Attributfehler bereinigt werden.

Dieses Verfahren zeigt jedoch auch Schwächen. Wird etwa eine Landesstraße vorgegeben, so ist in die Zusatzwidmung die Nummer des Verkehrsweges einzutragen, was häufig vergessen wird. Der vorgestellte Algorithmus lernt bei einem Fehlen der Straßenummer, jene Zahl einzutragen, welche zuerst vom Programm gelernt wurde. Das ist insofern problematisch, als Anwender dazu tendieren, die automatischen Korrekturen zu akzeptieren. Es entstehen falsche Attribuierungen. Diese Problematik tritt bei allen Nummerierungen und Aufzählungen auf.

In der nächsten Überarbeitung der Software, welche noch vor Abgabe des Dokumentes stattfindet, wird dieses Manko beseitigt. Das ist einfach möglich, indem das Lernen blockiert wird, sobald der fehlerfreie Begriff zumindest eine Aufzählung enthält.

Ein etwas allgemeinerer Ansatz besteht darin, aufzuzeichnen, wie häufig sich eine Korrektur bestätigt. Wir haben dazu die Tabelle Brain lediglich um zwei Spalten zu erweitern, in welcher gespeichert wird, wie oft diese Korrektur bestätigt wurde bzw. wie häufig andere ebenfalls gültige Werte eingetragen wurden.

Tabelle 9: Fehlerkorrektur – Realisierung von verzögertem Lernen und Vergessen

EBENE	SPALTE	FALSCH	KORREKT	VORKOMMEN	VERWORFEN
nut_f.shp	WIDMUNG	Verk	VERK	247	0
gefzo_1.shp	ZSW	R	RW	12	3

In der ersten Zeile der Tabelle Brain erkennen wir, dass die Vorgabe „Verk“ zu Großbuchstaben ausgebessert wurde, was 247-mal erfolgte. Eine Ausbesserung zu einem anderen Wert gab es nicht. Wir können mit hoher Sicherheit davon ausgehen, dass diese Verbesserung sinnvoll ist. Dabei handelt es sich um den häufigsten Fehler. Wie im Kapitel Analyse ausgeführt, wurde er über 3000-mal begangen. In der Ebenen gefzo\_1.shp werden die Gefahrenzonen der Wildbach- und Lawinenverbauung gespeichert. Mit der roten Gefahrenzone wird ein Bauverbot festgelegt und mit der gelben Gefahrenzone werden Gebiete ausgewiesen, in denen nur mit Auflagen gebaut werden darf. Mögliche Gefahrenursachen sind Wildbäche oder Lawinen, was in der Spalte ZSW mit folgenden Eintragungen beschrieben wurde<sup>50</sup>:

- RL ... Rote Gefahrenzone Lawine,
- GL ... Gelbe Gefahrenzone Lawine,
- RW ... Rote Gefahrenzone Wildbach,

<sup>50</sup> Der Auftretenswahrscheinlichkeit der Gefahr (Rot, Gelb) ist sowohl in der Spalte Widmung als auch in der Spalte ZSW kodiert.

– GW ... Gelbe Gefahrenzone Lawine.

Die Vorgabe R ist nicht zulässig und wurde 12-mal nach RW ausgebessert. Dreimal war diese Ausbesserung nicht korrekt. Die automatische Korrektur dieser Vorgabe ist riskant.

Die automatische Korrektur auf Basis der Tabelle Brain, kann erst dann zur Anwendung kommen, wenn der Eintrag in die Spalte VORKOMMEN einen Schwellwert überschritten hat. Das Lernen bedarf einiger Wiederholungen. Dieser Schwellwert sollte dramatisch erhöht werden, wenn es Ausbesserungen zu anderen Werten gab. Ein sinnvoller Schwellwert ist 5, wenn es zu keiner gegenteiligen Ausbesserung kam. Dieser geringe Grenzwert garantiert ein schnelles Lernen und ist bei einem geringen Planaufkommen zu empfehlen. Erfolgt auch alternative Ausbesserungen, verdoppelt sich der Schwellwert pro Ausbesserung. Er wächst also exponentiell an. Das gewährleistet, dass Lernen noch sinnvoll ist, wenn eine sehr geringe Anzahl an gegenteiligen Ausbesserungen vorliegt. Die Schranke für das Ausbessern wächst jedoch rapid an, falls mehrere anderslautende Ausbesserungen erfolgten. Wieweit sich die Strategie für den Schwellwert bewährt, wird die weitere Beobachtung der Schnittstelle zeigen.

#### 7.4.4. Semantische Transformationen

Eine wichtige Aufgabe im GI-Bereich ist es, Geodaten in andere Schemen zu überführen. Dabei müssen die Semantik (Bedeutung) und die Struktur des neuen Datenmodells erfüllt werden. Man spricht von einer semantischen Transformation. INSPIRE ist ein Beispiel für eine semantische Transformation. [79]. Im gegenständlichen Fall ist aus dem FWP das Bauland zu extrahieren. Das Schema Bauland besteht aus einer einzigen Ebene, welches neben der eigentlichen Widmung noch die Informationen der Sanierungsgebiete<sup>51</sup>, Aufschließungsgebiete<sup>52</sup> und der zeitlichen Folgenutzungen<sup>53</sup> führt. Diese werden durch entsprechende Erweiterungen der eigentlichen Widmung codiert. Liegt die Widmung WR (reines Wohngebiet) vor und überlagert sich diese Fläche mit einem Sanierungsgebiet aufgrund

---

<sup>51</sup> Sanierungsgebiete sind Baulandgebiete mit Mängeln. Mängel sind Lärm (LM), Hochwasser (HW), Abwasser (AW) und Luft (LU).

<sup>52</sup> Aufschließungsgebiete sind Baulandgebiete, die noch nicht vollwertiges Bauland sind, jedoch durch entsprechende Maßnahmen solches können und aus Sicht der Raumplanung auch sollen.

<sup>53</sup> Zeitliche Folgenutzungen sind zukünftige Nutzungen. Ist die zukünftige Nutzung eine Baulandkategorie, werden diese Objekte ebenfalls in die Ebenen Bauland übernommen.

von Lärm, wird dieses Gebiet im Schema Bauland als SG-WA bezeichnet. Der Sanierungsbedarf Lärm geht verloren. Die Gebiete müssen nicht deckungsgleich sein, weshalb ein Verschnitt notwendig ist. Ähnliche modellhafte Transformationen wurden auch für die Aufschließungsgebiete und zeitlichen Folgenutzen realisiert. Auffüllungsgebiete<sup>54</sup> sind nicht Teil des Baulandes.

Im Rahmen der Diplomarbeit „Historisierung von Geodaten“ wurden die einzelnen gemeindeweise vorliegenden Baulanddaten der verschiedenen Perioden zusammengeführt und in einem speziellen Schema abgelegt, sodass mit einem Schieberegler der Baulandstand zu einem gewissen Zeitpunkt visualisiert werden kann [80]. Die aus Flächenwidmungsplänen extrahierten Baulanddaten werden in dieses System eingebracht.

Abhängig von der Planart werden zusätzliche Informationen an jedes Feature angefügt. Das ist insbesondere eine eindeutige Nummer des Uploads. Aber auch die Gemeindenummer sowie die Verfahrensfallnummer werden ergänzt, um die Daten einfach bis auf das Objekt hin verwalten zu können. Um eine schnelle Beschriftung im Web-Service des Landes zu garantieren, wird ein eigenes Beschriftungsfeld angefügt, wo der Beschriftungstext aus den einzelnen Spalten generiert wurde, damit er performant geplottet werden kann.

In der Schnittstelle werden die in Fachkreisen etablierten Kodierungen verwendet. Die Abkürzung WA steht etwa für „Allgemeines Wohngebiet“ und KG+WA bedeutet, dass eine geschossweise Überlagerung eines Kerngebietes mit einem Allgemeinen Wohngebiet vorliegt. Die Abkürzungen sind notwendig, denn nur so können alle Flächen ohne Überlagerung beschriftet werden. Weite Kreise der Bevölkerung kennen diese Abkürzungen nicht. Das Webservice bietet einen Informationsbutton, der diesen Kurzformen erläutert. Diese Information wurde in einer getrennten Tabelle gehalten. Da die zutreffende Erläuterung nur mit einer Verknüpfung über mehrere Spalten hinweg gefunden werden kann, kommt es beim Web-Service manchmal zu „Timeouts“. Die Verknüpfung war zu zeitaufwendig und wurde abgebrochen. Deshalb wird nun auch diese Information in einer Spalte angefügt. Performantes Verhalten wird durch Redundanz erkaufte.

---

<sup>54</sup> Auffüllungsgebiet sind Sondernutzungen im Freiland. Sie füllen Lücken in einer zeilenförmigen oder weilerförmige rechtmäßige Bebauung auf und dürfen nicht größer als 3000 m<sup>2</sup> sein.

Diese Informationsspalte wäre nicht weiter erwähnenswert und stellt nur eine von vielen semantischen Transformationen dar, wenn sie nicht den Weg zu wesentlichen Erweiterungen aufzeigen würde. Wir wollen das Kürzel DO, welches für Dorfgebiet steht, genauer betrachten. Gehen wir in die Frühzeit der Raumplanung zurück, finden wir folgende Definition (Steirisches Raumordnungsgesetz 1974 §23 (5f) LGBl. Nr. 127/1974):

Dorfgebiete, das sind Flächen, die vornehmlich für Bauten land- und forstwirtschaftlicher Betriebe in verdichteter Anordnung bestimmt sind, wobei auch Wohngebäude und Gebäude, die den wirtschaftlichen, sozialen, religiösen und kulturellen Bedürfnissen der Bewohner dienen, errichtet werden können;

Entsprechend dem Steirischem Raumordnungsgesetz 2010 §30, LGBl. Nr. 49/2010 gilt nun:

„Dorfgebiete, das sind Flächen, die für Bauten land- und forstwirtschaftlicher Nutzung in verdichteter Anordnung bestimmt sind, wobei auch Wohnbauten und sonstige Nutzungen zulässig sind, die den wirtschaftlichen, sozialen, religiösen und kulturellen Bedürfnissen der Bewohner von Dorfgebieten dienen und sich der Eigenart des Dorfgebietes entsprechend einordnen lassen, soweit sie keine diesem Gebietscharakter widersprechenden Belästigungen der Bewohnerschaft verursachen;“

Die Erweiterung „Eigenart des Dorfgebietes“ schränkt die Bautätigkeit in diesen Gebieten in Richtung traditioneller Gebäude ein und das Verbot der „Belästigungen der Bewohnerschaft“ gibt den Wünschen der Mehrheit der Einwohner gegen dem Individuum mehr Gewicht.

Dorfgebiet hat einen Bedeutungswandel durchgemacht. Rechtlich gilt in diesem Fall die Versteinerungstheorie. Darunter wird verstanden:

„Nach der herrschenden Rechtsprechung des Verwaltungsgerichtshofes sind die Festlegungen in einem Planwerk grundsätzlich nach jener Rechtslage auszulegen, die zum Zeitpunkt der Erlassung z. B. eines Flächenwidmungsplanes gegolten haben.“ [81]

Dem Nutzer die umgangssprachliche Bedeutung Dorfgebiet bekannt zu geben, ist eine für viele Anwender befriedigende Lösung, die jedoch für Spezialisten wie Rechtsanwälte, Richter und Raumplaner zu kurz greift. Sie benötigen die genaue rechtliche Definition. Die derzeitige Lösung listet mit dem Informationsknopf die Verfahrensfallnummer auf, mit deren Hilfe das Rechtskraftdatum und damit die zutreffende Rechtslage erhoben werden können. Ein Aufruf des Rechtsinformationssystems (RIS) liefert die exakte Definition. Dieser Vorgang ist aufwendig und könnte durch den Computer erledigt werden, indem durch einen Klick auf ein Grundstück ein vollständiger und korrekter Report erstellt würde, der die Rechtssituation beschreibt. Die EDV-Anwendung „Terminevidenz“, welche in der Abteilung 13 – Umwelt und Raumordnung geführt wird, kann mit der Gemeindenummer und der Verfahrensfallnummer das Rechtskraftdatum liefern. Anstelle der Kurzbezeichnung könnte nun die korrekte rechtliche Information wiedergegeben werden. Natürlich muss intern ein Verschnitt mit allen anderen Ebenen des Planwerks erfolgen, die weitere wesentliche Informationen enthalten. Das wäre für den FWP z. B. die Ebene beschr\_f, welche Sauerungsgebiete enthalten. Das sind

„Gebiete, in denen Maßnahmen zur Beseitigung städtebaulicher oder hygienischer Mängel sowie zur Vermeidung der Gefährdung der Sicherheit oder gesundheitsschädlicher Folgen (§ 22 Abs. 7 Z. 3 und 4) erforderlich sind...“ (Steiermärkisches ROG 1974 § 123 Abs. 4).

Für jedes Grundstück kann nun ein auf den Gesetzen und Verordnungen beruhender Text erstellt werden, welcher die Planinhalte beschreibt.

Schon bei der Kurzinformation kam es zu unangenehmen Laufzeitproblemen. Ein solcher Reportgenerator würde dieses Problem noch wesentlich verschärfen. Es ist daher notwendig, einen Button im Web-Service zu gestalten, der im Sinne einer Anfrage agiert. Klickt man mit dem „Advanced-Info-Button“ auf ein Grundstück, wird die E-Mail-Adresse abgefragt. Der User erhält dann eine E-Mail mit der vollständigen Beschreibung der Widmung mit ihren Beschränkungen und Ersichtlichmachungen in Textform und einem Ausdruck des Grundstücks. Eine Wartezeit von einer Minute wird von den Nutzern bei diesem Vorgehen sicher akzeptiert und die Lösung ist mit diesen Zeitvorgaben und der verfügbaren Hardware machbar. Basis wäre eine Tabelle, welche die Planart (FWP, OEP), die Ebene, die Spalte und die Attributausprägung, die Textbausteine, die Reihenfolge im Bericht und die Gültigkeitsdauer enthält. Das vom Nutzer ausgewählte Grundstück wird mit allen Planebenen

verschnitten. Kommt es für eine Ebene zu einem Treffer und liegt dort die entsprechende Attributausprägung einer Spalte vor, wird der Textbaustein des entsprechenden Zeitfensters gesetzt. Jeder Textbaustein wird entsprechend seiner Reihenfolge eingefügt. Die Funktionalität der Reihenfolge liegt in der Klasse Mail, welche für das automatische Mailing konzipiert wurde, bereits vor.

Wir können derzeit nicht mit Sicherheit sagen, ob ein Grundstück tatsächlich so gewidmet ist, wie es die digitalen Daten zeigen. Dieser Umstand wurde im Kapitel „Die Rechtsrelevanz des digitalen FWPs“ abgehandelt. Verweise auf die Gesetzesstellen, wie sie im Reportsystem vorgesehen sind, implizieren beim Nutzer die Vermutung, dass die rechtliche Situation gezeigt wird, was jedoch nicht der Fall ist. Ein solches System würde erst Sinn machen, wenn durch Änderung im Verfahrensablauf sichergestellt wird, dass korrekte Daten vorliegen.

#### 7.4.5. Mailing

Am Beginn jedes Programmlaufes wird das Berichtssystem initialisiert. Während des Programmablaufes werden einzelne Textblöcke gesammelt, wobei jeder Block vom Programm mit einer Positionsnummer versehen wird. Je niedriger diese Zahl ist, desto weiter vorne wird der Textblock bei der Generierung des Gesamttextes eingefügt. So hat der Textblock, welcher das Echo der Eingabe und die Upload-ID enthält, die Nummer 0 und wird dadurch an den Anfang des Reports gestellt. Die Gesamtbeurteilung wird gleich nach dem Eingabe-echo aufgrund der Positionsnummer 1 aufgelistet. Der Leser des Protokolls, muss nicht alle Details durchlesen, um die für ihn wichtige Information zu erhalten. Textblöcke mit gleicher Positionsnummer werden entsprechend der Reihenfolge des Auftretens geordnet. Die einzelnen Fehlermeldungen, die idente Platznummern haben, werden dementsprechend in der Prüfreihefolge gelistet.

Jeder Report wird als Datei in einem für diese Planart vorgesehenen Ordner archiviert, wobei die Upload-ID ergänzt durch eine laufende Nummer als Name verwendet wird. Für jede Abgabe sind somit mehrere Protokolle möglich. Das ist nur notwendig, wenn aufgrund eines Programmfehlers ein Absturz erfolgen sollte und die Software noch einmal die Daten kontrollieren muss.

Der Report begründet jede Beanstandung und listet die Ebene sowie die Objekt-ID auf, sodass der Planer den Fehler leicht finden kann. Wurden vom Programmteil Hödlbert Reparaturen an Geodaten durchgeführt, werden diese Daten gezippt und dem Planer in einer eigenen E-Mail zugestellt. Der Planer verifiziert die Korrekturen. Entsprechen sie den Wünschen des Planers, kann er die Daten neu liefern. Automatische Korrekturen werden niemals direkt in den Geodatenbestand des Landes übernommen. Nur fehlerfreie Lieferungen des Planers werden dafür akzeptiert.

Daten können entweder im Prüfmodus oder im Abgabemodus, der auch die Prüfung inkludiert, hochgeladen werden. Ersteres hat zur Folge, dass nur der Datenlieferant ein Prüfprotokoll erhält. Zweiteres bewirkt, dass nicht nur der Planer eine Mail erhält, sondern auch die Gemeinde, falls eine fehlerfreie Lieferung erfolgte. Der Auftraggeber der Planung hat damit eine Bestätigung, dass der Vertragspunkt „Datenlieferung an das Land“ erfüllt wurde. Die Daten werden automatisch in die Datenbanken des Landes integriert.

#### 7.4.6. Paralleles Aufbereiten der Geodaten

Die Software kontrolliert alle fünf Minuten, ob neue Pläne abgegeben wurden. Liegen neue Daten vor, wird die Verarbeitung gestartet. Es ist unwahrscheinlich, aber zumindest einmal bereits eingetroffen, dass eine Verarbeitung noch nicht abgeschlossen war, als ein neuer Programmablauf gestartet wurde. Die Software muss also in der Lage sein, die Planverarbeitung parallel durchzuführen. Jede Abgabe wird in einem Ordner abgelegt, der pro Planart, Gemeindenummer und Verfahrensfall eindeutig ist. Wurde diese Abgabe noch nicht aufbereitet, werden weitere Lieferungen nicht akzeptiert. Jeder Lieferung steht also ein eigener Arbeitsordner zur Verfügung, was parallele Abarbeitung ermöglicht.

Dieser Arbeitsordner wird nicht von jedem Programmteil genutzt. Das Zippen von reparierten Geodaten erfolgt z. B. in einem Container, der nach erfolgter Arbeit wieder gelöscht wird. Um hier Parallelität zu erreichen, werden die Container durchnummeriert. Besteht zur Laufzeit ein temporärer Container, wird ein weiterer Container angelegt. Die Container werden durchnummeriert.

#### 7.4.7. Warnungen

Das System unterscheidet zwischen Fehlern und Warnungen. Jeder Fehler bedingt, dass die Daten nicht vom Land übernommen werden. Warnungen werden im Report aufgelistet, führen jedoch nicht dazu, dass die Lieferung abgelehnt wird. Typische Beispiele für Warnungen sind zu lange Textfelder. Es ist für die Übernahme der Geodaten irrelevant, wie groß die Textfelder definiert sind, sofern der Inhalt der Textfelder korrekt ist, was ebenfalls geprüft wird. Wichtig ist also eine korrekte Attribuierung. Die fehlerhafte Felddefinition ist nebensächlich und stellt lediglich eine Warnung dar. Das Protokollieren von Warnungen lässt sich mit einer Environmentvariablen ausschalten, was derzeit im Echtbetrieb der Fall ist. Der Planer soll nicht durch unnötige Informationen belastet werden.

