Master's Thesis

Development of an Online B2B Platform for the Integrated Vehicle Health Management Community

Robin D Waluschnig

Knowledge Management Institute (KMI)
Graz University of Technology, Austria

Advisor: Ass. Prof. Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic

Manufacturing Department, School of Applied Sciences
Cranfield University, United Kingdom

Advisor: Dr. Ip-Shing Fan

Cranfield and Graz, July 2011

This thesis is submitted in accordance with the
Double Degree Programme regulations.

Abstract

The Integrated Vehicle Health Management (IVHM) community despite joint development of IVHM systems still faces challenges in the area of information exchange and collaboration. A business context analysis showed that buyers face the major complication of high transaction and search costs related to locating IVHM technology whereas suppliers have difficulties in advertising their technology and broadening their customer base.

In order to tackle these challenges in this thesis an information intermediary new to the area of IVHM was created. Therefore the concept of online marketplaces was successfully adapted to this new context of IVHM and implemented as a prototype. The Weighted Scoring Method and the Analytical Hierarchy Process were employed to select the most suitable open source software solution as basis for the system. The chosen e-commerce platform Magento proved to be a very flexible and powerful framework for the implementation of the prototype.

Besides conventional software testing Thinking-Aloud tests and the System Usability Scale were employed to evaluate the system with actual test users. The successful implementation of the prototype proved the technical feasibility and allows further evaluation of the IVHM B2B Platform in a pilot phase at the IVHM Centre at Cranfield University. The created platform constitutes a valuable supplement to existing B2B collaboration services such as Exostar. It achieves its intended purpose and associated benefits in terms of efficiency improvements in information exchange and communication, intensification of buyer-supplier collaboration beyond IVHM and facilitation of general business initiation for the IVHM community.

Kurzfassung

Die Partner der Integrated Vehicle Health Management (IVHM) Community stehen trotz gemeinsamer Entwicklung von IVHM Systemen vor Herausforderungen in Informationsaustauch und Kollaboration. Eine Analyse des IVHM Umfeldes zeigte, dass für die Käufer hohe Transaktions- und Suchkosten zur Lokalisierung entsprechender Technologie anfallen, wohingegen es den Herstellern schwer fällt ihre Technologie zu bewerben und ihre Kundenbasis zu erweitern.

Um hinsichtliche dessen Abhilfe zu schaffen wurde in dieser Arbeit ein für das IVHM Umfeld neues Informationssystem entwickelt. Dafür wurde das Konzept eines virtuellen Marktplatzes erfolgreich für diesen Kontext adaptiert und als Prototyp implementiert. Die Weighted Scoring Method und der Analytical Hierarchy Process wurden angewandt um die geeignetste Open Source Lösung als Basis für das System zu selektieren. Die gewählte E-Commerce Plattform Magento erwies sich als flexibles und leistungsstarkes Framework für die Implementierung des Prototyps.

Neben konventionellen Softwaretestmethoden wurden Thinking-Aloud Tests und die System Usability Scale eingesetzt um das System mit Testnutzern zu evaluieren. Die erfolgreiche Implementierung des Prototyps zeigte die technische Machbarkeit und erlaubt eine weitere Evaluierung der IVHM B2B Plattform während einer Pilotphase am IVHM Centre an der Cranfield University. Die entwickelte Plattform stellt eine wertvolle Ergänzung zu bestehenden B2B Kollaborationsdiensten wie zum Beispiel Exostar dar. Außerdem erfüllt sie ihren Zweck und den damit korrespondierenden Nutzen in Bezug auf Effizienzsteigerungen im Informationsaustausch, Intensivierung der Käufer-Verkäufer Kollaboration auch über IVHM Technologie hinaus und die generelle Förderung von Geschäftsanbahnung zwischen den IVHM Partnerunternehmen.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

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 entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

……. ……..
date (signature)

Acknowledgements

First and foremost I would like to express my gratitude to my advisors, Dr. Ip-Shing Fan at Cranfield University and Ass. Prof. Dr. Denis Helic at Graz University of Technology for their ongoing support throughout the course of this thesis.

At the same time I would like to thank Dr. Kirby Keller, The Boeing Company, Steve Parker and Stéphane Benoit, both Meggitt PLC, for the inspiring discussions and their valuable input to this project.

Finally, my parents, Viola and Manfred Waluschnig, receive my deepest gratitude for supporting me in all aspects throughout my whole education and all my studies at different universities.

Cranfield and Graz, July 2011 Robin D Waluschnig

Contents

List of Acronyms

A&D Aerospace and Defence

AFDX Avionics Full-Duplex Switched Ethernet

AHM Asset Health Management

AHP Analytical Hierarchy Process

ASP Application Service Provider

BPO Business Process Outsourcing

B&M Bricks-and-Mortar

CAGR Compound Annual Growth Rate

CBM Condition-Based Maintenance

COTS Commercial Off-The-Shelf

CSS Cascading Style Sheets

B2B Business-to-Business

B2C Business-to-Consumer

B2G Business-to-Government

BIT Built-In-Test

C2B Consumer-to-Business

C2C Consumer-to-Consumer

C2G Consumer-to-Government

CBR Case-Based Reasoning

CRM Customer Relationship Management

EAV Entity-Attribute-Value

EDI Electronic Data Interchange

EDIFACT Electronic Data Interchange For Administration, Commerce and Transport

EHM Equipment Health Management

ERD Entity-Relationship Diagram

ERP Enterprise Resource Planning

G2B Government-to-Business

G2C Government-to-Consumer

G2G Government-to-Government

HCI Human Computer Interface

HTML HyperText Markup Language

ICT Information and Communication Technology

IIS Internet Information Service (Microsoft web server)

ISHM Integrated Systems Health Management

HW Hardware

IEEE Institute of Electrical and Electronics Engineers

IVHM Integrated Vehicle Health Management

KBS Knowledge-Based Systems

MECE mutually exclusive and collectively exhaustive

MCDM Multi Criteria Decision Making

MVC Model-View-Controller

N/A Not Available/Not Applicable

NGO Non-Governmental Organisation

NPO Non-Profit Organisation

NPV Net Present Value

OEM Original Equipment Manufacturer

ORM Object-Relational Mapping

PA Platform Administrator

PDF Portable Document Format

PHM Prognostics and Health Management

PSS Product Service System

RBR Rule-Based Reasoning

R&D Research and Development

RFA Request For Availability

RFQ Request For Quotation

RFP Request For Proposal

ROI Return-On-Investment

SA Supplier Administrator

SaaS Software as a Service

SDLC Software Development Life Cycle

SHM Subsystem Health Management

SME Small and Medium Enterprise

SQL Structured Query Language

SW Software

SRS Software Requirements Specification

SSS Server-Side Scripting

SUS System Usability Scale

TCO Total Cost of Ownership

THA Thinking-Aloud

TRL Technology Readiness Level

UAT User Acceptance Testing

UC Use Case

UI User Interface

URL Uniform Resource Locator

V&V Verification and Validation

WSM Weighted Scoring Method

WYSIWYG What-You-See-Is-What-You-Get

XHTML eXtensible HyperText Markup Language

XP eXtreme Programming

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Introduction

In recent years enormous developments in the area of Integrated Vehicle Health Management (IVHM) have been observed. Driven by the need to decrease maintenance costs, improve efficiency and technological step changes in mobile computing power, IVHM systems are deployed in many different industries and sectors. IVHM systems integrate with present concepts of maintenance, overhaul and repair and offer a total health monitoring for high-tech, high-value vehicles and assets such as aircraft, ships, high-speed trains or power plants.

1.2 Motivation

The Integrated Vehicle Health Management Centre at Cranfield University was launched 2008 by The Boeing Company and Cranfield University in order to facilitate and further integrate the development of IVHM systems between industry and academia. Since then, the centre has evolved steadily including new partners such as Alstom S.A., BAE Systems plc, Meggitt plc or Rolls-Royce plc. The centre now incorporates suppliers as well as buyers of IVHM systems.

Manufacturers of high-value vehicles usually have to work closely together with IVHM suppliers right from start of the vehicle design in order to integrate the IVHM system on the vehicle. Nevertheless, depending on the IVHM components or systems considered, different buyer-supplier relationships for IVHM system are observed. Even though the centre facilitates the communication and the development of IVHM systems of the partners involved, there are areas in collaboration and information sharing that require improvement.

There is no common platform for IVHM systems and components where existing IVHM capabilities and technologies including their specifications are centralised. It is thus difficult for IVHM buyers to search and compare existing IVHM technologies. At the same time suppliers do not have a central location where they can list and advertise their IVHM capabilities.

To tackle these challenges the possibility of an online Business-to-Business (B2B) platform as information intermediary shall be explored. The platform shall bridge the information exchange gaps between IVHM buyers and suppliers and facilitate their communication and collaboration in one central point.

1.3 Aim

The aim of this thesis is to *identify the features and assess the technical feasibility and benefits of an online business-to-business platform for the integrated vehicle health management community at the IVHM Centre* at Cranfield University.

1.4 Objectives

The *main objective* of this project is to *build a prototype of the business-to-business platform for the IVHM Centre.* Based on the analysis to specify and implement the prototype, and the evaluation of the prototype itself the project aim can be achieved.

The development of the prototype needs to cover the whole software development life cycle, from understanding the business context and its complications, requirements analysis, software (package) evaluation and selection to design, implementation, software testing, validation and deployment. Throughout the different phases of the software project life cycle different objectives are pursued.

Common for all phases is the objective for each step to acquire a thorough understanding of the approaches discussed in literature, to identify the most suitable approach for the context and then to customise and apply it for this specific project.

The sub-objectives that build up towards the main objective are outlined in the following itemisation.

- Conduct interviews with all key stakeholders to analyse the business context, its complications and the relationships between IVHM buyers and suppliers in order to identify the processes to be mapped on the platform, and the information and relation of IVHM products and technologies to be presented. This objective can be summarised in *understanding the background and identifying the problems to be solved.*
- Identify the features of the B2B platform to solve the problems in terms of 'doing the right thing' and produce a concise software requirements document for a platform prototype.
- Select the most suitable software evaluation and selection approach. Apply it to the researched solution possibilities (e.g. commercial software, customisation of an open source software package, custom development) and choose the optimum solution with respect to the requirements and the project constraints.

- Design and implement the software prototype according to best-practice software engineering methods.
- Test the software at all layers including User Acceptance Testing (UAT) and validate the software with the client in order to hand it over for the pilot phase.

1.5 Deliverables and Scope

The main deliverables are three-fold: the documentation of the business context and its complications, a concise software requirements specifications document and a complete software prototype that is fulfilling all functional requirements. The software solution is web-based and deployed in existing server architecture at the IVHM Centre.

The project is delivering a fully functional software solution but some areas are excluded from the scope. Firstly, the solution is a software prototype that is used solely internally for the IVHM Centre in order to evaluate the benefits and the business case for such a platform. Considering the time frame of the project the implementation of a high-security system, as it would be required for confidentiality reasons for software with interfaces outside the centre, is not in the scope of this project.

The project ends with the hand-over of the software prototype. Thus, the maintenance of the software pilot and the identification of additional requirements and benefits during the pilot phase are not covered in the scope.

The prototype allows to assess the feasibility of the platform in terms of technology and systems as well as operations, how well it solves the problems and how well it generates its intended benefits with regards to the initial requirements. The prototype's requirements specifications and the mentioned feasibility assessments lay the foundation for the business case evaluation for a potential final software product.

The initial requirements can only be refined during the pilot phase whereas also a quantitative assessment of the platform's benefits will be performed. At the same time the development costs for a potential final implementation with the refined requirements need to be estimated in order to feed into a cost-benefit analysis. This economic feasibility study for a potential final platform solution is not in the scope of this project.

1.6 IVHM B2B Platform Objectives

Based on the hypothesis that the *IVHM B2B Platform* is technically feasible, its purpose is to deliver benefits for the IVHM community. The IVHM B2B platform concept shall be assessed by a problem analysis and feasibility study supported by a software prototype. The prototype also allows to evaluate the achievement of the platform objectives and associated benefits. The platform objectives are outlined in the following list.

1. Efficiency improvements in the areas of information exchange and supplier-buyer communication.

    - Central presentation of all available IVHM products and capabilities to potential buyers inside the IVHM Centre community.
    - Clear overview for buyers what technologies are available having a central pool of available IVHM modules and systems with cross-compatibility information.
    - Supplier-buyer communication is improved in general due to the exchange of information and discussion on IVHM products and technologies on one central platform.
    - Improvements in the suppliers' understanding of the buyers and their requirements.

2. Possibility for platform users to intensify the business relationship and collaboration in terms of systems beyond IVHM.

    - Linking of IVHM systems with complementary technology.
    - Integrate products or technologies from other areas of certain suppliers for instance control systems on the platform.

3. Facilitation of general business initiation between partners inside the IVHM Centre.

In short, the platform as central pool of available IVHM modules and systems including compatibility information shall facilitate the efficiency of information exchange on IVHM products and support the initiation of business between buyers and suppliers inside the IVHM Centre.

1.7 Thesis Structure

This thesis is structured in three major parts. **Part I**, chapters 2-6, covers the literature review and the market research, **Part II**, chapters 7-12, comprises the actual feasibility study and the prototype implementation and **Part III**, chapters 13-14, discusses and concludes the findings of the thesis.

Part I is structured based on three major topics, **chapter 2** provides the background to IVHM, **chapter 3** discusses online marketplaces including the theoretical background and a market analysis and **chapters 4-6** comprise the literature review for selected topics in a software development life cycle. **Chapter 4** discusses software requirements in general and how to document them. **Chapter 5** explores the process of software evaluation and selection including software delivery models, software return-on-investment and methodologies to evaluate and select a software solution. Concluding Part I, **chapter 6** describes software verification and validation methods.

Part II comprises the actual feasibility study and prototype implementation covering all areas of the software development life cycle. **Chapter 7** gives an overview of the methodology applied and **chapter 8** analyses and documents the IVHM business context including interview results

and case studies. **Chapter 9** documents the software prototype requirements using the IEEE software requirements specification template and a detailed use case analysis. **Chapter 10** covers the market research on possible software packages, software evaluation criteria and the actual application of different software evaluation methodologies. **Chapter 11** then provides an in-depth analysis of the technical properties of the platform implementation including the system, software and database architecture as well as detailed software design including applied software design patterns. **Chapter 12** concludes Part II with the results from the software verification and validation comprising Thinking-Aloud tests and a usability questionnaire.

Part III finalises the thesis with a discussion of the results and overall conclusions. **Chapter 13** discusses the results and the applied methodology, argues the established platform objectives and elaborates the prototype's success factors and potential barriers of adoption. **Chapter 14** concludes the thesis with regards to its aims and objectives including its limitations and gives an outlook on future work.

Part I

Literature Review

Chapter 2

Integrated Vehicle Health Management

This chapter provides an introduction to the technological background of IVHM and discusses the industries where IVHM systems are deployed.

2.1 Introduction to Integrated Vehicle Health Management

The correct functioning of all critical components of any high-value vehicle such as spacecraft, aircraft, car, truck, ship, submarine or train during its operation is crucial to ensure the safety of its passengers. Vehicle health, the condition of a vehicle's systems and sub-systems, needs to be monitored in order to perform overhaul and maintenance of these components. At the same time vehicle maintenance attributes to a large portion of the overall costs of operating a vehicle. For commercial aircrafts Bird et al. [8] state that even close to 95 percent of the total life cyle costs are contributed by maintenance activities.

IVHM now subsumes the collection of data concerning the current and future condition and performance of a vehicle and its transformation into actionable information. Even though the term is not underpinned by a generally accepted definition in literature, Benedettini et al. [5] generalise the definition of IVHM in their extensive literature study as "the capture of vehicle condition, both current and predicted, and the use of this information to enhance operational decisions, support actions, and subsequent business performance".

The term IVHM is primarily used in the aerospace sector [5] but other terms such as Asset Health Management (AHM) [4], Equipment Health Management (EHM) [56], Integrated Systems Health Management (ISHM) [28] or Prognostics and Health Management (PHM) [40] may refer to the same or a similar concept that is applied for various high-value assets and complex systems.

The following sections will focus on IVHM based on the stated definition and its application to all kinds of vehicles.

2.1.1 IVHM System Technology and Configuration

IVHM evolved from Built-In-Test (BIT) systems in early analog aircraft systems where pushing a 'test button' would do a simple test on the aircraft circuitry, signalling success with a green light on the dashboard [8]. From this basic idea of checking a subsystem's health the IVHM concept has evolved. Figure 2.1 describes the basic architecture of an IVHM system as it is deployed today.

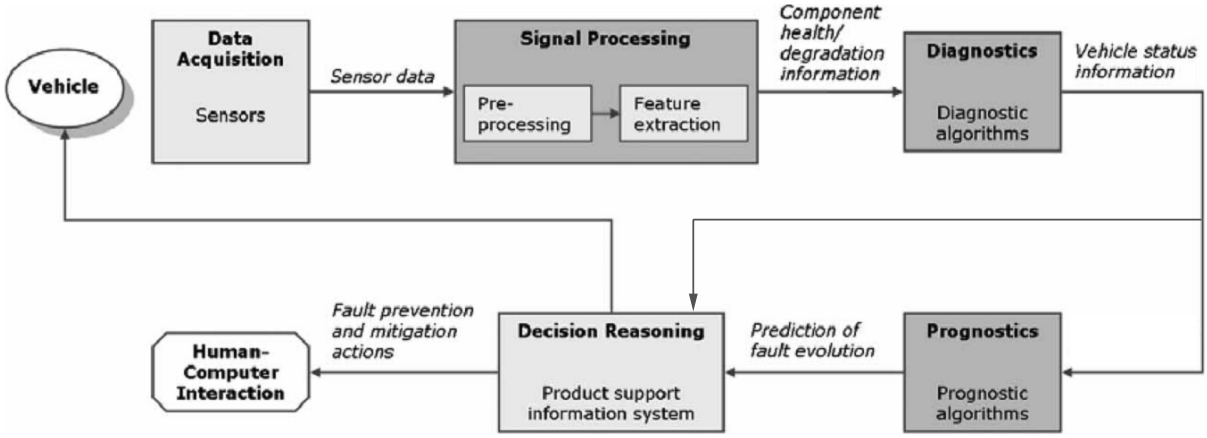

Figure 2.1: IVHM system architecture. (Source: adapted from Benedettini et al. [5, pg. 162])

Benedettini et al. [5] describe the IVHM data acquisition and processing phases as follows. In the first step *sensors* acquire raw data that directly or indirectly relate to a system's or component's health. Examples are temperature, vibration or flow-rate sensors for engines, avionics or vehicle structures.

The raw sensor data is then preprocessed in a *signal processing unit* in order to remove noise and 'clean' the data before feature extraction methods can be applied. Low-pass filtering and time-synchronous averaging are example methods for the former whereas for instance Fourier transform-based methods are often used to extract condition indicators.

The *diagnostics* module processes the health data to detect and identify impending and initiating failure conditions. Whereas diagnostics analyses the current health status, *prognostics* is concerned with the future condition of the system. Current and historical health data is used in combination in order to estimate the time-to-failure of components and systems. Prognostics employs model-based approaches such as Lagrangian dyamics or autoregressive moving average (ARMA) models or data-driven approaches such as artificial neural networks or expert systems that are based on known fault patterns.

Finally, diagnostics and prognostics data is fed into a *product support information system* that turns it into product support actions. Selected information is then transmitted to on-board automatic recovery systems, the vehicle operators, and support managers such as maintenance operators and supporting infrastructure.

Now that a typical IVHM architecture has been explored, it has to be integrated into the bigger context. Figure 2.2 illustrates a general example for airborne vehicles in the bigger picture

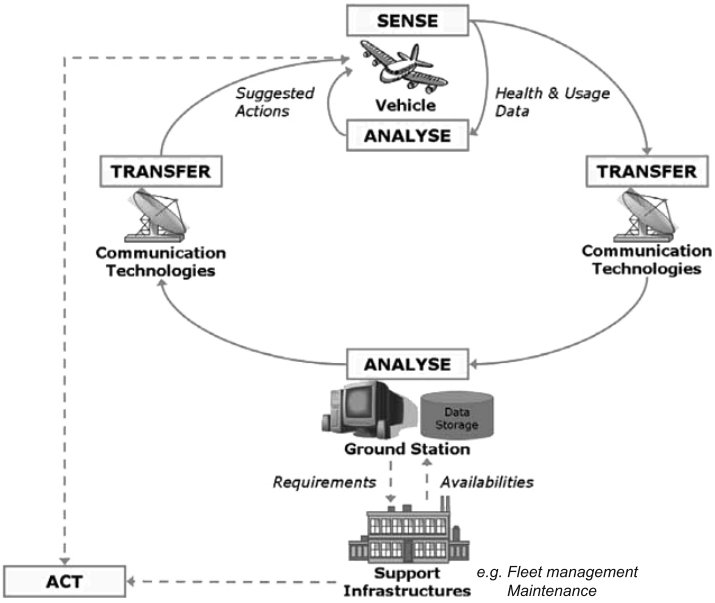

Figure 2.2: Illustration of an IVHM system configuration. (Source: adapted from Benedettini et al. [5, pg. 163])

according to Benedettini et al. [5]. Employing (wireless) communication technology data is transmitted to a ground support centre where further analysis capabilities are available and activities in the support infrastructure are planned and triggered. All variations between all data processing is performed on-board, and all health management functions are carried out on remote resources, are viable. The actual solution depends on the necessary autonomy and complexity of the vehicle and its operational environment.

2.1.2 IVHM Drivers

Driven by the need for safety, cost-effective vehicle operation, and automation of logistics coordination for e.g. spare parts, the IVHM concept evolved from early BIT systems [8]. Even though the cost pressure in commercial as well as military vehicle markets [1] is the major driver of the IVHM concept, today it is even seen as value proposition for aftercare service providers (Williams [101] citet in Benedettini et al. [5]). IVHM enables increased viability for performance based arrangements in terms of Product Service Systems (PSSs) where companies offer a mix of products and services [2]. There IVHM is considered as key enabler to achieve higher profit margins in the long term and reduce technical risk.

2.1.3 Benefits of IVHM

As already indicated in section 2.1.2 IVHM generates different benefits around vehicle operations.

For *mission operation* IVHM provides more accurate information on the vehicle status before and during the mission. This facilitates adaptive control and improvid survivability [5] and based on the vehicle's health and better informed decisions by the vehicle operators (e.g. pilots

in aerospace) and decision makers in command-and-control centres [101]. Thus, overall mission reliability and effectiveness is enhanced.

Due to the more accurate assessment of the current and future health of a vehicle potential failures can be identified early and treated accordingly, thus improving reliability and safety. (Fox and Glass [32] citet in Benedettini et al. [5])

Support functions are concerned with *fleet management* and *maintenance operations* [101]. Matching the requirements of a mission to the capabilities of a vehicle based on its condition is the purpose of fleet management [101]. IVHM allows the maximisation of vehicle utilisation balancing mission requirements and maintenance needs in 'fix-or-fly' decisions, configuration of maintenance programmes and mission reconfiguration [5]. In the area of maintenance IVHM facilitates the employment of Condition-Based Maintenance (CBM) where rather than fixed time intervals, actual usage and component condition drive maintenance operations [101].

The usage of diagnostics and prognostics in vehicle components reduces the need for inspections, fault ambiguity, time for repairs while increasing fault detection coverage. Overall the maintenance performance driven by maintenance downtime, cost of spare parts (less ambiguity, early replacement of cheaper components that would affect more expensive ones) and maintenance man-hours is improved. [5]

In support of maintenance, *logistics operations* also benefit from the continuous availability of accurate health information of single vehicles and the whole fleet [5]. Information on component usage and condition enables advanced notification for maintenance requirements, automatically triggering the order of spare parts and scheduling of maintenance. Employing IVHM the overall logistics footprint can be reduced [40].

Ultimately, IVHM is seen as enabler for improved vehicle or system design [8]. Field data captured during operations of previous vehicle models can be used by Original Equipment Manufacturers (OEMs) in order to upgrade or design components and systems that minimise the environmental impact, improve availability and reliability and thus reduce costs. Furthermore, based on field data even the IVHM systems and their integration in a vehicle can be improved. [5]

Figure 2.3 summarises the benefits of IVHM integrating Subsystem Health Management (SHM) systems in its improved decision support which benefits associated activities. From improved vehicle design, fleet management and on-board safety, improved maintenance and logistics, to reduced training of operators to interpret vehicle health and maintainers to detect failure root causes, IVHM benefits on the whole range of activities. [85]

2.1.4 IVHM Systems in Context

Looking at the basic IVHM architecture shown in figure 2.1 one might consider IVHM as a standalone subsystem built upon sensors, a related instrumentation system and some software. Scandura [85] strongly advocates that IVHM should not be treated as such. The author states:

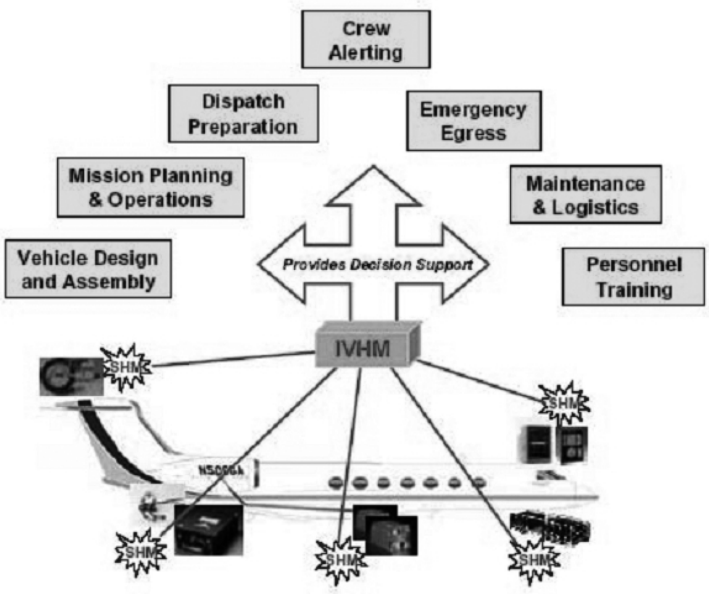

Figure 2.3: Summary of activites benefittnig from IVHM-enabled decision support. (Source: adapted from Scandura [85, pg. 7.D.1-5])

> "While IVHM utilizes these components to perform its intended function, a true IVHM system incorporates a philosophy, methodology and process that focuses on design and development for safety, operability, maintainability, reliability and testability. To be most effective, IVHM must be 'designed in' to the target system (i.e., the vehicle and its supporting infrastructure) from the beginning of the program, and not 'added on' along the way. IVHM must be elevated to the status of a system engineering discipline."

One can only agree with this statement considering the full context of IVHM. The on-board IVHM technology generates actionable data that used as decision support (automatically) triggers a whole set of activities and affecting all areas involved in vehicle operations and even vehicle design. Due to the complexity of these intertwined activities and systems IVHM as system engineering discipline has to be integrated from the beginning of the vehicle design.

2.2 IVHM Target Industries

The industries currently targeted by the IVHM Centre, aerospace, marine, railway and the high-performance land craft sector have very similar properties. Also the sectors in the extended scope of energy and health can be categorised similarly. Taking the aerospace industry as representative for these industries, Loukis et al. [57] states that the market in general shows 'oligopsonistic' and 'oligopolistic' conditions. Thus, sales is characterised by a small number of customers that purchase these sophisticated high-value often higly customised products and services that cannot be sold online, whereas procurements is characterised by a limited number of sellers (to a smaller extent) supplying different specialised and sophisticated raw materials, spare parts, high-value components and electronics.