#### 7.4.8. Geodaten verwalten

Das Verwalten der Geodaten ist wesentlich rechenintensiver als die Prüfung und Reparatur, weshalb die Verwaltung auch separat erfolgen kann. Das wird durch entsprechende Übergabeparameter beim Aufruf gesteuert, der periodisch durch einen geplanten Task erfolgt. Derzeit ist eine Trennung der Geodatenprüfung/Reparatur und der Verwaltung nicht geschaltet, da der Rechner imstande ist, beide Aufgaben gleichzeitig zu bewältigen.

Zur persistenten Speicherung der Geodaten steht einerseits eine Datenbank für die aktuellen Informationen zur Verfügung und andererseits eine Datenbank mit den historischen Geodaten. Zur Führung der aktuellen Geodaten ist es notwendig, die obsolete Information zuerst zu löschen und danach die neue Information einzufügen. Die historischen Geodaten entstehen, indem auf das Löschen obsoleter Information verzichtet wird und lediglich die neuen Features über die alten Features gelegt werden. Dadurch entstehen räumlich übereinander liegende Objekte. Jedes Objekt wird, wie im Kapitel 7.4.4 erwähnt, mit der Gemeindenummer und der Verfahrensfallnummer versehen. Die Kombination aus Gemeindenummer und Verfahrensfallnummer legt das Rechtskraftdatum des Planes fest. Jeder Plan ist solange gültig, bis ein neuer Plan für eine Gemeinde in Rechtskraft tritt. Das Projekt Terminevidenz speichert anfallende Termine im Bereich der Raumplanung. Es existiert also eine Tabelle, welche die Gemeindenummer, die Verfahrensfallnummer und das Rechtskraftdatum unter vielen anderen Informationen enthält. Um animierte Karten über die Zeit zu erstellen, ist jedoch die Gültigkeitszeit in Form zweier Datumfelder *von ... bis* für jedes

Objekt des FWPs notwendig. Wir kennen aus der Terminevidenz das Von-Datum, welches dem Rechtskraftdatum entspricht. Das Bis-Datum ergibt sich als Rechtskraftdatum des nächsthöheren Planes.

Tabelle 10: Aufbau der Gültigkeit

Gemnr	Vfnr	Von (Rechtskraft)	Bis
60652	1.00	2000-01-26	2005-01-23
60652	2.00	2005-01-23	2010-01-27
60652	3.00	2010-01-27	2016-01-26
60652	4.00	2016-01-26	2016-01-26

Zur Generierung der Spalte BIS existiert eine Standardfunktion in arcpy (CalculateEnd-Time\_management).

Der Plan wird im Intranet publiziert und vom Raumplaner oder von Gemeindebediensteten optisch überprüft. Dabei fallen Fehler auf. Diese werden korrigiert und eine neue Planversion wird hochgeladen. Falsche Versionen werden nicht aus der Historie gelöscht, um den Werdegang nachvollziehen zu können. Bei einer zeitanimierten Karte dürfen die fehlerhaften Versionen nicht berücksichtigt werden. Jeder Upload wird mit einer eindeutigen Nummer versehen, wobei spätere Uploads höhere Nummern erhalten. Diese Nummer wird ebenfalls zu jedem Objekt gespeichert. Für die Zeitanimation sind also nur die Uploads mit der höchsten Upload-ID pro Gemeindenummer und Verfahrensfall zu verwenden. Zur Ermittlung einer entsprechenden Hilfstabelle aller relevanten Uploads existiert auch eine Standardfunktion von arcpy (Statistics\_analysis), mit der die Ebenen dann gefiltert werden.

Der kritische Leser wird einwerfen, dass es nicht notwendig ist, sowohl aktuelle Geodaten als auch historische Geodaten vorzuhalten. Die aktuellen Geodaten ergeben sich aus allen Objekten, wo das Von-Datum existiert und noch kein Bis-Datum vorliegt. Der Einwand ist richtig. Viele Anwender im Land sind es jedoch gewohnt, auf Ebenen zuzugreifen und dort nur die (möglichst) aktuelle Situation vorzufinden. Davor eine SQL-Abfrage abzusetzen, wäre ein aufwendiger und lästiger Schritt, der den Mitarbeitern des GIS-Steiermark viel Kritik einbringen würde. Darüber hinaus sind die historischen Daten aufgrund der Menge

an Objekten langsamer. Da die historischen Informationen nur selten gebraucht werden, liegen sie auf einen verhältnismäßig schwachen Rechner.

In der Raumplanung wurde zur Erstellung digitaler Pläne früher gerne AutoCAD verwendet. In CAD-Systemen wird eher mit vielen Layern gearbeitet, die dann einheitlich dargestellt werden. Das Rendern in Abhängigkeit der Attributinformation gehört zu den Standardaufgaben in GI-Systemen und ist dort Usus. In den historisch gewachsenen Schnittstellen des FWP's und des OEP's ist der Einfluss der AutoCAD-Welt nicht zu übersehen. Viele unterschiedliche Ebenen liegen vor. Die Geodatenverwaltung der 40 Ebenen des FWP's und der 8 Grundebenen sowie der 39 Aufbauebene des OEP's sind unübersichtlich, weshalb sie landesintern zusammengefasst werden. Dabei wird zu jedem Objekt die ursprüngliche Ebene gespeichert, aus der es stammt. Der FWP besteht nach diesem Vorgang aus lediglich vier integralen Ebenen. Die Ebene `nutz_f` ist die wichtigste Ebene und beschreibt die Widmungen des gesamten Gemeindegebiets. Diese Nutzungen werden durch überlagernde Flächen, Punkte und Linien beschreiben, welche die Nutzungen beschränken oder andere relevante Sachverhalte, wie Gefahrenzonen zeigen.<sup>55</sup> Zwei zusätzliche Ebenen sind für die Gebäude und Grundstücksgrenzen vorgesehen. Analoges gilt für OEP's, die ebenfalls in vier Ebenen überführt werden.

#### 7.4.9. Implementierung der Gemeindestruktureform (GSR)

Am 16. Dezember 2010 fiel der Startschuss für eine GSR der Steiermark, wobei aus den bestehenden 542 Gemeinden 287 Kommunen geformt wurden [82]. Ziel der Initiative ist es, „wirtschaftliche und leistungsfähige Gemeinden zu schaffen, die dauerhaft in der Lage sind, ihre Angelegenheiten ohne Haushaltsabgang zu erfüllen“ (siehe Gesetz über die Neuregelung der Gemeinden im Land Steiermark §1 Abs. (1) LGBl. Nr. 31/2014).

159 Gemeinden waren von der GSR nicht betroffen. Die verbleibenden Gemeinden wurden hauptsächlich als Ganzes zusammengefasst. Einige wenige Gemeinden wurden zuvor geteilt und die Teilstücke unterschiedlichen Kommunen zugeordnet. Die neuen Teilungslinien

---

<sup>55</sup> FWP\_ueber\_f...überlagernde Flächen, FWP\_ueber\_l ... überlagernden Linien, FWP\_ueber\_p ...überlagernde Punkte

laufen entlang von Katastralgemeindegrenzen<sup>56</sup> oder entlang sonstiger Grundstücksgrenzen<sup>57</sup>. Das Amt der Stmk LREG hat eine übersichtliche Tabelle veröffentlicht [83], anhand der die Änderungen ersichtlich sind.

Die Verwaltung der Pläne erfolgt entsprechend ihrer Lieferung gemeindeweise. Kommt ein neuer Plan, werden alle Objekte mit dieser Gemeindenummer aus den steiermarkweiten Ebenen gelöscht und durch die neuen Objekte ersetzt. Durch die GSR scheint diese Vorgehensweise obsolet geworden zu sein, da eine neue Gemeinde mehrere alte Gemeinden ersetzt. Verschärft wird die Problematik durch Teilungen von Gemeinden. Die Gemeindenummern können scheinbar nicht mehr zum Löschen herangezogen werden.

Der klassische geografische Ansatz ist es, mit der neuen Gemeinde ein Loch in den alten Datenbestand zu schneiden und dieses Loch mit den neuen Daten auszufüllen (update bzw. erase, append). In den Randbereichen entstehen bei dieser Vorgehensweise Sliver, die mittels Toleranzen beim Verschnitt verringert werden können.<sup>58</sup>

Der gravierende Nachteil des Verfahrens liegt in dem Faktum, dass im Fall von Sliver auch Widmungsflächen der Nachbargemeinde geändert werden. Kleine Teile werden aus den Nachbargemeinden herausgeschnitten. Die Toleranzen verziehen die Vertices.

Die von der Gemeindestruktureform betroffenen Gemeinden wurden rechtlich aufgelöst und hören damit auf zu existieren. Aus ihnen wurden neue Gemeinden mit neuen Gemeindenummern gebildet. Bis zur Wahl eines neuen Gemeinderates wurden Regierungskom-

---

<sup>56</sup> Die alte Gemeinde Limbach bei Neudau wurde so geteilt

<sup>57</sup> Entsprechende Teilungen wurden in den Katastralgemeinden Kohlberg, Oberstorcha, Schlag bei Thalberg, Stocking durchgeführt.

<sup>58</sup> Bei der wichtigsten Ebene `nutz_f` des FWPs, welche die Landschaft ohne Überlagerungen und Löcher beschreiben muss, kann man das Verfahren noch verfeinern, indem zuerst das Loch für die Gemeinde ausgeschnitten wird, die entstehenden Sliver im Randbereich der Gemeinde selektiert werden (siehe auch 7.4.3.2), die man dann automatisiert in der benachbarten Fläche aufgehen lässt (eliminate). Da die neue Gemeinde noch nicht eingesetzt wurde, ist garantiert, dass die Sliver in den alten Widmungen aufgehen. Bei den Slivern handelt es sich um Reste der zu löschenden Gemeinde. Das Verfahren funktioniert nur für flächendeckende Ebenen, da dort immer ein Nachbar für die Funktion Eliminate vorhanden ist.

missäre eingesetzt. Sie erledigen die Geschäfte der Gemeinde. Häufig ist das einer der betroffenen Bürgermeister. In allen Fällen erklärten sie die alten FWP's und OEP's für das neue Gebiet als gültig.

Diese rechtliche Vorgehensweise kann auch digital nachgebildet werden und ermöglicht das Update der Planunterlagen in steiermarkweiten Registern, ohne Widmungsinformationen abzuändern. Als erster Schritt werden die Pläne digital fortgeschrieben, indem zu jedem Objekt nicht nur die alte Gemeindenummer, sondern auch die neue Nummer gespeichert wird. Das ist ohne viel Mühewaltung möglich, da hauptsächlich ganze Gemeinden zusammengelegt wurden. Eine Zuordnungstabelle existiert, wobei mit der Join-Operation die neue Gemeindenummer zugeordnet wurde. Es bleiben lediglich 5 FWP's bzw. OEP's, wo einzelne Objekte in verschiedenen Gemeinden zu liegen kommen, weshalb die Features zu spalten sind.

Kommt nun ein neuer Plan von einer GSR-Gemeinde, können die zu löschenden Objekte leicht gefunden werden, da jedes Objekt ihre alte und neue Gemeindenummer kennt. Sind die alten Daten gelöscht, brauchen die neuen Stammdaten lediglich eingefügt werden.

Wie konkret zu löschen ist, wird durch Regeln beschrieben, welche in einer Tabelle (loereg) abgelegt sind. Auch für das Speichern der Geodaten gibt es eine Regeltabelle. Jede Regel besteht aus dem Quell- und Zieldatensatz sowie aus zusätzlichen Kennungen, welche mit der Lieferung abgeglichen werden. So kann herausgefunden werden, ob für diese Lieferung die Regel anzuwenden ist. Das sind konkret die Planart, ein Kürzel, welches angibt, ob es sich um einen Stammpplan oder eine Änderung handelt, und ein weiteres Kürzel, das besagt, ob es sich um einen Planentwurf (Auflage) oder einen endgültigen Plan handelt. Darüber hinaus beschreibt die Regel die Löschmethode. Gelöscht wird entweder über das beschriebene Verfahren oder mithilfe eines Geodatensatzes, welcher das zu löschende Gebiet festlegt. Die zweite Methode kommt praktisch nicht zum Einsatz, wurde aber codiert, um die Software für verschiedene andere Anwendungsfälle gebrauchen zu können.

Im Wartungsfall ist neben dem Löschen mit der Gemeindenummer auch das Löschen mit der eindeutigen Kennung der Datenlieferung notwendig und realisiert.

In einigen Fällen werden vor der Abgabe der neuen GSR-Gemeindedaten noch die FWPs der alten Gemeinden upgedatet. Da jedes Objekt ihre alte Nummer kennt, kann auch dabei mit dem gleichen Verfahren gearbeitet werden.

Dieses Verfahren funktioniert auch dann, wenn sich die gemeindeweisen Pläne am Rande des Planungsgebietes (geringfügig) überlagern. Inselpläne können mit Definition-Querys erstellt werden.

Die Praxis hat gezeigt, dass alte Pläne mit der neuen Gemeindenummer hochgeladen werden. Im Fall des FWPs ist das unproblematisch. Die neue Gemeindenummer beschreibt ein wesentlich größeres Gebiet und die Kontrolle auf Flächenfüllung deckt den Fehler auf. Bei OEPs ist diese Kontrolle wirkungslos, da es keine flächenfüllende Ebene gibt. Deshalb wird die neue Gemeinde über einen Verschnitt in die alten Gemeindeteile zerlegt und jedes Teilstück muss zumindest ein Objekt des OEPs enthalten. Da auch die Teilstücke große Gebiete umfassen, kann davon ausgegangen werden, dass jedes Teilstück ein Planobjekt enthalten muss.

#### 7.4.10. Änderungen von Teilbereichen des Plans

Die Software verarbeitet neben Plänen, welche das gesamte Gemeindegebiet abdecken, auch Änderungen von Teilbereichen einer Gemeinde. Diese Teilbereiche werden in beinahe gleicher Struktur geliefert. Lediglich das Änderungsgebiet ist als zusätzliche Ebene abzugeben. Im Fall des FWPs wird dieser Änderungsbereich, im Rahmen der Datenreparatur, aus der flächenfüllenden Ebene `nutz_f` abgeleitet, falls er nicht abgegeben wird. Beim OEP besteht diese Möglichkeit nicht, da keine flächenfüllende Ebene existiert.

Änderungen sind sehr häufig und müssen laut Schnittstelle derzeit noch nicht als Geodatenätze abgegeben werden.<sup>59</sup> Das hat zur Folge, dass viele Änderungen lediglich in Form von PDF-Dateien vorliegen. Für einen Mitarbeiter des Landes wurde eine Software entwickelt, in der diese Gebiete erfasst werden können und mit dem Rechtsdokument elektronisch verknüpft werden.

---

<sup>59</sup> Eine neue Verordnung ist mit 1. Oktober 2016 in Rechtskraft getreten, wo nicht nur die Stammpläne, sondern auch die Geodaten der Änderungen zu liefern sind.

Wird der Änderungsplan in Form von digitalen Geodaten­sätzen hochgeladen, was derzeit noch nicht verpflichtend ist, erfolgen eine zu den Stammp­länen analoge Prüfung und die Korrektur. Diese Änderungsdaten werden in eigenen Ebenen abgelegt. Diese Daten werden im WebGIS elektronisch über den Stammp­lan gezeichnet, sodass im Teilbereich der Änderung die aktuelle Situation den alten Plan abdeckt. Durch Wegschalten der Änderungsebenen kann der Stammp­lan visualisiert werden.

Seit 1974, als mit der Flächenwidmungsplanung in der Steiermark begonnen wurde, kam es zu 17.000 Änderungsverfahren, was ca. 30 Änderungen pro Stammp­lan entspricht [84]. Durch die Planzeichenverordnung 2016 LGBL. Nr. 117/2016 ist ab 1. Oktober die Abgabe von Änderungen in Form von Geodaten­sätzen verpflichtend, weshalb mit einem hohen Datenaufkommen zu rechnen ist.

Wird ein neuer Stammp­lan geliefert, werden alle nun hinfällig gewordenen Änderungen gelöscht. Auch hier kann das Löschen über die Gemein­denummern erfolgen, wenn jedes Objekt die Gemein­denummer nach und vor der Strukturreform kennt. Es wird so vorgegangen, wie es im vorherigen Kapitel 7.4.9 beschrieben wurde.

Das Bauland benötigt einen eigenen Updatemechanismus. Mit dem digital vorliegenden Änderungsgebiet wird ein Loch in die Baulandebene gestanzt. Das erfolgt mit einer Toleranz von fünf Zentimetern. Das Bauland wird, wie im Kapitel 7.4.4 beschrieben, aus den Änderungsdaten extrahiert und in das Loch eingefügt. Sliver treten naturgemäß auf und werden, wie in Fn. 55 beschrieben, gelöscht. Damit kommt es zu geringfügigen Änderungen der Widmung, was jedoch nicht problematisch ist, da diese Ebene für die interne Verwendung zu Analyse­zwecken gedacht ist. Rechtsinformationen werden daraus nicht abgeleitet.

## 7.5. Abgrenzung zu anderen Softwareprodukten

Am Beginn jeder Geoanalyse wird geprüft, ob die Datenqualität für die Fragestellung ausreicht. Meist erfolgt die Prüfung manuell. Es existieren am Markt Produkte, welche diese Arbeiten (teilweise) übernehmen. Das ist etwa [checkgeodata.net](http://checkgeodata.net) der Firma Axmann geoinformation, der auf FME beruht. Die Dateio­rganisation, das Datenformat, das Koordinatensystem, der Koordinatenbereich, die Attributprüfungen, räumliche Zusammenhänge und

vieles mehr werden geprüft, wobei die Firma auch den speziellen Regelaufbau für beliebige Schnittstellen anbietet [85].

Die Firma ESRI vertreibt die kostenpflichtige Erweiterung ArcGIS Data Reviewer, welche räumliche Prüfungen<sup>60</sup>, attributive Prüfungen, die Integrität von Features<sup>61</sup> und Metadatenprüfungen den Layern abverlangt [86].

Der Service, Normen wie die Planzeichenverordnung in EDV-Regeln zu überführen, was die Firma Axmann geoinformation anbietet, erspart viele Arbeitsstunden. Auch wird eine Internetplattform angeboten, welche die Geodaten übernimmt. Auf FME basierend unterstützt dieses System nahezu alle gängigen Datenformate. Die Länder Salzburg, Burgenland und Tirol prüfen so ihre FWPs über das Portal und das Land Steiermark bedient sich im Bereich der Bestandsvermessungen und bei Verkehrsprojekten an dem großartigen Service [87]. Der praktische Einsatz bei Ländern zeigt, dass das Produkt durchaus in der Lage ist, die Aufgaben dieser im Rahmen der Dissertation entwickelten Software zu ersetzen.

Der Data Reviewer würde hervorragend in die Softwarelandschaft des Landes Steiermark passen, da die Geoinformationsverarbeitung hauptsächlich auf Basis von ESRI-Produkten erfolgt. Auch dieses Produkt wäre in der Lage, viele Komponenten dieser Software zu ersetzen.