The aerospace sector is dominated by a small number of primary airframers (in particular Boeing (US), Airbus (parent EADS, European) and BAE (British Aerospace, parent BAE Systems, UK), by three major engine system manufacturers (Rolls-Royce (UK), General Electric (US) and Pratt & Whitney (parent UTC Corp., US)) and by a larger number of major subsystem suppliers as for example Honeywell International, Inc. (US), Hamilton Sundstrand (parent UTC Corp., US), BAE Systems (UK), Finmeccanica (Italy), Smiths Group (UK), Thales (France) or MTU Aero Engines (parent KKR, Germany). [98] Similar structurs can be observed in the rail sector with Alstom (France), Bomardier (Canada), Kawasaki (Japan) and Siemens (Germany) as major OEMs. [58]

The capital intensity of these high-value-product industries bears high entry barriers for new players and also shows stark consolidation forces and thus 'oligopolisation' and 'oligopsonisation' due to increased cost pressure and economies-of-scale considerations [57]. Companies als tend to rationalize their supply base in order to minimise transaction costs and foster relationships with suppliers or even integrating their suppliers for subsystem development. [13], [54]

Thus, B2B relationships in these industries are oriented long-term based on (a history of) trust, good (informal) relationships and collaboration [77]. Considering the long life cycle and development costs of these high-value assets and the manufacturers' goal to minimise supplier risk in terms of quality, reliability and service this seems to be a logical consequence [37]. New entrants in the supplier market sometimes even with technological advantages have a difficult time to build up the necessary credibility and trust in order to receive business from OEMs and major subsystem manufacturers.

Chapter 3

Online Marketplaces

Since the rapid success of the internet, electronic commerce and online marketplaces have been growing at a similar pace. In support of this fact, Forrester Research, a global independent research company, estimates a Compound Annual Growth Rate (CAGR) for the US as well as European online retail market of 10 percent from 2010 to 2015. This results to a total online retail market volume of $279 billion for the US and €134 billion for Europe in 2015. [11]

Compared to the respective total retail market, for instance the US Census Bureau [97] states $4.1 trillion in its *2009 Annual Retail Trade Report* (excluding motor vehicles and parts), the online retail numbers may seem small, but many retail products are not typically sold online. For instance food and beverages, building materials and garden equipment or gasoline are not typically sold via online channels (even though some businesses are increasingly successful delivering the former[1]). Hence, the online retail market is growing in its conventional areas such as books, electronics or clothes at an enormous pace.

The online retail market serves as an example to stress the importance of electronice commerce and online marketplaces in general. Nevertheless, online retail covers only a small area of electronic business and electronic commerce. In the following sections both terms are defined and various types of e-commerce approaches and their business models are discussed.

3.1 Electronic Business and Electronic Commerce

The terms electronic business (e-business) and electronic commerce (e-commerce) without clear definition are often used interchangeably in media and sometimes even in literature, hence require clarification.

E-business is commonly referred to as the conduct of business both within an organisation as well as with external stakeholders using information and communication technology in support of *all* business processes of an organisation [12, pg. 14]. E-business may then refer to two different concepts within organisations. Related to strategy and operations e-business is the

[1] Tesco plc, UK's leading grocery chain, has become the fourth biggest online retailer in the UK. [29]

ICT-enabled conduct of business [7, pg. 465]. Related to the type of organisation e-business refers to organisations that mainly operate online, in contrast to Bricks-and-Mortar (B&M) businesses that have a physical presence like stores or outlets [12, pg. 14]. *Amazon* as prime example has no physical shops for its customers.

E-commerce is a widely used term as well, but its definition is less clear in literature. Haig [39, pg. 251] defines e-commerce short as "business transactions over the Internet". Gora and Mann [36, pg. 1] support a wider definition of e-commerce as "the trade and commerce of goods and services employing state-of-the-art information and communication technology, in particular the internet". Rayport and Jaworsky [76, pg. 644] define e-commerce even more widely as "technology-mediated exchanges between parties (individuals and organizations), as well as the electronically-based intra- or inter-organizational activities that facilitate such exchanges".

The term 'commerce' implies that goods or services including information are exchanged. All definitions also consent that for e-commerce at least two parties are involved which excludes purely internal transactions that do not support commercial transactions with external parties (in contrast to e-business). In conclusion, e-commerce is defined in support of Chaffey [12, pg. 8] as "all electronically mediated transactions between organizations and its external stakeholders". Thus, e-commerce constitutes a subset of e-business activities which also includes intra-organisational ICT-enabled activities [7, pg. 1ff.].

Reflecting on these findings in literature it is fair to say that ultimately all business activities (indirectly) support commerce activities (e.g. human resources) and since some organisations exclusively engage in electronic commerce, e-commerce can also be seen as equivalent to e-business. Thus, depending on the context and the type of business the terms e-business and are either equivalent or e-business is a superset of e-commerce as supported by Beynon-Davies [7, pg. 1ff.].

Figure 3.1 illustrates the latter case with e-commerce as a subset. Each organisation or entity is depicted by an indication of Michael Porter's (internal) value chain [74] symbolising the entity's (business) activities. The entities are in different e-commerce relationships with each other, namely B2B, Business-to-Consumer (B2C) and Consumer-to-Consumer (C2C), connected via supply chain, customer or demand chain and community chain . The following section will analyse these relationships and e-commerce transaction models.

3.2 E-Commerce Transaction Models

3.2.1 Business-to-Consumer (B2C)

B2C marketplaces, 'online stores' [95, pg. 45] or 'e-tailers' for 'online retailers' [11] are the most common e-commerce types. B2C describes a business model where goods and services are sold from business to consumer, thus directly serving the demand chain as shown in figure 3.1 [76, pg. 4f.]. This model reflects the simple online store concept as for example *Amazon.com* in its purest form for books, CDs and DVDs or *Dell.com* selling computers [7, pg. 11f.].

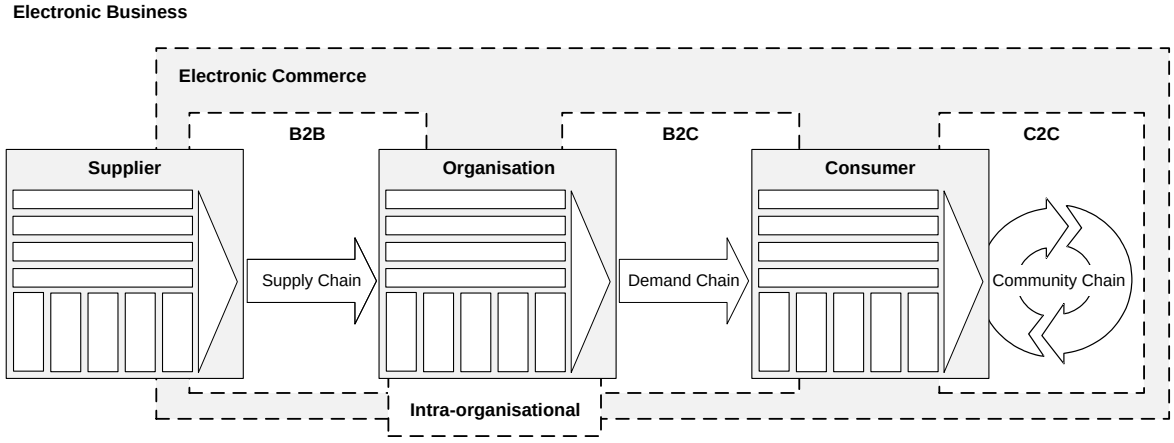

Figure 3.1: A model of the e-business and e-commerce domain. (Source: adapted from Beynon-Davies [7, pg. 3])

In the chapter introduction a double-digit growth in B2C online retailing is mentioned. The reasons for the enormous growth are evident since there are numerous advantages for an online shop to increase its bottom line compared to a conventional B&M business.

Access to a larger, even global market, wider geographical reach, '24x7' opening hours [36, pg. 293f.] and an e-shop driven expansion of the product portfolio (e.g. products that due to storage could have not been sold through B&M stores) facilitate an increase in revenue [25, pg. 8]. In addition to that the customers' convenience is enhanced. Now they can literally shop at anytime from anywhere where they have web access referring to the surge in sales of smart-phones and tablet computers. Improved inventory management and storage costs, reduced occupancy costs (no stores in premium locations required), less personnel costs as well as higher volume purchases and related discounts facilitate a significant cost reduction [25, pg. 8ff.].

Concluding, online shops are highly attractive sales channels for standardised, easy-to-store-and-ship products where consumers do not necessarily require product interaction before buying it. This conclusion and the advantages of online stores are directly reflected in the success of for instance *Amazon.com*. Nevertheless, not all product types are suitable for B2C without the necessary logistics integration (e.g. Tesco grocery online shop and home delivery [29]) or a consequential reduction in profitability.

Along with *Amazon.com* and similar organisations in particular start-ups and small enterprises benefit from the concept, since they can channel their usually tight cash into product development instead of maintaining a store network and writing paychecks to the necessary shop assistants [25, pg. 8].

3.2.2 Business-to-Business (B2B)

In B2B both sellers and buyers are business corporations [7, pg. 11]. Figure 3.1 depicts B2B as transactions between an organisation and its suppliers that are again businesses.

In contrast to B2C, B2B e-commerce has a much longer history. Long before the surge of the

internet era, companies in various industries already had their mainframes connected in order to conduct ICT-enabled business. Interfaces like Electronic Data Interchange (EDI) were and are still used to facilitate such application-to-application interaction. [92, pg. 4ff.]

EDI defines the electronic exchange of business and transaction data as for example order, invoicing, bank transfer or terms of payment data between business partners in a standardised format. The data is automatically exchanged without user interaction between the business partners' IT systems that are integrated via EDI. [45, pg. 12ff.]

The initial coarse definition of EDI led to an uncontrolled growth of customised solutions that were not interoperable, thus Electronic Data Interchange For Administration, Commerce and Transport (EDIFACT), an industry-independent norm, was developed. Whereas EDI focuses on the rationalisation of existing business relationships, electronic marketplace aim at initiating new business while at the same time rationalising them. [45, pg. 12ff.]

The success of the internet has not only led to the development of internet-based EDI [86] but also to an explosion of easily usable and low-cost forms of e-commerce and different types of online B2B marketplaces [92, pg. 4].

B2B marketplaces deliver numerous benefits facilitating the buying and selling process. Buyers can easily find and compare products and prices, whereas sellers can aggregate their supply and demand [39, pg. 53]. Businesses have the possibility to attract new customers relevant to their industry while keeping them consistently up-to-date with product information as well as industry news [92, pg. 7]. In addition to that, B2B marketplaces provide a secure environment for transactions [39, pg. 54] and create an online (often industry-specific) business community that facilitates vertical as well as horizontal communication between buyers and suppliers [92, pg. 10]. Ultimately, marketplaces help sellers to improve their understanding of their customers in a way that they can improve and customise their products and services [39, pg. 54].

B2B Models

Turban et al. [96, pg. 200ff.] and Timmers [92, pg. 35ff.] describe various models for online B2B e-commerce. This section shall discuss the most significant ones.

- *Supplier-oriented marketplace*
  Business customers and individual consumers use the same supplier-provided platform for interaction. Example: *Dell*. [96, pg. 204f.]
- *Buyer-oriented marketplace*
  A large buyer invites suppliers to its own marketplace platform to bid on the announced Request For Quotations (RFQs). Examples: Platform of *General Electric*. [96, pg. 205]
- *Third-party or intermediary-oriented marketplace*
  An intermediary is hosting a marketplace platform where sellers and buyers interact [96, pg. 206]. In the B2C context this concept is similar to an 'electronic mall' where an

intermediary hosts a collection of e-shops. B2B Example: *Alibaba* and *EC21* (see table 3.1). [92, pg. 36f, 39, 79f.]

- *Virtual corporation: collaboration and networking between business partners*
  In this model several business partners share costs and resources to develop a product or service. A B2B platform facilitates this process of collaborative design and engineering providing for example communication tools, information sharing and knowledge management, groupware and EDI capabilities. Business partners are either organised along the supply chain or each one creates a portion of the product or service according to its expertise. Example: *Exostar*. [92, pg. 38f.]

3.2.3 Other E-Commerce Transaction Models

Besides B2C and B2C there are other e-commerce transaction models worth exploring. Figure 3.1 illustrates the *C2C* model where consumers sell directly to other consumers. These transactions can also involve third parties as in the examples of *eBay* or *Craigslist* [12, pg.11f.]. In these two examples individuals auction or sell fixed-priced goods, services or information by means of these platforms as intermediary.

C2C is serving the community chain, a complex network between individual actors. Common C2C e-commerce models are e-auctions, virtual communities, collaboration platforms and information brokerage, for example in support of buying decisions. [7, pg. 337ff.]

In the inverted case of B2C, in the Consumer-to-Business (C2B) model the consumer sells products, services or information to a business. Either individual consumers or consumer groups approach a business with an offer or the consumer directly offers a service to the business [76, pg. 5f.]. An affiliation program where a consumer for instance advertises a product of a seller on his blog and receives a certain financial reward from the seller for each initiated sale is a prime example for a C2B model (e.g. the *Amazon Associates*[2] affiliation program). [49]

In order to complete the set of e-commerce transaction models also non-business institutions such as the governments, non-governmental (NGO) and non-profit organisations (NPO) need to be identified as actors [96, pg. 11]. Considering government, in Business-to-Government (B2G) and Consumer-to-Government (C2G) models businesses or consumers interact with or provide feedback to the government. In the inverted case of Government-to-Business (G2B) and Government-to-Consumer (G2C) the government provides services or information (e.g. tax processing or legal regulation information) via e-commerce. The last model, Government-to-Government (G2G), describes ICT-enabled inter-governmental services and exchange of information. [12, pg. 11ff.]

3.3 Online Marketplaces

Table 3.1 provides an overview of the most important B2C, B2B and C2C marketplaces including a short description and unique-visitors numbers. Comparing the 2009 and 2011 numbers the

[2] https://affiliate-program.amazon.com, accessed 25.05.2011.

e-commerce market growth stated in the chapter introduction is confirmed. The table also lists two B2B e-commerce platforms (*Alibaba* and *E21*) where due to their business model, revenue would be a much better performance indicator than unique visitors.

Amazon and *eBay* are the leading e-commerce platforms world-wide. Only *Craigslist* plays in the same league considering unique visitors, nevertheless generates much lower revenues due to its purely advertisement-based business model.

Marketplace	**Unique Visitors**	**Link**	**Comment**
eBay	56,199[1] 62,251[2]	www.ebay.com	Leading e-commerce platform for auctions and fixed-price sales.
Amazon	51,220[1] 78,383[2]	www.amazon.com	Leading product catalog and e-commerce services such as the Amazon WebStore, Amazon Marketplace and cloud computing services.
Craigslist	22,251[1] 72,909[2]	www.craigslist.org	Classified advertisement site that allows free listing for anyone.
Yahoo! Shopping	18,588[1]	shopping.yahoo.com	Online storefronts for products listed in Yahoo!'s shopping service.
Overstock.com	11,306[1] 19,001[2]	www.overstock.com	Auctions and fixed-price sales at lower fees than Ebay.
CafePress	4,665[1] 4,645[2]	www.cafepress.com	Design and sales of customised merchandise as e.g. t-shirts, posters, print-on-demand books and CDs.
Zazzle	2,120[1] 5,235[2]	www.zazzle.com	Design of merchandise that is made and sold on demand.
Etsy	2,074[1] 6,992[2]	www.etsy.com	Marketplace for hand-made goods and vintage.
Oodle	1,280[1] 3,896[2]	www.oodle.com	Collection of classified listings that are repackaged for third-party sites, e.g. newspapers and periodicals.
eCrater	462[1] 1,449[2]	www.ecrater.com	Online shopping mall.
Alibaba	4,005[2]	www.alibaba.com	Leading Chinese-based B2B e-commerce platform with 61 million registered users.[3]
EC21	242[2]	www.ec21.com	Leading Korean B2B e-commerce marketplace with more than one million member companies and 3.5 million monthly visits.[3]
Alibris	1,279[2]	www.alibris.com	US-based B2C online store for books CDs and DVDs.

Table 3.1: Top online marketplaces according to Tozzi [93] and The Economist [91].

[1] In thousands for January 2009. Source: Nielsen Online, The Nielsen Company, a global marketing and advertising research company cited in Tozzi [93].

[2] In thousands for January 2011. Source: Web traffic analysis service Compete.com. http://siteanalytics.compete.com/, accessed 18.05.2011.

[3] Source: [24].

3.3.1 B2B Marketplaces

In terms of online B2B marketplaces *E-Market Services* [3] , a reputable non-profit organisation, funded by the trade promoting organisations of Canada[4], Ireland[5], Norway[6], Spain[7] and The Netherlands[8] in order to make it easier for companies to use electronic marketplaces for international business (selling and sourcing), reports 686 international B2B marketplaces for different areas.

Considering related areas 14 marketplaces concerning aviation, 7 concerning defence and 20 concerning maintenance repair operations are listed [3] . Examples of such marketplaces are *Aeroxchange*[9], founded by leading airlines in North America, Asia and Europe that aggregate 45 billion USD in purchasing power, or *Aerochain*[10], owned by Brasilian aircraft manufacturer *Embraer S.A.*.

All of these marketplaces are focused on sales and purchase of products and spare parts often offering full integration with common Enterprise Resource Planning (ERP) systems. For all marketplaces a registration process is required to be approved as a member sometimes requiring certain accreditations for sellers. Their business models vary, some do not charge fees for buyers but for sellers as for instance *Aerospace Online*[11], but in most cases all marketplace users are subject to transaction-based or general membership fees. The annual subscription or membership fees are rather high, for instance up to 2500 USD for *Exostar*[12] or up to 8000 USD for *PartsBase*[13].

With regards to the IVHM context of this project *Exostar* is the leading B2B service for the stronlgy related Aerospace and Defence (A&D) industry serving BAE Systems, The Boeing Company, Lockheed Martin, Northtrop Grumman, and Raytheon as its major customers [27][14]. It offers multi-enterprise collaboration, interaction and business process management services with full ERP and design software integration underpinned with sophisticated security and identity management. The service integrates organisations along the extended value chain from sourcing processes, supplier certifications and auditing, design and development collaboration to supply chain execution and maintenance management serving 70,000 customers in the A&D industry. [27][15]

[3] http://www.emarketservices.com, accessed 11.07.2011.
[4] http://www.edc.ca
[5] http://www.enterprise-ireland.com
[6] http://www.innovasjonnorge.no
[7] http://www.emarketservices.es
[8] http://www.evd.nl/e-business
[9] http://www.aeroxchange.com, accessed 11.07.2011.
[10] http://www.aerochain.com, accessed 11.07.2011.
[11] http://www.aerospaceonline.com, accessed 11.07.2011.
[12] https://www.exostar.com/registration.asp, accessed 11.07.2011.
[13] http://www.partsbase.com/public/includes/Layout.asp?Page=MEMBERSHIP, accessed 11.07.2011.
[14] http://www.exostar.com/Exostar_Customers.aspx, accessed 20.07.2011.
[15] http://www.exostar.com/products.aspx, accessed 20.07.2011.

Chapter 4

Software Requirements

The elicitation and documentation of software requirements is a crucial part of any software development project. The following sections will discuss their importance, explore different types of requirements and elaborate a requirements documentation template.

4.1 Software Requirements in Context

Kulak and Guiney [55] define seven self-explanatory phases that are fairly consistently followed throughout the development of any software system. Gustafson [38] adds one preceding phase, business case considerations and feasibility study. The reason why Kulak and Guiney [55] are not mentioning this phase is the fact that it is in reality usually conducted by the business areas of an organisation with often not enough involvement from the IT side. Now these eight phases are defined as follows:

1. Business case and feasibility
2. **Requirements gathering**
3. Analysis
4. Design
5. Construction
6. **Testing**
7. **Deployment**
8. **Maintenance**

The **bold-faced items** in reality are often either ignored or only attributed with minimum attention. Engineers tend to perceive these steps as secondary activities whereas analysis, design

and construction receive the most attention, are perceived to require more imagination and are seen as more intellectual challenge. [55, pg. 2f.]

The ignorance or low appreciation of the requirements activity is also indicated by the yearly CHAOS Summary report of the Standish Group. According to them only 32% of all software projects delivered in time and budget with **all required features and functions** [89]. Surely, not all can be assigned to a lacking requirements phase but there is clear indication that this phase as mostly primary phase in any software project does not receive the necessary attention.

In terms of the definition of requirements themselves a differentiation can be made. Broadly speaking there are two different types of requirements:

- Functional requirements,
- Non-function requirements.

4.2 Functional Requirements

IEEE Standards Board [44, pg. 35] defines a functional requirement as: "a requirement that specifies a function that a system or system component must be able to perform." These are the features and functions of a system a user needs in order to perform his operations. *Use cases* are the common approach to document functional requirements [55, pg. 9].

4.2.1 Use Cases (UC)

UCs describe how the system can be used and what functions are expected to be executed in the system depending on the type of user and the scenario [103, pg. 291]. In othe words, a UC delineates the actions a user is taking and the responses the system generates [78, pg. 35f.].

Use Case Template

Jalote [48] provides a use case framework which is described in Table 4.1.

4.3 Non-Functional Requirements

Non-functional requirements for a software system are according to Chung et al. [14, pg. 2] "global requirements on its development or operational costs, performance, reliability, maintainability, portability, robustness and the like". Non-functional requirements address areas of a system that though hidden to the user, but are essential to the system itself. Since most of the non-functional requirement areas end in *-ility*, Kulak and Guiney [55, pg. 9] also calls them "collection of *-ilities*" such as scalability and the areas already mentioned.

Non-functional requirements are crucial during system development and also serve as decision basis for different possibilities of software design and implementation. If ignored or not taken

Use Case 0	***Name of Use Case 0***
Actor	The person or system which uses the system that is being designed.
Precondition	A certain state the system or the actor has to be in, or actions that must have been performed before this activity.
Main success scenario Description	Describes the use case and the interaction if nothing fails and all steps of the scenario succeed.
Required inputs	(Data) Inputs from the actor that are required for the use case.
Exception scenario	Describes possible fault cases in the scenario and the system behavior if some steps in the scenario where not completed successfully.

Table 4.1: Use Case 0.

properly into account "such requirements are generally acknowledged to be among the most expensive and difficult to correct once a software system has been implemented" (Brooks (1987) and Davis [19] cited in Chung et al. [14, pg. 1]).

4.4 Software Requirements Specification

The definition of software requirements is crucial in each development methodology. Some only require high-level descriptions and documentation of the requirements, as for example agile methods, other methods need a precise requirements documentation. Such a document is call Software Requirements Specification (SRS). [48, pg. 37f.]

4.4.1 Software Requirements Specification Template

The IEEE Computer Society [43] has defined the "*IEEE Recommended Practice for Software Requirements Specifications*", IEEE Standard 830-1998, which will be outlined as follows.

---

1. **Introduction**
   provides an overview of the SRS.
   (a) *Purpose*
   Explain the purpose and the intended audience of the SRS.
   (b) *Scope*
   Describe the product, what it will do and will not do, the application of the software including benefits, objectives, and goals.
   (c) *Definitions, acronyms, and abbreviations*
   Define the terminology used in the SRS.

(d) *References*
Specify all documents that are referenced in the SRS.

(e) *Overview*
Describe what the SRS contains and the organisation of it.

2. **Overall description**
describes the background to understand the rest of the requirements.

(a) *Product perspective*
Put the product into perspective with other products usually in a block diagram showing the context of the system and its relations. Specify the constraints in terms of user, hardware, software interfaces, memory, operations and adaption of the software for different sites.

(b) *Product functions*
Summarise the major functions of the product.

(c) *User characteristics*
Describe educational level, experience, and technical expertise of users.

(d) *Constraints*
Describe other constraints not mentioned in 2a.

(e) *Assumptions and dependencies*
Describe assumptions that would affect the requirements in the SRS.

3. **Specific requirements**
"contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements" [43]. Furthermore this section should explain all inputs, outputs of the system and the functions performed by the system.

(a) *External interface requirements*
Describe inputs and outputs of the system in detail.

(b) *Functions*
describes all the functions of the system for all modes of operation, essentially how are inputs processed and outputs generated. Describes the Use Cases (UCs) (see section 9.3.2 for a short introduction to UCs).

i. Use Case 1
ii. Use Case 2
iii. ...
iv. Use Case N

The functional description is recommended to use the terminology "*the system shall*" perform a certain activity. [55]

(c) *Performance requirements*
Describe static and dynamic performance requirements.

(d) *Design constraints*
Identify any design constraints.

(e) *Software system attributes*
Define non-functional requirements such as reliability, availability, security, maintainability and portability.

(f) *Other requirements*
Specify other requirements not mentioned yet.

4. **Appendices**

5. **Index**

[43], [38, pg. 103f.]

Jalote [48, pg. 38ff.] defines four reasons why a high-quality SRS that has been validated with the client is so important:

- "An SRS establishes the basis for agreement between the client and the supplier on what the software product will do.
- An SRS provides a reference for validation of the final product.
- A high-quality SRS is a prerequisite to high-quality software.
- A high-quality SRS reduces the development costs."

These statements might seem as logical and common sense but still requirements documentation is often lacking the necessary amount of attention as already mentioned in section 4.1. It should be stressed that the SRS is the cornerstone of the contract between supplier and customer and the more imprecise the SRS the lower the quality of the final product or the higher the (unexpected) development costs. Furthermore, with regard to the testing phase of a project, without a high-quality SRS there is no reference against the actual implementation that can be validated. Validation is essential to assure that the produce fulfills all the requirements before it can be handed over to the customer.

Chapter 5

Software Evaluation and Selection Methodology

When acquiring new software various aspects have to be considered. This chapter discusses different software delivery models and methods how to evaluate and select the most suitable software solution for a certain purpose.

5.1 Software Delivery Models

Once the need for a new software in order to support business operations has been identified, with often many possible solutions available on the market the optimum software package as well as software delivery model has to be determined.

Sledgianowski et al. [87] outline three different sourcing strategies for software in terms of delivery models and licensing schemes. Even though this specific article is focusing on ERP software, the mentioned sourcing strategies apply with some variation to all software sourcing decisions. The authors describe the software sourcing options as:

1. "Implementing a purchased or leased packaged software system on-site;
2. implementing an application service provider (ASP) model offering a packaged system delivered and supported by a remote data centre; and
3. outsourcing development of a custom system which is then implemented on-site."

5.1.1 Conventional Software Package on Premises

Option one refers to conventional Commercial Off-The-Shelf (COTS) software that is purchased, licensed or leased and maintained by the organisation itself in its own data centres. [87]

5.1.2 Software as a Service

Option two describes a model that especially in recent years has become enormously popular spreading from Small and Medium Enterprises (SMEs) to larger corporations [53]. Sabbah [83] describes this model, which is also referred to as Software as a Service (SaaS) , as an approach where the software vendor delivers the software application mostly via the internet, on demand to the organisation. The Application Service Provider (ASP) hosts the software in its own data centres and provides software maintenance as well as the corresponding software support. The customer, i.e. the organisation, uses the application for instance in its web browser as a service without being concerned with installing or implementing, maintaining or supporting the software. [87]

At first glance one might think SaaS models are more expensive than conventional ones, but considering the Total Cost of Ownership (TCO) of software and economies of scale on the ASP side, this is not necessarily true. The TCO for a SaaS Customer Relationship Management (CRM) implementation for instance is up to 30% less [22]. Concerns countering SaaS solutions are security, the integration with current on-site systems and the risk of software reliability since the in-house IT department is not hosting the application any more and thus cannot support it [87]. Nevertheless, the (financial) benefits outweigh the concerns especially for standardised applications as for example CRM systems and smaller or medium enterprises offering the opportunity to convert capital expenditures into operational expenses without worrying about installation and maintenance [53]. In other areas where process standardisation is not as common, or where software requirements and security concerns are high, SaaS might not be the right choice [87].

5.1.3 Custom Software Development

Option three then considers development of custom software. This includes software development from scratch or even customisation of existing software packages for a specific purpose that is more than just the configuration of a ready-to-use software packages. The development can be outsourced which is often the case for ERP systems [87]. For general software applications this fact may vary.

If the required resources and skills are available inside an organisation (for example in the IT department) and the project size is reasonable, the software might as well be developed, deployed and maintained in-house. If this is not the case, the development is outsourced to IT consulting providers. Those often not only develop the software but also host it. In such a way full business processes can be outsourced (Business Process Outsourcing (BPO)) [23]. Concluding, all variations between in-house and outsourced development and hosting are possible.

5.1.4 Conclusions on Software Delivery Models

The decision for a specific sourcing model in the end depends on various factors. The results of a requirement-based evaluation of different software solutions including the software Return-

On-Investment (ROI), a software's fit into the overall IT strategy, criticality of the software for the business and non-functional requirements are pivotal. IT decision-makers face a complex decision with many different criteria and factors to consider. In the course of the decision-making process, if conducted properly, the most appropriate software sourcing model meeting the business requirements and respecting the boundary conditions will be identified. An approach to support this decision-making process is discussed in section 5.3.

5.2 Software Return-On-Investment

Whatever the software sourcing options are, the pre-condition is a positive software business case. Software development has to be considered as an investment that is supposed to create value. As an investment it also bears certain risks and has the clear objective of generating a ROI. Software ROI models justify a decision to purchase a software or invest in its development. [20, pg. 13ff.]

A simple definition of ROI is shown in equation 5.1 [100]. This definition only illustrates the idea of software ROI. More sophisticated approaches based on Net Present Value (NPV) considerations in terms of cash flow and the integration of intangible benefits are discussed in Denne and Cleland-Huang [20, pg. 13ff.] but go beyond the scope of this document. The following sections will give a short overview of important factors when modelling a software business case.

$$ROI = \frac{Benefits - Costs}{Costs} \tag{5.1}$$

5.2.1 Software Business Case Model

Melillo [61] discusses different factors that influence a software ROI business case model, namely software type, benefits, costs and model parameters.

Software Type

The type of software has to be identified first. *Application software* supports specific business processes and thus directly creates value (e.g. supply chain management module of an ERP system), whereas *infrastructure software* supports business applications without providing a direct business value (e.g. middleware or network management tools). For the latter it is more difficult to quantify the added value. [61]

Benefits

Software delivers two types of benefits: hard benefits and soft benefits. Hard benefits can be measured with tangible cash flows whereas soft benefits cannot be easily quantified but still improve the business case. Examples are reduced head count and increased revenue for the former and improved quality or customer satisfaction for the latter type of benefit. [61]

Costs

In the next step the costs of the software investment have to be identified. The concept of TCO covers the whole life cycle costs including software development, maintenance and service, hardware, systems management and even employee training. [61]

Model Parameters

The determination of model parameters complete the ROI model for a software investment. The useful life span of a software has to be estimated, i.e. how long can a software be used productively also influencing software depreciation considerations. To complete an NPV calculation for the ROI model a certain (interest) rate has to be defined in order to take the time-value of money into account. Since an investment should exceed an organisation's cost of capital, a hurdle rate as rate of return required by the business shareholders higher than the cost of capital is usually set to analyse the ROI of a software to be purchased. [61]

5.3 Software Selection Methodology

A software selection methodology is a structured and standardised approach in order to select the optimum software solution [90]. Based on the work of Nikoukaran and Paul [68], Jadhav and Sonar [46], Jadhav and Sonar [47], Franch and Carvallo [33], Franch and Carvallo [34], Sarkis [84] and Tewoldeberhan et al. [90] a six-step software selection methodology is defined.

1. **Step:** *Determination of software need and requirement identification*

   In the first step the need for a software system is determined and the functional as well as non-functional requirements are identified. The requirements need to be documented as meticulously as possible since they serve as decision foundation for the software selection.

2. **Step:** *Preliminary investigation of available solutions*

   This step delivers a list of available solutions including high-level features and major functionalities. Web-based research, online catalogues and third party reports are valuable contributors to this list.

3. **Step:** *Shortlisting and elimination of candidate solutions*

   The exhaustive list of possible solutions is shortlisted based on essential requirements (hard criteria), fit into existing systems and IT architecture and basic price and vendor elimination. For instance, a small enterprise looking for an accounting software will not consider a multi-million-£ SAP installation.

4. **Step:** *Identification of evaluation criteria set*

   In this step evaluation criteria have to be identified in order to holistically evaluate a certain solution. The criteria are grouped into hard, i.e. essential, and soft, i.e. optional, criteria. Hard criteria are used in the screening step to eliminate solutions immediately.

One obvious requirement for the criteria set is to be as mutually exclusive and collectively exhaustive (MECE) as possible. This means criteria should exhaustively cover all areas to be evaluated but at the same time not overlap or correlate.

5. **Step:** *Screening and application of evaluation methodology including empirical evaluation*

   Step 5 at first eliminates all solutions that do not meet hard criteria requirements. Secondly, based on the criteria set an evaluation methodology is applied. Section 5.5 discusses evaluation methodologies in more detail. In order to provide input for the evaluation methodology and certain criteria (e.g. user interface) software demos and trial setups are used in order to empirically evaluate solutions.

6. **Step:** *Selection of solution*

   Step 5 produces an aggregate total score for each solution. All solutions are ranked based on this aggregate score and the top solution(s) can be selected. Now, all factors that are not reflected in the criteria are taken into account. Input from price negotiations with different vendors, price/performance trade-off, software ROI, alignment with IT strategy and human dependable factors (e.g. decision-maker's preference) influence the final decision. With the final decision at hand all necessary agreements with the vendors can be made and the software solution can be purchased and implemented.

Now that a software selection methodology has been elaborated section 5.4 will discuss evaluation criteria in more detail corresponding to phase four of the selection methodology. Section 5.5 will then elaborate evaluation approaches based on the following criteria set referring to phase 5 of the selection methodology.

5.4 Software Evaluation Criteria

Literature proposes a vast amount of criteria to evaluate different types of software ([34], [84], [47]). The aspiration for the evaluation criteria is to be MECE with respect to the type of software project delivered. This section discusses relevant criteria and criteria groups identified in literature review with regards to the scope of this project. The criteria are described in table 5.1 whereas the following remarks need to be taken in to account

Functional criteria fully depend on the specific software requirements and cannot be generalised. Criteria in terms of benefits in contrast to most literature are not included in the criteria list. It is assumed that all software packages should should deliver the same benefits if they are fulfilling all requirements. If direct and indirect benefits differ for different software solutions corresponding evaluation criteria are introduced during the prototype and feasibility study.

5.5 Software Evaluation Methodology

The evaluation and selection of a software package is very complex and time consuming process and can be formulated as Multi Criteria Decision Making (MCDM) problem [46]. Now that a

Criteria group	Criterion	Description	Metrics
Functionality	Included functionality	Individual functional areas the software has to cover. Different functional criteria are elaborated in a real world example.	Specific[1]
Quality	Technology and programming language	Is state-of-the-art technology, architecture and programming language used that is most suitable for the requirements?	Qualitative[2]
Quality	Security	How much security features does the software offer in terms of encrypted connections, data storage, logging and auditing capabilities, user rights management etc.?	Qualitative[2]
Quality	Adaptability and source code quality	How structured and well-written is the source code? Are best practice design patterns used? Is the source code sufficiently commented and is it easy to adapt and maintain? Is the software considered bug-free?	Qualitative[2]
Quality	Reliability	How robust is the software, thus running without crashing?	Qualitative[2]
Usability	User interface and ease of use	How easy is it for a (new) user to navigate and perform actions in the software? How attractive is the design of the user interface?	Qualitative[2]
Usability	Error reporting	Is there an error reporting functionality for the user as well as the administrator?	Boolean
Vendor	References	Number of references for a certain software package.	Qualitative[3]
Vendor	Demo	Availability of software demo and free-trial	Boolean
Cost	License cost	Software license costs in £	Numerical
Cost	Installation and implementation cost	Cost in terms of money or man-days, whereas man-days can be transformed into monetary terms.	Numerical[4]

Table 5.1: Evaluation criteria and metrics.

[1] depends on the respective functionalities in the specific requirement context.
[2] Very poor, poor, fair, good, very good
[3] Very poor (<3), poor (4-10), fair (11-20), good (21-30), very good (>30)
[4] In £, man-days or even a qualitative scale (see[2]). Metric depends on the detail of project planning.

selection methodology and software criteria have been defined the actual evaluation technique needs to be chosen. Jadhav and Sonar [47] state that the Weighted Scoring Method (WSM) and the Analytical Hierarchy Process (AHP) have been widely adopted successfully for the evaluation of software packages. Another more complex method applied to this type of MCDM problem is a Knowledge-Based Systems (KBS) approach [47]. A KBS is an artificial intelligence system employing Case-Based Reasoning (CBR) and Rule-Based Reasoning (RBR) in oder to support the decision making process [71]. Due to the lower complexity of the decision in this project only WSM and AHP are considered relevant and will be explored further.

5.5.1 Weighted Scoring Method (WSM)

The idea of WSM is a rather simple. It is assumed that there are n alternatives $\{A_1, A_2, \ldots, A_n\}$ for a decision that are evaluated using m different criteria $\{C_1, C_2, \ldots, C_m\}$. Each alternative needs to be rated for each criterion numerically with a score S_{ij}, $\{S_{ij} | i = \text{index of } A_i \wedge j = \text{index of } C_j\}$. Attributing relative weights $\{W_1, W_2, \ldots, W_m\}$ to each criterion the final score for each alternative $S(A_i)$ is then calculated as weighted sum of its criteria scores as shown in equation 5.2. [47]

$$S(A_i) = \sum_{j=1}^{m} S_{ij} W_j \tag{5.2}$$

The simplest version of WSM attributes a uniform weight for each criterion, which is equivalent to the mean value of the criteria scores as shown in equation 5.3.

$$S(A_i) = \frac{1}{m} \sum_{j=1}^{m} S_{ij} \tag{5.3}$$

WSM Example

Let's assume a simple decision to buy a new car. There are two alternatives $\{A_1, A_2\}$ with A_1 as a premium sports car and A_2 as a cheaper family wagon. The three decision criteria are {Price, Engine, Brand}. The decision-maker rates the importance of the weights at $\{50\%, 30\%, 20\%\}$. A Likert-type scale [15] is used to assign a score for each criterion. The decision-maker evaluates the statement, *"Alternative A_i has an excellent C_j"*, for each alternative and criterion with the options {*Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree*} that correspond to numerical values of 1 to 5.

	Price	**Engine**	**Brand**	**Weighted Score**	**Mean Score**
A_1	2	4	5	3,2	3,67
A_2	4	3	2	3,3	3,0

Table 5.2: WSM example scores.

The scores for each criterion and overall scores for both weighting methods are shown in table 5.2. Option A_2 outperforms using different criteria weighting due to the *price* focus whereas A_1

outperforms for the uniform weighting approach. The example shows the weakness of the WSM in the arbitrary assignment of weights. Decreasing the weighting for *price* by only 3% attributing it to *brand* would change the ranking of the alternatives. This observation is supported by Maiden and Ncube [60] who state that using the simple WSM approach, weightings are often inconsistent and do not show clearly what are the most essential requirements for a software.

5.5.2 Analytical Hierarchy Process (AHP)

The AHP offers a more organised and approach to assign weights and scores based on pairwise comparison. The approach was pioneered by [81]. The idea is to construct a decision hierarchy with the goal of the decision at the top, followed by the criteria and sub-criteria and the alternatives at the bottom level [82]. Based on the previous example such a hierarchy is shown in figure 5.2.

In the next step a set of pairwise comparison matrices is constructed. To do this, each element is subject to pairwise comparison with all elements at one level with respect to each element of the higher level [82]. That means all criteria need to be compared pair-wisely with respect to the goal, and all alternatives with respect to each criterion.

Intensity of Importance	*Definition*	*Explanation*
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgement slightly favour one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgement strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation
Reciprocals of above	If activity *i* has one of the above non-zero numbers assigned to it when compared with activity *j*, then *j* has the reciprocal value when compared with *i*	A reasonable assumption
1.1–1.9	If the activities are very close	May be difficult to assign the best value but when compared with other contrasting activities the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities.

Figure 5.1: Saaty's AHP scale of fundamental numbers. (Source: Saaty [82])

To do this pairwise comparison Saaty [82] devises a scale of numbers shown in figure 5.1. Assuming n alternatives and m criteria, the comparison matrices are then constructed as follows. $a_{i,j}$ denotes the pairwise comparison value between alternative A_i and A_j considering a criterion C_i. With the decision-maker assumed to be consistent the value of the inverse comparison results

to $a_{j,i} = \frac{1}{a_{i,j}}$, and $a_{i,i} = 1$ when compared to itself [47]. The resulting matrix for all alternatives considering criterion C_i is shown in equation 5.4.

$$A_{n,n}^{C_i} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,n} \\ \frac{1}{a_{1,2}} & 1 & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1,n}} & \frac{1}{a_{2,n}} & \cdots & 1 \end{bmatrix} \tag{5.4}$$

The comparison matrix for all criteria compared with respect to the goal is constructed in the same way as $A_{n,n}^{C_i}$. The resulting pairwise criteria matrix is shown in equation 5.5. In total AHP requires $m(\frac{n(n-1)}{2} + \frac{(m-1)}{2})$ pairwise comparisons for all criteria with respect to all alternatives and all criteria with respect to each other [47].

$$C_{m,m} = \begin{bmatrix} 1 & c_{1,2} & \cdots & c_{1,m} \\ \frac{1}{c_{1,2}} & 1 & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{c_{1,m}} & \frac{1}{c_{2,m}} & \cdots & 1 \end{bmatrix} \tag{5.5}$$

Now in the next step Saaty [81] proves that the principal eigenvector v_{max} (i.e. the eigenvector corresponding to the maximum eigenvalue λ_{max}) reflects the decision maker's actual opinion and relative weighting of criteria and the alternatives considering a certain criteria. By normalising v_{max} by the sum of its elements, the priority vector p is calculated. Let p_c denote the priority vector for all criteria, and p_j the priorities of each alternative A_i for criterion C_j, the total AHP score for each alternative $S(A_i)$ is shown in equation 5.6.

$$[p_1, p_2, \cdots, p_m] \times p_c = \begin{bmatrix} S(A_1) \\ S(A_2) \\ \vdots \\ S(A_n) \end{bmatrix} \tag{5.6}$$

AHP Example

Applying AHP for the 'buying a car' example, the problem is structured as shown in figure 5.2. The comparison matrices for the alternatives with respect to each criterion and the corresponding priorities are shown in table 5.3. The criteria comparison matrix, the resulting criteria priority and the overall scores for the alternatives are shown in table 5.4. In total $3(\frac{2(2-1)}{2} + \frac{(3-1)}{2}) = 6$ pairwise comparisons need to be filled for this simple example, but this number grows polynomially with the number alternatives and criteria. Not only the weighting but also the overall weighting is much more significant compared to WSM. In this example *Car A2* is clearly recommended by the AHP with a score of 72%.

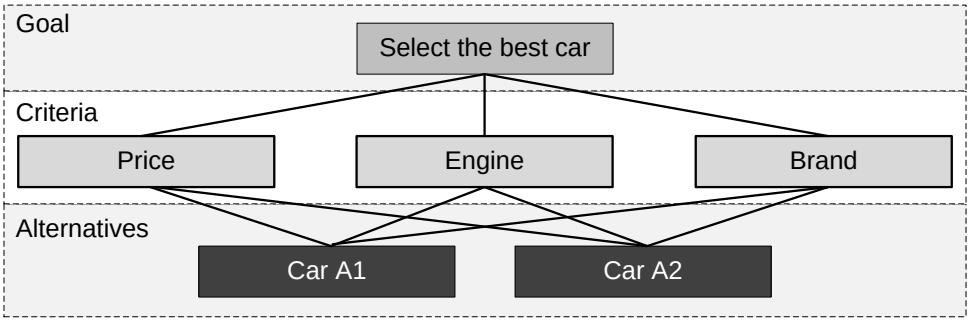

Figure 5.2: Decomposition of the AHP example problem into a hierarchy. (Source: adapted from Saaty [81])

Price	*Car A1*	*Car A2*	*Eigenvector*	*Priority*
Car A1	1	1/8	0,124	11%
Car A2	8	1	0,992	89%
	λ_{max}	2		
Brand	*Car A1*	*Car A2*	*Eigenvector*	*Priority*
Car A1	1	5	0,981	83%
Car A2	1/5	1	0,196	17%
	λ_{max}	2		
Engine	*Car A1*	*Car A2*	*Eigenvector*	*Priority*
Car A1	1	1/4	0,243	20%
Car A2	4	1	0,970	80%
	λ_{max}	2		

Table 5.3: Comparison matrices, eigenvectors and priority vectors for alternatives with respect to each criterion of the AHP example.

Criteria	*Price*	*Engine*	*Brand*	*Eigenvector*	*Priority*	**Car A1**	**Car A2**
Price	1	4	6	0,946	69%	11%	89%
Engine	1/4	1	3	0,298	22%	83%	17%
Brand	1/6	1/3	1	0,125	9%	20%	80%
	λ_{max}	3,05			**Total Score**	28%	72%

Table 5.4: Criteria comparison matrix, criteria priority vector and overall score of alternatives for AHP example.

Chapter 6

Software Verification and Validation

Software verification and validation (V&V) is the critical process to ensure that a software meets the requirements and provides the functionality according to the customer's expectations [88, pg. 515]. The process is performed during as well as after the implementation and often more than 50% of a software project's costs are spent on V&V [65, pg. xiii].

The terms of verification and validation are often confused they need clarification. IEEE Standards Board [44] defines *verification* as

> "The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase."

wheras *validation* is defined as the evaluation of the software at the end of the implementation in order to ensure it is meeting the user requirements [50]. Boehm [9] (cited in Sommerville [88, pg. 516ff.]) defines these terms more concisely:

- "Validation: Are we building the right product?"
- "Verification: Are we building the product right?"

Thus, verification checks that the software or its components meets its functional and non-functional requirements, hence the correctness of the software [67, pg. 7f.]. Validation on the other hand is concerned with the product meeting customer expectations. In short, the ultimate goal of the V&V process is to ensures that the software product is 'fit for purpose' [88, pg. 516].

6.1 Verification and Validation Approaches

Software V&V comprises two different approaches. The differentiation depends on the fact if for the V&V process the software has to be actually run or not [67, pg. 7].

Static analyis as the term 'static' suggests is concerned with checking the software in terms of requirements documents, design documents and source code [67, pg. 7]. To do so, the software doe not need executed. Static analysis methods comprise software inspection, peer reviews, walk-through or algorithm analysis. It involves reviewing the source code and reasoning over all possible system behaviours during run-time and it is performed throughout all stages of the software development process. [88, pg. 517], [80], [6]

Dynamic analysis on the other hand involves execution of the actual software while checking its output and behaviour. Using test data the behavioural and performance properties of a system in operation are observed in order to expose possible failures [6]. *Testing* is the common term used for dynamic verification and validation [88, pg. 517].

Static methods and testing are complementary approaches for V&V whereas both must be performed repeatedly and alternated [67, pg. 7]. Nevertheless, from a practical point of view based on empirical studies for defect detection strategies in software Runeson et al. [80] recommend to use static analyis such as inspection particulary for defects in requirements and design, and dynamic analysis, i.e. testing, for defects in the source code. The following sections describe testing as essential software verification and validation method in more detail.

6.2 Software Testing

Myers et al. [65, pg. 6] in his book *The Art of Software Testing*, first published in 1979, defines testing as "the process of executing a program with the intent of finding errors", whereas 'errors' should be interpreted broadly for the verification as well as validation context. The author constitutes this definition as crucial considering the psychological implications and economics of testing. This definition ensures that the tester will not subconsciously design test cases that are not comprehensive enough, since based on the concept of self-fulfilling prophecy the aim is to develop tests that will *not* pass successfully [67, pg. 10].

Software testing is the last opportunity to correct (potential) software defects at low cost. This implies that software defects or requirements misunderstandings that are discovered only during system operations are much more expensive to correct (Davis [19] cited in Juristo et al. [51, pg. 8]). Perry [72, pg. 67] states the costs to fix an error in the operations phase are 12.5 times higher compared to the actual implementation or coding phase.

An essential fact of software testing is that software can not be exhaustively tested [65, pg. 10ff.]. Just considering the domain of different input possibilities for a system, it is usually too large to be exhaustively tested since the possible input configuration are nearly infinite [67, pg. 13]. For that reason, it can never be guaranteed that a software is completely error-free, testing needs to be considered from an economic point of view. Thus, the goal of testing should be to maximise the 'return on investment of testing' in terms of number and criticality of errors found traded off against the resources invested to design and execute tests [65, pg. 5].

The goal of testing is to detect as many defects as possible with as few test cases as possible [48, pg. 236]. In test design different information is incorporated such as software specification,

features of the input and output domain of the system and the source code itself [67, pg. 20f.]. These complementary information sources are used to feed into the two major testing concepts: functional and structural testing ([6].

6.2.1 Functional Testing

Testing the functionality of a system no knowledge of the system's internal logic or structure is necessary [72, pg. 69f.]. The system can be tested or test cases can be developed simply considering the functionality of the system stated in the requirements specification [75, pg. 448f.]. This is why this approach is also called *black-box testing*. Similarly the strategy is also named *behavioural*, *data-driven*, or *input/output-driven* testing [65, pg. 9ff.]. With each test performed a specific functionality of the system is validated without being concerned with the internal matters of the system [75, pg. 448f.].

6.2.2 Structural Testing

Black-box testing is only concerned with the behaviour of the system ignoring its structure. *Structural testing* on the other hand is concerned with the structure, i.e. the actual implementation of the software [88, pg. 557f.]. The primary focus here lies on the source code and its control and data flow [48, pg. 248]. Given the visibility of the internal structure of the system the test cases are designed. This is why structural testing is often referred to as *white-box* or *glass-box testing* [88, pg. 557] because now the tester is concerned with what is going on inside the system.

It is important to note that the scope of functional and structural testing is different [67, pg. 21]. Structural testing is applied to individual components of a system in order to check its control and data flow at source code level. Functional testing on the other hand being only concerned with a component's or system's external interfaces and input and output is applied to components as well as combination of components and the whole system. [67, pg. 21]

To provide an effective testing method both structural and functional testing must be employed at the same time [6]. The employment of only structural testing does not discover a lack of source code that should handle a possible condition (missing path), and the employment of only functional testing will disregard many possible defects at source code level [67, pg. 21].

6.3 Testing Strategy

For testing to be most effective it must start at the beginning of the project and be carried on in parallel with the development of the software [67, pg. 16]. Software needs to be tested, that means verified and validated, on different levels. The approach is called V-model and is depicted in figure 6.1 [17, pg. 101].

The different stages of testing in a bottom-up manner are known as *unit*, *integration*, *system* and *acceptance* testing [67, pg. 16f.]. Nevertheless, the testing plans for each stage should be

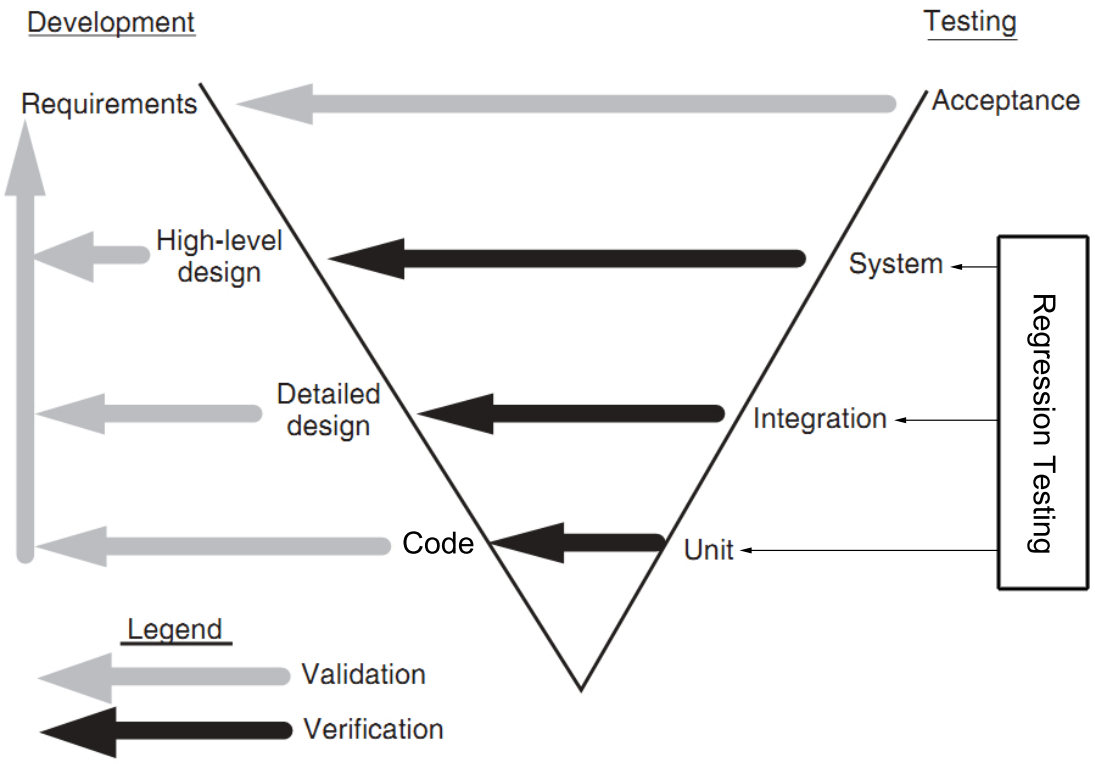

Figure 6.1: Development and testing phases in the V-model. (Source: adapted from Naik and Tripathy [67, pg. 16] and Bertolino and Marchetti [6])

developed in reverse order of the execution, since the requirements are available first [72, pg. 101]. Going down the left-hand side of the V-model each design and development phase is based upon the earlier stages. Thus, the corresponding testing plans for each level also depend on all previous design phases.

The first three testing phases, unit, integration and system testing, are used to verify that the different levels of the software satisfy the conditions imposed on them. The tests proof the formal correctness of the corresponding software design layers. [67, pg. 16f.]

The aim of the acceptance testing stage is to validate the software in terms of user requirements [72, pg. 102ff.]. In contrast to the earlier stages, this phase does not verify the actual formal correctness of the system at all levels, but ensures that the system fulfills the customer's quality expectations [67, pg. 17]. Nevertheless, as shown in figure 6.1 by means of the grey arrows, eventually all phases of software verification feed into the validation of the system in terms of requirements.

The following sections will outline the different testing stages and associated regression testing in more detail.

6.3.1 Unit Testing

Unit testing is concerned with testing each unit or component of the software as implemented in the source code [75, pg. 473f.]. It represents the lowest level of testing since the actual source

code components are verified. Unit tests are designed to focus on testing one single component while ignoring the rest of the system [79]. They are conducted to test the program's structure in terms of white-box testing and are often used in test-driven design approaches [16] where unit tests are written before the actual components are developed .

This '*test-first*' approach is particularly advocated in eXtreme Programming (XP), an agile software development methodology [16]. In practice unit tests are implemented with the help of unit testing frameworks, as for example JUnit[1] for Java. These frameworks simplify the process of unit testing such as implementation, automation and reporting [67, pg. 73f.].

6.3.2 Integration Testing

A software system is developed as a collection of components and processes. Once the individual pieces are implemented and verified using unit tests, they need to be tested in working together [73, pg. 345]. Combining different components of the product and testing them is referred to as *integration testing* [75, pg. 345f.]. Components and processes can either be incrementally combined and then tested (*incremental integration testing*), or they are not tested until full system integration (*big bang integration testing*) [52, pg. 44]. The latter is often used in order to save costs to write test scaffoldings [73, pg. 345] but is not recommended in literature [52, pg. 45]. In the big bang approach it is difficult to localise what actually caused the failure, programmer team dynamics could deteriorate in finding the responsible code fragments and automation is difficult since a software system changes continuously over the development life cycle. [52, pg. 45f.]