Die in dieser Arbeit erstellte Prüfroutine basiert auf einer Analyse der Regelverstöße realer Datenlieferungen. Ein Schwerpunkt liegt bei der Attributkontrolle, wo Bedingungen und Folgebedingungen möglich sind. Neben dem Abgleich gegenüber gültigen Einzelwerten und Listen gültiger Werte kann durch Integration des RE-Moduls von Python auf nahezu unbeschränkte Weise gegenüber allen Mustern in Textfeldern geprüft werden. Mit dem RE-Modul gibt es eine eigene Beschreibungssprache, die exklusive Listen, inklusive Listen, Prüfung auf Sonderzeichen, Ziffern und vieles mehr unterstützt. Weiterführendes ist in der Literatur zu finden [64, S. 340]. Mustererkennung in Textfeldern wird auch von checkgeodata.net unterstützt. Welche Möglichkeiten bestehen, ist nicht klar.

---

<sup>60</sup> Überlagern, Berühren, Enthalten etc. von Features zweier Ebenen

<sup>61</sup> Mindestentfernungen von Vertices, Multipart – Singlepart, ...

Die Geodatenverwaltung der Software berücksichtigt alle notwendigen Bedürfnisse des Landes. Das Bauland wird aus FWPs extrahiert, die Geodaten werden historisch geführt. Änderungen von Daten werden in eigenen Featureklassen abgelegt, welche bei einem neuen Stammplan gelöscht werden müssen. Neben den eigentlichen Plänen können auch Planentwürfe, sogenannte Auflagen, verwaltet werden. Die Entwicklung solcher Spezialwünsche müsste bei der Firma Axmann bestellt werden und würden nur auf einen Fall hin entwickelt werden. Datenverwaltung ist nicht Gegenstand des Data Reviewer. Beide Produkte weisen eine Bruchstelle auf.

Die Vollständigkeit und Lagerichtigkeit der Geodaten kann durch einen Vergleich mit anderen Geodatenbanken des Landes durchgeführt werden. Dabei wird eine Unschärfe in den zu vergleichenden Geodaten berücksichtigt. Das wäre ebenfalls ein Alleinstellungsmerkmal. Der Data Reviewer sowie checkgeodata.net bieten keine speziellen Werkzeuge dafür an.

Alle verglichenen Produkte erkennen Fehler gegenüber Schnittstellen. Die im Rahmen der Dissertation erstellte Software geht jedoch weiter und repariert die fehlerhafte Lieferung, welche dann dem Planer zur Kontrolle zugeschickt wird. Dieser Service ist weder bei dem Programm checkgeodata.net noch bei der Software Data Reviewer vorhanden und wohl der bedeutendste Vorsprung. Darüber hinaus beruhen die Reparaturen auf einer Analyse realer Fehler, was die Praxisrelevanz zur Folge hat. Die Daten werden durch die entwickelte Software auch verwaltet und historisch geführt, was die Produkte am Markt ebenfalls nicht erledigen.

## 7.6. Abgrenzung zur künstlichen Intelligenz (KI)

Durch Programme, welche Spracherkennung leisten, Autofahrer unterstützen oder Schach spielen, ist künstliche Intelligenz alltagsrelevant geworden. Auch in dieser Arbeit wurde ein selbstlernender Algorithmus vorgestellt und einige Anregungen von Expertensystemen übernommen, weshalb dieses Kapitel sinnvoll erscheint.

Der bekannteste Test, der nachweisen soll, ob es sich um ein KI-System handelt, ist der Turing-Test. Eine Person sowie ein Computer beantworten Fragen eines Nutzers. Kann der

Nutzer nicht erkennen, ob er mit einem Menschen kommuniziert oder mit einem Computer, wird von KI gesprochen. C. Meier stellt fest:

„Bis heute hat kein Programm den Test wirklich und zweifelsfrei bestanden“ [88].

Die Bedeutung des Tests dürfte somit eher in der Zielsetzung liegen.

Beckheimer greift auf McCarty's Förderantrag für KI zurück und definiert KI abgekürzt als „Nachahmen von intelligentem Verhalten in künstlich geschaffenen Systemen“. Es werden die Komponenten „Wahrnehmen, Kommunizieren, Schlussfolgern, Lernen, Handeln in komplexen Systemen“ aufgezählt [89]. Expertensysteme, neuronale Netze, Fuzzy Logic-Systeme gelten als KI-Lösungen.

„Expertensysteme sind Programme, mit denen das Spezialwissen und die Schlussfolgerungsfähigkeit qualifizierter Fachleute auf eng begrenzten Fachgebieten nachgebildet werden soll.“ [90]

Sie bestehen aus einer Benutzerschnittstelle, einer Erklärungskomponente, einer Wissenserwerbskomponente, einer Wissensbasis und einem Inferenzmotor. Die Erklärungskomponente dieser Systeme war eine große Anregung. Auch hier wird den Usern erklärt, warum ihre Pläne nicht stimmen und welche Reparaturen durchgeführt wurden, die dann vom User zu verifizieren sind. Das notwendige Wissen wird einerseits von Raumplanern vorgegeben, indem ihre attributiven Ausbesserungen aufgezeichnet werden und als Vorlage für automatische Korrekturen verwendet werden, und zum anderen liegen fixe Reparaturpläne vor, wie beispielsweise bei Geometriereparaturen (multipoint to singlepoint), Korrektur des Koordinatensystem, Spellchecker, Attributstrukturbereinigung etc. Wir haben also durch die Vorgabe von Raumplanern eine Wissenserwerbskomponente, welche eine Wissenstabelle aufbaut, und andererseits auch „fix verdrahtetes“ Know-how in einer Wissensbasis. Als Benutzerschnittstelle wurde ein Mailingsystem aufgebaut.

„Der Inferenzmotor stellt die zentrale Problemlösungskomponente dar. Hier werden die, aus der Problemstellung extrahierten Fakten entsprechend der Regeln verknüpft und auf neue Fakten geschlossen. Die Reihenfolge in der dies geschieht, wird von der Inferenzmaschine selbst festgelegt.“ [91]

Inferenzmotoren sind komplexe Komponenten, welche Schlüsse ziehen und eine Vielzahl an Handlungsmöglichkeiten besitzen. Im gegenständlichen Fall waren die Handlungsmöglichkeiten bescheiden. Liegt eine Multipoint-Struktur vor und war laut Schnittstelle eine Singlepoint-Struktur gefordert, ist der Reparaturschritt „*multi to single*“ durchzuführen. Wir haben es mit simplen Wenn-dann-sonst-Abfragen zu tun, die nicht aus den Programmablauf herausgeschält wurden und in einem eigenen Modul behandelt werden. Ein Inferenzmotor liegt also nicht vor und damit fehlt eine zentrale Komponente des Expertensystems.

In der Regel werden Expertensysteme von speziellen Fachleuten angefertigt und eine ganze Gruppe von Experten befüllt die Wissensbasis. Die Vielzahl an Personen und die Größe des Systems sind nicht gegeben, um von einem Expertensystem zu sprechen.

## 8. Fazit

Am Beispiel des FWP's wurde eruiert, wo Planer Schwierigkeiten bei der Erstellung von Geodaten haben. Dazu wurden 331 Abgaben von Plänen analysiert, in denen über 12000 Fehler gefunden wurden. Damit liegt eine solide Datenbasis vor, um Aussagen über die Problemfelder bei der Erstellung von rechtsrelevanten Geodaten zu tätigen. Überraschend war die Erkenntnis, dass der Großteil der Fehler (73 %) mit hoher Zuverlässigkeit automatisch bereinigt werden kann, da keine speziellen Situationskenntnisse dafür notwendig sind. Diese Bereinigung umfasst Arbeiten, wie das Löschen überflüssiger Spalten, die Korrektur von Buchstabenstürzen und Ähnliches. Durch die Entwicklung von einer Software (Hödlbert<sup>62</sup>), welche die Ausbesserarbeiten durchführt, konnte dem Datenproduzenten viel aufwendige Routinearbeit abgenommen werden. Dabei wurden zwei prinzipiell unterschiedliche Methoden angewandt und miteinander verglichen. Zum einen wurden für Verstöße gegen die Datenschnittstelle Lösungsalgorithmen implementiert, wobei Expertensysteme mit Erklärungskomponenten als Vorbild herangezogen wurden. Zum anderen wurden die fehlerhaften Abgaben von Planern und die danach folgenden Korrekturen elektronisch aufgezeichnet. Mehrfach angewandte Lösungsstrategien, im Sinn von Aktivierungsfunktionen neuronaler Netze, wurden für automatische Korrekturvorschläge herangezogen. Korrigierte Geodatenätze werden dem Planer mit einem entsprechenden Fehlerprotokoll zur Kontrolle rückübermittelt.

Es wurde also ein vereinfachtes Expertensystem einem selbstlernenden System gegenübergestellt, wobei beide Varianten zu brauchbaren Ergebnissen führten. Die erste Variante ist in der Lage, die Korrektur zu begründen. Das zweite System kann keine sachlichen Gründe für die Korrektur anführen, was für behördliche Verfahren nicht ausreichend erscheint. Darüber hinaus werden beim selbstlernenden System Lösungsstrategien eines Planers zur Unterstützung eines möglicherweise im wirtschaftlichen Wettbewerb stehenden Konkurrenten eingesetzt. Dadurch entsteht das Problem des Urheberrechtes, welches derzeit auf europäischer Ebene diskutiert wird.

Regelverstöße gegenüber Schnittstellen können aufgedeckt werden und den Datenerstellern kann mit Hilfe automatischer Korrekturprogramme viel Arbeit abgenommen werden.

---

<sup>62</sup> Das Programm heißt Hödlbert nach dem Kaktus meiner jüngsten Tochter.

So wie Schreibprogramme über Rechtschreib- und Grammatikkorrekturen verfügen, sollten Prüfprogramme Datenkorrekturen durchführen. Die Ergebnisse sind weitestgehend brauchbar. Der Planinhalt ist damit nicht geprüft. Hierfür gibt es weitere Möglichkeiten.

Viele Dateninhalte von Plänen sind in anderen Datenbanken mit anderen Maßstäben und Dimensionen gespeichert. Nichtsdestotrotz können damit Omission und Commission von Objekten geprüft werden. Dabei werden die Objekte der höheren Dimension um einen den unterschiedlichen Maßstäben entsprechenden Betrag gebuffert und diese Bufferflächen müssen die Vergleichsobjekte enthalten [70]. Dieser Ansatz wurde einer Fuzzy-Lösung gegenübergestellt, die bei entsprechender Pixelauflösung zum selben Ergebnis führt. Weil mit unscharfen Mengen gearbeitet wird, ist die Argumentation schwierig, wenn Objekte beanstandet werden. „Das übermittelte Objekt O weicht gegenüber dem Register R um mindestens 100 m in der Position ab“, ist ein leicht kommunizierbarer Satz, der dem ersten Ansatz entspricht. „Das übermittelte Objekt O weicht mit hoher Wahrscheinlichkeit in nicht tolerierbarer Weise vom Register R ab“, entspricht dem zweiten Ansatz und vermittelt dem Datenproduzenten, dass er willkürlicher Kritik unterworfen ist. Für eine behördliche Anwendung disqualifiziert sich dadurch der Fuzzy-Ansatz.

Der planungsautonome Teil entzieht sich gänzlich der Prüfung. Datenproduzenten kontrollierten die abgegebenen Geodaten des Flächenwidmungsplans in einem bereitgestellten Web-Service mit ihren Papierplänen. Bei der Analyse konnte festgestellt werden, dass bei 8% der Abgaben inhaltliche Fehler vorliegen. Die optische Kontrolle ist unabdingbar, wenn die Geodaten als Gleichstücke zur Rechtskarte fungieren sollen. Das hohe qualitative Ziel kann erreicht werden, indem die Rechtsverfahren adaptiert werden. Für Verordnungen des Landes mit Raumbezug gibt es einen generellen Weg, der ohne wesentlichen Mehraufwand zu Registern von rechtsrelevanten Geodaten führt. Der Landesverfassungsdienst, welcher alle Verordnungen des Landes in das Rechtsinformationssystem einbringt, akzeptiert nur raumrelevante Verordnungen mit Kartenmaterial einer behördlichen Zentralstelle für Geoinformatik<sup>63</sup>. Diese Zentralstelle sammelt ihrerseits diese Geodaten und stellt sie zentral im Internet bereit, was hohe Publizität und leichte Zugänglichkeit für den Bürger garantiert. Ein zusätzlicher Kommunikationskanal zwischen beiden Einheiten stellt sicher, dass nur tatsächlich von der Regierung beschlossene Verordnungen in die publizierten Register

---

<sup>63</sup> Im Land Steiermark bietet sich dafür das GIS-Steiermark der Abteilung 17 an, welches für alle Landesdienststellen und Bezirksverwaltungsbehörden Geodienste bereitstellt.

aufgenommen werden. Damit würde eine Sammlung aller raumrelevanten Verordnungen des Landes entstehen, auf die ebenso zentral und einfach zugegriffen werden kann, wie auf das Grundbuch. Nicht nur privatrechtliche Beschränkungen auf Grund und Boden wären dann zentral über das Internet abfragbar, sondern auch ein Großteil der öffentlich-rechtlichen räumlichen Beschränkungen. Zur praktischen Umsetzung wäre lediglich eine Weisung des Landesamtsdirektors notwendig, was ohne besonderen Aufwand vonseiten der Behörde durch formlose Ansuchen und Gespräche eingeleitet werden kann.

Bei Plänen, die von vielen autonomen Körperschaften wie den Gemeinden erstellt werden, kommt der übergeordneten Prüfbehörde eine entscheidende Rolle zu. Am Beispiel des Flächenwidmungsplanes wurde gezeigt, wie ein Verfahren aussehen müsste, das neben der Schnittstellenentsprechung auch die Richtigkeit des planungsautonomen Teiles garantiert, und die Daten Eingang in ein Register öffentlich-rechtlicher Beschränkungen finden könnten. Dazu müssen die unverzichtbaren Papierpläne<sup>64</sup> von einer zentralen Behörde für das Genehmigungsverfahren aus bereitgestellten Geodaten erstellt werden, was bei entsprechender Adaption der Schnittstellen und Planzeichen vollautomatisch erfolgen könnte. Bei dieser zentralen Behörde werden diese Geodaten gesammelt, welche sich nicht von den Papierplänen unterscheiden. Wie das Beispiel des Flächenwidmungsplans zeigt, wird hierfür eine Änderung der gesetzlichen Grundlage benötigt, da der Verfahrensablauf über das Raumordnungsgesetz festgelegt ist. Dafür ist ein Landtagsbeschluss erforderlich.

Es ist davon auszugehen, dass auch viele Entwürfe entstehen, die nicht Rechtskraft erlangen. Technische Vorkehrungen stellen sicher, dass die Vielzahl an Planversionen zu keinen Verwechslungen führen. Die Richtigkeit und die Vollständigkeit der Informationen (Integrität) können vor illegaler Manipulation oder unbewussten Bearbeitungen durch digitale Signaturen geschützt werden.

Die diskutierten Verfahren können nur funktionieren, wenn sie Geodaten der höchsten Qualität garantieren und die Verarbeitung der Geodaten möglichst fehlerfrei ablaufen. Es stehen theoretische Qualitätsmodelle für Geodaten und Software bereit, die für die Schnittstellenprüfung als auch Verarbeitungssoftware herangezogen wurden. Vollständigkeit, logische

---

<sup>64</sup> Geodaten selbst können nur von Spezialisten gelesen und verarbeitet werden. Es kann Politikern nicht zugemutet werden, Beschlüsse auf Grundlagen zu fassen, die für sie nicht zugänglich sind. Daher ist ein Papierplan oder eine entsprechende unveränderbare PDF-Datei im Verfahren unabdingbar.

Konsistenz, Positionsgenauigkeit, Zeitgenauigkeit, Thematische Genauigkeit, Zweck, Verwendung und Herkunft bilden die qualitative Basis der Geoinformatik und Korrektheit, Robustheit, Zuverlässigkeit, Effizienz, Sicherheit, Verfügbarkeit, Wartbarkeit, Flexibilität und Gebrauchstauglichkeit die Basis der Software. Beide Modelle werden integral miteinander verwoben und führen zu den gewünschten Ergebnissen, wobei die Größe des Projektes ein geplantes Vorgehen bedingt. Im Bereich des Software Engineering sind für verschiedene Problemstellungen Modelle vorhanden, die beschreiben, wie Software zu entwickeln ist. Diese Modelle wurden evaluiert und ein inkrementelles Vorgehen mit starker Anlehnung an agilen Grundsätzen war die Methode, welche ein schnelles Fortschreiten bei hoher Qualität garantiert. Insbesondere auf geändertes Userverhalten, das häufig vom Einsatz der Software selbst hervorgerufen wird, kann gut eingegangen werden. Die Überwachung von Softwareindizes garantiert eine homogene Qualität, wobei dem Maintainability-Index eine hohe Bedeutung zukommt, da ein Großteil der Softwarearbeit auf Wartungstätigkeiten fällt. Für die Zuverlässigkeit wurde ein eigener Index entwickelt.

Kritische Stellen wurden nach dem Verfahren des Test-Driven Developments entwickelt. Kontrollprogramme ermöglichen das Prüfen des Codes einer gesamten Klasse, wenn an einer Stelle eine Änderung vorgenommen wird. Blackbox-Tests des Gesamtsystems mit Echtdateien sind trotzdem unverzichtbar. Viele bereits vorhandene Pläne ermöglichten ausreichende Lasttests mit Echtdateien.

Die Arbeit zeigt, wie man Vorgehen muss, um zu rechtsrelevanten Geodaten zu kommen und wie die notwendigen Qualitätsbedingungen bei den Daten und der Verarbeitungssoftware garantiert werden können. Rechtsrelevante Geodaten haben zur Folge, dass viele Rechtsgeschäfte ohne aufwendige Wege zu Ämtern erfolgen können, da die notwendigen räumlichen Informationen durch Kartenservice im Internet von den Behörden gebündelt angeboten werden können. Diese Reglementierungen können einerseits den Wert eines Grundstücks wesentlich beeinflussen und sind andererseits im Internet transparent. Die Arbeit stellt neben der Verwaltungsvereinfachung einen Schritt in Richtung Rechtssicherheit für Bürger dar: man weiß was man kauft.

## Anhang - Fehlerbericht

### 1. Vorgehensweise

Mit der Verordnung der Schnittstelle 2007 für den FWP und dem Örtlichen Entwicklungsplan (OEP) wurde die erste Version einer Software zur automatischen Prüfung und Abarbeitung dieser Pläne entwickelt [1]. Diese Software wurde seinerzeit von mir konzipiert und vom Büro DI Mussack & DI Skalicki-Weixelberger programmiert. Das Programm erstellte Prüfprotokolle, welche dem Planer, der die Daten erstellt hat, zugeschickt wurden. Jede dieser Mails listet die begangenen Schnittstellenverstöße bei einer Lieferung auf. Sie wurden in einem Container gesammelt. Da die Mails von einem Computer erstellt wurden, haben sie einen systematischen Aufbau. Analysiert man diese Schriftstücke, kann man Aussagen über die Probleme treffen, die Planer mit der Erfüllung der Norm haben. Für die Analyse dieser Prüfprotokolle wurde ein Programm in Python geschrieben, welches die Dateien sequenziell abarbeitet und Fehler summiert sowie kategorisiert.