6.3.3 System Testing

Once all (sub-)systems are integrated *system tests* simulate the operation of the entire system [72, pg. 70]. Integration tests provide a reasonably stable system whereas system testing verifies that the whole system runs correctly [67, pg. 17]. Functional as well as various non-functional system features are tested, such as security, robustness, load and stress capabilities, stability, performance and reliability of the system [75, pg. 483ff.]. Systems tests are critical since they are the final phase of software verification. They ensure that the system is almost 'fault-free' (see chapter introduction for the concept of 'complete software testing') before it is validated in user acceptance testing. [67, pg. 17]

6.3.4 Regression testing

Software development whatever methodology is applied is an evolutionary process [6]. Throughout the whole process components are added, modified and integrated. To ensure that all the previously implemented components are still working, and that modifications do not have any negative side effects, they have to be retested [65, pg. 147]. Saving previous tests and re-running them is called *regression testing* [6]. They determine if any modifications regress to any other

[1] http://www.junit.org/, accessed 15.05.2011.

aspects of the system [65, pg. 147]. Figure 6.1 shows that regression tests are run in parallel to all software verification stages in the V-model and are not considered as a separate level of testing [6].

6.3.5 Acceptance Testing

In acceptance testing or User Acceptance Testing (UAT), the last testing stage, the goal is to validate the whole system with reference to the requirements specification. Essentially the quality of the product towards the user expectations is examined rather than the detection of defects. [67, pg. 17]

For acceptance testing usually realistic client data in the eventual client setup is used in order to demonstrate that the system provides what it actually has been commissioned for and what the client is expecting from it. [48, pg. 230ff.]

Usually the customer or its end user perform the acceptance testing under the supervision of the design team. The tests focus on major functional and performance requirements, usability and man-machine interaction, external interfaces, reliability and specified system constraints. [42]

During acceptance testing the test cases for different scenarios are usually organised in a test suite. The challenge is to make sure that the test suite covers all the requirements (completeness). The customer should have software his own acceptance criteria ready based on which he accepts or rejects the software system for further development. [67, pg. 17]

Sofware Usability

Considering usability as major factor in user acceptance testing different approaches are mentioned in literature. Besides expert reviews and field usability testing with the actual user base, *thinking-aloud (THA) usability tests* are the most dominant approach [70]. Holzinger [41] even calls it "the single most valuable usability engineering method". The users verbalise their thoughts while using the system [70].

THA enables the developers to understand how the users perceive each interface item and it identifies which parts in the man-system interaction cause the most problems. It also reveals why users want to interact in a certain way with the system providing hints on the actual use of the system in practice. In addition to that, in contrast to field testing only a small number of users needs to be engaged in THA to gather significant results. [41]

Part II

Feasibility Study & Prototype

Chapter 7

Methodology Overview

The area of IVHM is rapidly evolving and is facing challenges in collaboration and doing business between buyers and suppliers. A business-to-business platform tackling these challenges for the IVHM community does not exist at this point in time. Even though the context of such a platform is still evolving its feasibility shall be evaluated. To do so the business challenges need to be analysed in detail and a software prototype is developed. Therefore, a research methodology is applied that incoporates those approaches discussed in the literature review which are most suitable to support this project. In the following sections the research methodology is elaborated.

In order to **analyse** and **understand the business context** and the challenges in IVHM buyer-supplier relationships *semi-structured interviews* with key responsibles from both buyer as well as supplier side are employed. This analysis then feeds into what problems actually should be solved in terms of 'doing the right thing' and the requirements for the software platform.

'Doing the thing right' in terms of the development of the prototype the commonly known *waterfall model* [88, pg. 66-68] is employed as **software development methodology** building upon an existing software package. The single-developer setup with a clear time frame and the requirements documentation already required for the initial software package selection make this approach most appropriate for this project.

With the business context analysed, the actual initial **software requirements** are explored and documented based on the *IEEE Software Requirements Specification* template which is discussed in section 4.4.1. The template is considered as an industry-wide guideline and best practice for software requirements documentation. Thus, the initial SRS with a concise *use case* analysis lays the foundation for the choice of software package and technology and the actual software implementation.

Based on the initial SRS *software evaluation criteria* are developed. After a *market research* of available software packages and frameworks these criteria are fed into a **software evaluation and selection methodology**. Both the *Weighted Scoring Method* and the *Analytical Hierarchy Process* are employed and compared in order to evaluate and select the most appropriate solution. Other software evaluation and selection methodologies were not considered since they are too

complex and do not deliver additional benefit for making this decision in the context of this platform prototype.

A *gap analysis* then identifies the functional gaps of the selected solution. During the subsequent implementation phase best-practice methods of *system architecture design*, *database design*, *software architecture design* including *software design patterns* and *user interface design* are adopted in order to deliver a high-quality software prototype. A good software design and documentation already for the prototype reduces the costs for a potential follow-up development of a final software product.

Software tests and validation with the client ensure that the 'right thing' was 'done right' along the whole software development life cycle. Therefore *unit-testing* at source code level, *integration and system testing* along with *regression tests* are employed in order to provide verification of the software platform. To evaluate and validate the software towards client expectations *User-Acceptance Testing* based on the *System Usability Scale* as *test-user questionnaire* and *Thinking-Aloud* tests are adopted.

With the prototype proving success to the platform's technical feasibility, the subsequent pilot phase allows to evaluate the economic feasibility. However, this analysis serving as a decision-making basis for the actual implementation of a final software product is a follow-up task to this project and thus not included in the scope of this document.

Chapter 8

Business Context

This chapter provides the results of the analysis of the business context for the IVHM B2B platform. Section 8.1 sets the context in terms of the environment where the platform is deployed. The remaining sections (8.2 - 8.3) summarise the findings of interviews with key employees of major IVHM system suppliers and customers defining the business context for the platform.

8.1 Cranfield Integrated Vehicle Health Management Centre

Cranfield University and the multinational aerospace and defence corporation, The Boeing Company[1], founded the Cranfield IVHM Centre[2] supported by the East England Development Agency (EEDA) in 2008. The aim of the centre is to deliver generic IVHM solutions based on highly leveraged core research and to facilitate commercialisation of developed IVHM technologies. [18]

Logo. (Source: Cranfield IVHM Centre (2011))

Figure 8.1 puts the focus of the IVHM Centre in context based on the Technology Readiness Level (TRL). TRL is a concept promoted by some US government agencies to evaluate the maturity of evolving technologies before they are integrated into a (sub)system. It defines nine levels of technology maturity starting from the understanding of basic principles as foundation for a technology (TRL 1) to the highest level, a system being proven through successful operations (TRL 9). [99]

The IVHM Centre as shown in figure 8.1 focuses on highly leveraged, pre-competitive (core) research (TRLs 4-6) and leveraged commercialisation (TRLs 7-9) of IVHM technology. Along all TRL levels an effective technology transfer to industry is facilitated.

Since the launch of the centre a number of additional partners from various industries have joined [18] (in alphabetical order):

[1] http://www.boeing.com/
[2] http://www.cranfield.ac.uk/ivhm/

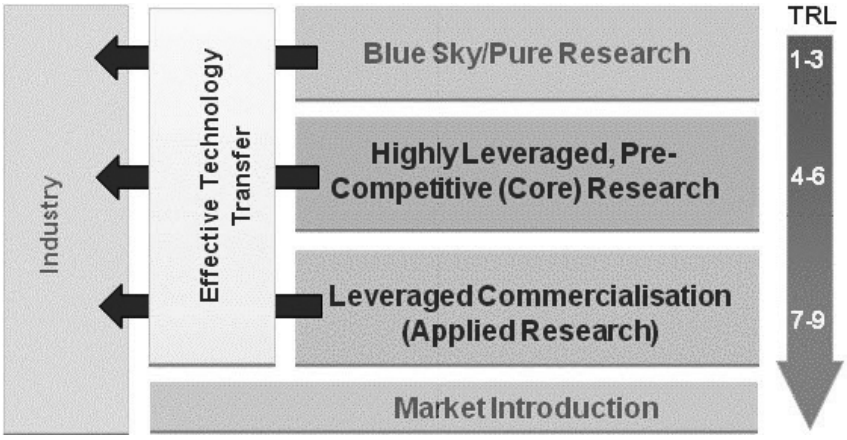

Figure 8.1: Cranfield IVHM Centre business model. (Source: adapted from Cranfield IVHM Centre [18])

- *Alstom Transport*[3], a divison of French multinational conglomerate Alstom S.A., supplying railway systems, equipment and services.
- *BAE Systems plc*[4], a British multinational aerospace, security and defence corporation.
- *EPSRC*[5], the British government agency Engineering and Physical Sciences Research Council.
- *Meggitt plc*[6], a British-based aerospace equipment supplier.
- *UK Ministry of Defence*[7].
- *Rolls-Royce plc*[8], a British-based global power systems and services company providing solutions for aerospace, marine and energy markets.
- *Thales S.A.*[9], a French electronics company providing IT systems and services for the aerospace, defence and security sector.

The Cranfield IVHM Centre leverages the collaboration between academia and its industrial partners to develop sustainable IVHM technology that is commercially deployed. Industrial and academic researchers, engineers and business experts provide expertise in systems engineering and architecture, system integration, analysis and algorithm development, demonstration and fast prototyping, business modelling, simulation and business transformation. [18]

[3]http://www.alstom.com/transport/
[4]http://www.baesystems.com/
[5]http://www.epsrc.ac.uk/
[6]http://www.meggitt.com/
[7]http://www.mod.uk/
[8]http://www.rolls-royce.com/
[9]http://www.thalesgroup.com/

8.2 IVHM System Components

In figure 2.1 the basic architecture of an IVHM system is illustrated. In order to provide the required functionalities in this architecture certain technical system components are necessary. An IVHM system's major technological elements are outlined as follows:

- Sensors
- Raw data preprocessing unit including for instance charge amplifiers, low-band filters, analog-digital converters and the required electronic circuitry
- Feature extraction unit with signal processing hardware and software
- Data processing unit for diagnostics, prognostics and potentially decision reasoning consisting of:
  - Hardware platform,
  - Operating system for the hardware platform and
  - Processing software
- Electronics and required (wireless) communication systems to integrate all components and to connect to central data systems (on-board or base station)

Depending on the environment where the system is deployed and the consequential (strict) requirements in cost, functionality, power usage, cooling, wiring, weight and size etc., the setup may vary [8]. For example, the feature extraction unit could be incorporated in the general data processing unit.

Also the requirements for the setup vary vastly since for instance an IVHM for airborne vehicles has totally different requirements in terms of e.g. weight, volume, power consumption and network connection (for instance Avionics Full-Duplex Switched Ethernet (AFDX)) compared to an industrial system monitoring e.g. a power generation turbine. In the industrial setup factors such as weight and size do not matter as much and most of the data processing is performed in a remote data centre using simpler networking technology.

Concluding, the full configuration of an IVHM system as depicted in figure 2.2, where and how much data processing is done, on-board or remote, if data is transmitted during operations or retrieved with the vehicle grounded, determines the actual setup of the IVHM architecture.

8.3 IVHM Buyer-Supplier Relationships

When corporations do business with each other, various factors such as type of product/service delivered or bargaining power on either side influence the B2B relationship. The features of the business relationships between buyers and suppliers in the IVHM system context are explored in the following sections.

8.3.1 Business Scenarios

Understanding the drivers and benefits of IVHM systems (see sections 2.1.2 and 2.1.3) the ultimate goal of an IVHM system is to generate value. Since IVHM systems often have to be integrated into a product from the beginning and its research and development costs are high, they are only deployed in critical areas where the business case is profitable. Thus, any development of an IVHM system starts with a modelling of the business case.

Standardisation and requirements of IVHM systems and components then strongly depend on their deployment environment and the industry. Thus, also buyer-supplier relationships vary from joint development of a full system to simple cost-driven relationships for COTS products as for example sensors. Figure 8.2 illustrates the spectrum of business relationships between both parties.

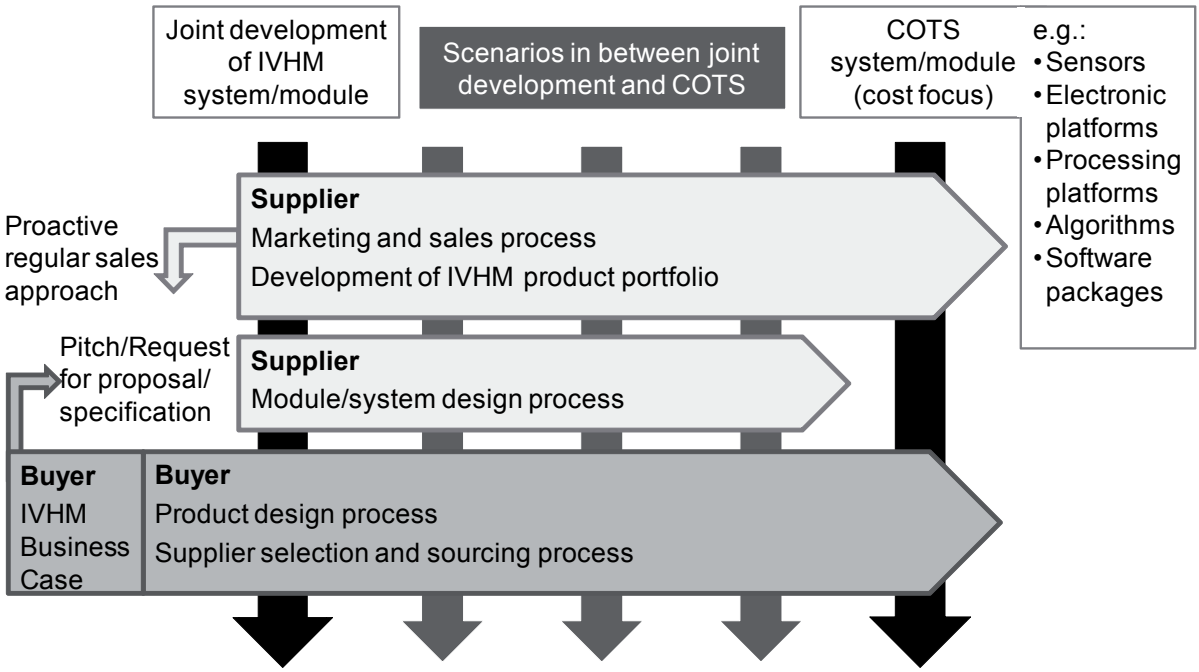

Figure 8.2: IVHM buyer-supplier relationship.

In the beginning either the supplier proactively approaches the potential buyer through its regular (often informal) sales and marketing process or the buyer approaches its suppliers directly. The former approach helps suppliers to elicit information and understand its buyers better, whereas the latter is more often the case. There, for instance an aerospace manufacturer approaches the (potential) supplier with a fixed specification, issues a Request For Availability (RFA) or Request For Proposal (RFP), or even has different suppliers pitching for a solution.

In a joint development even the business case might be developed together in conjunction with price negotiations. The system is engineered and integrated collaboratively in the target platform. This is mostly the case for IVHM systems with strong requirements such as airborne vehicles.

On the other end of the spectrum standardised COTS IVHM components and products are consi-

dered. For sensors, standardised hardware platforms, or software packages usually cost-focussed business relationships are observed. Especially in an industrial context where requirements are lower, systems tend to be more standardised and prices need to be fixed for a sound business case calculation, business relationships are more inclined to the end of the spectrum.

8.3.2 IVHM System Sourcing Case Studies

To illustrate the variations of IVHM sourcing three sample systems for British and US engine manufacturers that are supplied with certain components from an IVHM supplier will be discussed. Figure 8.3 illustrates the examples with the IVHM components supplied. The components marked with a numbered circle are supplied to the aircraft manufacturer.

1. **Case study**
   - Buyer defines and provides electronics box and data processing unit.
   - Supplier only implements software algorithms and tests with buyer? test data.
2. **Case study**
   - Buyer only provides health processing software.
   - Supplier provides sensors, on-board electronics, preprocessing unit and processing hardware platform.
3. **Case study**
   - Buyer provides preprocessing unit, processing hardware platform and processing software.
   - Supplier provides sensors, electronics and operating system for processing hardware.

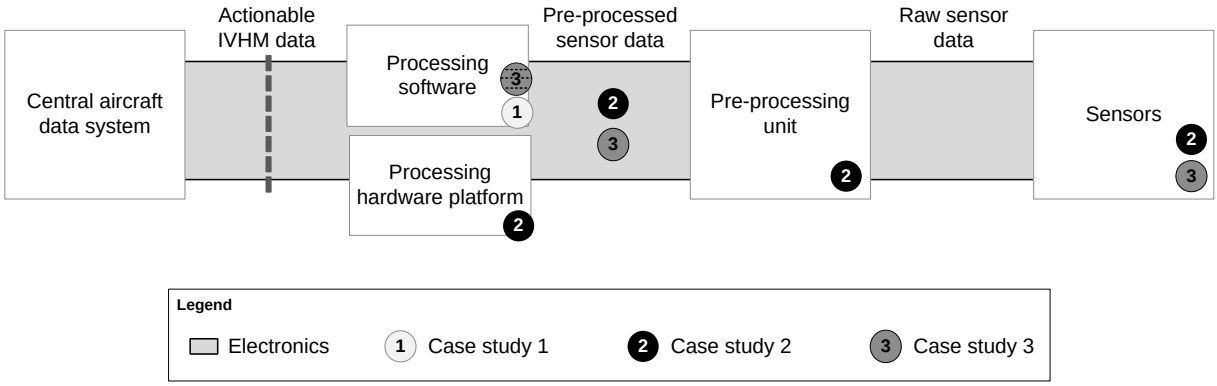

Figure 8.3: Sourcing examples for aircraft engine IVHM components.

Even though the IVHM supplier is integrated into the development from the beginning only certain parts of the system are supplied.

8.3.3 IVHM Systems for Different Industries

The requirements for IVHM systems for different industries vary significantly. In the following paragraphs the properties of two extremes of IVHM systems are summarised.

- Industrial IVHM system
  (e.g. gas turbines in power plants)
  - Highly standardised systems and products.
  - Low requirements in terms of e.g. size, weight or power usage.
  - Buyers prefer a fixed system setup up with fixed costs for business cases evaluation. Business relationships vary from supplying only COTS modules to full service IVHM systems. For instance:
    * only supply of sensors to
    * full system setup including service of data centre operations in dedicated supplier data centres (example: power plant with IVHM systems and computer racks in one country; data preprocessed and fed back via internet to IVHM supplier data centre in another country).
- Airborne IVHM system
  (e.g. aircraft engine monitoring)
  - Mostly highly customised systems and often single sourcing (development to specification).
  - Strict requirements in terms of size, weight, power consumption, environment and electronic interfaces.
  - Business relationships tend to be of early R&D-to-specification type during the design of the end product.
    * In the civil aircraft sector a great proportion of IVHM R&D costs and risks are often transferred to the supplier due to the airframers' bargaining power.
    * In the military sector sometimes public R&D funding is available.

Concluding, only in rare cases the full IVHM system is supplied by one company. Depending on intellectual property issues (e.g. test data), competency and cost, different components are sourced from different IVHM suppliers.

8.3.4 Complication

Business relationships between IVHM suppliers and OEMs vary from joint development to cost-focussed of-the-shelf purchasing of IVHM products. In the process of joint development, sourcing and selling of IVHM components buyers and suppliers face different complications.

IVHM Buyers

As already mentioned in section 2.2 business sourcing is based on trust, prior successful engagements and strong (informal) relationships with the suppliers. Thus, a buyer looking for a certain IVHM component or system would first contact previous suppliers if there is availability. This incorporates engineers to spend time on eliciting information from previous suppliers and worst case even searching in supplier directories or just the internet for potential suppliers of products with a certain specification.

Suppliers advertise their IVHM components and capabilities on (often outdated) brochures that can be downloaded from their website whereas IVHM capabilities that are spread out over different divisions of a supplier are difficult to find or only with enormous investment of time. Thus, there is low convenience and high search and transaction costs in gathering information on IVHM components and capabilities that are available on the market. Furthermore, in case a previous supplier can offer a certain capability, the engineers would not be aware of potential higher-quality or lower-cost products, since then no time on product research would be spent.

The complications for IVHM sourcing and development decisions on the *buyer* side can be summarised as follows.

- High transaction and search costs
- Low convenience in the interaction with suppliers
- Access to information of technology only for a few suppliers based on earlier relationships
- No transparency of potential suppliers of higher-quality, lower-cost or new-technology substitutes
- Low competition between suppliers due to already established relationships and thus potentially higher costs

IVHM Suppliers

Suppliers of IVHM components and technology face different complications. They need to keep up relationships with (potential) buyers and print costly product catalogues to be disseminated regularly. Due to the relatively fixed relationships with OEMs it is difficult to gain access to other potential buyers, in particular for small suppliers with potentially higher-quality products and capabilities. Information exchange on IVHM products is merely based on relationships so suppliers have difficulties to understand the OEMs better to improve and customise their products.

In summary, IVHM *suppliers* face the following complications.

- High transaction costs in sales and marketing processes to keep up buyer relationships
- Low convenience in interaction with buyers

- Access only to a limited amount of buyers
- Difficulties to broaden customer base and high entry barriers especially for small suppliers that potentially provide higher-quality products
- Difficulties to improve understanding of buyer requirements to improve and customise products

8.3.5 Implications of IVHM Buyer-Supplier Relationships

The complications for OEMs and suppliers match each other well in terms of information exchange and interaction. On the one hand buyers would like to broaden their market overview of suppliers whereas at the same time suppliers would like to access a wider range of OEMs.

The latter complications of buyers and supplier bear some strategical contradictions. The buyers would like to increase the competition for their suppliers while 'locking in' competitive-edge technology from certain suppliers that should not be shared with other OEMs. This contradicts with the goal of suppliers to broaden their buyer base and also sell to other OEMs in order to reduce dependency from a few OEMs.

To solve the mentioned issues in the context IVHM buyer-supplier relationships the concept of an intermediary will be explored in the following chapters. Nevertheless, the strategic implications have to be kept in mind throughout the development of such an intermediary.

Chapter 9

Software Requirements Analysis

This chapter defines the requirements for the B2B platform. The feature analysis of existing platforms sets the background for the precise documentation of the functional and non-functional requirements for the IVHM B2B platform.

9.1 Feature Analysis of Existing Marketplaces

In section 3.3 major existing online marketplaces were introduced. No marketplace dedicated to IVHM products and technology could be identified. At the same time existing marketplaces of related areas are solely focused on sales and purchase of products without facilitating the collaboration and the sharing of knowledge on systems and technology during the design process.

This section analyses the features of these marketplaces and serves as pre-study that feeds into the requirements specification for the IVHM B2B platform. Table 9.1 describes the features identified in the online marketplaces described in table 3.1 in section 3.3. All mentioned features seem beneficial and are integrated in the platform except for a *recommender system* for platform items and a full forum-like *community* which are not regarded to deliver additional value in this specific context.

9.2 Software Requirements Specification

The SRS for this project is documented based on the structure proposed by IEEE Computer Society [43] which is described in section 4.4.1.

9.2.1 Introduction

Purpose

The purpose of this document is to provide a precise and complete description of the functional as well as non-functional requirements for a prototype of the IVHM Centre Online B2B Platform.

Product catalogue	An intuitive categorisation of items in a catalogue allows simple browsing and navigation.
Simple search and advanced search	A search feature for simple keywords or even for specific item or supplier attributes (e.g. item type or certification) simplifies quick localisation of items.
Personal account	A personal account restricted by a login system offers platform personalisation and the possibility to save user related information and interests.
Wishlist/Watchlist	Items of interest can be saved in a a dedicated list in a user account for later reference.
Product comparison	The user can add (similar) items of interest to a comparison feature. The feature then provides a clear overview of the different attributes and descriptions of the items.
Shopping cart/Enquiry basket	Items to be purchased or a items that a marketplace user wants to enquire further information about are stored in this feature. With a 'basket checkout' or 'enquiry submission' the purchase is closed or the enquiry sent to the suppliers.
Recommender system	Based on previously viewed or searched products an artificial intelligence system recommends items a user could be interested in.
Item tags/Tag cloud	Items can be tagged by the suppliers as well as the users with key words. The tags are then represented in a tag cloud where more popular tags are shown bigger or in different color. By clicking on a tag the marketplace shows all related items.
Recently viewed/searched or new/popular views/-searches	Boxes on the user interface display recently viewed or searched items by the user, or new or popular items and searches on the whole marketplace. This way the user can easily refer to his previous marketplace interaction and always be up to date with new developments on the platform.
Comments and reviews on items	A commenting and reviewing system allows users to share their thoughts on certain items with other users on the marketplace. This also offers a way to interact with the supplier.
Community	Some marketplaces offer a digital forum for users to share and discuss their thoughts on items or general marketplace-related topics and the possibility to communicate with each other via a messaging system.
Company microsites	Suppliers host their own microsite on the marketplace to centralise and advertise all their products in one page inside the platform.
Item management	A supplier can manage his items that are advertised on the marketplace. A separate user interface allows the supplier to add new items and edit or delete existing ones.

Table 9.1: Features of major online marketplaces.

After the validation of the requirements with the key users and project stakeholders, this SRS serves in first instance as basis for the development of the functional prototype of the platform. Upon completion of the development the document is used to validate the software with the

requirements defined by this document.

The platform is then deployed on a test basis and used by the IVHM community. If the platform delivers its benefits and the business case proves profitability, this requirements document will be refined based on the test user feedback and platform experience. The refined SRS will then be used as foundation for a professional makeover of the platform. The makeover will provide a final software product that fulfills all refined functional as well as non-functional requirements, as for example security. The deliverable of a possible re-implementation of the prototype is a fully operational final platform product with enhanced features identified during the prototype testing and evaluation phase.

The intended audience of this document consists of all stakeholders of the IVHM B2B platform, in particular the key user groups at the IVHM Centre, the sponsor of this project and the developers.

Scope

The scope of this software is to provide a web-based information and interaction platform for the buyers and suppliers in the IVHM community at the IVHM Centre at Cranfield University. The major benefit of the platform is the central availability and the easy access to IVHM technology and product data from different suppliers. This shall facilitate the buyer-supplier communication in terms of IVHM products & capabilities and support the initiation of business between these parties.

The IVHM customer community has the possibility to review products and technologies in one central location and can feed back desired combinations of technology/modules and nice-to-have features. On the other hand, the platform supports the suppliers in order to improve their understanding of buyer requirements and provides a possibility to advertise systems beyond IVHM technology, for instance control systems components.

The platform shall facilitate the initiation of business but is not handling the actual business transaction processing. There will be no possibility to purchase or sell IVHM products directly on the platform. Organisations have their own processes and systems for order and payment processing. Thus, the integration of these functionalities is not in the scope of this platform.

A major goal during the development of the system is to minimise the development costs both in monetary terms, for instance license fees, as well as man hours since the scope is to develop system prototype that serves feasibility considerations and is not a final operational software product. Nevertheless, the (software) quality of the prototybe shall not be compromised by this constraint.

Definitions, acronyms, and abbreviations

Table 9.2 describes the terminology used in this document.

(Platform) Administrator	Responsible that is configuring and maintaining the IVHM platform at the IVHM Centre.
Buyer User	Company representative that is representing the buyer side of IVHM technology, for instance an aircraft engine or train manufacturer and has access to the front end of the platform.
Supplier User (Supplier Administrator)	Company representative that is representing the supplier side of IVHM technology, for instance a manufacturer of IVHM electronics or aircraft engine vibration sensors and has restricted access to the backend of the platform.
(Platform) User	General term for a user of the B2B platform.

Table 9.2: Software requirements specification document terminology.

References

- IEEE SRS Standard [43]
- The functional requirements in terms of use cases are described in section 9.3.

Overview

The remaining parts of the SRS are organised as follows: Section 9.2.2 provides an overall description of the software and describes the user characteristics. It specifies general constraints to be considered during the development of the platform and states inherent dependencies and assumptions made.

Section 9.2.3 delineates all functional requirements the software is expected to deliver including the description of use cases. Furthermore design constraints, performance requirements and other system attributes are discussed.

9.2.2 Overall description

Product perspective

The IVHM B2B platform is an online application for all users in the IVHM community. It is intended to bridge the gap between the suppliers of IVHM technology and their potential customers. The system has to be user-friendly, easy to learn and reliable for this purpose. It shall support any user platform, e.g. a Windows, Linux or mobile platform, that provides web functionality. The platform itself is a standalone application that is not dependent on any other software systems.

The system is based on a client-server architecture where a web server uses a database system as persistence layer and serves the users via web interface in their web browser. The basic architecture of the system is depicted in figure 9.1.

The database holds all static data of the B2B platform, as for instance user or product data. A priori there are no constraints in terms of database system from the client side. The most appropriate system is to be chosen in the software evaluation and selection phase.

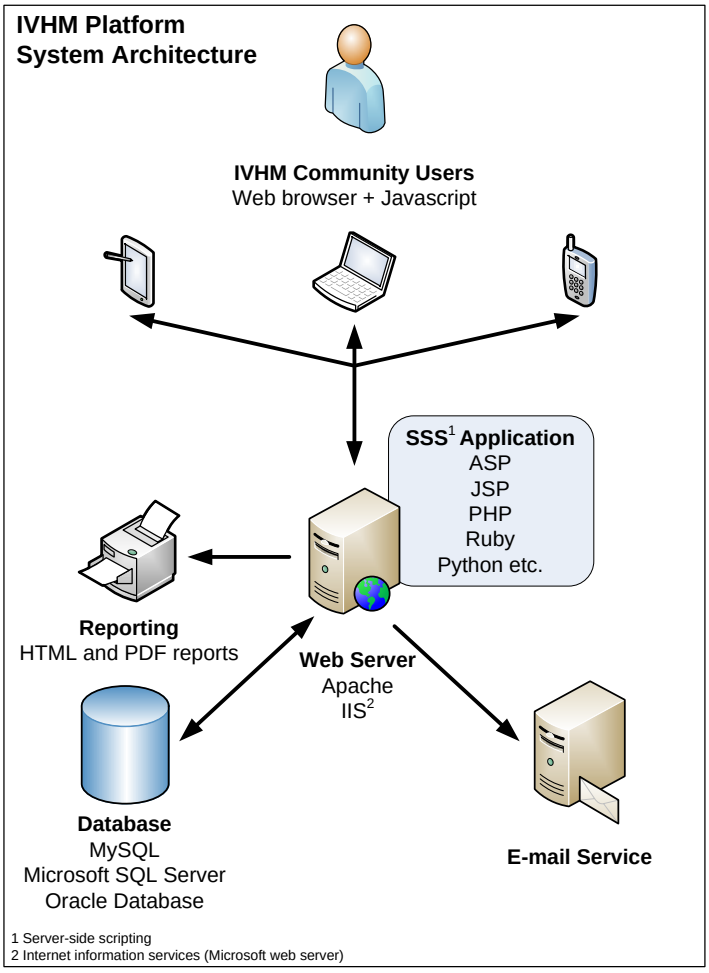

Figure 9.1: IVHM platform system architecture with possible technologies and vendors.

The same applies for the web server and the technology to be used. There are no constraints in terms of server application as well as server-sided scripting. The most suitable technology and programming language have to be identified for the system. The web server is using an e-mail service in order to notify and inform users and supports reporting functionalities to generate different kinds of reports.

Since the system is a web-based application there is no adaption for different sites necessary. All users inside the IVHM network can easily access the system via web interface using their preferred web browser.

Product functions

The initial functional requirements of the system inspired by the analysis of existing marketplaces in section 9.1 are described as follows:

- General platform functionality
  - Secure registration and access to the system for platform users. Possibility to manage and restrict access and permissions for users on the platform.

  - Presentation of IVHM products, technologies and capabilities with all specifications in an online platform, in a browser-independent web interface.
  - Storage of product and technology descriptions in a central database.

- IVHM buyers - 'Buyer users'
  - Possibility to browse through different categories and 'tags' of offered IVHM products and technologies.
  - Search mechanism for quick localisation of particular products or technologies.
  - Creation of an 'IVHM basket' where interesting products and technologies can be stored for follow up with the suppliers.
  - News functionality that notifies buyers of new IVHM products on the platform.
  - Buyer feedback and supplier contact features to easily get more information on certain technologies.
  - Buyer communication area that allows to define and communicate specific IVHM requirements.

- IVHM suppliers - 'Supplier users'
  - Adding, editing and viewing products and technologies including categorisation and tagging.
  - Interface to process customer feedback and questions.
  - Representation and analysis of platform usage data, for example which technologies are most popular, for administrator and suppliers on the platform.
  - Management and customisation of supplier micro site on platform.

- Platform administrator
  - Possibility to manage platform users (add, edit, view, delete). Definition and assignment of roles to grant users specific rights and permissions on the platform.
  - Interface to manage product categories and tags.
  - Interface to set up a new supplier including the establishment of a corresponding micro site.

User characteristics

- The user should be familiar with navigating and interacting with websites.
- The platform administrator and 'supplier users' that maintain micro sites for different organisations on the platform need to have basic web design skills in order to customise the micro sites. Knowledge in (X)HTML and CSS is favourable.

Assumptions and dependencies

- Product and IVHM technology details on the platform prototype are provided manually. An automated import function might be added later.
- Platform administrator is already set up in the system.
- Roles and user rights are already predefined in the system.
- The administrator manages the user base, roles and user rights on the platform.
- Setting up a new supplier on the platform the administrator is assumed to support the establishment of a supplier micro site. The supplier organisation is assumed to provide the necessary (X)HTML, CSS and image editing skills to support the creation of a micro site that complies with their corporate design standards.

9.2.3 Specific requirements

External interface requirements

The main interface to interact with the system is the browser interface. Users can enter, edit, view and delete data via web interface. Also platform reporting functionalities are provided via this interface.

In order to notify and send updates to users e-mails are used. The web server utilises an e-mail service to serve this interface.

For the initial platform data import a corresponding data import interface using scripting languages could be employed. The details of this interface are defined based on the data format provided by the client for the initial upload.

Functions

The basic functional requirements what the system shall do is described in section 9.2.2. The detailed description of the use cases and all modes of operation of the system are described in section 9.3.

Performance requirements

In terms of static performance requirements the number of simultaneous users of the platform should cover all employees of the IVHM Centre and possible users in the partner organisations. The limitations in a client-server architecture with a web-based client interface in terms of user is given by the server technology and capacity. Technologies in question easily support far more users than considered for the prototype (less than 100 users).

The dynamic performance requirements as for instance response or throughput times for this low-data-volume application are inconsiderable. In order to have the user feel the system reacts

instantaneously, the maximum response time for any user action must not exceed 100 milliseconds [64].

Design constraints

A priori no design constraints for the functional system prototype are defined.

Software system attributes

The system is a fully functional prototype which is deployed within the IVHM Centre's network of systems. Thus, from a security point of view the system is protected by the overall IT security infrastructure of the IVHM Centre. Due to this circumstance and the fact that the prototype will not be used as final operational product the security requirements for the platform are lower. Nevertheless, special attention has to be paid to cross-site scripting and SQL injections.

In terms of availability and reliability, the system shall be fully available 24/7 because of the global user base but 'can' be taken down for maintenance and updates. Users need to be informed about the maintenance enough time in advance. During operational hours the system has to run fully reliable. Special attention is paid to user inputs not to compromise the reliability of the platform.

9.3 Use Cases Analysis

The use cases describe the interaction of the user with the system and its functionality. The functionalities of the IVHM B2B platform are are strongly inspired by the features of existing online marketplace platforms as described in section 9.1.

9.3.1 Use Case Terminology

Users have been defined in section 9.2.2. To further clarify the 'platform administrator' represents the platform manager and maintainer with the maximum access rights and permissions. A 'buyer user' can only interact with the front end of the system, thus searching and browsing products et cetera. A 'supplier user' or 'supplier administrator' has access to the backend of the system and can manage its products and micro site.

9.3.2 Use Cases

Use Case 1	***Platform user login***
Actors	• Administrator • Buyer User • Supplier User
Precondition	Actor has openend the platform frontend or backend start page in his browers and is not logged in.
Success scenario	1. Actor enters correct user name and password and confirms. 2. The system validates the input data and logs the actor on. 3. Depending on the actor type the system displays the corresponding frontend or backend start interface.
Required inputs	• User name • Password
Exception scenario	If the actor enters a wrong user name/password combination the system notifies him of the mistake and redirects him to the login area. This also applies if a user with insufficient rights tries to login to the system backend. If a user has forgotten his user name/password combination, it can be retrieved by clicking on 'Forgot your password?' and providing the registered e-mail address. The system will then deliver the login information via e-mail.

Table 9.3: Use Case 1.

Use Case 2	***Manage platform users and roles***
Actor	Administrator
Precondition	Actor is logged with administrator account and in front of 'user management' interface. Role 'SupplierAdmin' granting a supplier user access to the system backend to manage one's products and categories is predefined in the system.
Success scenarios	• To add a new user the administrator clicks the 'Add new user' button, enters the required information and saves the user. The system then creates the user in the system. • If the user is a special supplier user the administrator assigns the role 'SupplierAdmin' in the sub-interface 'Roles'. Then the user has special backend access to the system. • To change and remove users the administrator navigates to the 'View users' interface. There a click on the corresponding icon for a certain user 'removes' him after a confirmation, another icon allows to 'edit' the user details. The 'edit' interface is equal to the 'add interface' with predefined inputs.
Inputs	• User name • User's e-mail address (required for new user) • User's first and last name (required for new user) • User's company affiliation (required for new user) • Role association • Contact details, website,
Exception scenario	If one of the required inputs is missing, the system notifies the administrator and redirects to the input form.

Table 9.4: Use Case 2.

Use Case 3	***Manage platform content and static pages***
Actors	• Administrator • Supplier User
Precondition	Actor is logged in to system backend.
Success scenarios	• The actor navigates to the content management system (CMS) interface where all pages associated in terms of access rights with the actor are shown (all pages for the platform administrator). For the role 'SupplierAdmin' the associated platform micosite is shown. • To edit the actor clicks 'edit' on a certain page and either uses the integrated WYSIWYG editor or adapts the page source code directly to edit the page. The interface also offers a form to upload images and files to be linked in a page. • To add a new page the actor clicks 'add page' and can start designing the page in an empty edit interface. • To remove a page the actor clicks 'remove' on a certain page and the system deletes the page after user confirmation.
Inputs	• Page name (required for adding a page) • Page content including (X)HTML and CSS code, images or PDF files.
Exception scenario	If the users does not provide a page name while adding a new page, the system notifies the user and does not save the page.

Table 9.5: Use Case 3.

Use Case 4	***Browse IVHM products and technologies***
Actor	Buyer User
Precondition	Actor is logged in to front end.
Success scenarios	• The user browses through the platform's IVHM product and technology portfolio by clicking on the corresponding (sub-)categories in the catalogue tree in the front end. • Once the user has navigated to a certain catalogue category the system displays all platform items in this category. Furthermore, the user can choose how the products are displayed, for instance as list or grid. • In the category view the system also offer criteria to further narrow the items shown. Subcategories are displayed as well as product or technology attributes that can be used to filter, for instance connector features of an IVHM sensor.
Inputs	• No data inputs. The user provides input by choosing specific categories and product attributes.

Table 9.6: Use Case 4.

Use Case 5	***Simple platform search***
Actor	Buyer User
Precondition	Actor is logged in to front end.
Success scenarios	• The user enters a search term to the search form in the front end and hits enter. • The systems lists all products, capabilities or categories that are associated with the search term, for instance due to product name, attribute name or attribute value.
Inputs	• Search term(s)
Exception scenario	If no products, capabilites or categories associated with the search term are found the system shows empty search results and informs the user.

Table 9.7: Use Case 5.

Use Case 6	***Advanced platform search***
Actor	Buyer User
Precondition	Actor is logged in to front end.
Success scenarios	• The user navigates to the 'advanced search' interface by clicking the corresponding link in the front end. • In this interface the user can define search criteria for product and technology attributes in addition to simple search terms. • Upon search execution the systems lists all products, capabilities or categories that fulfill the advanced search criteria.
Inputs	• Search term(s) • Search criteria for product attributes
Exception scenario	If no products, capabilites or categories associated with the search terms constrained by the search criteria are found the system shows empty search results and informs the user.

Table 9.8: Use Case 6.

Use Case 7	***Tag IVHM products and capabilities***
Actor	All
Precondition	Actor is logged in.
Success scenarios	• A front end user while browsing through IVHM products and capabilities can easily add 'tags' to an item by entering the tag name in the corresponding form on the product page. • By clicking an item in the tag cloud in the front end, the system shows all IVHM products and capabilities that are 'tagged' accordingly. • A backend user navigates to the 'manage products' interface and for each product he can easily manage all tags with a simple form, i.e. add, edit or remove tags.
Inputs	• Tag name
Comments	A tag cloud is shown in the front end on a side bar in all product related pages.

Table 9.9: Use Case 7.

Use Case 8	***Add and remove items to 'IVHM product basket'***
Actor	Buyer User
Precondition	Actor is logged in the front end.
Success scenarios	• While browsing through IVHM products and capabilities the actor can add each item to the 'IVHM product basket' with a click the corresponding link. The feature is similar to a shopping cart where an interesting set of products can be grouped. • The system displays the items in the basket in a dedicated block in the front end. • The actor can easily remove items from the basket by clicking the corresponding icon. • Once the actor is satisfied with the product set in the basket he can save it to his account. The system then saves the product set with the user account such as the user can access and organise it later. • At the same time the user can 'submit' a basket to the suppliers. The system then sends the product set to all manufacturers in this product group. The suppliers can then follow up with the buyer on this configuration and also discuss it with the other suppliers in the set.
Inputs	• Clicks on IVHM products and capabilities to be saved or submitted in 'IVHM product basket'.
Comments	If no items are present in the 'IVHM product basket' the save and submit buttons are disabled by the system.

Table 9.10: Use Case 8.

Use Case 9	***Compare IVHM products or capabilities***
Actor	Buyer User
Precondition	Actor is logged in the front end.
Success scenarios	• While browsing through IVHM products and capabilities the actor can add each item to a 'compare basket' with click the corresponding link. • All items to compare are shown in a 'compare basket' next to the normal content of the B2B platform. • The actor can remove unwanted items from the 'compare basket' by clicking the corresponding icon. • Once the actor is satisfied with the products/capabilities to compare, he clicks 'compare' and a new page opens where all items are presented in a well-arranged way such that all features can be easily compared.
Inputs	• IVHM products or capabilities to compare.
Comments	If no items are present in the 'compare basket', the compare items link is disabled by the system.

Table 9.11: Use Case 9.

Use Case 10	***Add/edit/view/delete IVHM product, technology or capability***
Actor	Supplier User
Precondition	Actor is logged in and in front of the 'manage products' interface.
Success scenarios	• 'Manage products' interface lists all available products for the supplier and for each item buttons to view, edit and delete the item. In addition an 'add new product' button is provided. • The actor clicks *view* for a certain product. The system provides the representation of the product as it would appear to the buyer. • The actor clicks *edit* a product. The system display the edit interface. In a form the user can edit all attributes of the product. Upon saving the system validates the user entries and updates the product information and attributes. Possibility to cancel or reset are given. • The actor clicks *delete* for a certain product. After a user confirmation the system removes all product information from the platform and notifies the user of successful deletion. • The actor clicks *add new product*. The system displays an empty form environment where all product attributes can be set and all product information uploaded. Upon saving the system validates the user entries, adds the product to the platform and display the 'manage products' interface where the new product is listed. Cancel and reset possibilities are given.
(Optional) inputs for *add* and *edit*	• Product name • Description • Product datasheet • Association with categories/tags • Image gallery • Case study information • Compatibility with other products • Contact details for the product
Exception scenario	In case the user enters invalid information in the 'add' and 'edit' case the system highlights the affected fields and notifies the user.

Table 9.12: Use Case 10.

Use Case 11	***Manage IVHM catalogue categories***
Actor	Supplier User
Precondition	Actor is logged in and in front of the 'manage categories' interface.
Success scenarios	• 'Manage categories' interface lists a tree structure of all catalogue categories available in the system. • The actor can access a certain category and view all of his products listed in this category. • In this category view the actor can add and remove his products to the category by assigning them in the provided category box. • The actor can add new sub-categories under the root category by clicking the respective button in the 'manage categories' interface. • The actor can change, for instance rename or reassign to a different parent category, and remove categories containing only his products.
Inputs	• Category name • Category location in catalogue tree
Exception scenario	The actor can change and remove only those categories that exclusively contain products that the actor manages (i.e. products from the same supplier). If this is not the case the actor is notified by the system. To change categories with mixed products the actor has to file a request to the platform administrator.

Table 9.13: Use Case 11.

Use Case 12	***IVHM product and technology platform reporting***
Actor	Supplier User
Precondition	Actor is logged in to the back end.
Success scenarios	• The actor navigates to the reporting interface in the back end. • There the system provides statistics for all products on the platform associated with the supplier in terms of number of views, number of submits in a 'IVHM product basket' and also corresponding statistics for the associated categories.
Inputs	• User click on product or category to generate platform report for.

Table 9.14: Use Case 12.

Chapter 10

Solution Evaluation and Selection

In this phase the most appropriate software package shall be chosen from available solutions. Following the software selection methodology elaborated in section 5.3, six steps need to be followed in order to identify the most appropriate solution.

Step 1 is already fulfilled due to the analysis and documentation of the software requirements. The following sections will further follow the methodology and eventually identify the most suitable software solution.

10.1 Analysis of Available Solutions and Shortlisting

Based on online research 6 solutions including custom development from scratch considering the requirements were identified and shortlisted (*Steps 2 and 3*). The analysis already excludes full commercial solutions. From an economic point of view these are not suitable for such a prototype which is intended to assess the feasibility of such a platform in general, with the economic aspects this system only assessed during the prototype pilot phase. Thus, the analysis focused on open source packages. The results are shown in table 3.1 excluding custom development.

Besides the shortlisting for non-commercial solutions two more options can be eliminated. The requirements state that the costs for the prototype development in terms of license fees and man hours need to be minimised. Thus, the *B2B Marketplace Script* with license fees of 67 USD and *custom development from scratch* are eliminated as possible solutions since existing frameworks can be leveraged to reduce development costs significantly. This leaves four possible options that need to be evaluated: *Magento*, *OSCommerce*, *Zen Cart* and *PHPB2B*.

All solutions are open source, the source code is freely available, thus can be easily analysed and used for the prototype implementation. Another observation is that all packages are using the same technology, PHP as Server-Side Scripting (SSS) language and MySQL, the most common open source database. This is not really surprising since PHP has been there since the early days of SSS, it is easy to learn and supported by a large community has strongly evolved over the years. In the next step, *Step 4*, the software package evaluation criteria are defined.

					
Name	*B2B Marketplace Script*[1]	*Magento Community Edition*[2]	*OSCommerce*[3]	*Zen Cart*[4]	*PHPB2B*[5]
Version	-	1.5.1.0	2.3.1	1.3.9h	4.0.1
Type	Open Source with license	Open Source	Open Source	Open Source	Open Source
Technology	PHP, MySQL	PHP, Zend Framework, MySQL	PHP, MySQL	PHP, MySQL	PHP, MySQL
License	67 USD	GPL	GPL	GPL	GPL

Table 10.1: Overview of available solutions excluding custom development.

[1] http://www.b2bmarketplacescript.com
[2] http://www.magentocommerce.com
[3] http://www.oscommerce.com
[4] http://www.zen-cart.com
[5] http://www.phpb2b.com

Criteria group	Criterion	Description
Functionality	Login and access system	Is the login and access system for administrators and users suitable?[1]
Functionality	Access rights and permission management	Does the software offer a flexible user rights management system with administrators and users with different access rights and permissions?[1]
Functionality	Content management system	Does the system incorporate a content management system and offer micro sites for different users?[1]
Functionality	'Product basket' feature	How suitable is the 'product/capability basket' to store products and enquire further information?[1]
Functionality	Product catalogue management	How suitable is the system to manage the product catalogue and its categories?[1]
Functionality	Product comparison feature	Does the software offer a product comparison feature and is it suitable?[1]
Functionality	Product management	How suitable is the system to manage products/capabilities and its attributes?[1]
Functionality	Product tagging feature	Does the system incorporate a structure for tagging products and capabilities and how suitable is it?[1]
Functionality	Search feature	How suitable is the system's search feature for products and capabilities?[1]
Functionality	Reporting feature	How suitable is the reporting system of the system e.g. number product/page views etc.?[1]

Table 10.2: Evaluation criteria and metrics with respect to system functionality.

[1] Qualitative metric: very poor, poor, fair, good, very good.

10.2 Solution Evaluation Criteria

The criteria identified during the literature review in table 5.1 need to be refined according to the requirements analysis. These final criteria are used to analyse the available software packages for our project and identify the most suitable solution using a software evaluation methodology.

After the requirements analysis the criteria group that requires refinement is *functionality*. The criterion *included functionality* is thus split and enhanced by the criteria shown in table 10.2.

The combination of tables 5.1 and 10.2 results to 5 criteria groups (*Functionality*, *Quality*, *Usability*, *Vendor* and *Cost*) and 20 evaluation criteria in total. However, since all considered software packages are open source, two criteria can be eliminated: *Vendor/Demo* and *Cost/License cost*. The final list of the 18 evaluation criteria is shown in table 10.3.

10.3 Application of Software Evaluation Methodology

In *Step 5* different software evaluation methodologies are applied based on the identified criteria. WSM and AHP are the most appropriate approaches to solve this specific decision making

Criteria group	Criterion	Code
Functionality	Login and access system[1]	F0
Functionality	Access rights and permission management[1]	F1
Functionality	Content management system[1]	F2
Functionality	'Product basket' feature[1]	F3
Functionality	Product catalogue management[1]	F4
Functionality	Product comparison feature[1]	F5
Functionality	Product management[1]	F6
Functionality	Product tagging feature[1]	F7
Functionality	Search feature[1]	F8
Functionality	Reporting feature[1]	F9
Quality	Adaptability/Source code quality[1]	Q0
Quality	Reliability[1]	Q1
Quality	Security[1]	Q2
Quality	Technology/Programming language[1]	Q3
Usability	Error reporting[1]	U0
Usability	User interface/Ease of use[1]	U1
Vendor	References[2]	V0
Cost	Installation and implementation cost[1]	C0

Table 10.3: Full evaluation criteria set.

[1] Very poor, poor, fair, good, very good - corresponding to numerical values 1 to 5.

[2] Very poor (<3), poor (4-10), fair (11-20), good (21-30), very good (>30) - corresponding to numerical values 1 to 5.

problem because the complexity of the decision, that is the number of criteria and options, is rather low.

10.4 Weighted Scoring Method (WSM)

In the first step each software package needs to be numerically evaluated for each criterion. To do so the software is installed, executed, tested and its source code analysed. The results of the detailed analysis of each software package for each criterion can be found in the appendix in section A.1, in table A.1 for *Magento*, in table A.2 for *OSCommerce*, in table A.3 for *Zen Cart* and in table A.4 for *PHPB2B*. The detailed analysis justifies the score for each package and each criterion.

Secondly, the individual weighting for each criterion needs to be determined in order to calculate the weighted sum, which is the score for each package. In the simplest approach each criterion has the same weighting (i.e. averaging criteria scores). With more sophistication the weighting is adapted to the specific needs of this project in correspondence with the weighted scoring method.

Table 10.4 summarises the results for the WSM evaluation. Column two represents the simple average weightings, column three the more sophisticated relative weightings tailored to this project and column 4 normalises them to serve as basis for the weighted scoring method. Columns

	Criteria Weighting			SW Package Score			
Code	**AVG[1]**	**RW[2]**	**WSM[3]**	**Magento**	**OSCommerce**	**Zen Cart**	**PHPB2B**
F0	6%	3	8%	5	3	2	5
F1	6%	3	8%	2	1	1	5
F2	6%	1	3%	5	1	1	3
F3	6%	1	3%	4	3	3	4
F4	6%	3	8%	4	3	3	5
F5	6%	1	3%	5	1	1	1
F6	6%	3	8%	5	3	3	3
F7	6%	1	3%	5	1	1	1
F8	6%	2	5%	4	4	4	4
F9	6%	1	3%	4	3	3	3
Q0	6%	5	13%	5	3	3	1
Q1	6%	2	5%	5	5	5	1
Q2	6%	1	3%	5	5	5	2
Q3	6%	4	10%	5	3	3	1
U0	6%	1	3%	4	3	3	2
U1	6%	2	5%	5	5	5	2
V0	6%	1	3%	5	5	4	3
C0	6%	5	13%	3	3	3	2
			AVG Score	**4,44**	3,06	2,94	2,67
			Norm. AVG Score	**34%**	23%	22%	20%
			WSM	**4,33**	3,05	2,95	2,65
			Norm. WSM	**33%**	24%	23%	20%

Table 10.4: Scores of software packages based on WSM.

[1] Simple average weighting (AVG) for all criteria.
[2] Relative weighting (RW) for each criterion with 1 to 5, low to high importance.
[3] Normalised weights derived from RW for the weighted scoring method (WSM).

5 to 8 contain the score for each criterion based on the analysis of each software package.

The WSM's clear winner is *Magento*, whereas *OSCommerce* and *Zen Cart* have not only very similar overall score but also similar scores for each criterion. The reason is that *Zen Cart* is based on *OSCommerce* and has thus very similar properties and functionalities.

One at first rather surprising finding is the fact that differences between the score based on average weights and the WSM weights are almost negligible. Since the distribution between the scores for each criterion and the distribution between the relative weightings is rather small (1-5), with the number of criteria high (18), small changes in the weighting of the criteria do not have a significant effect on the total score.

Figure 10.1 illustrates the normalised and weighted scores for each criterion and software package. Criteria *Q0* (adaptability/source code quality) and *C0* (installation and implemenation cost) with a weighting of 12,5%, together with *Q3* (technology/programming language) at 10% were rated as most important criteria. *Magento* scored at the top for all criteria, except for *F1* where it was strongly outperformed by *PHPB2B*. Overall all the figure shows clearly why *Magento* is to be chosen.