### 2. Beobachtungsperiode und Endergebnisse Flächenwidmungspläne

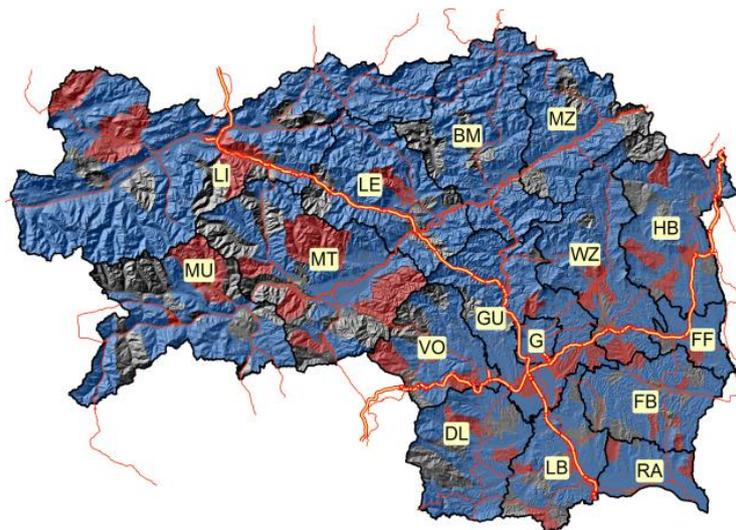


Abbildung 19: Gemeinden mit digitalen FWPs

Für die Analyse der begangenen Fehler standen Protokolle des Zeitraumes vom 27.3.2009 bis zum 19.11.2011 zur Verfügung. Es handelt sich also um 967 Tage, was 2,6 Jahren entspricht. In dieser Periode wurden insgesamt 53 FWP in die Datenbank des Landes eingefügt, wobei pro korrekter Abgabe im Allgemeinen mehrere Versuche notwendig waren. In der Steiermark gab es im Beobachtungszeitraum 542 Gemeinden, sodass über die Schnittstelle 9,8 % der möglichen Planwerke übermittelt wurden. Abbildung 1 zeigt die Flächendeckung bei FWP am Ende der Beobachtungsperiode, wobei rot eingefärbte Gemeinden Planwerke der Schnittstelle 2007 aufweisen (11,2 % der Landesfläche) und blau dargestellte Flächen (71,5 %) vorherigen Schnittstellen genügen. Für die rot dargestellten Gemeinden lagen also die Prüfprotokolle vor.

### 3. Örtliche Entwicklungspläne

Mit der Schnittstelle 2007 wurde auch die digitale Abgabe der OEPs verordnet, wobei keine Strafbestimmungen für die Nichtabgabe vorgesehen sind. Am Ende der Beobachtungsperiode waren 27 OEPs vorhanden, was 5 % aller möglichen Pläne darstellt. Die Flächendeckung ist zu 7,5 % gegeben.

Es wurden also deutlich weniger örtliche Entwicklungskonzepte als Flächenwidmungspläne von den Gemeinden oder beauftragten Raumplanern abgegeben. Es wäre zu erwarten gewesen, dass fast mit jedem FWP der übergeordnete OEP hochgeladen wird. Mögliche Ursache der Diskrepanz ist, dass die OEPs, auf denen die FWP gründen, älter als diese sind und noch nicht unter die Abgabepflicht fallen. Auch ist die Akzeptanz einer neuen Vorschrift nicht von Beginn an gegeben. So ist anzunehmen, dass die Abgabepflicht der OEPs teilweise ignoriert wurde. Das zuständige Referat wurde über diesen Umstand informiert und es wurde auf Wunsch des Referates ein Abgleich mit der Terminevidenz durchgeführt, wo alle Verfahren und die Termine des FWP und OEPs gespeichert sind. Als Ergebnis dieser komplexen Auswertung lag eine Liste der einzumahnenden Pläne vor, welche alle Ausnahmen des Gesetzes berücksichtigt. Betroffene Gemeinden wurden aufgefordert, die Daten nachzuliefern. Daraufhin wurde ein Großteil der fehlenden OEPs geliefert.

Es wird zwischen Stammplänen, welche das ganze Gemeindegebiet beschreiben, und Abänderungen, welche nur Teilbereiche der Gemeinde betreffen, unterscheiden. Innerhalb der Beobachtungsperiode wurde kein Stammpfad (FWP und OEP) upgedatet. Das entspricht

der Erwartung, da der Planungshorizont für den OEP 15 Jahre und für den FWP 10 Jahre beträgt. Es lagen nur Protokolle von Stammpänen vor.

FWPs wurden zur Gänze analysiert, da eine große Datenbasis besteht und die Probleme an ihnen aufgezeigt werden können. OEPs wurden nicht in voller Tiefe untersucht, wobei sich für beide Planarten, soweit die Untersuchung erfolgte, etwa das gleiche Bild zeigt. Da OEPs aus deutlich weniger Ebenen und Features bestehen, sind auch entsprechend weniger Fehler registriert worden. Schwierigkeiten zeigen sich bei dieser Planart nicht so deutlich.

## 4. Beschreibung der Analysen und Ergebnisse

### 4.1. Fehler in den abgeleiteten Gesamtdaten

Bevor die Protokolle untersucht wurden, sind die aus den Abgaben resultierenden Ergebnisse durchleuchtet worden. Die Datenbestände des Landes wurden auf ihre Konsistenz überprüft.

Die gemeindeweise gelieferten Pläne werden zu einer landesweiten Sicht zusammengespielt. Es erfolgt eine historische Datenführung. Die Praxis hat gezeigt, dass auch ein Datenbestand notwendig ist, der lediglich die aktuelle Situation zeigt. Diese wird aufgebaut, indem die alten Ebenen der Gemeinde gelöscht werden und durch die neue Lieferung ersetzt werden.<sup>65</sup> Der historische Datenbestand entsteht, indem der neue Plan einfach über den alten Plan gespeichert wird, ohne den alten Plan zu löschen. Da die Pläne mit einem Zeiger versehen sind, der ihre zeitliche Gültigkeit angibt (Verfahrensfallnummer), entsteht so ein historischer Datenbestand.

Bei der Analyse der historischen Pläne wurde festgestellt, dass drei FWPs im historischen Datenbestand fehlen. Die Fehler wurden korrigiert. Die Ursache ist nicht geklärt, dürfte jedoch in einem Absturz der entsprechenden Softwarekomponenten zu finden sein. Eine Adaption der Software, die Abbrüche anzeigt, ist notwendig. Vermutlich liegt die Ursache

---

<sup>65</sup> In der Software sind auch Gemeindefusionen zu berücksichtigen. Die Gemeindefusionen der Steiermark fasst nicht nur Gemeinden zusammen, sondern, zerschneidet auch einige Gemeinden entlang von „beliebigen“ Grundstücksgrenzen, weshalb obige Beschreibung als eine grobe Vereinfachung anzusehen ist.

für den Absturz an kurzzeitigen Netzwerk- oder Rechnerausfällen, gegenüber denen das System besonders anfällig ist. Die neue Software ist robuster zu konzipieren und umzusetzen.

Das Gesamtsystem, von dem der Nutzer abhängig ist, benötigt 6 verschiedene Computer, das Netzwerk und dessen Router. Von vielen Anwendungen des Landes gemeinsam genutzte Komponenten kommen zum Einsatz und führen zu dieser Komplexität.<sup>66</sup>

Moderne Softwarearchitektur greift gerne auf das „Cloud Computing“ zurück. Der Name geht auf grafische Darstellungen von Computerarchitekturen zurück, in denen gewisse Leistungen als Wolke dargestellt werden, die irgendwo im Netz durchgeführt werden. Deren genaue Lokalisation interessiert im Allgemeinen nicht. Dabei unterscheidet man zwischen „Infrastructure as a service“, „Platform as a Service“ und „Application as a Service“. Die Übernahme von Flächenwidmungsplänen unterliegt einer gesplitteten Organisation und nutzt zentrale Services, zentrale und dezentrale Datenbanken und im Netz verteilte Speicher. Infrastructures as services werden von der zentralen EDV genutzt. Es liegt ein internes „Cloud Computing“ vor, was auch als private Cloud bezeichnet wird. Cloud Computing bringt Synergieeffekte, aber auch Abhängigkeiten, wobei die verteilte Persistenz das Hauptproblem für die Betriebssicherheit darstellt und auch zu Performanceeinbußen führt. Eine genaue Untersuchung wäre notwendig. Das im Rahmen dieser Arbeit entwickelte Programm schreibt ein Systemprotokoll, welches auch Probleme dieser Art aufzeichnet. Nach einer gewissen Betriebsdauer kann auf Basis einer Auswertung des Systemprotokolls eine

---

<sup>66</sup> Der externe Raumplaner meldet sich über den Portalrechner an, wo die Authentifizierung erfolgt. Die Daten werden am Applikationsserver entpackt und auf den Platten des Abteilungsrechners abgelegt, wobei eine Protokollierung in einer Oracle-Datenbank durchgeführt wird, welche ebenfalls am Applikationsserver läuft. Am Rechner der GIS-Abteilung werden die Daten kontrolliert, verwaltet und das Mailing wird durchgeführt, wofür ESRI – Komponenten verwendet werden, was Kommunikation mit dem Lizenzrechner nach sich zieht. Die Prüfung und Verwaltung wird als automatischer Task gestartet und erfolgt in periodischen Abständen von 10 Minuten. Dafür wird eine Authentifizierung des Programms am Rechner der zentralen Userverwaltung durchgeführt.

auf Erfahrungen gestützte Abwägung zwischen Synergieeffekten und Abhängigkeiten erfolgen.

#### 4.2. Fehler der Einzelpläne

Die Geodaten können entweder im Testmodus oder im Abgabemodus abgegeben werden. Der Testmodus erstellt den Prüfbericht und sendet ihn dem Planer zu. Der Abgabemodus führt nicht nur die Prüfung durch, sondern integriert die Daten auch in die Datenbank des Landes. Nur vollständig richtige Geodaten werden in den Datenbestand des Landes integriert. Der FWP einer Gemeinde wird in der Regel mehrfach übermittelt, bis er fehlerfrei vorliegt. Bei einer fehlerfreien Lieferung im Abgabemodus werden die Daten übernommen. Als offen gelten Gemeinden, für welche Uploads erfolgten, aber noch keine positiv geprüfte Übermittlung im Abgabemodus stattfand.

Für 62 verschiedene Gemeinden wurden FWPs abgegeben. Da im Beobachtungszeitraum 53 Gemeinden in die Datenbestände des Landes integriert wurden, kann man vermuten, dass derzeit noch 9 Gemeinden bearbeitet werden. Zwei Datenlieferungen waren jedoch lediglich Softwaretest und eine Gemeinde wurde in der Zwischenzeit positiv getestet. Die restlichen wurden als fehlerhaft zurückgewiesen. Um keine Verfälschungen in Aussagen wie „positive Abgaben zu Abgabeversuchen“ zu erhalten, wurden nur vollständig abgeschlossene Gemeinden einer Analyse unterzogen. Ein kurzer Blick auf die unvollständigen Gemeinden ist trotzdem interessant.

Fünf der nicht abgeschlossenen Planlieferungen wurden von einem Mitarbeiter des Landes übergeben, der diese für den Raumplaner freundlicherweise testete. Die Raumplaner vermuteten, dass mit der Übermittlung der Geodaten an einen Landesbediensteten die Abgabe erledigt ist. Das Prüfprotokoll wurde von dem Bediensteten bzw. dem Planer ignoriert. Der Kollege wurde kontaktiert und auf die Problematik aufmerksam gemacht. Zukünftig verwaltet er, wie es immer seine Aufgabe war, die Zugangsrechte und führt keine Prüfungen für Planer durch. Die Planer wurden kontaktiert und haben in der Zwischenzeit bereits positive Daten hochgeladen. Die Initiative vieler Mitarbeiter ist lobenswert und wurde wahrscheinlich durch eine fehlerhaft interpretierte Weisung von mir hervorgerufen.

In weiterer Folge werden nur abgeschlossene Pläne beachtet, da nicht gesagt werden kann, wie viele fehlerhafte Versuche in einer Abgabe noch folgen werden, und so die Aussagen verfälscht wären.

Jeder Upload wird in einer Oracle-Tabelle festgehalten, welche statistische Interpretationen zulässt.

Tabelle 12: Anzahl an positiven sowie negativen Abgaben und Systemabstürzen

Planart	Anzahl	Prüfung/Abgabe	Fehlerfrei	Falsch	Systemfehler
FWP	331	176/155	95 (28,7 %)	231 (69,8 %)	5 (1,5 %)

Planart	Flächenwidmungsplan (FWP)
Anzahl	Anzahl der Uploads von Geodaten
Prüfung/Abgabe	Die Geodaten wurden nur zur Prüfung hochgeladen/ein Abgaberversuch wurde durchgeführt.
Fehlerfrei	Die Daten waren fehlerfrei.
Falsch	Es wurden Fehler registriert.
Systemfehler	Es kam zum Absturz der Software.
Alle Prozentangaben beziehen sich auf die Anzahl.	

Systemfehler traten bei 2 % aller Uploads auf und wurden vom System als solche erkannt. Hier war manuelles Eingreifen notwendig. Damit liegen nicht nachvollziehbare Manipulationen am System vor, worunter seine Rechtsrelevanz leidet. Es kann im Nachhinein nicht mehr gesagt werden, was geändert bzw. repariert wurde, da in der Hektik eines Systemausfalls auf Dokumentationen verzichtet wurde.

Die damit verbundene Arbeitszeit fällt nicht ins Gewicht, wenn man gegenüberstellt, dass vor dieser Schnittstelle die Daten zu Gänze manuell bearbeitet wurden. Diese Tätigkeit hat einen Vollzeit Arbeitsplatz benötigt. Der Benutzer des Portals ist nur insofern betroffen, als er länger auf das Prüfprotokoll warten muss. Das macht im Allgemeinen keine großen Probleme, da die Datenlieferanten noch die „händische“ Kontrolle gewöhnt sind, die generell so lange gedauert hat, wie jetzt die Ausnahmefälle dauern.

Nichtsdestotrotz spricht dieser Umstand gemeinsam mit den Sachverhalten, die im Anhang – Fehlerbericht 4.1 dargestellt werden, für eine Überarbeitung des Softwarepaketes in Hinblick auf Korrektheit (siehe Kapitel 4.5.1), Robustheit (siehe Kapitel 4.5.2) und Zuverlässigkeit (siehe Kapitel 4.5.3).

In weiterer Folge wurde untersucht, wie viele Abgaben pro Gemeinde notwendig waren, und diese wurden in Klassen eingeteilt. So entstand die Häufigkeitsverteilung, wie sie in Abbildung 2 dargestellt wird. Die Abszisse zeigt die notwendigen Uploads bis zur positiven Abgabe und die Ordinaten die Summe an Gemeinden, welche diese Anzahl an Uploads notwendig hatten.

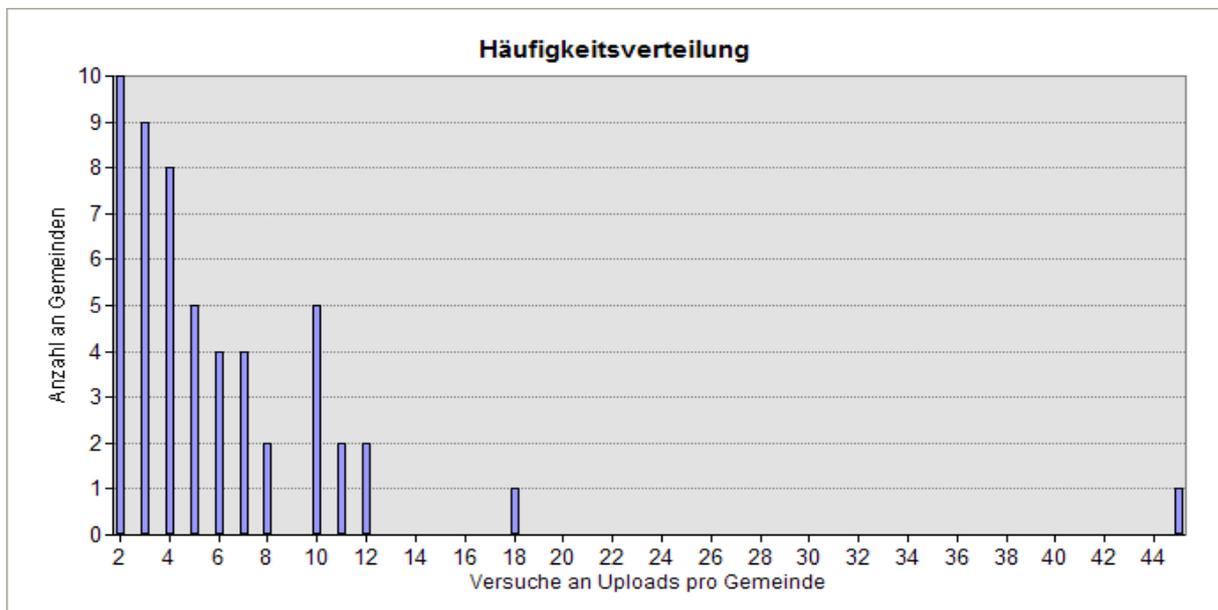


Abbildung 20: Histogramm der Uploads pro für Gemeinde

In der Flächenwidmungsplanung kommen auf eine positive Abgabe ca. 2,4 fehlgeschlagene Versuche. 10 Gemeinden benötigten lediglich zwei Übergaben. Nur bei einer Gemeinde war die erste Übergabe bereits korrekt. Sie erfolgte im Prüfmodus und musste deshalb nocheinmal im Übergabemodus hochgeladen werden. 9-mal wurden fehlerhafte Daten zurückgewiesen. Beim zweiten Versuch war die Lieferung in Ordnung und erfolgte im Abgabemodus.

Für die Abgabe eines FWP sind im Schnitt 6,2 Uploads notwendig. Dem aufmerksamen Leser ist möglicherweise nicht entgangen, dass jeder positiven Abgabe nur 2,4 Fehlschläge

gegenüberstehen, was ein Widerspruch zu diesem Mittelwert sein könnte. Das ist jedoch nicht der Fall, denn mitunter benötigt eine Gemeinde mehrere positive Abgaben.

Zumindest zwei positive Abgaben pro Gemeinde sind notwendig, wenn die Abgabe solange im Prüfmodus erfolgt, bis ein positives Ergebnis vorliegt. Dann müssen die korrekten Daten nochmals im Übergabemodus hochgeladen werden. Das war 33-mal der Fall.

Zu mehreren Uploads pro Gemeinde kann es auch kommen, wenn nach der positiv bewerteten Übergabe inhaltliche Korrekturen notwendig waren. Das ergab sich meist aufgrund einer optische Kontrolle der Daten im WebGIS des Landes durch Planer oder Gemeinden. Der Plan wird dann überarbeitet nochmals hochgeladen.

Damit kann festgehalten werden, dass die Erstellung eines schnittstellenkonformen Geodatenatzes dem Planer Schwierigkeiten bereitet. Insgesamt wurde die Schnittstelle nur von 13 verschiedenen Personen benutzt. Mit nur 2 Versuchen zur Erstellung eines schnittstellengerechten Geodatenatzes kamen lediglich Mitarbeiter von Firmen aus, deren Schwerpunkt die Geoinformatik darstellt. Sie wurden von Raumplanern beauftragt, die Geodatenätze zu erstellen. Führten die Raumplaner bzw. Gemeinden die Erstellung des Geodatenatzes selbst durch, waren bis dato zumindest 3 Versuche notwendig.