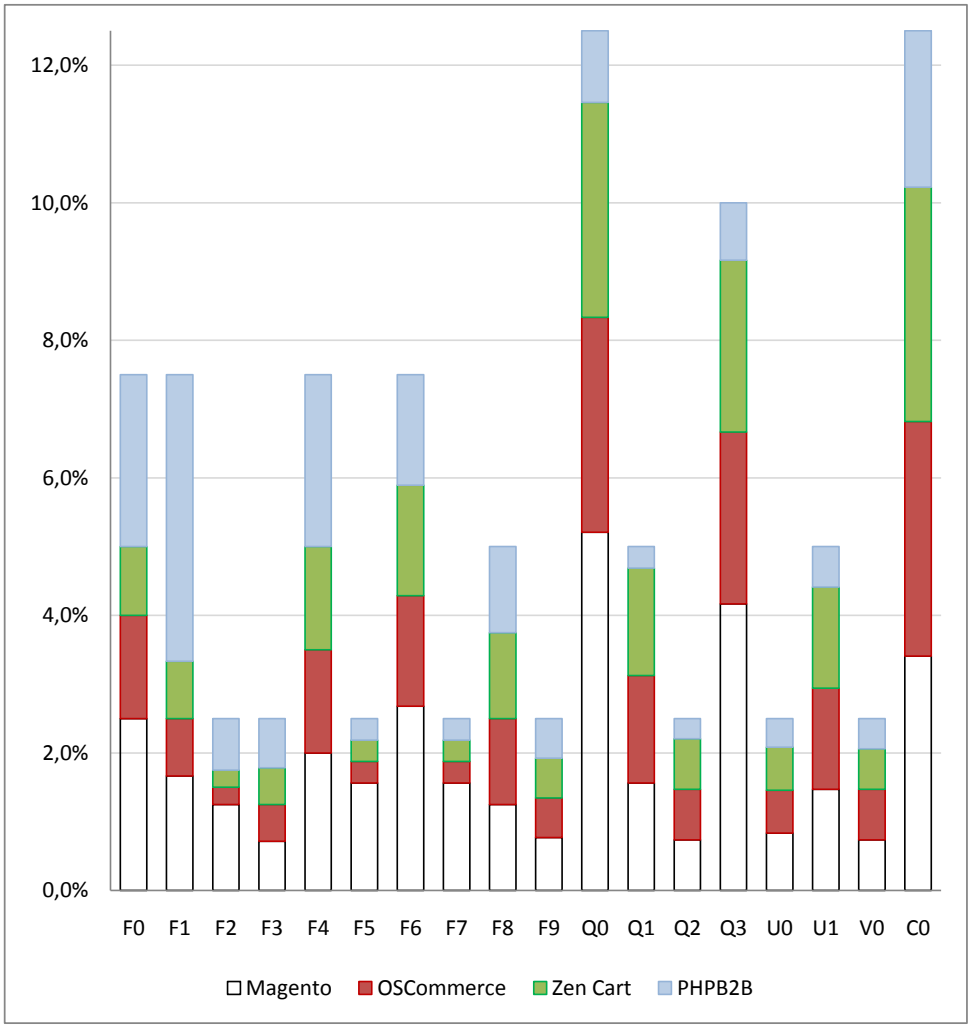

Figure 10.1: Normalised WSM scores of software packages per criterion. Aggregate values reflect WSM weighting for each criterion.

10.5 Analytical Hierarchy Process (AHP)

By means of the AHP all criteria as well as all options for each criterion are subject to pairwise comparison. The normalised principal eigenvector of this comparison matrices provides a relative weighting for each criterion on the one hand, and a relative score for each option for each criterion on the other hand.

The results of the AHP analysis are shown in table 10.5. The detailed AHP matrices and calculations can be found in the appendix in section A.2 in tables A.5, A.6, A.7 and A.8. The AHP analysis also suggests *Magento* as clear winner with a starkly higher score compared to the competing software packages. Figure 10.2 illustrates the weighted normalised AHP scores for each criterion and each package.

Considering the aggregate normalised value for each criterion AHP ranks *Q0*, *F6*, *C0*, *F4* and *Q3* as most important criteria. These 5 criteria alone attribute 60% of the AHP score, reassuring that the quality and technology of the software, together with implementation cost are much more important than already implemented functionalities. For instance *PHPB2B* stron-

		AHP Score per Criterion[1]			
Code	**AHP[2]**	**Magento**	**OSCommerce**	**Zen Cart**	**PHPB2B**
F0	5%	27,8%	9,8%	4,8%	57,5%
F1	7%	17,8%	6,3%	5,5%	70,4%
F2	3%	67,4%	7,1%	7,1%	18,4%
F3	3%	33,3%	16,7%	16,7%	33,3%
F4	10%	26,2%	10,5%	10,5%	52,8%
F5	2%	75,0%	8,3%	8,3%	8,3%
F6	12%	66,2%	9,1%	9,1%	15,5%
F7	3%	44,0%	13,1%	21,5%	21,5%
F8	4%	25,0%	25,0%	25,0%	25,0%
F9	2%	49,5%	19,4%	19,4%	11,7%
Q0	16%	55,4%	19,0%	19,0%	6,5%
Q1	3%	32,1%	32,1%	32,1%	3,6%
Q2	2%	30,8%	30,8%	30,8%	7,7%
Q3	10%	69,7%	12,9%	12,9%	4,5%
U0	2%	51,8%	21,3%	21,3%	5,5%
U1	4%	31,6%	31,6%	31,6%	5,3%
V0	1%	59,3%	26,1%	10,6%	4,0%
C0	11%	42,4%	22,7%	22,7%	12,2%
Overall Score		**46%**	**16%**	**16%**	**22%**

Table 10.5: Software evaluation results of AHP.

[2] Weighting of each criterion resulting from AHP.
[1] Relative score of each software package for the criterion considered resulting from AHP.

gly outperforms for the functional criteria *F0*, *F1*, *F4* but scores quite low on quality and cost. *Magento* might not offer all functionalities but the technology used, software architecture and source code quality together with a functional edge in product catalogue management make it the most attractive solution.

10.6 Discussion and Conclusion of Solution Evaluation and Selection

Both software evaluation methods suggest *Magento* as optimum software package considering the evaluation criteria, whereas AHP assigns *Magento* a normalised score that is relatively 30% higher in comparison to WSM. Also for the further ranking both methods differ. Even though they both assign *OSCommerce* and *Zen Cart* virtually the same score, WSM rankes them slightly better than *PHPB2B*, whereas AHP ranks them other way round, but with a more significant distance.

With AHP being less restricted by a scoring scale and employing pairwise comparison the results are more intuitive and accurate than those provided by WSM. Even though AHP and WSM show a very similar structure in the individual weighting of the criteria (same highest weights *Q0*, *C0*, *Q3*, *F6* and similar lowest weights), the weighting distribution for AHP (min 1% - max

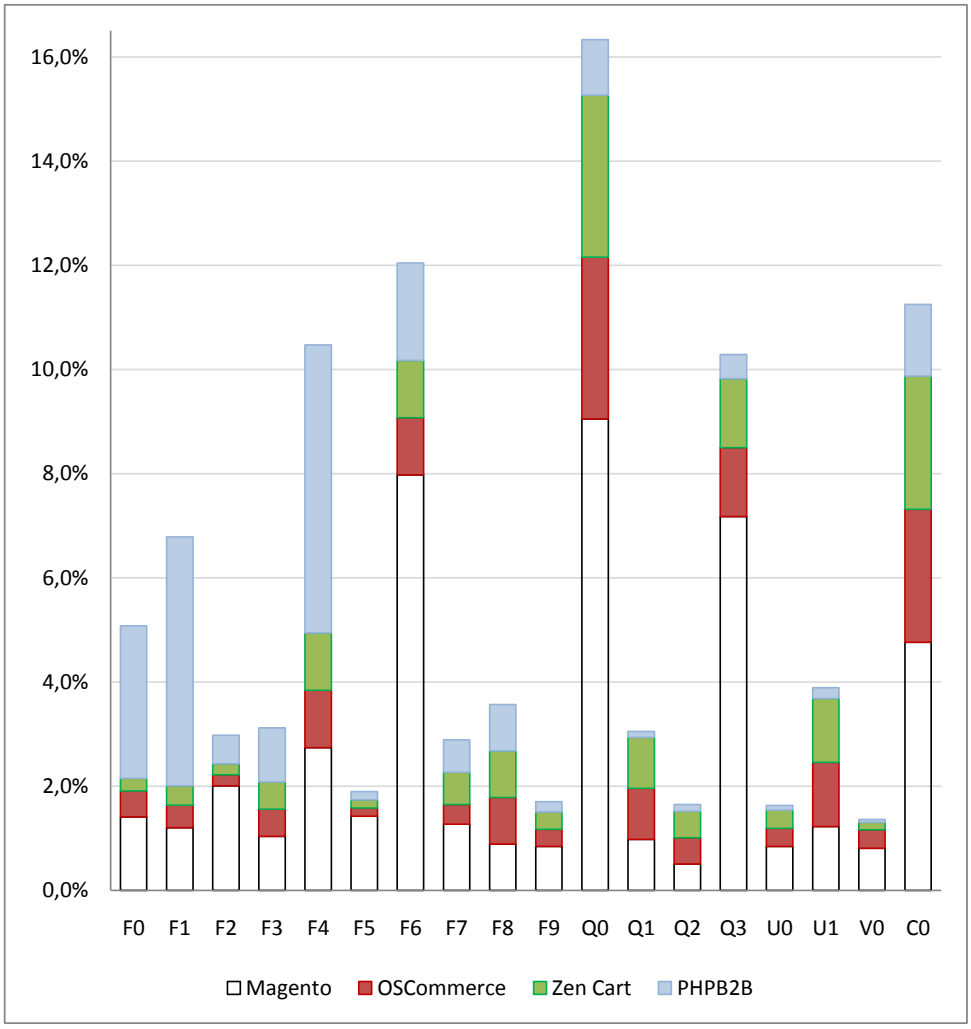

Figure 10.2: AHP scores of software packages per criterion. Aggregate values reflect AHP weighting for each criterion.

16%) is much more distinct and less uniform compared to WSM (min 3% - max 13%). Thus, AHP better reflects the actual and more intuitive importance ranking of the criteria set.

The same applies for the actual normalised scores, AHP with min 1% - max 75%, and WSM with min 6% - max 63%. AHP with its pairwise comparison thus offers more significant results than WSM with its scaling and weighting-assignment bias.

AHP's downside is though that it requires tedious work to fill all comparison matrices. In this application 261 matrix fields had to be filled assuming the remaining ones are filled through software support (total of 612 matrix elements). WSM on the other hand required only 72 inputs for the average weighting approach and 90 inputs otherwise.

For the weighted scoring method it was also concluded that if the scoring scale is too small for a high number of criteria, the actual weighting of the criteria does not have a significant influence. Thus, the quite tedious task of assigning individual weights for each criterion can be replaced with simple averaging over the results. Otherwise the number of criteria could be reduced (grouping) or the scoring scale increased in order have the relative weighting really

influence the outcomes.

In conclusion, WSM offers possibly biased results due to difficulties in assigning overall weights for criteria and the influence of scoring scale decisions but a faster and less tiresome process. AHP on the other hand requires the slow and tedious process to create the comparison matrices but provides more reliable results. The trade-off between these two needs evaluated based on the complexity of the decision taken and the time available, whereas in general, due to the results AHP would be preferred. In this particular analysis both approaches delivered the same result in recommending *Magento* as software package for this project.

Now that the most suitable software package and framework has been identified, the next section will elaborate the analysis of the software from a technical point of view, the required customisation of the package and the actual implementation of the IVHM B2B platform.

Chapter 11

Implementation

This chapter discusses the software engineering aspects of the IVHM B2B Platform.

11.1 Development Environment

The following software and software tools were used on *Windows 7* to develop and generate documentation during the implementation phase. *Eclipse Helios 3.6.0* with PHP plugin *Eclipse PDT 2.2.0* was used as integrated development environment. *Zend Server 5.1.0* was used as PHP web application server employing *Apache 2.2.16* and *PHP 5.3.5*. *MySQL 5.1.50 Community Edition* was deployed as relational database and *phpMyAdmin 3.3.3* and *MySQL Workbench 5.0 Community Edition* were used to visualise and manipulate the database. *Exim 4.76* based on Cygwin was used as mail transfer agent and *Subclipse 1.6.18* was used as SVN plugin for Eclipse to synchronise with the SVN repository.

11.2 System Functionalities and Gap Analysis

11.2.1 Functional Gaps

The requirements analysis identified 3 major users for the system: the buyer user, the platform administrator and the supplier administrator. Thus, the system needs to provide two major User Interfaces (UIs), a *frontend* for the buyer user and a *backend* for the administrators.

Considering the set of functionalities required due to the use case analysis, some are already given in the chosen software package *Magento*. Figure 11.1 provides an overview of the functionalities required and to what extent they are covered by a plain Magento installation.

The access system is given for frontend and backend. In the frontend b*rowsing items* and *tagging* is provided, *simple search* and *products comparison* only need minor customisation whereas the *advanced search* and *product basket* require major software customisation.

General	
UC1 Login/Logout	

Frontend		Backend		SA	PA
UC4 Browse products		**UC2** Advanced permission for administrators		-	x
UC5 Simple search	~	**UC3** Content Management System (microsites)		x	
UC6 Advanced seach	x	**UC7** Tagging		x	
UC7 Tagging		**UC10** Manage products		x	
UC8 Product basket	x	**UC11** Manage categories		x	
UC9 Compare products	~	**UC12** Reporting		x	

Legend		feature given, only simple SW package configuration required.
	~	feature somewhat given, small source code customisation required.
	x	feature not given or major SW package customisation required.
	SA	supplier administrator.
	PA	platform administrator.

Figure 11.1: Overview of functional gap analysis based on use cases.

In the backend the functionalities for the *platform administrator* are mostly given, however there is *no* module providing *advanced rights management* for the backend users. This is why all modules in the backend need major customisation for the supplier administrators (SA).

The *CMS* module needs to be restricted for the SA's microsite. *Tagging*, *product management*, *category management* and *reporting* require strong customisation to implement the *advanced permission management system*. Each module needs to be adapted in order to provide the possibility to access and adapt exclusively the products, technologies, tags, microsites and reporting the SA 'owns'.

11.2.2 User Interface and Magento Functionality Elimination

Along with the functional enhancement and customisation of the system, *many functionalities* given in Magento need to be *deactivated* since the platform is not a marketplace setup where items can be actually bought and sold. For instance the *sales*, *ordering*, *tax*, *pricing* and *billing* functionalities need to be deactivated. In particular the *pricing* requires (major) changes in many areas since it is a core functionality of Magento. Furthermore the modules for *promotions*, *newsletters* and *mobile applications* need to be removed.

Besides the functional adaptations of Magento the *user interface* needs to be customised to fit the requirements of the IVHM B2B platform. Thus, the *UI design* in terms of color schemes, fonts etc. and in particular the *layout* for the different interfaces of the frontend are subject to (major) adjustments.

11.3 Systems Architecture

Figure 11.2 illustrates the system environment representing a classic client-server architecture. The user interacts with the system via a web browser using any web-enabled device. *PHP*

is employed as server-side scripting language on a *Apache* web server. As persistence layer a *MySQL* database is employed which stores all system related data. Both Apache as well as the database are deployed on a Windows server with Cygwin-based *Exim* as mail transfer agent.

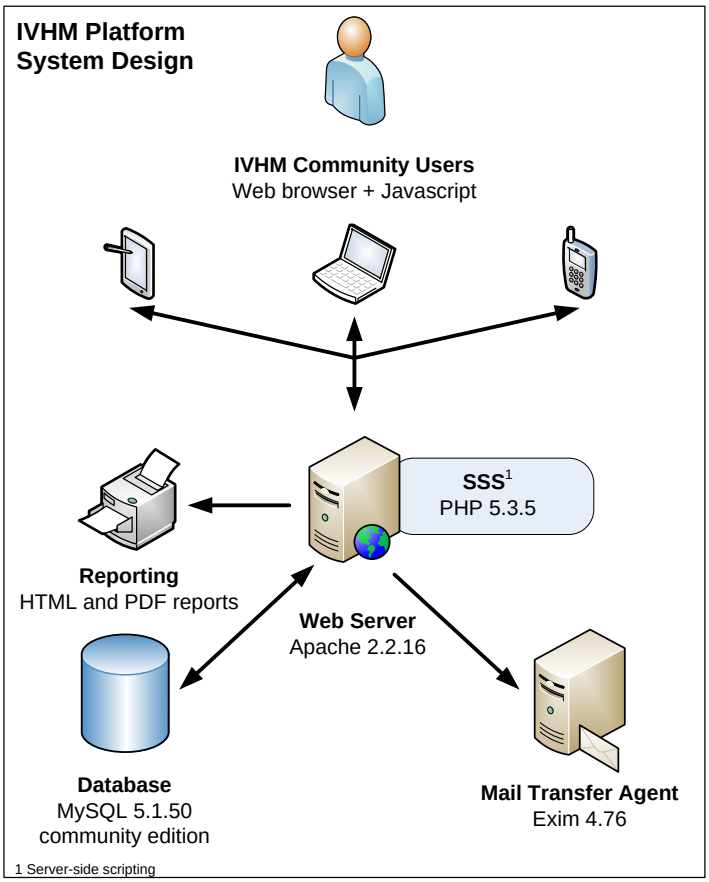

Figure 11.2: IVHM system architecture and technology.

11.4 Software Architecture

Magento is based on a three-tier software architecture consisting of a data, logic and presentation layer as depicted in figure 11.3. The *data layer* comprises MySQL as relational database where an Entity-Attribute-Value (EAV) approach, representing an open database schema, is mapped. The details of this database design are discussed in section 11.5.

The *logic layer's* Object-Relational Mapping (ORM) layer maps the data layer's simple tables or more complex EAV models into domain objects. The advantage of this data abstraction layer is that the physical data model and data access complexity is hidden from the actual application logic and is thus streamlining the application code and separating the application logic from the actual persistence technology used [69].

The primary libraries used in the logic layer especially supporting the ORM are the *Zend Framework (ZF)* and *Varien*. The Zend Framework is a leading open source, PHP web application framework. It is fully object-oriented and offers amongst others rapid application development,

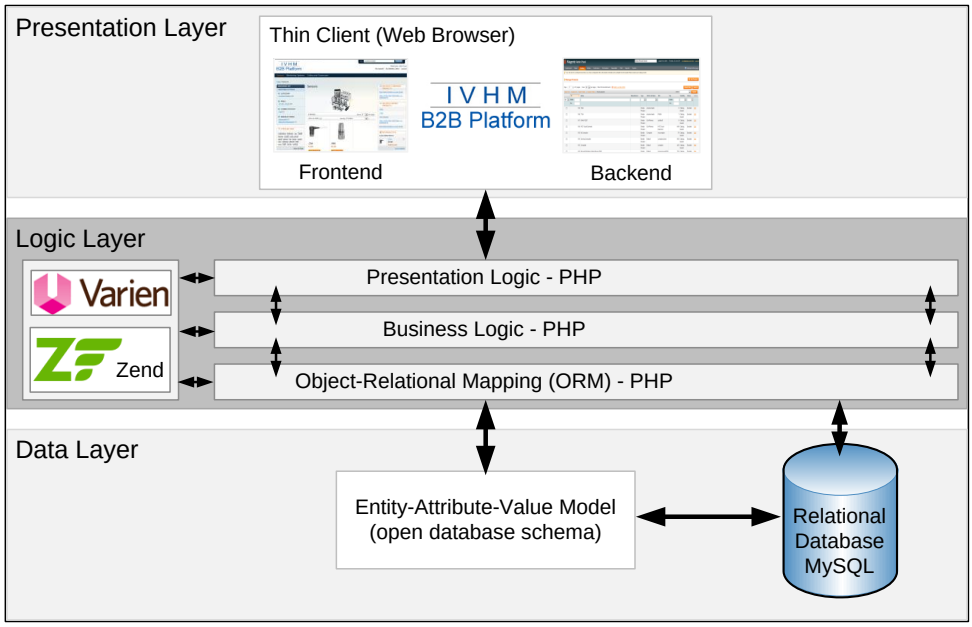

Figure 11.3: Three-tier software architecture and major libraries.

model-view-controller components, database abstraction and session management [102][1]. Varien is the former name of the company that now develops Magento and strongly building on the ZF it is the core library of Magento. It offers for instance the classes *Varien_Object* and *Varien_Data_Collection*, that represent the base class all model objects in Magento inherit from, and a convenient model container respectively.

In the sublayer *business logic* the data models are requested, manipulated and prepared for the *presentation logic* layer. The latter generates the HTML and related website code that is then presented in the user's web browers in the *presentation layer*. Due to the fact thin client is used the presentation layer is absolutely simple just rendering the supplied website source code. Thus, the prior presentation logic layer is actually implementing the 'view' in the model-view-controller sense. Software design details will be elaborated in section 11.6.

The advantages of the layered software approach are manifold. Layers are loosely coupled with defined interfaces and the separation between the components improves flexibility, maintenability and extendability. Furthermore testing is simplified. [62]

11.5 Database Design

Instead of conventional relational data modelling, the Magento framework employs EAV modelling that in turn is mapped in a simple relational database schema. The following sections elaborate the database design and show Entity-Relationship Diagram (ERD) representations of the most important tables.

[1] http://framework.zend.com/about/components, accessed 30.06.2011.

11.5.1 Entity-Attribute-Value Model

The EAV model is a very popular data modelling approach for heterogeneous data where out of a potentially large number of attributes, only a small fraction is applied to given entity. The data is referred to as 'sparse' or 'sparse matrix' in mathematics. [21]

In conventional representation each table holds a certain type of data with one column identifying an entity (e.g. 'ID' or the primary key) and a column for each attribute-value pair. Thus, each row contains a different entity with all different attribute-value pairs defined in the table [66]. In this design, sparse data imposes the problem that most of the table fields will be empty. Furthermore, if a new attribute-value pair for an entitiy needs to be stored either a new table is created or the schema of the old table needs to be altered [21].

EAV design solves these problems by conceptually storing the data triple *(entity, attribute, value)* in three different columns or even three different tables, thus 'normalising' the database schema. Then the physical storage of the data strongly differs from the conventional logical representation of the data. Hence, the EAV system needs to store the logical schema in *EAV metadata* tables. [66]

11.5.2 Magento Entity-Attribute-Value Design

Figure 11.4 shows the main EAV tables in an ERD in Crow's Foot Notation for a product or capability advertised on the platform. In the table *catalog_product_entity* each product represents a row identified by the primary key *entity_id*. *All* the attributes for the entire system are stored in *eav_attribute*. Now the actual value of an attribute for an entity is stored in 'value tables'. For reasons of data consistency these values are stored in 5 different tables depending on the value's data type, e.g. *catalog_product_entity_varchar* or *catalog_product_entity_int*.

This 'open schema' allows an arbitrary number of attributes per entity that can be added during runtime without redesigning the database schema while at the same time being highly storage-efficient for sparse data [66].

The ERD in figure 11.5 further refines the EAV model of the platform in particular the association of *attributes* to a certain *entity*. Note that in this figure table *catalog_product_entity* represents the same table as in figure 11.4. Each entity is of a certain type defined in table *eav_entity_type*. A *catalog_poduct_entity* or a *customer_entity* define a certain entity type and at the same time are tables to hold all entities of this type. Each entity type is associated with a set of attributes (*eav_attribute_set*), which holds all attributes associated with an entity.

The attribute set is organised in attribute groups (*eav_attribute_group*), which offers the possibility to organise a large number of attributes into logical subgroups (e.g. an accelerometer has attribute groups concerning dynamic, electrical and environmental properties). Finally, *eav_entity_attribute* maps all *eav_attributes* in the corresponding attribute group. This table also holds redundant information (*entity_type_id* and *attribute_set_id*) which violates the normal form, but speeds up data lookup.

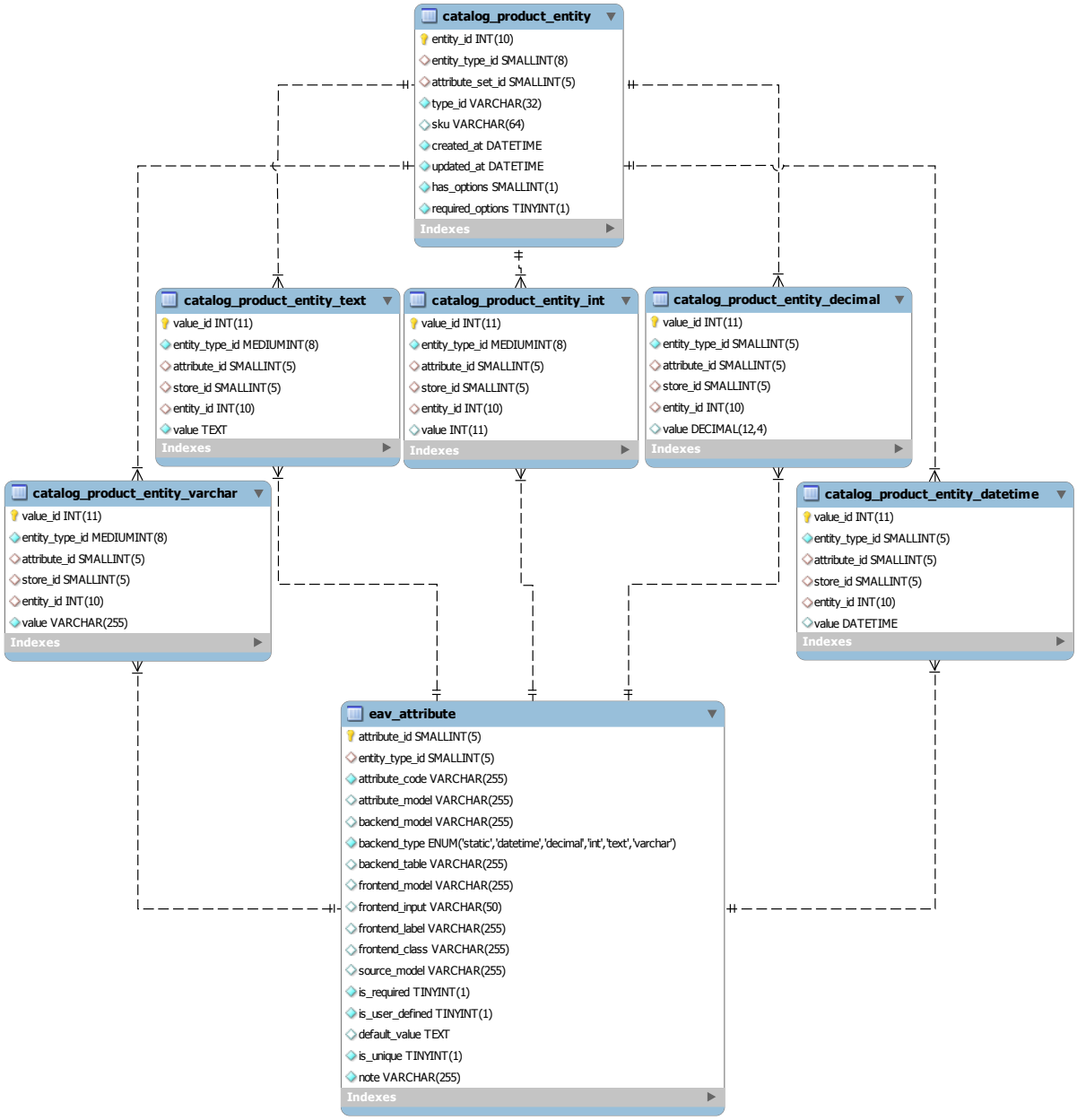

Figure 11.4: ERD of entity-attribute-value model for a platform item in Crow's Foot Notation.

Summarising, even though this database design adds complexity in storing and retrieving data, the flexibility of the model perfectly fits the platform's requirements. Suppliers can add different kinds of IVHM products and technologies with different kinds of attributes to the platform without requiring changes to the data model. Thus, no limitation in the storage and representation of IVHM product and capability data is given.

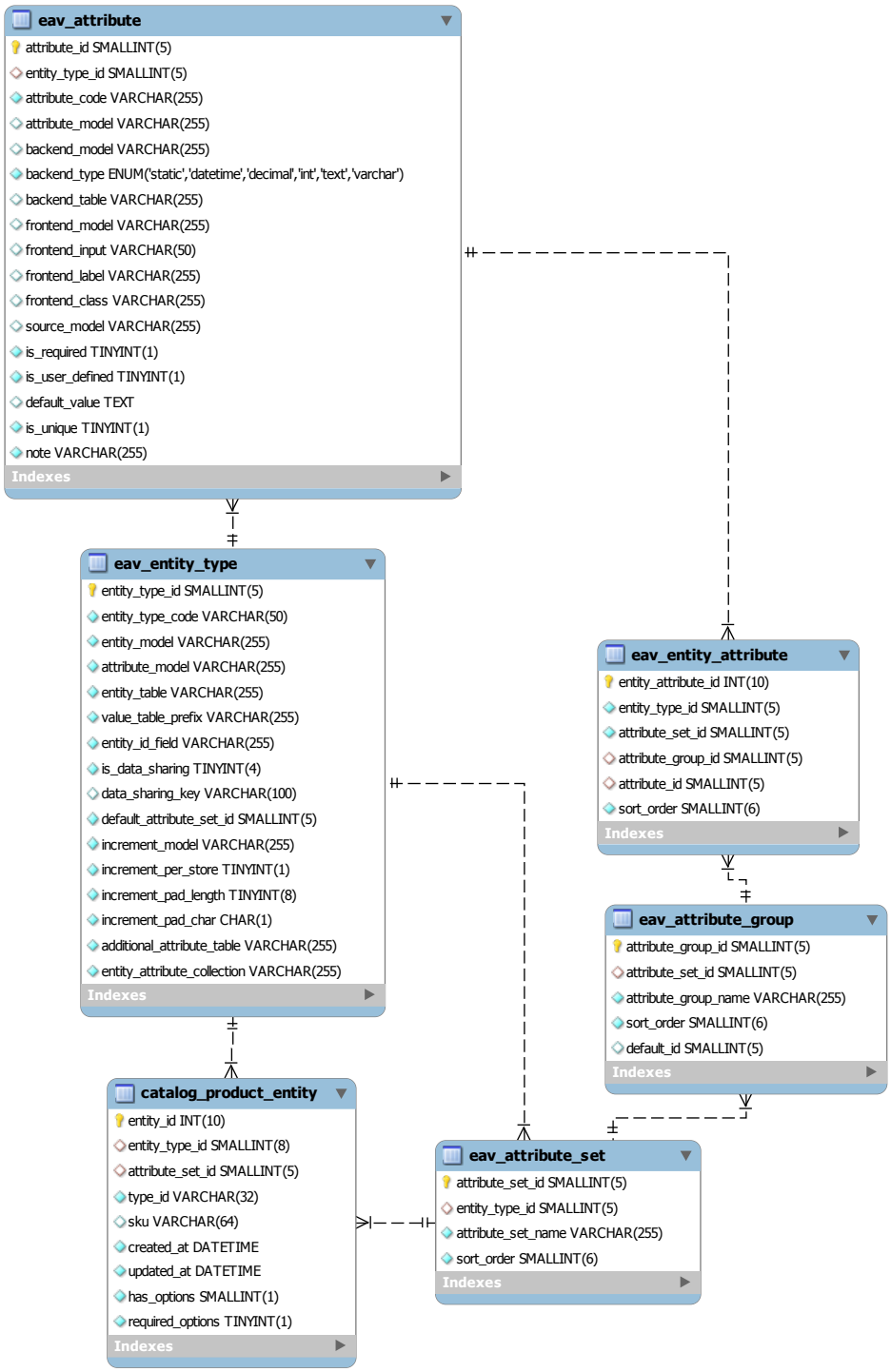

Figure 11.5: ERD of entity-attribute associations for an IVHM item in Crow's Foot Notation.

11.6 Software Design

The Magento framework is implemented based on best-practice design methods and design patterns. It incorporates a *Front Controller*, the *Model-View-Controller (MVC) pattern*, the

Event-Observer pattern, the *Singleton pattern* and various *Factories* to instantiate different kinds of objects, in particular for the *Object-Relational Mapping*. The major building blocks of the software design are shown in figure 11.6.

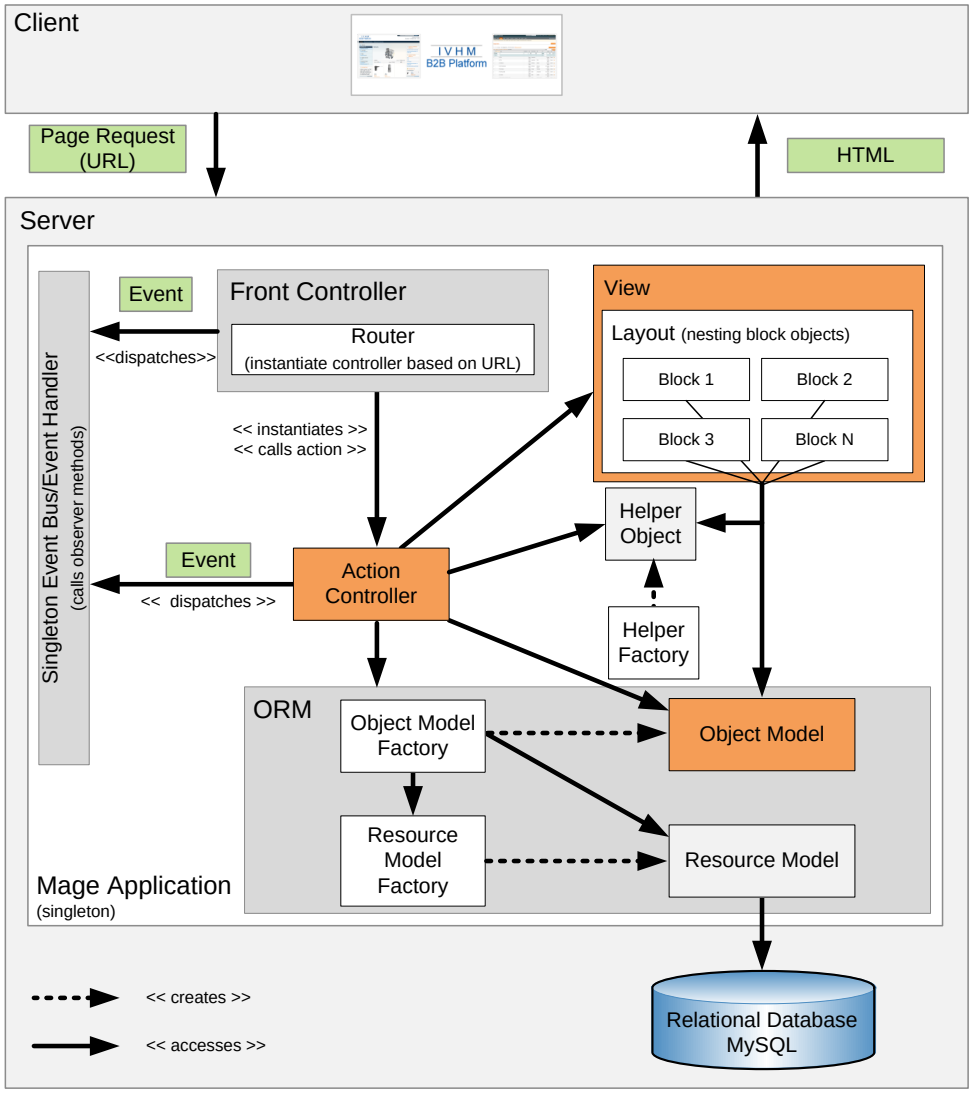

Figure 11.6: Major building blocks of Magento software design.

Due to the thin client configuration the *Client* side is very simple. It either displays website source code including HTML, CSS and Javascript in a web browser or sends *Page Requests* including Uniform Resource Locator (URL), HTTP GET and POST request to the server.

The *Server* contains all the system logic and the persistence layer. The major components on the server are the singleton instantiation of *PHP Class Mage* as core of the *Magento Application* and the *MySQL database*.

The *Mage Application* has three major components: the *Front Controller*, the *ORM* and the *MVC* including event handling and controlling. In addition it incorporates an *Abstract Helper Factory* together with a *Factory Method pattern* that generates singleton *Helper Objects*. This setup is used to group objects with a common purpose, defer instantiation to subclasses and make sure that only a single instantiation of a *Helper Object* is created [35].

Helper Objects typically contain (generic) functions and methods to retrieve and manipulate data in the model and to prepare the presentation output for instance in HTML. *Helpers* are used by the *Action Controller* to manipulate the *Object Models* and *Blocks* inside the *View* to provide presentation logic.

11.6.1 Front Controller

The *Front Controller* represents the central entry point for handling *Page Requests* and its URL.

According to Fowler et al. [31] complex web sites use this pattern if many things such as security, internationalisation, or providing particular views for certain users need to be handled upon page request. In this pattern all request handling is channeled through a single handler object which then instantiates the corresponding object and delegates the action [31].

The process where the URL is translated into the corresponding controller triggering the corresponding action is called *Routing* [59][2]. Depending on the *Routing* the (newly instantiated) *Action Controller* performs the requested action.

11.6.2 Model-View-Controller (MVC) Pattern

Magento implements a best-practice MVC pattern. MVC offers a clear separation between the domain objects themselves (the model) and their actual presentation to the user (the view) [30]. This decoupling increases flexibility and reuse and simplifies development, testing and maintenance [35].

Magento implements a *configuration-based* MVC, that means it is required to explicitly inform the system about new classes. Each *module* in Magento has a configuration file that contains all relevant information for the module. Then at runtime all Magento modules are loaded and combined in one configuration setup. The advantage is that this approach allows to override and easily enable and disable classes and functionalities. [59][3]

Model

The model represents the domain object, the actual data to be represented, often also the business logic to generate this data. Domain objects are fully self-contained without any relation to a specific presentation. Thus, the same domain object can be presented in different ways by different representations. [30]

The model is a passive object, responding to the view for state updates, and receiving instructions from the controller to change its state [63].

[2] http://www.magentocommerce.com/knowledge-base/entry/magento-for-dev-part-1-introduction-to-magento/, accessed 04.07.2011.

[3] http://www.magentocommerce.com/knowledge-base/entry/magento-for-dev-part-1-introduction-to-magento#2, accessed 01.07.2011.

In Magento the model is a *Object Model* that is generated out of the ORM layer and manipulated by the methods provided in its model object class. The ORM will be discussed in more detail in section 11.6.4.

View

The *view* is the actual representation of the data in the user interface. It must always ensure to reflect the current state of the model. Upon a change of the model the associated views are notified. [35]

In Magento the view is instantiated by the controller in form of a *Layout Object*. The layout nests an arbitrary number of *Block Objects* that directly retrieve the data from the model.

The *Block Object* contains the presentation logic to transform the model data into a certain representation. Furthermore, the *Template View Pattern* is implemented. The idea of this pattern is to use markers in static HTML pages, that are replaced during run time when servicing a request [31]. *Blocks* make use of static *Templates* to integrate the data retrieved from the model in a predefined layout.

Considering that the blocks are nested, each block could actually perceived as a view by itself, representing data from different model objects that were instantiated by the controller. Eventually, triggered by the controller the layout with its nesting tree is rendered to HTML and delivered to the client.

Controller

In general the controller manages the model and the view. Depending on the user inputs and other events the model and/or the view are informed to change accordingly. [63]

In Magento the *Action Controller* is instantiated by the *Front Controller* and performs a requested action based on the *Routing*.

There the controller instantiates or manipulates its models and views (*Layout*) supported by *Helper Objects* that are requested from a *Helper Factory*. The models themselves are retrieved from the ORM layer using the *Object Model Factory*. Furthermore, in performing different actions the controller also might dispatch *Events* to the event bus (e.g. event 'customer_login') that trigger the *Event Handler*.

11.6.3 Event-Observer Pattern

Magento implements the Event-Observer pattern using an *Event Bus* in order to simplify the source code and enable asynchronous communication.

The basic idea is that in one-to-many dependencies between objects as used here, if one object changes or a certain event happens (e.g. customer login), all dependent objects are notified and updated automatically [35].

In Magento in a certain application module (e.g. 'product catalogue') one can register for an event (e.g. 'customer_login') in the module's configuration file. There the method is specified that is called in the module's observer class.

Now if for instance an *Action Controller* dispatches this event on the *Event Bus* the *Event Handler* in the top-level *Mage Application* will run through all observers for this action and trigger the observers' event methods. This methods then adapt an existing model or trigger the instantiation of a new *Action Controller*.

11.6.4 Object-Relational Map (ORM)

Magento makes use of ORM to separate and simplify the interaction with the persistence layer. ORM is used to encapsulate the mapping between domain objects and its relational or even EAV data thus decoupling the application from the underlying data model and the data access details [69].

In Magento the ORM is separated in the *Object Models* that represent the actual domain objects and provide the interface for the business logic to retrieve and manipulate data, and *Resource Models* that handle the actual communication with the database. Both objects are instantiated out of their corresponding factories that have the same purpose as the *Helper Factory*.

Using this differentiation between *Object* and *Resource Models* the logical data objects are decoupled from the actual database interaction [59][4]. This has the advantage that if for instance the database is changed, only the *Resource Models* that implement the database access need to be adapted.

11.6.5 Sequence Diagram

In addition to the structure of the software design, figure 11.7 shows the actual sequence sequence diagram for the different components and objects of the Magento application. It clarifies the sequence of interaction between *Model*, *View* and *Action Controller*, the relation between the *ORM* objects and the construction of a *View* from *Layout*, *Blocks* and *Templates* all supported by *Helper* objects.

User Interface

The user requests a page via web *Browser* and receives rendered *HTML, CSS* and *images* in return.

Controller

Based on the page request the correct *(Action) Controller* instantiates and manipulates the corresponding *Object Models*. In the next step (5), the *Layout* is instantiated. When the *Layout*

[4] http://www.magentocommerce.com/knowledge-base/entry/magento-for-dev-part-5-magento-models-and-orm-basics, accessed 04.07.2011.

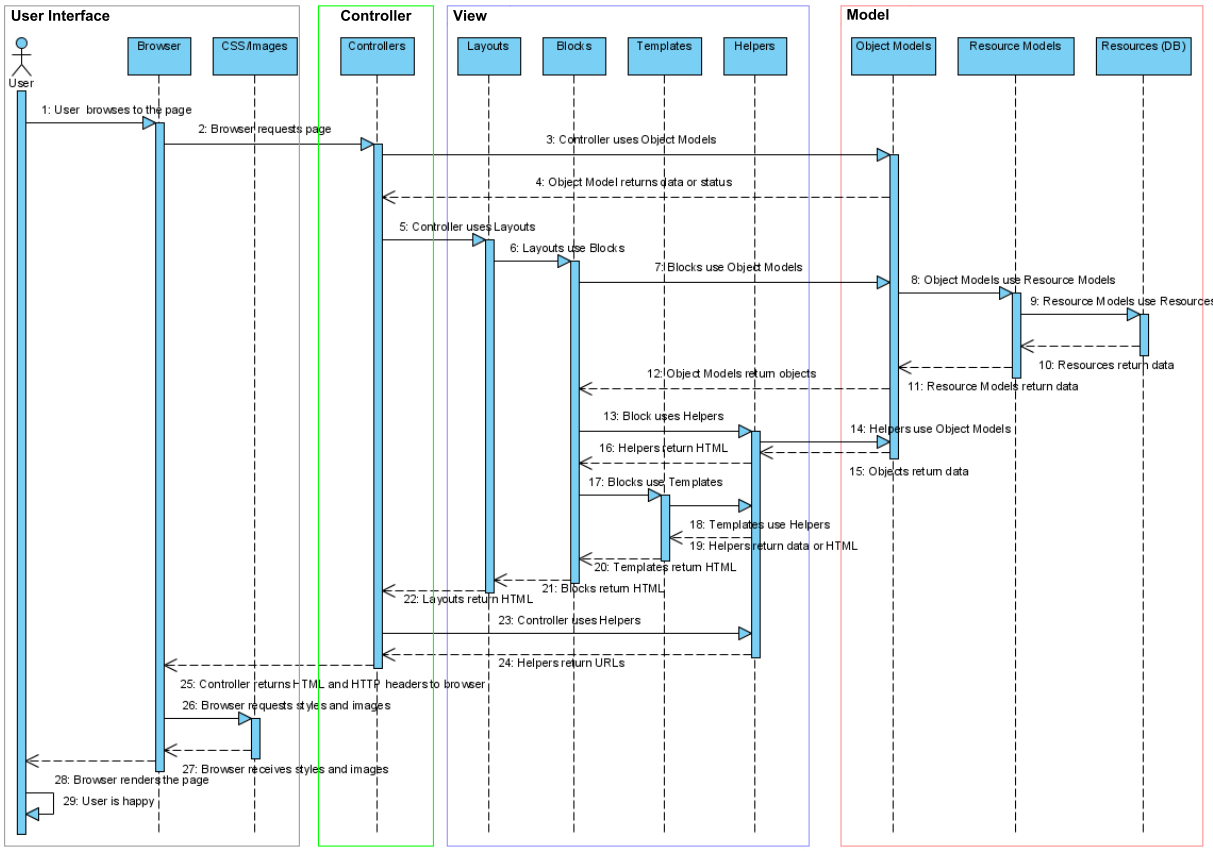

Figure 11.7: Magento page request sequence diagram and model-view-controller interaction. (Source: adapted from Magento Inc. [59][a])

[a] http://www.magentocommerce.com/wiki/_media/general/doc/page-request-flow.png, accessed 29.06.2011.

is returned, a *Helper* is used to generate the corresponding URL to redirect, and eventually the rendered HTML and HTTP headers are submitted to the user.

View

The *Layout* uses its *Blocks* which together with their *Helpers* retrieve data from the *Object Models*. At they same time they use static *Templates* to generate HTML that is integrated in the top-level *Layout* that is rendered and returned to the controller.

Model

In Magento database access operations are grouped and only executed when they are actually needed, that is when the data is read (not when the controller instantiates the object model (step 3)) [59][5]. This is why only when the *Blocks* retrieve data from the model to generate their content, the *Resource Model* and as consequence the *Database Resource* are accessed (steps 7, 8, 9, 10, 11).

[5] http://www.magentocommerce.com/knowledge-base/entry/magento-for-dev-part-5-magento-models-and-orm-basics, accessed 04.07.2011.

Chapter 12

Testing and Software Validation

In this chapter the approaches to verify and validate the software are discussed including the results from user-acceptance tests.

12.1 Unit Testing

Unit testing represents the lowest level of testing verifying the actual source code. For PHP the de-facto standard framework for unit testing is PHPUnit developed by Sebastian Bergmann[1]. The framework simplifies the creation and the actual running of the test including easy analysis of the test results.

Employing a *test-first approach* PHPUnit was used for all unit tests concerning new modules or module customisations during the implementation of the IVHM B2B platform. A example of such a unit test can be found in listing 12.1. The custom *AdvancedAdminPermissions* module is tested if it only returns the number of items permitted for one manufacturer in the backend.

```
1  <?php
2  require_once 'PHPUnit/Framework.php';
3  require_once '../app/Mage.php';
4
5
6  class Rdw_AdvancedAdminPermissions_Test extends
       PHPUnit_Framework_TestCase {
7
8    /**
9     * Init the test case and the Magento application.
10    *
11    */
12   public function setUp()
13   {
14     Mage::app('default');
```

[1] http://www.phpunit.de, accessed 10.06.2011.

```
15      }
16
17      /**
18       * Checks the number of products delivered by the restricted backend
19       * model for a certain SupplierAdmin - 'Manufacturer A'.
20       *
21       * @test
22       */
23      public function testGetRestrictedProductCollection()
24      {
25        $source = Mage::getModel('catalog/product')->getCollection()
26        ->getResource()
27        ->getAttribute('manufacturer')
28        ->getSource();
29
30        $manufOptionId = Mage::getSingleton('catalog/config')->
              getSourceOptionId($source, 'Manufacturer A');
31
32        $helper = Mage::helper('catalog/product'); // SUT
33        $collection = $helper->getRestrictedProductCollection($manufOptionId)
              ;
34
35        $this->assertEquals(3, $collection->count()); // assertion with 3
              items in the DB
36      }
37  }
38  ?>
```

Listing 12.1: Unit test for supplier admin restrictions to manage products in the backend.

12.2 User Interface and Use Case Validation

In this section the most important functionalities and user interfaces of the platform are presented in order to validate the use cases.

The system has to major user interfaces the *frontend* and the *backend*. The *frontend* is the interface for the actual users of the platform where they can can browse and search for IVHM produts and technologies, group them in a product basket or save them for later reference.

The *backend* is the administrator interface of the platform for two different types of administrators. There is the *platform administrator*, that can configure and maintain all parts of the system, and the *supplier administrators*, suppliers of IVHM capabilities that manage their products and technologies, maintain their microsite and communicate with users on the platform.

12.2.1 Frontend

Main Frontend User Interface

Figure 12.1 shows a screenshot of the major platform user interface in the frontend.

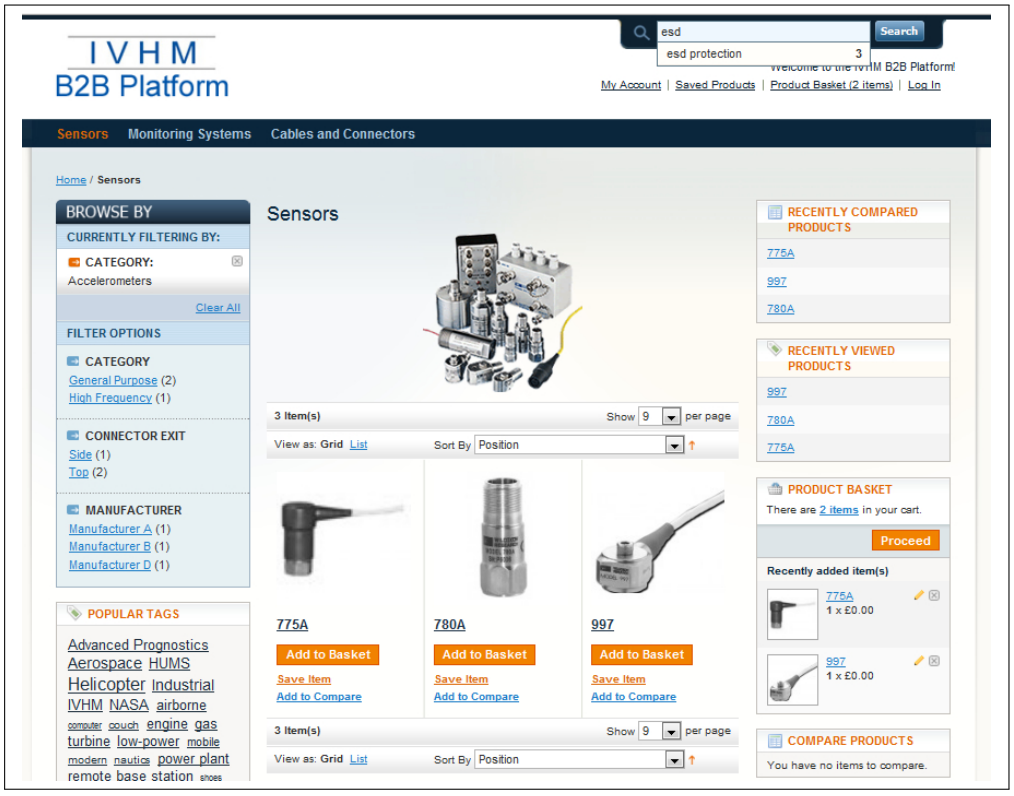

Figure 12.1: Screenshot of main frontend user interface for product browsing and simple product search.

In the top right there is a *search field* for searching the whole platform for key words (in this example 'esd'). The system is offering an auto-completion feature and delivers all the search results in the content area of the interface.

Below the platform user can *manage his account*, access *saved products* or products in the *product basket*.

To navigate on the platform the links at the top (e.g. 'Sensors' or 'Monitoring Systems') provide access to the *catalogue's category tree*, where as the *'browse by'* box at the left-hand side offers *filtering* options (e.g. by manufacturer or category).

At the bottom left a *tag cloud* display the most popular tags for IVHM items on the platform. In the central content area all the products or technologies for the specified category, filtering options, tags or search results are displayed. There the items can be added to the *product basket*, *saved*, or added the *item comparison feature*.

The right-hand side shows boxes for *recently compared* and *recently viewed products* for easy reference, the *product basket* with the stored items, and the *compare box*. There those items are stored the user would like to compare.

Frontend Advanced Search

Figure 12.2 shows the 'Advanced search' user interface and sample search results for an attribute-related search.

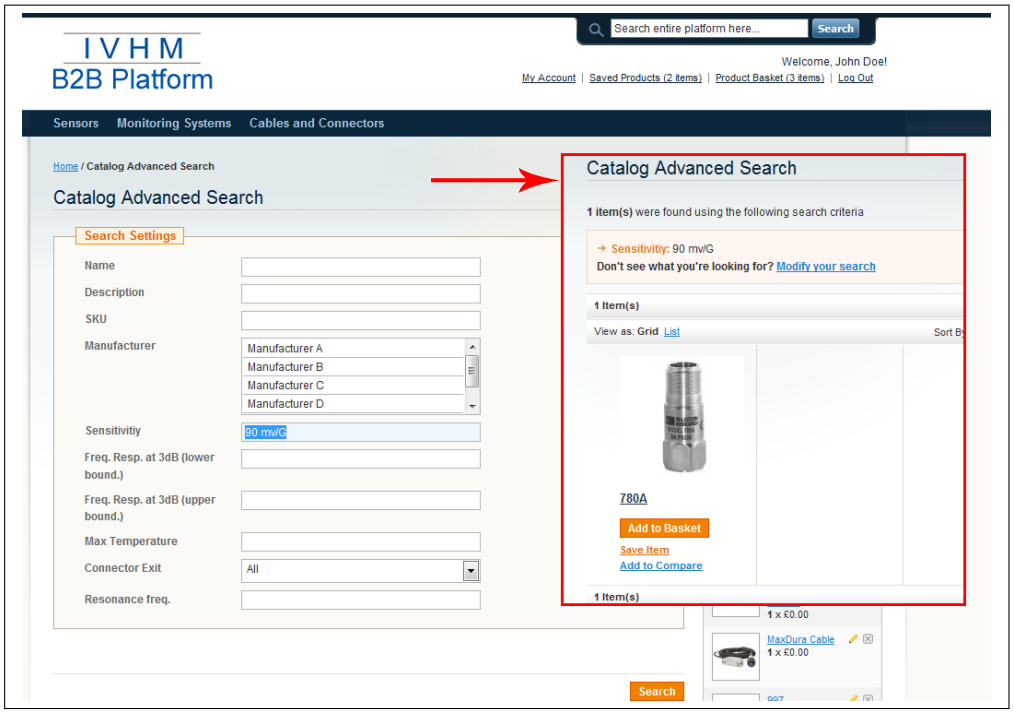

Figure 12.2: Screenshot of advanced search with inline search results in the platform frontend.

Frontend IVHM Item View

Figure 12.3 shows the IVHM product view on the platform. It provides all the details and specifications for an IVHM product or technology. The user has the opportunity to directly *forward the item by email* or *comment* on it on the platform to engage with the supplier (hyperlinks next to product image).

At the bottom the item's *tags* are displayed whereas the user can add self-defined tags. At the top right the system displays *related items*. There the system recommends compatible components and systems. The remaining boxes on the right-hand side are equal to the ones from the main user interface.

Frontend Product Comparison

Figure 12.4 displays the interface for product comparison. After the user has added items to the *comparison box*, by clicking 'Compare' the system opens the comparison interface. The user can easily compare each attribute or feature of the item, print the comparison, save an item or store it in the product basket.