Der Extremfall von 45 Uploads enthielt 5 positive Abgaben, welche im Internet publiziert wurden. Von Usern – hauptsächlich Gemeindebedienstete der betroffenen Gemeinde – wurden in diesem konkreten Fall immer wieder Abweichungen gegenüber dem rechtsgültigen Papierplan entdeckt, die dazu führten, dass die Daten neuerlich vom Raumplaner hochgeladen werden mussten. Hauptsächlich hat es sich um fehlende Festlegungen gehandelt, die im Papierplan vorhanden waren, im digitalen Plan jedoch fehlten. Diese Löcher im Geodatenatz, waren sehr groß und mussten deshalb korrigiert werden. Es ist notwendig, in den Prüfkanon Topologieprüfungen aufzunehmen. Dieser Fall hat auch ein Problem bei der historischen Datenführung aufgezeigt. Jede positive Abgabe wird in der historischen Datenbank gespeichert. Diese Datenbank enthält somit auch inhaltlich fehlerbehaftete Planversionen. Das trifft auf 5 Gemeinden zu.<sup>67</sup> Es stellt sich die Frage, ob es Sinn macht, die historischen Versionen von falschen Versionen zu bereinigen. Für die Bereinigung spricht, dass

---

<sup>67</sup> Gemeinde 61751 5-fach, 60813 3-fach, 60821 2-fach, 61415 2-fach, 61434 2-fach.

Zeitreihen damit einfacher zu erstellen sind. Gegen eine vollständige Beseitigung dieser Information spricht, dass die lückenlose Dokumentation verloren geht.

Man kann davon ausgehen, dass noch unentdeckte Fehler in den digitalen Daten vorhanden sind. Bei zumindest 8 % der Gemeinden wurden nach der positiven Abgabe noch inhaltliche Fehler gefunden. Um rechtlich relevante Daten zu erhalten, ist es also unabdingbar, die elektronischen Daten nocheinmal optisch zu kontrollieren.

Für eine Gemeinde waren 18 Uploads notwendig, und betrachtet man die Abgaben genauer, entdeckt man, dass jede Ebene einzeln hochgeladen und getestet wurde. Nachdem alle Ebenen abgearbeitet waren, wurde der Gesamtplan abgegeben. Diese Vorgehensweise unterscheidet sich von den anderen Abgaben, bei denen immer versucht wurde, einen Gesamtplan abzugeben.

Eine gewisse Verfälschung der Statistik liegt auch vor, weil die Schnittstelle ausprobiert wird. Vorläufige Geodaten werden zur Prüfung hochgeladen. Auf diese Weise ist der Planlieferant imstande, eine korrekte Datenbasis zu bauen, die in späterer Folge mit rechtsrelevanten Informationen aufgefüllt wird.

#### 4.3. Fehler innerhalb eines Planes

Im Folgenden werden die Prüfungen der Geodaten beschrieben sowie quantitativ bewertet und es werden Überlegungen angestellt, ob automatische Reparaturen möglich sind. Die quantitative Erhebung unterscheidet unter der Anzahl an fehlerhaften Lieferungen und der Anzahl an auftretenden Fehlern. Diese Anzahl ist im Allgemeinen unterschiedlich, da in einer Lieferung ein Fehler mehrfach vorkommen kann.

##### 4.3.1. Abgleich mit der Baulandbilanz

Unter der Baulandbilanz versteht man die aufsummierten Flächen zusammengefasster Baulandwidmungen einer Gemeinde. Diese Bilanz gibt also an, wie groß die Gesamtausmaße an den Widmungen „Wohnen Allgemein“, „Wohnen Rein“ usw. in der Gemeinde sind. Diese Baulandbilanz muss vom Benutzer vorgegeben werden. Es wird geprüft, ob die Geo-

daten diesen Vorgaben entsprechen. Dabei wird eine Toleranzschranke von 15 % angewandt. Die Schranke ist so hoch, da die Baulandbilanz in Hektar mit 2 Nachkommastellen vorgegeben wird. Die Summenkontrolle erfolgt zuerst für die Gesamtfläche und ein zweites Mal für die unbebauten Flächen, die sogenannte Baulandreserve einer Gemeinde. Dafür wird die Ebene der Baulandreserven mit der Nutzungsebene verschnitten. Die Fehlerprotokolle können diesbezüglich ausgewertet werden, aber die ermittelte Anzahl (5) ist nicht aussagekräftig, da dieser Kontrollmechanismus nicht immer eingeschaltet war. Ein quantitativer Vergleich mit anderen Fehlerarten ist deshalb nicht möglich.

Je nach Ursache ergeben sich verschiedene Korrekturmöglichkeiten. Liegen identische doppelte Flächen vor, können diese gelöscht werden. Im Randbereich kommt es häufig zu sehr kleinen Löchern oder sich überlagernden Flächen. Diese Löcher bzw. Überlagerungen könnten automatisch bereinigt werden, indem Koordinaten innerhalb eines Toleranzbereiches gleichgesetzt werden (ESRI Geoprocessing Function: Integrate). Als Toleranzbereich würde sich 1 mm anbieten, da die Katasterdaten, welche Basis der Erfassung sind, auf diesem Raster aufsetzen. Slivers können an ihrer kleinen Fläche selbst, aber auch an dem sehr kleinen Fläche/ (Umfang\*Umfang)-Verhältnis erkannt werden. Auf diese wichtige Kenngröße wird im Kapitel 7.4.3.2 eingegangen.

#### 4.3.2. Liegen prüfbare Elemente vor?

Zu Beginn werden alle Dateien gelöscht, welche keine gültige Dateierweiterung des Shape-Formates aufweisen. Ausgenommen ist das Informationsfile. Es wird geprüft, ob noch Geodaten vorliegen. Liegen keine Geodaten mehr vor, wird folgende Meldung per Mail abgesetzt und die Prüfung abgebrochen.

Fehlermeldung: Zuwenig Ausgangsdaten vorhanden – Programmabbruch !

Anzahl: 0

Jede Abgabe hat zumindest formal Geodaten enthalten.

#### 4.3.3. Unvollständiger Fileaufbau

Es wird überprüft, ob die Dateien .dbf, welche die Attribute enthalten, und die Dateien .shx, welche die Referenzen zu den Geometriedateien .shp enthalten, vorhanden sind.

Fehlermeldung: \*\* Unvollständiger Geodatensatz (.shx oder .dbf fehlt): Name der Ebene

Anzahl der Lieferungen mit diesem Fehler (FWP): 1

Anzahl des Auftretens dieses Fehlers (FWP): 1

Dieser Fehler trat nur bei einem einzigen Geodatensatz auf und hat zu einem Systemabsturz geführt, der daraufhin behoben wurde. Es liegt dementsprechend keine protokollierte Mail vor. Eine automatische Reparatur ist schwer vorstellbar. Die Daten müssten zur Gänze aus anderen Datenbeständen erstellt werden können.

#### 4.3.4. Anzahl der Elemente eines Geodatensatzes ist nicht bestimmbar

Ist die Anzahl der Elemente eines Geodatensatzes nicht bestimmbar, liegt ein schwerer Fehler vor. Das kann der Fall sein, wenn nicht jedem Geometrieelement eine Zeile in der beschreibenden Sachdatentabelle (\*.dbf) zugeordnet werden kann oder umgekehrt. Dieser Fehler passiert, wenn die DBASE-Datei des Shape-Formates mit Excel nachbearbeitet wird. Dabei wird häufig ungewollt eine zusätzliche leere Zeile angelegt, welche zu diesem Problem führt.

Fehlermeldung: \*\* Die Anzahl der Elemente im Geodatensatz kann nicht bestimmt werden: Name der Ebene

Anzahl der Lieferungen mit diesem Fehler (FWP): 1

Anzahl des Auftretens dieses Fehlers (FWP): 1

Der Fehler trat nur bei einem einzigen Geodatensatz auf und hat zu einem Systemabsturz geführt, der sofort behoben wurde. Es liegt dementsprechend keine protokollierte Mail vor.

Eine automatische Korrektur dieses Umstandes ist aufwendig. Eine Möglichkeit ist, dass der Geodatensatz in die Coverage-Struktur übergeführt wird und dann nochmals in eine

Shape-Datei rücktransformiert wird. Dabei entstehen viele zusätzliche Spalten, die gelöscht werden müssen.

#### 4.3.5. Groß- und Kleinschreibung in der Ebenenbezeichnung

Die Norm sieht vor, dass alle Ebenen mit Kleinbuchstaben bezeichnet werden.

Fehlermeldung: \*\*Fehler - Ebenenname Großbuchstaben Kleinbuchstaben: ist  
Muster.shp / soll muster.shp  
Anzahl der Lieferungen mit diesem Fehler (FWP): 5  
Anzahl des Auftretens dieses Fehlers (FWP):60

Dieser Fehler kann einfach behoben werden. Es kann davon ausgegangen werden, dass die Ebene mit entsprechenden Kleinbuchstaben gemeint war. Da es sich um einen häufigen Fehler handelt, kann man mit einer automatischen Korrektur dem Lieferanten deutlich Arbeit ersparen.

#### 4.3.6. Fehlerhafter Geometrietyp

Für jede Ebene ist ein Geometrietyp vorgesehen. Hat die Geometriespalte eine andere Definition, kommt es zu einer Fehlermeldung.

Fehlermeldung: \*\*Fehler – Prüfung Geometrietyp: Muster.shp  
Anzahl der Lieferungen mit diesem Fehler (FWP): 17  
Anzahl des Auftretens dieses Fehlers (FWP): 21

14-mal wurde die Struktur *Multipoint* geliefert, wo der Geometrietyp *Point* vorgesehen war. Beim *Multipoint* handelt es sich um eine Zusammenfassung mehrerer räumlich verschiedener Punkte zu einem Geometrieelement, welches durch einen einzigen beschreibenden Satz von Attributen näher bestimmt wird. Auf diese Weise wurden Punkte mit gleicher Attribuierung zusammengefasst. Eine solche Zusammenfassung kann mit einer Standardfunktion des Geoprocessors von ArcGIS in die geforderte Struktur „Point“ übergeführt werden, wo jeder Einzelpunkt einen eigenen Satz an Attributen hat. Eine Behebung dieses Fehlers ist leicht möglich und kann automatisiert werden.

Die Ebene `ersl_wawi_f` muss laut veröffentlichter Schnittstelle als Polygonebene geliefert werden. Durch einen Fehler in einer Steuerdatei, welche die gültigen Ebenen und ihren Geometrietyt auflistet, wurde diese Ebene 6-mal falsch gerügt. Die Steuerdatei wurde korrigiert.

Einmal wurden die Gefahrenzonen der Wildbach- und Lawinenverbauung als Polylines geliefert. Es sind jedoch Flächen gefordert. Eine automatische Flächenbildung dieser Daten, die ursprünglich aus CAD-Systemen stammen und dort als Linien geführt werden, ist aufgrund von sehr seltenen, aber doch auftretenden Datenfehlern nicht ohne optische Kontrolle möglich.

#### 4.3.7. Leere Ebenen

Die Schnittstelle schreibt vor, dass keine leeren Layer übergeben werden dürfen. In vielen Gemeinden sind nicht für alle Ebenen Objekte vorhanden. Beispiele sind etwa denkmalgeschützte Objekte bzw. Naturschutzgebiete, die nicht in jeder Gemeinde anzutreffen sind. Hier wird geprüft, ob diese Ebenen weggelassen wurden und ob irrtümlich leere Datensätze übermittelt wurden.

Fehlermeldung: **\*\*Fehler – Prüfung Anzahl Elemente: Es sind keine Objekte in dieser Ebene vorhanden!**

Anzahl der Lieferungen mit diesem Fehler (FWP): 26

Anzahl des Auftretens dieses Fehlers (FWP): 308

Um es den Planern möglichst einfach zu machen, korrekte Daten abzugeben, werden vorgefertigte leere Ebenen bereitgestellt, die dann vom Planer auszufüllen sind. Der Plan wird schrittweise erstellt, indem ein Teil der benötigten Ebenen ausgefüllt wird. Danach wird die Arbeit gegenüber dem Portal geprüft. Alle Ebenen werden abgeschickt. Diese Hypothese erklärt die hohe Anzahl, mit welcher dieser Fehler auftritt. Die Annahme wird durch die Tatsache gestärkt, dass auch Ebenen betroffen sind, welche den zentralen Kern eines FWP ausmachen (Nutzungen, Kataster).

Eine Auflistung der betroffenen Ebenen ist aus dem oben beschriebenen Grund nicht zweckdienlich, da lediglich die Reihenfolge der bearbeiteten Ebenen dadurch beschrieben wird.

Nichtsdestotrotz wird evident, dass die Vollständigkeit der übermittelten Geodaten noch zu prüfen ist. Genaueres wird bei den Qualitätskriterien von Geodaten gesagt (siehe Kapitel 5.1).

#### 4.3.8. Objekte außerhalb der Gemeinde

Hier wird überprüft, ob gelieferte Objekte oder Objektteile außerhalb eines Rechtecks liegen, welches die Gemeinde umschreibt. Dabei wird der Extend jedes einzelnen Objektes herangezogen und mit dem Extend der Gemeinde verglichen. Der kritische Leser wird sich fragen, warum nicht der Extend des gesamten Layers gegenüber dem Extend der Gemeinde geprüft wurde. Jedes Objekt einzeln zu prüfen, kostet Rechenzeit. Die Praxis zeigt, dass der Extend des Layers oftmals unreal groß ist.

Mit dieser Prüfung wird neben Überständen einzelner Objekte auch kontrolliert, ob die richtige Gemeindenummer vorgegeben wurde und die Daten im korrekten Koordinatensystem geliefert wurden.

Man könnte bei dieser Kontrolle das die Gemeinde umschreibende Rechteck durch dessen konvexe Hülle ersetzen. Die Funktion „Selection by Location“ wählt alle Objekte innerhalb der Gemeinde aus. Entspricht die Anzahl der ausgewählten Objekte nicht der Gesamtzahl an Objekten, liegen Features außerhalb der Gemeinde. Diese Prüfung hat den Vorteil, dass sie genauer arbeitet, da nicht mehr umschreibende Rechtecke verglichen werden, sondern mit den exakten Geometrien gearbeitet wird. Alle Objekte einer Ebene werden auf einmal geprüft. Man wendet Standardfunktion des ESRI-Geoprocessors an, anstatt eine Schleife über alle Objekte einer Ebene zu rechnen, wodurch man einen Geschwindigkeitsvorteil hat. Diese Gründe sprechen für eine geänderte Vorgehensweise, was in der Neuentwicklung auch erfolgte.

Fehlermeldung: \*\*Fehler – Lageprüfung: Elemente außerhalb der Gemeinde gefunden!

Anzahl der Lieferungen mit diesem Fehler (FWP): 48

Anzahl des Auftretens dieses Fehlers (FWP): 132

Beim Upload ist anzugeben, um welche Gemeinde es sich handelt. Wenn man hier eine falsche Auswahl trifft, sind im Allgemeinen alle Ebenen außerhalb des umschreibenden Rechtecks. Dieser Umstand tritt auch ein, wenn die Planung in einem falschen Koordinatensystem erfolgt. Das war nur zweimal der Fall. In allen anderen Fällen, bei denen diese Prüfung negativ ausfiel, lagen einzelne Objekte außerhalb der Gemeinde, wobei mit einer Toleranz von 50 m gearbeitet wurde.

Im Rahmen einer automatischen Korrektur könnten alle Objektteile, die über die Gemeinde hinausreichen, abgeschnitten werden. Das kann mit der Standardfunktion Clip erfolgen und setzt voraus, dass Grenzänderungen der Gemeinden penibel nachgeführt werden. Da jede Gemeinde aus einer Summe von Katastralgemeinden besteht und damit Gemeindegrenzänderungen auch Änderungen von Katastralgemeinden darstellen, werden diese im Amtsblatt für Vermessungswesen bekannt gegeben.

Hat ein Geodatensatz ein Element ohne Ausdehnung, wird neben der entsprechenden Fehlermeldung auch die Meldung ausgegeben, dass ein Fehler aufgrund der Lageprüfung entdeckt wurde. Doppelte Meldungen von Fehlern sollten bei der Neuentwicklung beseitigt werden.

Da Objekte ohne Geometrie in allen beobachteten Fällen ohne Bedeutung waren, können diese Fehler auch automatisch bereinigt werden, was mit einer Standardfunktion des Geoprocessors möglich ist (`arcpy.RepairGeometry_management`).

Wurde die Gemeindenummer falsch vorgegeben, kann sie durch einen Verschnitt der Lieferung mit der Ebene der Gemeinden ermittelt werden. Zuerst wird dafür aus der flächenfüllenden Ebene `nutz_f` des FWPs der geometrische Mittelpunkt ermittelt. Dazu werden alle Objekte zusammengeführt (`dissolve`) und aus dem Ergebnis wird das Centroid inside abgeleitet. Dieser Zentralpunkt kann mit den Gemeinden verschnitten werden (`join by location`), woraus sich die korrekte Gemeindenummer ergibt. Mit dem Centroid zu arbeiten,

bringt den Vorteil mit sich, dass kleine Ungenauigkeiten im Randbereich, welche hier überprüft werden, keine Rolle spielen. Aufgrund des geringen Auftretens des Fehlers ist diese Korrekturmöglichkeit nicht dringend notwendig.

Ein fehlerhaftes Koordinatensystem kann ähnlich korrigiert werden. Zuerst erstellt man sich eine Ebene, die keine Projektionsdefinition besitzt und welche die Gemeinden der Steiermark in allen relevanten Koordinatensystemen enthält. Jede Gemeinde ist dort also mehrfach mit ihren unterschiedlichen Koordinatenwerten der einzelnen Systeme gespeichert. Der Datensatz speichert neben der Gemeindenummer auch noch das Koordinatensystem als Attribut. Diese Ebene wird als Systemdatensatz vorgehalten und muss nicht im Bedarfsfall erstellt werden, was Rechenzeit spart.

Im Anwendungsfall wird dieser Geodatensatz auf die benötigte Gemeinde mit einer SQL-Abfrage eingeschränkt, sodass die Ebene lediglich aus einer Gemeinde in allen relevanten Systemen besteht. Die Projektionsinformation der gelieferten Daten wird gelöscht, um die „Projection on the Fly“ auszuschalten.<sup>68</sup> In analoger Weise zur vorherigen Korrektur wird der Zentralpunkt der Lieferung ermittelt und ein Verschnitt mit der Gemeindeebene liefert das Koordinatensystem. Auch hier gilt, dass diese Korrektur aufgrund der Seltenheit des Fehlers nicht zwingend notwendig ist.

#### 4.3.9. Nicht definierte Ebenen

Werden Ebenen geliefert, welche nicht in der Liste der gültigen Layer aufscheinen, wird eine Fehlermeldung abgesetzt.

---

<sup>68</sup>

Löschen der prj-Dateien

Fehlermeldung: \*\* Fehler: Es liegen nicht definierte Ebenen vor!

Anzahl der Lieferungen mit diesem Fehler (FWP): 17

Anzahl des Auftretens dieses Fehlers (FWP): 28

Tabelle 13: Fehlerhafte Layerbezeichnungen

Fehlerhafte Ebene	Richtige Bezeichnung	Anzahl des Auftretens	Kommentar
aufbp.shp bew.shp ewgr.shp immi.shp im- mip.shp ssp.shp vinf.shp vinfp.shp vorr.shp	–	1	Eine als FWP deklarierte Abgabe enthielt einen OEP, weshalb 9 Fehler ausgewiesen wurden.
ersl_imm_p.shp	ersl_imm_f.shp, ersl_imm_l.shp	7	In den Ebene ersl_imm_f werden flächenhafte Immissionen und in der Ebene ersl_imm_l linienhafte Darstellungen von Immissionen abgelegt. Punktförmige Immissionen, wie durch die Änderung _p angedeutet, sind nicht vorgesehen.
erls_natur_l.shp	ersl_natur_l.shp	2	Zwei benachbarte Buchstaben wurden vertauscht.
ersl_versorf.shp	ersl_versorg.shp	2	Ein Buchstabe ist falsch. Auf der Schreibmaschinentastatur liegen sie neben einander (f, g).
DKM_GST.shp DKM_NUTZ_FL.shp	kat_grundstück.shp kat_nutzung.shp	1	Hier wurden gänzlich verschiedene Bezeichnungen verwendet. Einem erfahrenen Bearbeiter ist jedoch sofort klar, welche Ebenen gemeint sind. Die Abkürzung DKM steht im österreichischen Geodatenumfeld für „Digitale Katastralmappe“. GST kürzt Grundstück ab und NUTZ_FL steht für Nutzungen in Flächenstruktur. Es handelt sich um die Namen, mit denen die Daten vom BEV abgegeben werden.
ersl_verk_f.shp	ersl_verkehr_f.shp	1	In der Ebenen nutz_f werden Verkehrsflächen mit Verk abgekürzt, wodurch dieser Fehler verständlicher wird. Die Ebene enthält Eisenbahnstrecken und Straßen bzw. Bahnprojekte.
ers_gew_l.shp	ersl_gew_l.shp	1	Ein Buchstabe wurde weggelassen.
ersl_immi_l.shp	ersl_imm_l.shp	1	Hier ist ein Buchstabe zu viel.
???.shp	?	1	Hier konnte nicht herausgefunden werden, welche Ebene gemeint war.
ersl_bila_z.shp	ersl_bila_f.shp	1	Flächenhafte Überlagerungsebenen enden immer mit einem _f. Hier liegt ein falscher Buchstabe vor.

Automatische Fehlerkorrekturen der Namen scheinen möglich, wenn durch geringfügige Manipulationen ein gültiger Name erzeugt werden kann, wie etwa durch

- das Vertauschen zweier benachbarter Buchstaben,
- das Löschen eines Buchstaben,
- den Austausch eines Buchstaben.

Da der letzte Buchstabe im Ebenennamen den Geometrietyt beschreibt (f..Fläche, l..Linie, p..Punkt), sollte bei einem Fehler auch geprüft werden, welcher Geometrietyt tatsächlich vorliegt, was eine zusätzliche Absicherung darstellt.

Darüber hinaus kann durch Auslesen der Attribute gesagt werden, wie die Ebene heißen müsste. Damit wären auch völlig falsche Benennungen auflösbar. Es wird jedoch vorausgesetzt, dass die Einträge in den Spalten zu einem hohen Maß korrekt sind und in den verschiedenen Ebenen in einem hohen Ausmaß unterschiedliche zulässige Einträge gefordert werden. Eine geringe Unschärfe ist tolerierbar, wenn viele Einträge in der Ebene vorliegen. Weisen 99 % der Objekte auf den Layer A hin und auf den Namen B lediglich 2 %, kann mit hoher Wahrscheinlichkeit angenommen werden, dass es sich um die Ebene A handelt. Die Summe der Prozentsätze muss nicht 100 ergeben, da gleiche Attributeinträge in verschiedenen Ebenen möglich sind. Der Geometrietyt ist zur Prüfung der Korrektur ebenfalls heranzuziehen. Die Spaltennamen auszuwerten, ist im Fall des FWPs nicht ergiebig, da die Attributnamen nahezu aller Ebenen gleich festgelegt wurden.

Wird die Planart beim Upload falsch vorgegeben, werden Ebenenbezeichnungen einer anderen Schnittstelle zur Bewertung herangezogen. Das System prüft z. B. auf die Namen des OEPs anstatt auf Namen des FWPs. Die definierten Layerbezeichnungen der verschiedenen Pläne unterscheiden sich. Nur in Ausnahmefällen haben verschiedene Planarten gleiche Layernamen.

Die fehlerhafte Vorgabe der Planart kann angenommen werden, wenn ein Großteil (z. B. die Hälfte) der Layernamen falsch ist. Eine automatische Bereinigung ist denkbar. Das erfolgt, indem als erster Schritt die Ebenenbezeichnungen aller realisierten Schnittstellen

(OEP, FWP, Beleuchtungskörper) eingelesen werden. Danach wird abgeglichen, mit welcher Planart die Ebenennamen am besten übereinstimmen. Mit der neu festgelegten Planart kann weitergearbeitet werden.

Wurden fehlerhafte Bezeichnungen der Ebenen gerügt, hat das niemals zu Nachfragen von Planern geführt. Die Fehlermeldung und ihre Beseitigung machen offensichtlich kein Problem im Unterschied zu Spaltendefinitionen, weshalb Planer Kontakt mit der Behörde aufnehmen. Die automatische Korrektur ist hilfreich, jedoch nicht essenziell.

#### 4.3.10. Keine zu prüfenden Daten vorhanden

Es wird geprüft, ob Geodatenätze überhaupt vorhanden sind.

Fehlermeldung: \*\*Fehler: Es liegen keine zu prüfenden Daten mehr vor, nachdem fehlerhafte Ebenen gelöscht wurden!

Anzahl der Lieferungen mit diesem Fehler (FWP): 0

Diese Prüfung wurde eingeführt, um das Portal vor Lieferungen etwa von Musikdateien und Ähnlichem zu schützen. Liegen keine Geodaten vor, ist die einzige sinnvolle Reaktion, die Software zu stoppen und eine entsprechende Meldung abzusetzen. Reparaturmöglichkeiten sind nicht gegeben.

#### 4.3.11. Nicht erlaubter Attributtyp

Hier wird geprüft, ob die Definition eines vorgegebenen Attributes stimmt. Innerhalb des Beobachtungszeitraums kam es zu einer Abänderung dieser Prüfung. War die Spaltendefinition zu Beginn strikt einzuhalten, wurde die Feldlänge von Textfelder später als nicht mehr relevant angesehen, denn nach der Verifikation des Attributtyps erfolgt die Kontrolle des Feldinhaltes. Wie lange Felder definiert sind ist ohne Bedeutung, wenn die Feldinhalte, um den es eigentlich geht, korrekt sind. Wichtig ist lediglich, dass der Inhalt stimmt. Analoges gilt für numerische Felder. Sowohl die Feldtypen Float als auch Double<sup>69</sup> können die

---

<sup>69</sup> Float: Kommazahlen mit einem Bit für das Vorzeichen, 7 Bit für die Hochzahl und 24 Bit für die Mantisse

praktisch auftretenden Kommazahlen aufnehmen, weshalb anstatt des Datentyps Float auch Double geliefert werden kann und umgekehrt. In numerischen Feldern werden die Bebauungsdichten gespeichert, die Werte von 0,2 bis 2,5 annehmen können (siehe steirische Bebauungsdichteverordnung 1993 LGBl. Nr. 38/1993 geändert mit LGBL. Nr. 87/1994, LGBl. Nr. 61/2003, LGBl. Nr. 58/2011). Analoges gilt für Ganzzahlen, wofür die Typen Short Integer und Long Integer synonym füreinander verwendet werden können.

Fehlermeldung: **\*\*falscher Attributtyp Attributname**

Anzahl der Lieferungen mit diesem Fehler (FWP): 97

Anzahl des Auftretens dieses Fehlers (FWP): 130

Fehlerhaft definiert wurden die Spalten KG (45-mal), MAXH (48-mal), ZSW (29-mal) und ZSW1 (8-mal).

Die beanstandeten Spalten KG, welche für Katastralgemeindenummern vorgesehen sind, wurden mit Feldtypen für reelle Zahlen definiert (Float, Double). Da es sich bei Katastralgemeindenummern um natürliche Zahlen handelt, sind nur die Feldtypen Small Integer bzw. Long Integer zulässig.

Die Spalte MAXH macht am meisten Probleme. Sie kommt in der Ebene nutz\_f vor und beschreibt die maximal zulässige Bauhöhe in ganzen Metern und ist dementsprechend als Small Integer definiert. Damit kann die Spalte Zahlen bis ca. 32000 aufnehmen. Diese Spalte steht in einer sinnhaften Einheit mit der maximalen und minimalen Bebauungsdichte, wobei die Bebauungsdichte das Verhältnis zwischen der Summe der Geschosflächen zur Bauplatzfläche angibt. Möglich sind laut Bebauungsdichteverordnung Werte von 0,2 bis 2,5. Es liegen also reelle Zahlen vor, welche in dem Zahlentyp Float abgelegt werden. Für die Höhe wird deshalb auch gerne dieser Typ verwendet.

Die Zusatzwidmungen ZSW bzw. ZSW1 sind für Texte vorgesehen. In wenigen Ausnahmefällen enthalten die Zusatzwidmungen auch Reihenfolgen, welche in Form von Ziffern

---

Double: Kommazahlen mit einem Bit für das Vorzeichen, sieben Bit für die Hochzahl und 56 Bit für die Mantisse

Small Integer: ca. -32.000 bis +32.000

Long Integer: ca. -1.000.000.000 bis +1.000.000.000

[96]

vorzugeben sind. In den beanstandeten Fällen handelte es sich um solche Fälle. Der Datenlieferant hat irrtümlich ein Zahlenformat gewählt, und nicht, wie in der Schnittstelle gefordert, ein Textfeld festgelegt. Automatische Korrekturen sind möglich.

Werden Textfelder gefordert, aber Zahlenfelder geliefert könnten die Ziffern in ein Textfeld konvertiert werden. Die nachfolgende Prüfung der Feldinhalte würde ausreichen.

Im Bereich der maximalen Bauhöhe wurden Ganzzahlen gefordert und Kommazahlen geliefert. Die gelieferten Werte entsprachen jedoch im Wert Ganzzahlen (z. B. 8,0), sodass die gelieferten reellen Zahlen in natürliche Zahlen ohne Rest überführt werden können. Der Transformationsalgorithmus hat den Rest zu prüfen, und nur Überführungen, welche einen Rest von Null bzw. in Größenordnungen der Rechenunschärfen aufweisen, sind zu akzeptieren. Damit sind ungewollte Manipulationen ausgeschlossen.

#### 4.3.12. Nicht erlaubte Attributbezeichnungen

Es wird überprüft, ob die Attributbezeichnungen korrekt vorliegen.

Fehlermeldung: \*\*falsches Attribut Attributname

Anzahl der Lieferungen mit diesem Fehler (FWP): 50

Anzahl des Auftretens dieses Fehlers (FWP): 601

Folgende Attribute wurden als Falsch erkannt:

Tabelle 14: Fehlerhafte Spaltennamen

Spalte	Anzahl	Spalte	Anzahl	Spalte	Anzahl
AREA	1	BEZ	1	BLOCK	1
DATUM	1	DREHUNG	1	FeatId1	3
FI	1	FLAECHE	1	Geometry1_	1
GEOMETRY1_	1	GSTNR	1	ID	4

ID1	62	KG	6	LAYER	1
FID	114	MAXD	21	NUMMER	6
Nummer	1	NUTZ_ART	2	NUTZART	1
NUTZU_ART	1	NUTZUNG	2	OBJECTID	13
OBJECTID_1	8	ORIG_FID	6	PERIMETER	1
Shape	114	SHAPE_AREA	2	Shape_Area	5
SHAPE_LE_1	11	SHAPE LENG	2	Shape_Leng	10
SpatialAre	2	STATUS	1	WDIMUNG	1
WIDMUNG	105	X_KOORD	1	X_SKAL	1
Y_KOORD	1	Y_SKAL	1	ZSW	74
ZSW_BP	1	ZSW_UO	1	ZSW1	3
ZSW2	3				

Betrachtet man die falschen Attributnamen, erkennt man, dass sie häufig Reste von anderen Dateiformaten sind. Das File Geodatabase, wie auch die Personal Geodatabase von ESRI, legt die Standardspalten Shape\_area und Shape\_length an, die nach einer Überführung in das Shape-Format als Attribute erhalten bleiben. Häufig handelt es sich auch um Schlüsselfelder, die vom Lieferanten nicht gelöscht wurden (OBJECTID\_1, ORIG\_FID, ID1 ...), oder es sind Hilfsspalten, welche zur Datenerstellung oder bei der Analyse notwendig waren (X\_KOORD). All diese Felder können ohne Beeinträchtigung des zu übergebenden Inhaltes gelöscht werden.

Tippfehler in Spaltenbezeichnungen treten vergleichsweise selten auf (0,8 %), woraus sich eine Strategie zur Bereinigung des Fehlers ergibt (WDIMUNG anstatt WIDMUNG, NUTZU\_ART NUTZUNG NUTZART anstatt NUTZ\_ART).

Man kontrolliert auf Vollständigkeit der notwendigen Attribute. Ist diese gegeben, werden weitere zusätzliche Spalten automatisch gelöscht. Sind nicht alle notwendigen Spalten vorhanden, können überzählige Spalten auf Schreibfehler wie Buchstabenstürze, Synonyme

etc. untersucht werden. War diese Strategie erfolgreich, kann wie zuvor vorgegangen werden.

Alternativ dazu kann auch über den Spalteninhalt gearbeitet werden, der Aufschluss gibt, welche Spalte gemeint sein könnte. Das erfolgt, indem eine Schleife über alle definierten Spalten exekutiert wird. Der unbekanntes Spalte wird nun unterstellt, die gerade abgearbeitete definierte Spalte zu sein. Es erfolgt eine Spalteninhaltsprüfung. Liefert diese keinen Fehler, ist mit sehr hoher Wahrscheinlichkeit die richtige Spaltenbezeichnung gefunden.

#### 4.3.13. Fehlende Spalten

Es wird kontrolliert, ob alle geforderten Spalten vorhanden sind.

Anzahl der Lieferungen mit diesem Fehler (FWP): 9

Anzahl des Auftretens dieses Fehlers (FWP): 10

Der Fehler trat sehr sporadisch und nur bei einer Lieferung zweimal auf. Tabelle 4 beschreibt die fehlenden Spalten. Die zulässige maximale Gebäudehöhe wurde dreimal weggelassen, was wahrscheinlich daran liegt, dass sie mit der Schnittstelle 2007 eingeführt wurde. Ältere Schnittstellenversionen kennen diese Festlegung nicht und ein Planer, der nur sehr selten FWPs erstellt, arbeitet noch nach den alten Vorgaben. Die übrigen Fehler sind auf fehlerhafte Attributbenennungen zurückzuführen. Eine automatisierte Korrektur, wie im vorherigen Abschnitt erläutert, würde in diesen Fällen die Meldung obsolet werden lassen.

Tabelle 15: Fehlende Spalten

Spalte	Auftreten
MAXH	3
NUTZ_ART	3
NUMMER	2
WIDMUNG	2

#### 4.3.14. Falsche Attributwerte

Auf fehlerhafte Attributinhalt wurden nur Textfelder geprüft. Die Inhalte der maximalen Bebauungsdichte und der minimalen Bebauungsdichte sowie die maximale Gebäudehöhe waren nicht Gegenstand der Prüfung. Entsprechende Prüfungen sind noch zu implementieren. Die Grenzwerte, zwischen denen sich die maximale und die minimale Bebauungsdichte bewegen müssen, hängen von der Baulandkategorie ab und wurden erstmals in der steirischen Bebauungsdichteverordnung 1993 LGBl. Nr. 38/1993 festgelegt und vielfach novelliert.

War die Ebenenbezeichnung oder die Spaltenbezeichnung falsch, wurden alle Attribute als unzulässig ausgewiesen. Es handelte sich also um Folgefehler, was beim Auswerten der Mails zu berücksichtigen war. Insgesamt traf das für 44 % der Attributfehler zu. Die entwickelte Auswertesoftware könnte das berücksichtigen, indem Auflistungen mit falsch bezeichnete Ebenen und Spalten aus der Prüfung der Attributwerte ausgeschlossen werden (ausgewiesene Fehler: 19423; tatsächliche Fehler: 10923; Folgefehler: 8500).

Die Schnittstelle erlaubt weder führende noch nachfolgende Leerzeichen bei den einzelnen Attributwerten. Dieser Fehler wäre leicht automatisch zu bereinigen, indem alle diese Blanks gelöscht werden. Mit einer solchen automatischen Korrektur würde es auch nicht zu Sinnveränderungen kommen, denn ein unsichtbares Leerzeichen wird nirgends als Unterscheidungsmerkmal herangezogen. Entgegen meine Vermutung, dass es sich um eine häufige Fehlerursache handelt, waren nur 25 Fehler oder 0,2 % auf diese Ursache zurückzuführen.

Häufig werden Spalten, die in Kombination mit anderen Werten keinen Eintrag haben sollen, mit einem Blank ausgefüllt, was ebenfalls nicht zulässig ist und optisch für den Planlieferanten schwer erkennbar ist. Diese, dem obigen Fall sehr ähnliche, Problemfälle traten 2174-mal auf, was ca. 20 % ausmacht. Dieser Fehlerfall wird extra angeführt und nicht zum obigen Fall dazu gezählt, weil dieses überflüssige Leerzeichen bei der Korrektur der Geodaten speziell behandelt werden muss. So wird ein einziges alleinstehendes Blank nicht durch die Funktion Trim von VBA entfernt. Andere Sprachen haben ähnliche Probleme (z. B. Python).

Um es den Planern leichter zu machen, wurde die Schnittstelle erweitert, sodass nun auch ein Leerzeichen als gültiger Wert für keinen Eintrag akzeptiert wird.

Bei den verbleibenden 10.707 Fehlern kam es zu Häufungen gewisser Fehler. Das augenfälligste Beispiel sind die Verkehrsflächen in der Ebenen nutz\_f. Eine solche Fläche wird definiert, indem in der Spalte Widmung der Wert „VERK“ eingetragen wird. 1.925-mal wurde die Widmung jedoch mit beginnenden Großbuchstaben und folgenden Kleinbuchstaben geschrieben („Verk“). Insgesamt lagen 830 verschiedene Arten von Fehleintragungen vor.

Ein FWP enthält viele Verkehrsflächen und somit kann der Fehler häufig auftreten. Fehlerursache dürfte auch sein, dass diese Schreibweise in der Vorgängerschnittstelle gefordert war. Mit der Überarbeitung der Schnittstelle wurde die Groß- und Kleinschreibung konsequent für eine sinngebende Kategorisierung verwendet. Im Kapitel „Verbesserung der Struktur-Logik sowie der Schreibweise der Textaussagen“ der Anlage 2 der Planzeichenverordnung 2007 wird festgehalten:

„Großbuchstaben bei den Baulandausweisungen, Verkehrsflächen sowie beim land- und forstwirtschaftlichen Freiland bzw. bei der Ödlandnutzung; bei Vorbehaltsflächen, wichtigen überörtlichen Nutzungen, Hochwasserschutz und militärischen Anlagen sowie sämtlichen Nutzungsbeschränkungen, bei den Ver- und Entsorgungsbereichen für Wasser, Abwasser und Energie.

Kleinbuchstaben bei sämtlichen Freilandsondernutzungen und bei der Baulandzonierung im Ergänzungsplan.“ [1]

Natürlich sind auch andere Widmungen als Verkehrsflächen von der Problematik der „Case sensitivity“ betroffen.

Man sieht an diesem Beispiel das große Potenzial, das in einer automatischen Fehlerkorrektur steckt. Es ist klar, welche Widmung der Planer festlegen wollte, da keine andere Widmung mit dieser Buchstabenkombination beschreiben wird. Korrigiert man die Groß- und Kleinschreibung, hat man mehr als 3000 Fehler automatisiert bereinigt (siehe Tabelle 5).

Deshalb wurden die 830 verschiedenartigen falschen Eintragungen händisch untersucht, mit dem Ziel, herauszufinden, was die Planer gemeint haben. Dabei wurde möglichst konservativ vorgegangen. Eine Korrekturmöglichkeit wurde nur ausgewiesen, wenn diese eindeutig war. Um die Komplexität zu beherrschen, wurde in zwei Schritten vorgegangen. Im ersten Schritt wurde die Korrektur möglichst genau verbal in einer Excel-Datei beschrieben und im zweiten Schritt wurden die Beschreibungen in Klassen zusammengefasst. Nun lässt sich die Fehleranzahl pro Klasse ermitteln.

Tabelle 16: Fehlerklassen der Textattribute

Nr.	Kategorie der Fehlerkorrektur	Anzahl
a	Groß-und Kleinschreibung	3066
b	nicht erruierbar	2776
c	Syntax, wie fehlende Klammern etc.	1481
d	fehlendes Einheitszeichen	868
e	Zahlenfeld -> Textfeld; hier wurde die Spalte falsch definiert und das Attribut als Folge nicht erkannt. Der Attributtyp müsste korrigiert werden.	781
f	Der korrekte Spalteneintrag ist aus der Ebene ableitbar, da nur ein Eintrag in dieser Ebene möglich ist oder aus dem Wert einer anderen Spalte zu erschließen.	644
g	Formular	305
h	Eintrag in einer falschen Spalte	203
i	Regel erläutern	191
j	Schnittstelle	116
k	Vorgesehen; fehlerhafte Definition in der Steuerdatei des Prüfprogramms	98
l	Spalteninhalt nicht benötigt – Namen etc.	73
m	Verdoppelung	41
n	Straße und Straßenummer auftrennen und in die vorgesehen Spalten einordnen	32
o	römische Ziffer in arabische Ziffer umwandeln	9
p	sinnverwandtes Wort	8
q	Umlaut	6
r	Buchstabensturz	4

s	Sonderzeichen	3
t	Leerzeichen löschen	2

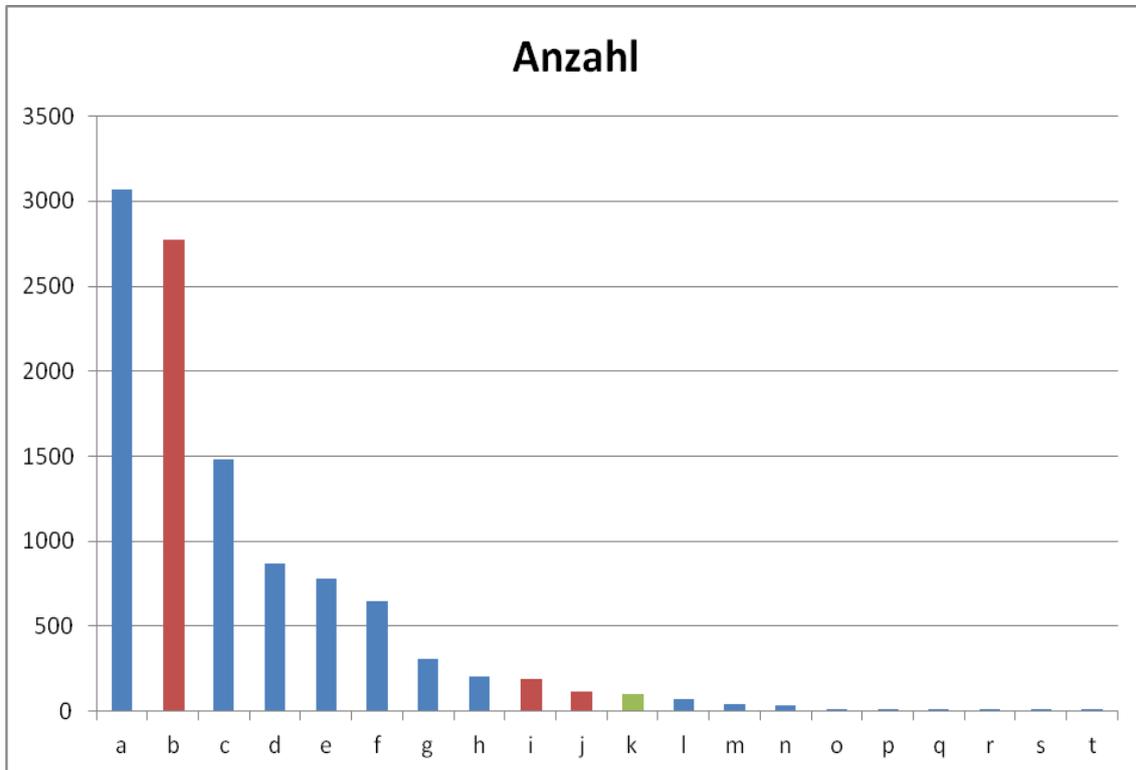


Abbildung 21: Histogramm des Auftretens von diversen Attributfehlern

Tabelle 5 sowie die Abbildungen 3 und 4 verwenden Blau für beseitigbare Fehler und Rot für Kategorien, die nicht berichtigbar sind. Die grüne Farbe steht dort für Fehler in der Schnittstelle, die im Lauf des Systembetriebes durch Raumplaner reklamiert wurden und nun beseitigt sind.

70,94 % der fehlerhaften Inhalte können korrigiert werden.<sup>70</sup>

<sup>70</sup>

Die Fehler in der Schnittstelle wurden aus der Prozentzahl herausgerechnet.

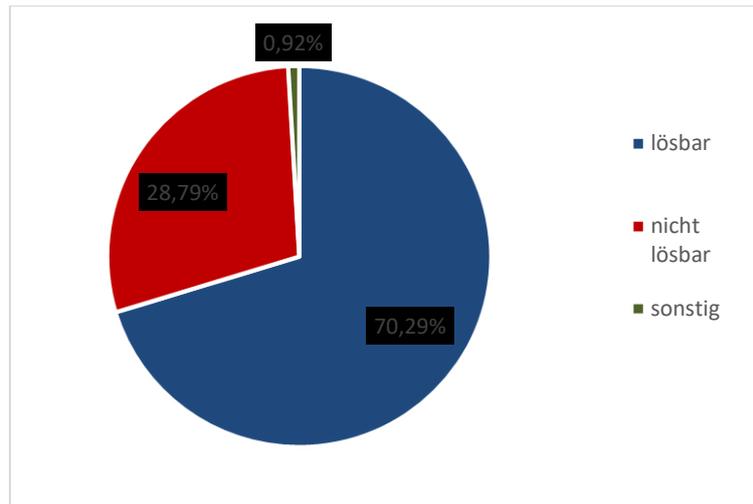


Abbildung 22: Tortendiagramm der Korrigierbarkeit von Attributfehlern

Mit rund 29 % handelt es sich bei der Groß- und Kleinschreibung um die größte Fehlerquelle. Wie am Beispiel VERK zu Beginn des Kapitels bereits ausgeführt, wird vermutet, dass die Umstellung einiger Planzeichen in ihrer Schreibweise Ursache der Fehler ist. Die mit der Planzeichenverordnung 2007 erfolgte Kategorisierung und die damit verbundenen Vereinheitlichungen sollten jedoch gegenüber der früheren willkürlichen Verwendung von Groß- und Kleinbuchstaben ein Vorteil sein. Insbesondere verbessert sich die Lesbarkeit der Pläne, welche mit diesen Abkürzungen beschriftet werden.

Angesichts der großen Häufigkeit der Fehler kann es sinnvoll sein, die Planersteller von der Kategorisierung und der zugeordneten Schreibweise noch einmal zu unterrichten. Das Gesamtdokument [1], welches bereits auf diesen Umstand hinweist, dürfte aufgrund der großen Seitenanzahl nicht zur Gänze gelesen werden. Ein kurzes Informationsschreiben oder ein kleiner Vortrag in Fachkreisen ist häufig effizienter.

Eine automatische Korrektur der Groß- und Kleinschreibung setzt voraus, dass die Schnittstelle keine Buchstabenkombinationen enthält, deren einzige Unterscheidung in diesem Kriterium liegt. Idealerweise sollte eine Eindeutigkeit über alle Ebenen hinweg vorliegen, da sich der Planer in der Zuordnung der Ebenen irren kann.

Mitunter wurde die Syntax nicht richtig vorgegeben. Z. B. werden in der Ebene `beschr_f` Ausschlussbereiche für das Einkaufen mit der Widmung „(EA)“ gekennzeichnet. Eine fehlende schließende Klammer, eine fehlende öffnende Klammer etc. werden als Syntaxfehler

angesehen und lassen sich automatisch bereinigen. Hier sind viele verschiedene Fälle möglich.

Ein Sonderzeichen wie z. B. Ö, ö, Ä, ä, wird je nach eingestellter Code Page unterschiedlich binär gespeichert. Übernimmt man Daten, können aus der binären Zeichenfolge nur dann korrekte Sonderzeichen rekonstruiert werden, wenn man die ursprüngliche Code Page kennt. Sonderzeichen waren ein besonderes Problem, da die zu verwendende Code Page nicht genormt ist. Bei Datenlieferungen des FWP's kann jedoch immer aus den restlichen Zeichen geschlossen werden, welches Sonderzeichen gemeint war.

Neben den verpflichtenden Daten des Shape-Formates gibt es zusätzliche Dateien. Diese wurden vor etwa einem Jahr um die .cpg Datei erweitert, in welcher der verwendete Zeichensatz steht. Das „character encoding“ ist damit problemlos möglich.

Bei der Angabe von Größen fehlen häufig die geforderten Einheitsangaben. Römische Ziffern lassen sich in arabische Ziffern transformieren, Buchstabenstürze können automatisch bereinigt werden. Ein Sonderproblem sind Straßenbezeichnungen. Die Weizerstraße – B 72 ist korrekterweise mit der Widmung B und der Zusatzwidmung 72 zu kodieren. Häufig wird der Fehler begangen, dass die gesamte Straßenbezeichnung in der Spalte der Widmung abgelegt wird (Widmung = B-72), wie es in der Schnittstelle 2003 gefordert war.

Im Bereich der Attributkorrektur sind viele verschiedene Strategien anzuwenden, um auf einen korrekten Geodatensatz zu kommen. Im Kapitel 7.4.3.7.2 wird ein selbstlernendes System vorgestellt, das nahezu alle Fehler bereinigen kann. Eine genaue Analyse aller Attributfehler ist trotzdem nicht obsolet, da das Wissen über die Probleme bei der Planerstellung in den Entwurf einer neuen Schnittstelle einfließen kann.

## Literaturverzeichnis

- [1] Amt der Steiermärkischen Landesregierung: „Raumordnungsinformationssystem Steiermark Planzeichenverordnung 2007“, 1.11.2007. [Online]. Available: [http://www.raumplanung.steiermark.at/cms/dokumente/10882364\\_19834135/2a721adc/PZVO %202007 %20Gesamtdokument\\_.pdf](http://www.raumplanung.steiermark.at/cms/dokumente/10882364_19834135/2a721adc/PZVO%202007%20Gesamtdokument_.pdf). [Zugriff am 15.11.2016]
  - [2] Wikipedia: „Cloud Computing“, 21.10.2015. [Online]. Available: [http://de.wikipedia.org/wiki/Cloud\\_Computing](http://de.wikipedia.org/wiki/Cloud_Computing). [Zugriff am 15.11.2016]
  - [3] T. Noack: „Unterschriebene Pläne – digitale Daten in der Nutzungsplanung: Wie werden Geodaten rechtsgültig?“, 13.7.2006 [Online]. Available: [http://www.vlp-aspan.ch/sites/default/files/expos\\_noack.pdf](http://www.vlp-aspan.ch/sites/default/files/expos_noack.pdf). [Zugriff am 15.11.2016]
  - [4] Wikipedia: „Eintragung“, 8.4.2014. [Online]. Available: <http://de.wikipedia.org/wiki/Eintragung>. [Zugriff am 15.11.2016]
  - [5] A. Lienhard, J. Zumstein: „Kataster der öffentlich rechtlichen Eigentumsbeschränkungen (ÖREB) Vertiefungsstudie zu den rechtlichen Aspekten“, Basel: AG SIK-GIS & swisstopo, 2005.
  - [6] T. Noack: „Unterschriebene Pläne – digitale Daten in der Nutzungsplanung: Wie werden Geodaten rechtsgültig?“, Zürich: ETH - Zürich, 2006.
  - [7] R. Ortner, M. Mitter: „Flächenwidmungsplan – eFWP“, 12.6.2013. [Online]. Available: [http://e-government.adv.at/2013/pdf/32\\_Ortner\\_Mitter-eGovernmentKonferenz20130612.pdf](http://e-government.adv.at/2013/pdf/32_Ortner_Mitter-eGovernmentKonferenz20130612.pdf). [Zugriff am 15.11.2016]
  - [8] T. Noack: „Rechtsgültige Geodaten in der Nutzungsplanung“, 13.7.2006. [Online]. Available: [http://www.vlp-aspan.ch/sites/default/files/expos\\_noack.pdf](http://www.vlp-aspan.ch/sites/default/files/expos_noack.pdf). [Zugriff am 15.11.2016]
  - [9] Bundesamt für Landestopographie swisstopo: „Der Kataster der öffentlichen-rechtlichen Eigentumsbeschränkungen (ÖREB-Kataster)“, 2015. [Online]. Available: <http://www.cadastre.ch/internet/kataster/de/home/services/publication/rdppf12.parsys.93931.downloadList.79683.DownloadFile.tmp/broschuereorebkde.pdf>. [Zugriff am 15.11.2016]
  - [10] D. Spangl: „Notwendigkeit und Möglichkeiten eines Katasters öffentlich-rechtlicher Eigentumsbeschränkungen in Österreich“, 7.12.2011. [Online]. Available: [publik.tuwien.ac.at/files/PubDat\\_206119.pdf](http://publik.tuwien.ac.at/files/PubDat_206119.pdf). [Zugriff am 15.11.2016]
  - [11] o. V.: „ÖREB-Kataster – die ersten Kantone sind online“ cadastre Fachzeitschrift für das schweizerische Katasterwesen No14, S. 20, 2014.
  - [12] J. Ludewig, H. Lichter: „Software Engineering Grundlagen, Menschen, Prozesse, Techniken“, Heidelberg: dpunkt.verlag GmbH, 2013
-

- 
- [13] E. W. Dijkstra: „The Humble Programmer“, 1972. [Online]. Available: [www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF](http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF). [Zugriff am 25.8.2014].
- [14] M. Wetzel: „Ein Experiment zur Ermittlung der Programmierproduktivität von Studenten“, 2007. [Online]. Available: [www.iste.uni-stuttgart.de/fileadmin/user-download/SEUH07\\_Wetzel.pdf](http://www.iste.uni-stuttgart.de/fileadmin/user-download/SEUH07_Wetzel.pdf). [Zugriff am 26 August 2014].
- [15] W. W. Royce: „Managing the Development of Large Software Systems“, 1970. [Online]. Available: <http://www.cs.umd.edu/class/spring2003/cms??c838p/Process/waterfall.pdf>. [Zugriff am 04 09 2014].
- [16] Ö. Bischofskonferenz: „YouCat“, München, Pattloch Verlag GmbH & Co. KG, 2010, S. 181.
- [17] I. Schäfer: „Software Engineering 1 - WS 2010/2011“, 2010. [Online]. Available: <https://www.tu-braunschweig.de/Medien-DB/isf/sse/v1-einleitung.pdf>. [Zugriff am 25 09 2014].
- [18] J. Grabowski: „Vorlesung Softwaretechnik“, 1.4.2007. [Online]. Available: <http://www.swe-old.informatik.uni-goettingen.de/notes/SS2007/grabowski/IV-Prozessmodelle-2-auf-1.pdf>. [Zugriff am 13.10.2014].
- [19] T. Horn: „Vorgehensmodelle zum Softwareentwicklungsprozess“, 2004. [Online]. Available: <http://www.torsten-horn.de/techdocs/sw-dev-process.htm>. [Zugriff am 16.10.2014].
- [20] K. Beck, J. Grenning, R. C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. van Beenekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler, B. Marick: „Manifesto for Agile Software Development“, 2001. [Online]. Available: <http://agilemanifesto.org/>. [Zugriff am 20.10.2014].
- [21] R. C. Martin: „Clean Coder“, München: Addison-Wesley Verlag, 2011.
- [22] S. Middeke: „Die Integration von Akzeptanztests in Agilen Vorgehensmodellen“, 2007. [Online]. Available: [http://swt-www.informatik.uni-hamburg.de/uploads/media/AgileAT\\_Middeke.pdf](http://swt-www.informatik.uni-hamburg.de/uploads/media/AgileAT_Middeke.pdf). [Zugriff am 25.10.2014].
- [23] N. Bartelme: „Geoinformatik Modelle Strukturen Funktionen“, Berlin – Heidelberg: Springer-Verlag, 2005.
- [24] R. Wave: „McCabe Cyclomatic Complexity“, 2012. [Online]. Available: [http://docs.klocwork.com/Insight-10.0/McCabe\\_Cyclomatic\\_Complexity](http://docs.klocwork.com/Insight-10.0/McCabe_Cyclomatic_Complexity). [Zugriff am 15.11.2016]
- [25] M. H. Halstead: „Elements of Software Science“, New York: Elsevier 1977.
- [26] Verifysoft Technology GmbH: „Halstead“, 1.7.2013. [Online]. Available: [http://www.verifysoft.com/de\\_halstead\\_metrics.html](http://www.verifysoft.com/de_halstead_metrics.html). [Zugriff am 4.9. 2015].
- [27] K. Lambertz, X. Cullmann: „Software – Verify Soft“, 2007. [Online]. Available: [www.verifysoft.com/de\\_cmtpp\\_msocoder.pdf](http://www.verifysoft.com/de_cmtpp_msocoder.pdf). [Zugriff am 15.11.2016].
-

- 
- [28] Microsoft Corporation: „Code Metrics Values“, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb385914.aspx>. [Zugriff am 15.11.2016].
- [29] K. Lipinski, H. Lackner, O. Laure, G. Kafka, A. Niemann, E. Raasch, B. Schoon, A. Radonic: „Das große Online Lexikon für Informationstechnologie – McCabe Metrik“, 25.11.2014. [Online]. Available: <http://www.itwissen.info/definition/lexikon/McCabe-McCabe-Metrik.html>. [Zugriff am 15.11.2016]
- [30] H. Balzert: „Lehrbuch der Softwaretechnik Band 2: Softwaremanagement, Software-Qualitätssicherung, Unternehmensmodellierung“, Heidelberg: Spektrum Akademischer Verlag, 1998, S. 257.
- [31] J. Jürjens: „Vorlesung Softwarekonstruktion im Wintersemester 2011/2012“, 2011. [Online]. Available: [https://www-secse.cs.tu-dortmund.de/secse/pages/teaching/ws11-12/swk/vorlesungsfolien/SWK\\_V02\\_Software\\_Engineering.pdf](https://www-secse.cs.tu-dortmund.de/secse/pages/teaching/ws11-12/swk/vorlesungsfolien/SWK_V02_Software_Engineering.pdf). [Zugriff am 15.11.2016]
- [32] T. McCabe: „A Complexity Measure“, IEEE Transaction on Software Engineering, S. 314, 1976.
- [33] H. Steitzer: „Einsatz von CAME-Tools zur Qualitätssicherung in großen Software-Projekten“, Norderstedt: diplom.de, 2000.
- [34] E. Bonabeau, G. Theraulaz: „Virtuelle Ameisen als Software-Agenten“, Spektrum der Wissenschaften 5, S. 72, 2000.
- [35] Wikipedia: „Agent-based model“, 8.12.2014. [Online]. Available: [http://en.wikipedia.org/wiki/Agent-based\\_model](http://en.wikipedia.org/wiki/Agent-based_model). [Zugriff am 11.12.2014].
- [36] K. Lipinski, H. Lackner: „Wissen – Das große Online Lexikon der Informatik“, 10.01.2015. [Online]. Available: <http://www.itwissen.info/definition/lexikon/Antwortzeit-response-time.html>.
- [37] ESRI: „ArcGIS Help 10.1“, 18.6.2012. [Online]. Available: <http://resources.arcgis.com/en/help/main/10.1/index.html#//01m100000005000000>. [Zugriff am 15.11.2016]
- [38] D. Goldberg: „Understanding Coordinate Management in ArcGIS“, 2009. [Online]. Available: <http://resources.esri.com/help/9.3/arcgisengine/dotnet/4716A382-71FA-463d-833D-A9D57277637C.htm>. [Zugriff am 15.11.2016]
- [39] ESRI: „ArcGIS Resouce Center – Grundlagen zu Feature-Classes“, 2010. [Online]. Available: <http://help.arcgis.com/de/arcgisdesktop/10.0/help/index.html#na/003n00000005000000/GUID-69FA8570-7C64-4213-B822-68716CD9E764/>. [Zugriff am 29.01.2015]
- [40] R. Aschauer: „Übungsskriptum Geoinformatik II“, Rottenmann: Universitätszentrum Rottenmann, 2005.
- [41] ESRI: „ArcGIS Hilfe 10.1“, 2012. [Online]. Available: <http://help.arcgis.com/de/arcgisdesktop/10.0/help/index.html#//00170000003v000000>. [Zugriff am 30.01.2015].
- [42] Bundesamt für Eich- u. Vermessungswesen: „Formatbeschreibungen und sonstige Informationen“, 01.6.2014. [Online]. Available: [http://www.bev.gv.at/portal/page?\\_pageid=713,1604439&\\_dad=portal&\\_schema=PORTAL](http://www.bev.gv.at/portal/page?_pageid=713,1604439&_dad=portal&_schema=PORTAL).
-

- 
- [43] M. Winand: „Use the Index, Luke A guide to database performance for developers“, 15.11.2011. [Online]. Available: <http://use-the-index-luke.com/de/sql/where/gleichheit/zusammengesetzte-schluessel>. [Zugriff am 15.11.2015]
- [44] ESRI: “Settings spatial Indexes,” 2009. [Online]. Available: <http://web-help.esri.com/arcgisserver/9.3/java/index.htm#geodatabases/settin-1197445025.htm>. [Zugriff am 17.02.2015]
- [45] ESRI: „How To: Improve the performance of ArcSDE with SQL Server“, 5.05.2016. [Online]. Available: <http://support.esri.com/technical-article/000008548>. [Zugriff am 15.11.2016]
- [46] Microsoft: „Developer Network – Back Up a Transaction Log (SQL Server)“, 5.08.2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ms179478.aspx>. [Zugriff am 15.11.2016]
- [47] G. Tudor: „ArcSDE (Enterprise) Database Administration: Configuration, Storage, Tuning, Monitoring, Management, and Maintenance“, . [Online]. Available: [https://s3.amazonaws.com/webapps.esri.com/esri-proceedings/proc06/papers/papers/pap\\_1183.pdf](https://s3.amazonaws.com/webapps.esri.com/esri-proceedings/proc06/papers/papers/pap_1183.pdf). [Zugriff am 15.11.2016]
- [48] ESRI: „Improving symbol drawing performance“, 16.6.2014. [Online]. Available: <http://resources.arcgis.com/en/help/main/10.2/index.html#//00s600000014000000>. [Zugriff am 15.11.2016]
- [49] ESRI: „Creating Line Symbols“, 2015 [Online]. Available: <http://desktop.arcgis.com/de/arcmap/10.3/map/styles-and-symbols/creating-line-symbols.htm>. [Zugriff am 15.11.2016]
- [50] ESRI: „Optimieren von Kartentext und -beschriftungen für eine verbesserte Performance“, 2015. [Online]. Available: <http://desktop.arcgis.com/de/arcmap/10.3/map/working-with-layers/optimizing-map-text-and-labels-for-performance.htm>. [Zugriff am 15.11.2016]
- [51] Wikipedia: „Lastverteilung (Informatik)“, 4.12.2015. [Online]. Available: [https://de.wikipedia.org/wiki/Lastverteilung\\_\(Informatik\)](https://de.wikipedia.org/wiki/Lastverteilung_(Informatik)). [Zugriff am 15.11.2016]
- [52] EUR-Lex: „Amtsblatt der Europäischen Union - VERORDNUNG (EG) Nr. 976/2009“, 19.10.2009 [Online] Available: <http://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex:32009R0976> [Zugriff am 15.11.2016]
- [53] M. G. Bernhard: „IT Security Management nach ITIL“, in Praxishandbuch IT-Sicherheit, Düsseldorf, Symposion Publishing GmbH, 2005, S. 103.
- [54] H. Rieger, J. Schoolmann: „Wege zu mehr IT-Sicherheit“, in Praxishandbuch IT Sicherheit, Düsseldorf, Symposion Publishing GmbH, 2005, S. 437.
- [55] P. Aschauer: „Whistleblowing und Corporate Governance – der mitbestimmte Börsegang?“, Graz, Dissertation, 2012, SS. 6–9.
- [56] J. Hromkovic: „Sieben Wunder der Informatik“, Wiesbaden: GWV Fachverlag GmbH, 2009.
-

- 
- [57] Center of Computing Technologies – Uni Bremen: „Digitale Signatur, Zertifikate, PKI, Sicherheitsprotokolle“, 2005. [Online]. Available: [www-rn.informatik.uni-bremen.de/lehre/itsec/itsec05-11a.pdf](http://www-rn.informatik.uni-bremen.de/lehre/itsec/itsec05-11a.pdf). [Zugriff am 15.11.2015]
- [58] ESRI: „ESRI Shapefile Technical Description“, 1998. [Online]. Available: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. [Zugriff am 14. 08. 2015].
- [59] R. C. Martin: „Clean Coder“, Addison – Wesley Verlag, 2011, S. 87.
- [60] R. Aschauer, N. Bolter, B. Schößer: „Zoning Plans“, in European User Conference Florenze 7-9.Okt 1998, Graz, 1998.
- [61] T. Fischer: „Risikodiagramm“, 24.8.2015. [Online]. Available: <http://risikomanager.org/methodenassistent/risikodiagramm-risikograph-risikolandschaft-risikoportfolio-risikomatrix/>. [Zugriff am 14. 08. 2015].
- [62] H. Balzert: Lehrbuch der Softwaretechnik Entwurf, Implementierung Installation und Betrieb, Heidelberg: Spektrum Akademischer Verlag Heidelberg, 2011.
- [63] B. Breutmann, R. Burkhardt: Objektorientierte Systeme, München Wien: Carl Hanser Verlag, 1992.
- [64] M. Weigend: Objektorientierte Programmierung in Python, Bonn: mitp-Verlag, 2005.
- [65] E. Stangl, M. Kurzmann: Geoprocessing mit Python, Rottenmann: Seminararbeit, 2006.
- [66] S. Shaw: „What Makes Software Flexible“, 19.6.2012. [Online]. Available: <http://c2.com/cgi/wiki?WhatMakesSoftwareFlexible>. [Zugriff am 15.11.2016]
- [67] M. Hedtstück, R. Reyl: „Tschernobyl – Fakten, Ursachen, Hintergründe“, 2001. [Online]. Available: <http://www.reyl.de/tschernobyl/>. [Zugriff am 13.12. 2014].
- [68] E. W. Dijkstra: „Structured Programming“, 1969. [Online]. Available: <http://www.cs.utexas.edu/users/EWD/transcriptions/EWD02xx/EWD268.html>. [Zugriff am 15.11.2016].
- [69] R. Schmidberger: „Grundlagen des Testens“, 10.5.2016. [Online]. Available: [http://www.iste.uni-stuttgart.de/fileadmin/user\\_upload/iste/se/people/schmidberger/downloads/Grundlagen\\_Testen-110\\_k.pdf](http://www.iste.uni-stuttgart.de/fileadmin/user_upload/iste/se/people/schmidberger/downloads/Grundlagen_Testen-110_k.pdf). [Zugriff am 15.11.2016].
- [70] M. Brucher, M. Moser: „Automatisierte Inhaltliche Kontrolle von Flächenwidmungsplänen“, Rottenmann: Seminararbeit, 2007.
- [71] Bundesamt für Eich- u. Vermessungswesen: „KATASTRALMAPPE DXF Schnittstellenbeschreibung“ 2014. [Online]. Available: [http://www.bev.gv.at/pls/portal/docs/PAGE/BEV\\_PORTAL\\_CONTENT\\_ALL-GEMEIN/0200\\_PRODUKTE/SCHNITTSTELLENBESCHREIBUNGEN/BEV\\_S\\_KA\\_Katastralmappe\\_DXF\\_V2.4.pdf](http://www.bev.gv.at/pls/portal/docs/PAGE/BEV_PORTAL_CONTENT_ALL-GEMEIN/0200_PRODUKTE/SCHNITTSTELLENBESCHREIBUNGEN/BEV_S_KA_Katastralmappe_DXF_V2.4.pdf). [Zugriff am 15.11.2016]
- [72] Amt der Steiermärkischen Landesregierung: „Steirisches Raumordnungsgesetz“ 2015. [Online]. Available: <http://www.raumplanung.steiermark.at/cms/beitrag/10231089/241551/>. [Zugriff am 15.11.2016]
-

- 
- [73] W. Siegert: „expert Praxislexikon Management – Training: 111 Stichwörter für Managementtraining“, Renningen: expert, 2001.
- [74] R. Aschauer: „Die Schwierigkeit eine Norm zu erfüllen“, in AGIT 2014 – Symposium und Fachmesse Angewandte Geoinformatik, Salzburg, 2014.
- [75] N. I. Quazi: „Mastering PyCharm“, Birmingham: Packt Publishing Ltd., 2015.
- [76] ESRI: „GIS Dictionary Sliver Polygon“, 2016. [Online]. Available: <http://support.esri.com/other-resources/gis-dictionary/term/sliver%20polygon>. [Zugriff am 15.11.2016]
- [77] A. Tereshenkov: „Tereshendov’s Blog“, 11.03.2016. [Online]. Available: <https://tereshenkov.wordpress.com/?s=sliver>. [Zugriff am 11.03.2016]
- [78] J. Schuster: „Rechtschreibkorrektur – Probabilistic models of pronunciation and spelling“, 21.11.2002. [Online]. Available: [http://www.cis.uni-muenchen.de/~micha/praesentationen/rechtschreibkorrektur/Rechtschreibkorrektur\\_Home.html](http://www.cis.uni-muenchen.de/~micha/praesentationen/rechtschreibkorrektur/Rechtschreibkorrektur_Home.html). [Zugriff am 16.11.2016]
- [79] T. Kutzner: A. Donaubaue, M. Müller, A. Feichtner, S. Goller, „Erfolgreiche Transformaton von Geodaten nach INSPIRE in der grenzüberschreitenden Region Bodensee“, in zfv – Zeitschrift für Geodäsie, Geoinformation und Landmanagement Nr 2, SS. 103 - 109, 2014.
- [80] G. Zahrer: Historisierung von Geodaten, Graz: Masterarbeit, 2012.
- [81] Amt der Steiermärkischen Landesregierung: „Raumordnungsgesetz“, 9.4. 2016. [Online]. Available: <http://www.raumplanung.steiermark.at/cms/ziel/241551/DE/>. [Zugriff am 16.11.2016]
- [82] I. Farcher: „Ab 1.1. 2015 gibt es 287 steirische Gemeinden“, 30.12.2014. [Online]. Available: <http://www.gemeindestrukturreform.steiermark.at/cms/beitrag/12165076/69389369/>. [Zugriff am 16.11.2016]
- [83] Amt der Stmk LREG: „Übersicht GSR-Konstellationen“, 17.5.2016. [Online]. Available: [http://www.gemeindestrukturreform.steiermark.at/cms/dokumente/11683218\\_114864047/1993499b/2015\\_GSR\\_%20Konstellationen\\_.xls](http://www.gemeindestrukturreform.steiermark.at/cms/dokumente/11683218_114864047/1993499b/2015_GSR_%20Konstellationen_.xls) [Zugriff am 9.4.2016].
- [84] M. Redik, L. Pistotnik: „Der rote Faden der örtlichen Raumplanung im Spannungsfeld zwischen Fortführung und Festlegung neuer Entwicklungsziele - rechtliche und fachliche Ansprüche aus der Sicht der Aufsichtsbehörde“, in Gemeindestrukturreform 2015 – von der Neuorientierung zur konkreten Raumplanung“, Graz: Vortrag, 09.06.2016.
- [85] Axmann geoinformation: „checkgeodata.net“, 15.4.2016. [Online]. Available: [http://www.axmann.at/checkgeodata\\_net.de.111.htm](http://www.axmann.at/checkgeodata_net.de.111.htm). [Zugriff am 16.11.2016]
- [86] ESRI: „What is data Reviewer“, 15.4.2016. [Online]. Available: <http://desktop.arcgis.com/de/desktop/latest/guide-books/extensions/data-reviewer/what-is-data-reviewer.htm>. [Zugriff am 16.11.2016]
- [87] M. Brunner: „Das Land Steiermark: Geodatenprüfung“, in axmann insider Nr 2 , S. 2, 2011.
-

- 
- [88] C. Meier: 25.3.2016. [Online]. Available: <http://www.welt.de/kultur/article153688321/Wie-der-Microsoft-Bot-uns-den-Spiegel-vorhaelt.html>. WeltN24 GmbH, „Weltbild“ [Zugriff am 14.11.2016]
- [89] A. Beckheimer: „Künstliche Intelligenz“, 30.5.2016. [Online]. Available: [http://www.cs.hs-rm.de/~linn/fachsem0910/breki/157609\\_Breckheimer\\_KI\\_Vortrag.pdf](http://www.cs.hs-rm.de/~linn/fachsem0910/breki/157609_Breckheimer_KI_Vortrag.pdf). [Zugriff am 16.11.2016]
- [90] F. Puppe: „Einführung in Expertensysteme“, Heidelberg: Springer Verlag Berlin, 1991.
- [91] G. Greif: „Moderne Aspekte des Wissensverarbeitung“, Graz: Institut für Informationsverarbeitung und Computergestützte neue Medien (IICM), 2000.
- [92] IEEE Institute of Electrical and Electronics Engineers: „IEEE Standard Glossary of Software Engineering Terminology“, 1990.
- [93] J. Grabowski: „Softwaretechnik I“, 1.4.2007. [Online]. Available: <http://www.swe-old.informatik.uni-goettingen.de/notes/SS2007/grabowski/IV-Prozessmodelle-2-auf-1.pdf>. [Zugriff am 13.10.2014].
- [94] „Wikipedia: Unified Process“, 10.9.2014. [Online]. Available: [http://en.wikipedia.org/wiki/Unified\\_Process](http://en.wikipedia.org/wiki/Unified_Process). [Zugriff am 18.11.2014].
- [95] Synergis: „Geodaten erfassen, analysieren und präsentieren“, Wien: Eigenverlag Schulungsunterlagen, 2015.
- [96] M. Pratt: „Adding Fields to a Shape Attribute Table in ArcGIS“, in ArcUser Nr 4, S. 44, 2002.
- [97] Amt der Tiroler Landesregierung: „eFWP Ein Dienst des Landes Tirol“, Available: <https://portal.tirol.gv.at/eFWPWeb/public/verordnungsplan.xhtml?cid=493> [Zugriff am 2.10.2016]
- [98] G. Booch, I. Jacobson, J. Rumbaugh: „The unified software development process“, Indianapolis: Addison-Wesley Professional, 1999.
- [99] ESRI: „ArcGIS Caches“, 10.9.2014. [Online]. Available: <http://resources.arcgis.com/de/help/main/10.1/index.html#/na/006600000181000000/> [Zugriff am 18.11.2014].
- [100] K. Lipinski, H. Lackner, O. P. Laué, G. Kafka, A. Niemann, E. Raasch, B. Schoon, A. Radonic: „ITWissen Digitale Signatur“, 7.11.2016. [Online]. Available: <http://www.itwissen.info/definition/lexikon/Digitale-Signatur-DSig-digital-signature.html> [Zugriff am 16.11.2016].
- [101] W. Riggert: „Digitale Signatur“, Available: <http://www.wi.fh-flensburg.de/fileadmin/dozenten/Riggert/bildmaterial/Dokumentenmanagement/4-Deliver-Digitale-Signatur.pdf>. [Zugriff am 7.11.2016]
-

- 
- [102] ESRI: „Integrate“, 26.08.2014. [Online]. Available: <https://pro.arcgis.com/de/pro-app/tool-reference/data-management/integrate.htm> [Zugriff am 8.11.2016]
- [103] Wikipedia: „Cupertino-Effekt“, 26.02.2015. [Online]. Available: <https://de.wikipedia.org/wiki/Cupertino-Effekt> [Zugriff am 31.12.2016]
- [104] P. Staub: „Über das Potenzial und die Grenzen der semantischen Interoperabilität von Geodaten“, 2009. [Online]. Available: <http://e-collection.library.ethz.ch/eserv/eth:41507/eth-41507-02.pdf> [Zugriff am 14.1.2017]
- [105] European Commission: „D2.8.I.6 Data Specification on Cadastral Parcels – Technical Guidelines“, 17.4.2014. [Online]. Available: <http://inspire.ec.europa.eu/id/document/tg/cp> [Zugriff am 14.1.2017]
- [106] Europäisches Parlament: „Erste EU-weite "Robotergesetze": Interview mit Mady Delvaux“, 12.1.2017. [Online]. Available: <http://www.europarl.europa.eu/news/de/news-room/20170109STO57505/erste-eu-weite-robotergesetze-interview-mit-mady-delvaux> [Zugriff am 21.1.2017]
- [107] G. Reif: „6. Expertensystemes“, 1.2.2000. [Online]. Available: <http://www.iicm.tugraz.at/greif/node8.html> [Zugriff am 23.1.2017]
- [108] Bundesministerium für Justiz: „Help.gv.at Grundbuchsabfragen im Internet“, 1.1.2016. [Online]. Available: <https://www.help.gv.at/Portal.Node/hlpd/public/content/60/Seite.600340.html> [Zugriff am 25.1.2017]
- [109] derStandard.at: „Grundstücksumwidmung bringt jährlich 2,7 Milliarden Euro Gewinn - derstandard.at/1317019815089/Bauland-Grundstuecksumwidmung-bringt-jaehrlich-27-Milliarden-Euro-Gewinn“, 10.10.2011. [Online]. Available: <http://derstandard.at/1317019815089/Bauland-Grundstuecksumwidmung-bringt-jaehrlich-27-Milliarden-Euro-Gewinn> [Zugriff am 25.1.2017]
- [110] Amt der Steiermärkischen Landesregierung: „Digitaler Atlas der Steiermark“, 2012. [Online]. Available: <http://gis2.stmk.gv.at/atlas/init.aspx?karte=adr&ks=das&cms=da&masssta> [Zugriff am 26.1.2017]
-