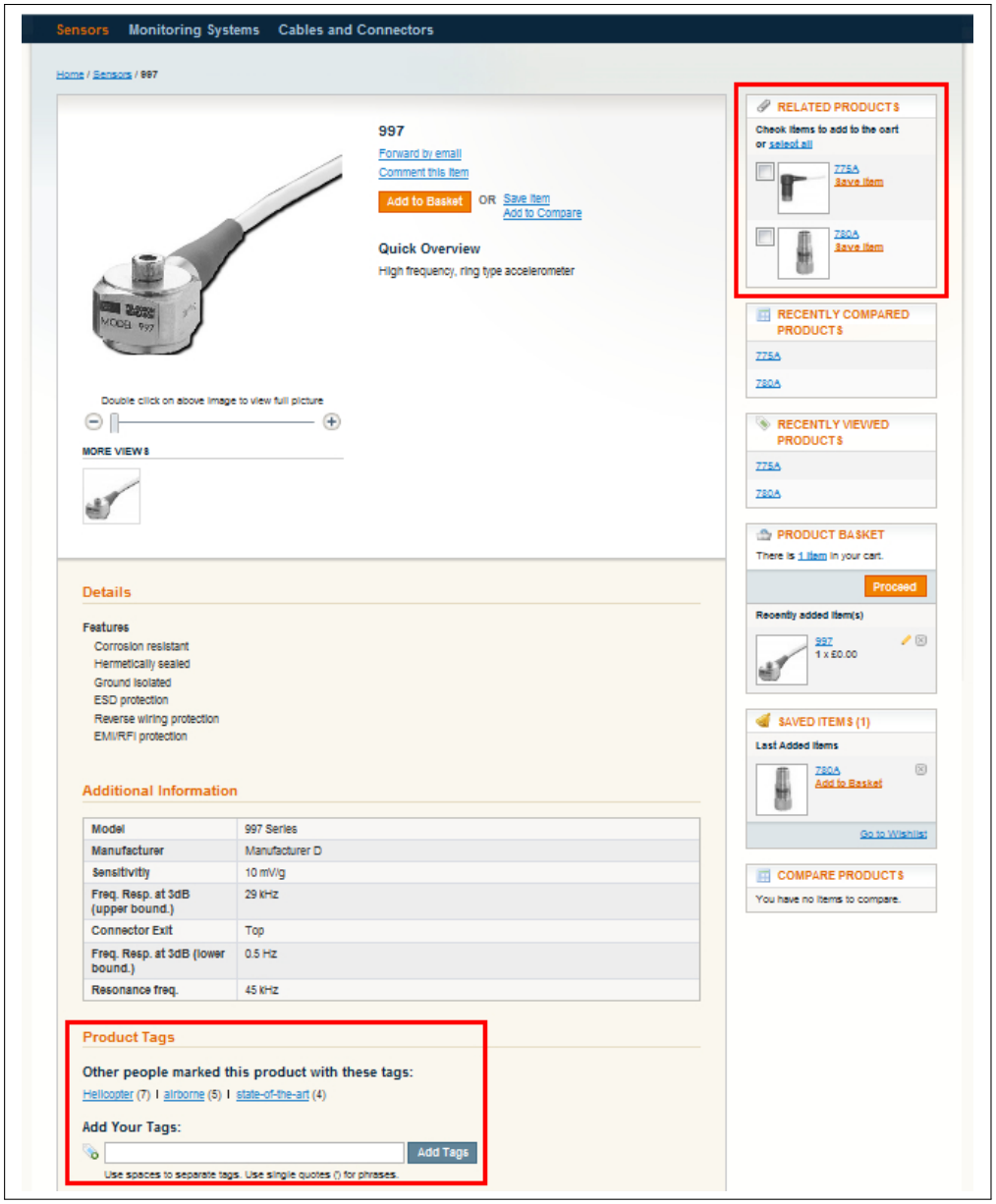

Figure 12.3: Detailed product view including related platform items and tagging feature in the frontend.

12.2.2 Backend

Figure 12.5 shows the main backend user interface for the platform administrator who has full access to all features and the full configuration of the system. The navigation pane is marked with a red box. The screenshot displays the *administrator user management* interface displaying three administrators: the *platform administrator* himself and two *supplier administrators (SupplierAdmin)* with restricted rights.

The inline screenshot marked with a red arrow and box shows the role the user 'ManufacturerA' is given. As 'SupplierAdmin' the user has only limited access to the backend.

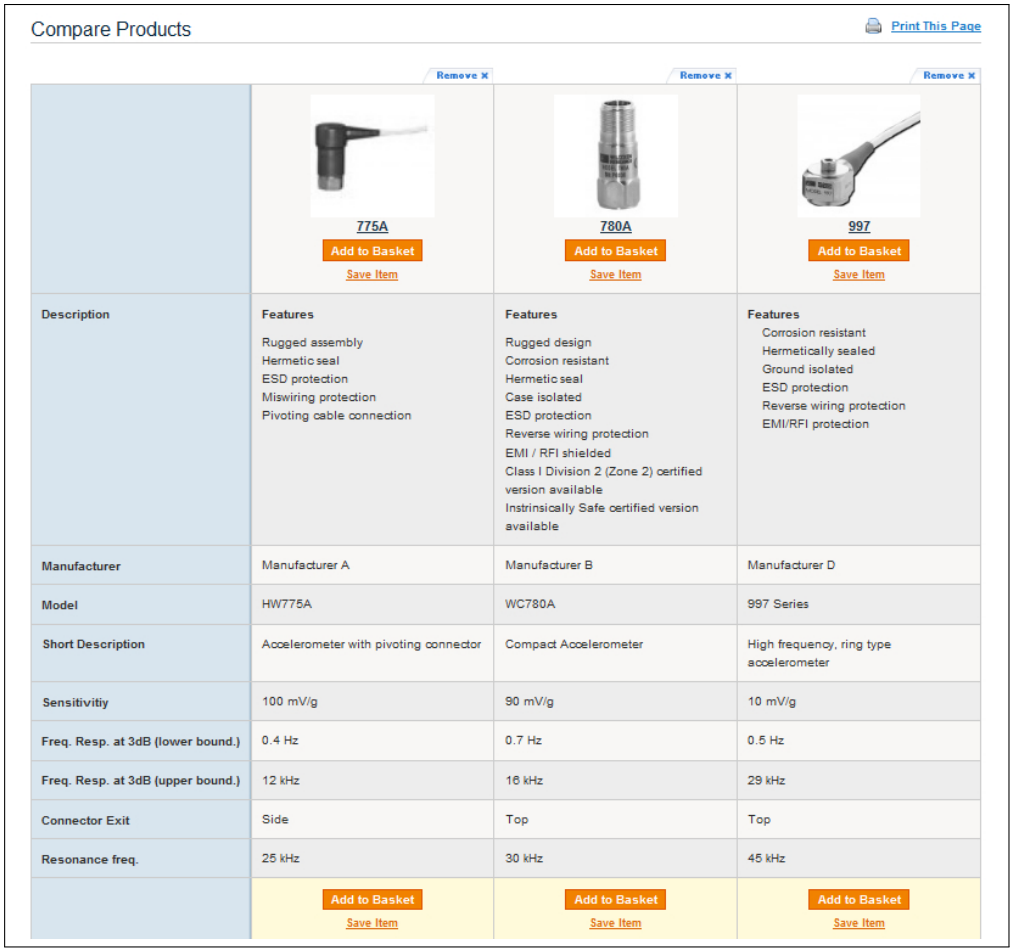

Figure 12.4: Screenshot platform comparison functionality with three different products from different manufacturers in the frontend.

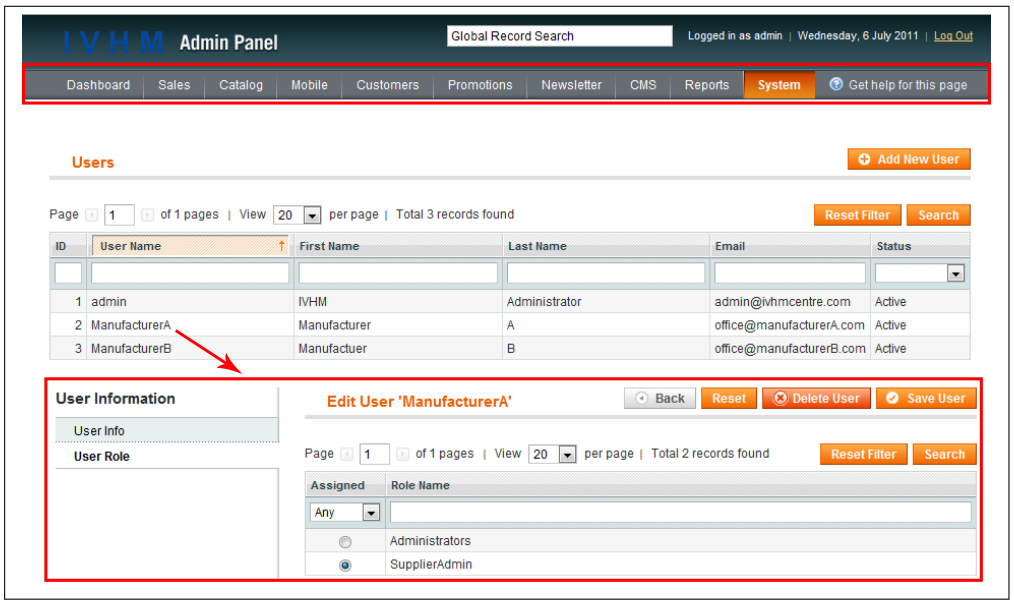

Figure 12.5: Screenshot of administrator backend with full permissions and user management interface.

Backend Product Management as Restricted Administrator

Figure 12.6 shows the backend user interface for the restricted *SupplierAdmin* administrator.

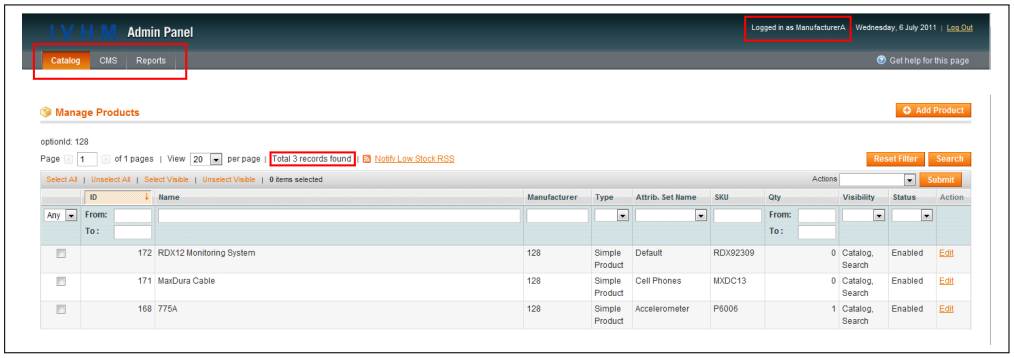

Figure 12.6: Screenshot of platform backend for a supplier user (SupplierAdmin) and the 'Manage Products' interface.

Due to the implementation of the advanced rights management system the user has only access to three areas marked by the red box in the administrator navigation: the *Catalog* including product and category management, the *CMS* to manage the supplier's microsite and the *Reports*.

Whereas the platform administrator has access to all products and CMS pages, the SupplierAdmin can only access areas and items that concern him, thus only his microsite, reports and communication concerning his products, and categories and IVHM products and technologies this supplier 'owns'. The figure shows the only three products of 'ManufacturerA' on the platform that holds several hundred items.

Figure 12.7 displays the user interfaces for the management of a specific product. On the left-hand site all the different attribute categories are shown with the 'Dynamic Properties' of item '775A (Accelerometer)' opened to be configured.

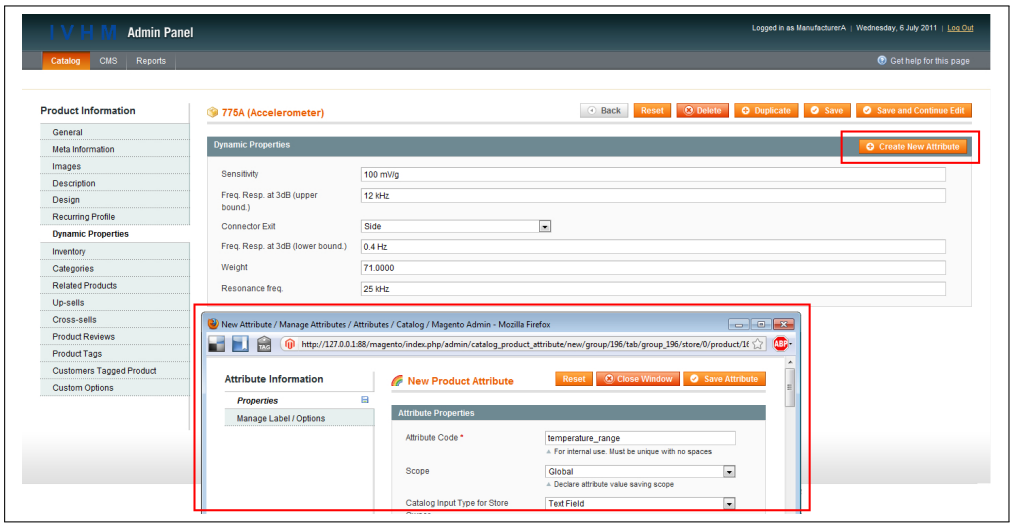

Figure 12.7: Screenshot of platform backend showing the interface to edit an item and to add a custom item attribute.

In addition to that the user can simply add a *new attribute* to an *attribute group* of an *entity* by

clicking *Create New Attribute*. The system then opens a new window as shown in the red box, where the user defines the new attribute (in this example 'temperature_range').

Backend Microsite Management

Figure 12.8 shows the interface where a supplier can manage its platform microsite. Employing a WYSIWYG editor to edit and format the microsite (including inserting pictures as for example the logo) does not require the user to have profound web design skills.

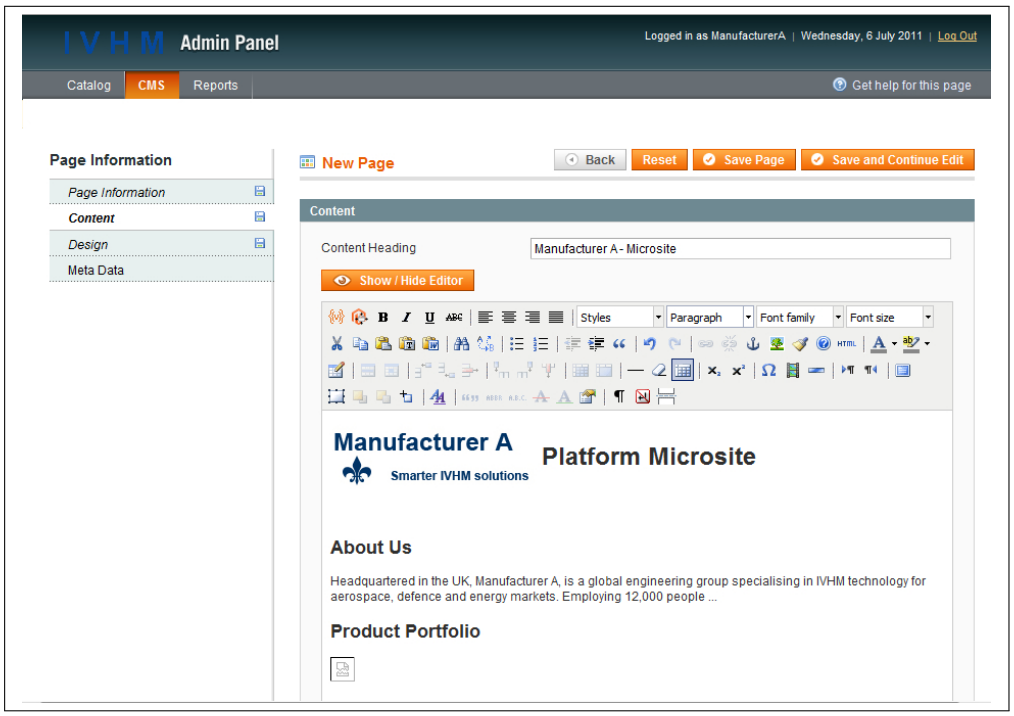

Figure 12.8: Screenshot of platform backend showing the interface to edit a supplier's platform microsite with the included WYSIWYG editor.

12.3 Software Validation and User Acceptance Tests

12.3.1 Software Validation Strategy

Software validation is the final step to ensure that the system meets the user expectations. To do so, *UATs* are employed in order to validate the quality as well as the usability of the system having real users interact with the system.

A three-fold approach is used to validate the software and its usability. A a *pre-defined test set* (see section 12.3.2) is employed which covers all required functionalities of the system. The test users are asked to perform the defined actions in the frontend and the backend of the system.

In order to gather feedback from the test users, two techniques are used: *Thinking-Aloud (THA)* tests and a *usability questionnaire*. Each test user is asked to fill a questionnaire after the test run. See section 12.3.3 for the questionnaire. In addition to that, for a subset of the users

THA is used, thus the developers observe the test users and ask them to express their thoughts aloud while interacting with the system. This way valuable feedback for the improvement of the system can be gathered.

12.3.2 Test Set

The test set comprises 7 actions for the frontend as well as for the backend. This test set covers all platform functionalities.

Frontend

1. *Login* as user 'johndoe@testbuyer1.com' with password 'asdfasdf1'.
2. *Browse* platform using the top navigation and find *Sensor* '775A'.
   - Navigate to item information page and *tag* the item with the keyword 'jet_engine'.
3. Using the platform's compare feature *compare* accelerometers '775A' and '780A'.
4. Add accelerometer '775A' and 'RDX12 Monitoring System' to the *Product Basket* and 'Proceed' sending an enquiry with random text to their suppliers.
5. Perform an *advanced search* for an item with a *Sensitivity* of '90 mV/g'.
6. *Save* the found item '780A' in the system for later reference.
7. *Logout*.

Backend

1. *Login* as supplier administrator 'ManufacturerA' with password 'asdfasdf1'.
2. Goto content management system (*CMS*) in the top navigation and *edit page* 'Supplier 1'.
   - Modify the contact details in the 'Content' of the microsite, change the phone number to '+44 1234 1234' and save the page.
3. Goto top 'Catalog→Manage Categories'. Navigate to category 'Accelerometers' and *add a new sub-category* 'Low Frequency'.
4. In 'Catalog→Manage Products' find and open the *edit* interface for item '775A'.
   - *Remove Product Tag* 'jet_engine' from item '775A'.
   - Goto 'Categories' for item '775A' and *change* its *category association* to 'Low Frequency'. Save the item.
5. In 'Manage Products' *add a new item*.

- Attribute set: 'Accelerometer'; Product type: 'Simple Product'.
- Name: 'Test Product'; Model: 'TP1'; Status: 'Enabled'; Short and full description: random.
- In 'Categories' associate item with the category 'Low Frequency'.
- 'Save and Continue Edit' the item.
- In attribute group 'Dynamic Properties' add a *new attribute* with code 'amplitude_nonlinearity'.
  - In 'Manage labels' set *Admin Title* to 'Amplitude nonlinearity'
  - *Save* the attribute.
- Go to attribute group 'Dynamic Properties' set a value of '1%' for 'Amplitude Nonlinearity' of the new product 'Test Product'.

6. *Check* the buyer *enquiries* for your products.
7. *Logout*.

12.3.3 Usability Questionnaire

A good questionnaire is the basis to have significant results in terms of platform usability. Tullis and Stetson [94] in their study *A Comparison of Questionnaires for Assessing Website Usability* studied 5 different usability questionnaires and identified System Usability Scale (SUS), which was developed by John Brooke for DEC Ltd. in 1986 [10], as delivering the most reliable results.

The SUS questionnaire comprises 10 statements using a Likert scale from 1 to 5 representing variations between *Strongly disagree* and *Strongly agree*. Theses statements cover a variety of system usability aspects such as complexity and the need for training and support. [10]

The SUS questionnaire is shown in appendix B. In addition to the original set of items the following statement was added for this particular usability study:

I feel proficient in using online e-commerce platforms such as Amazon or Ebay.

The same Likert scale applied measuring the proficiency of a tester in using similar systems. Nevertheless, the item is not included in the overall SUS scoring.

The SUS itself yields a single score for the overall usability of the system between 0 and 100Out of these 10 statements the SUS. The statements contribute to the final score differently. For statements {1, 3, 5, 7, 9} the sale position needs to be reduced by 1 to receive the score contribution. For statements {2, 4, 6, 8, 10} the scale position is subtracted from 5 to receive the contribution. The overall sum of score contributions yields the final SUS score. [10]

12.3.4 Thinking-Aloud Test Results

Six test users were employed in thinking-aloud tests. The users followed the test set instructions while expressing their thoughts aloud about using the system. The main feedback was that the system is in general easy to use and intuitive. The users especially noted the similarity to known website interfaces such as Amazon as beneficial for its usability. In addition to that, each test user stated explicitly that due its vast feature set the system appeared to be quite complex at first sight, but proved to be intuitive while using it. All THA users agreed that one would get used to the system very quickly when using more often.

Positive areas that two are more THA test users emphasised was the efficiency of the search feature and its intuitive positioning in the top right, the easy-to-use compare feature and its added value. Furthermore, the backend despite its first-sight complexity was perceived logically structure and quite intuitive in general.

Areas for Improvement

The areas for improvement that were identified by the test users are mostly concerned with the design and layout of certain system interfaces. In only a few cases a feature was misunderstood or an additional feature requested. The identified areas for improvement and their treatment is discussed in the following list.

- **Frontend**
  - The links in the top right of the interface were perceived as too small in font size. The corresponding design was adapted.
  - The advanced search link was moved from the page footer to the link area in the top right of the frontend due to test user feedback.
  - The compare feature widget was perceived to be too far down on the right-hand side of the user interface. It was thus placed closer to the header.
  - One of the test user would have preferred to right-click an item in order to tag it with a keyword. This is a feature that could be considered for the next phase of the platform.
- **Backend**
  - The system messages such “Successfully saved item” were perceived as too small and not highlighted enough. The font size as well as the highlighting was adapted accordingly.
  - In the *Manage Products* interface the text fields used for filtering platform items were intuitively misinterpreted by two test users as a way to add a new item. A posteriori the users perceived the feature as good especially when managing many items.

- A global search feature was requested for the backend in order to simplify navigation through the menus. This feature could be considered for the next phase of the platform.
- The majority of the users felt overwhelmed with the amount of features and configuration possibilities of the backend at first sight. Nevertheless at the same time they stated that they would get used to the interface very quickly.
- Performing the task of adding a new item to a sub-category one user was confused that instead of navigating first to the category and adding the item there in contrast to his intuition the new item is created first and afterwards associated with its category.
- Two test users, being both new to the system and the context of IVHM proposed to provide examples and more detailed descriptions for the attributes to be filled for platform items. This fact is acknowledged for the test users, but is not considered an issue for the environment where the system will be deployed.
- One test user was not used to using small 'plus signs' in order to expand the category tree. A remark in the user manual could mitigate this issue.

12.3.5 Usability Questionnaire Results

In addition to the 6 THA test users, 8 test users were handed the usability questionnaire after interacting with the system without a thinking-aloud protocol. 14 users have tested the system and filled the usability questionnaire. The results are shown in table 12.1. Before discussing the results general SUS scores are explored further with reference to literature.

Bangor et al. [3] developed and adjective rating scale for the SUS thus providing an absolute judgment of the usability of a system due to its SUS score. In their study they found a SUS mean score of 68.2 (n=1433) for web applications and 69.5 (n=3463) for all user interfaces considered. Regarding the absolute scale the authors developed the adjective rating as shown in figure 12.10. The figure also shows the acceptability and grading scale ratings in relation to the average SUS score. Bangor et al. [3] found that for a system's usability to be considered as *good* the mean SUS score is 71.4, and 85.5 for *excellent* usability.

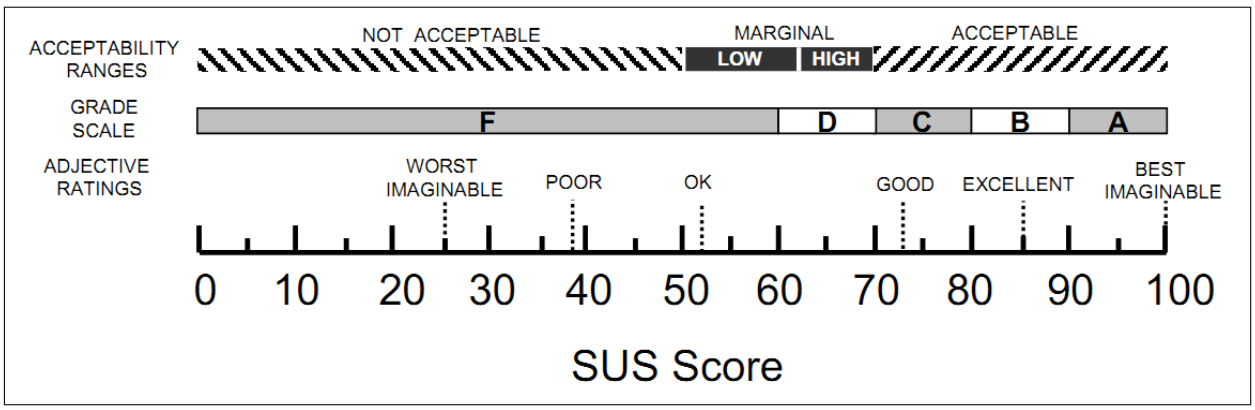

Figure 12.9: Average SUS scores mapped onto adjective ratings, a school grading scale and acceptability scores. (Source: adpated from Bangor et al. [3])

The questionnaire test results for the IVHM B2B prototype show an average SUS score of 70.7 which is only slightly below the score that Bangor et al. [3] consider as *good* in terms of usability. Nevertheless, the SUS score for test users that considered themselves as proficient, agreeing or strongly agreeing (rating 4 or 5) with the statement in section 12.3.3, is 77.5. This score is significantly higher than the general average and suggests that users with more experience in using these kinds of systems also perceive it as more usable. The scatter chart in figure 12.10 shows a linear relation between proficiency self-assessment and SUS score for the results of this specific system. Furthermore, the average score of *Q7*, if the system is perceived to be easy to learn, shows that all test users agree on this fact.

Even though the number of 14 test users is not enough in order to prove statistical significance the scores give a good indication of how the usability of the system is perceived. A prototype rated *good* in terms of SUS is a good starting point for the pilot phase and the potential implementation of a final software product.

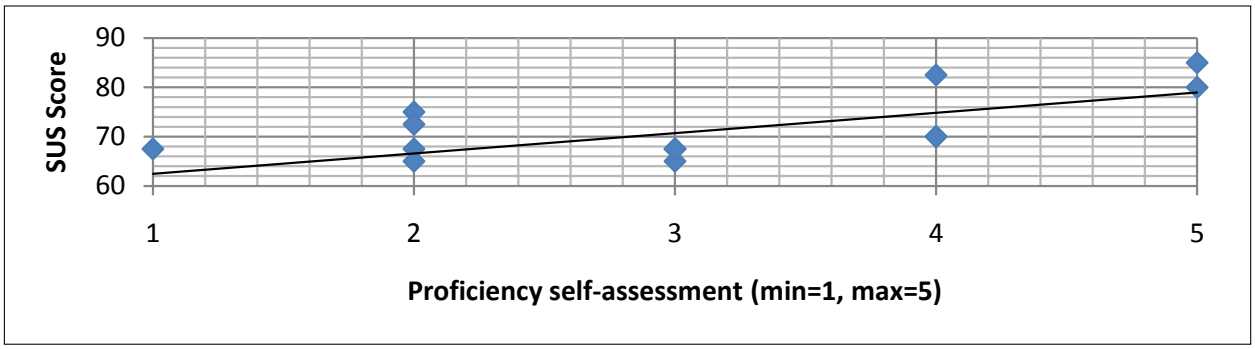

Figure 12.10: Scatter chart of the test users' SUS scores and corresponding proficiency self-assessment in interacting with e-commerce websites including regression line.

12.3.6 Software Validation and User-Acceptance Tests Conclusions

The thinking-aloud tests showed that the platform prototype is perceived well by the users, provided valuable feedback for minor corrections of the prototype and inputs for features of a potential final platform. The tests employing the SUS questionnaire also rated the usability of the prototype *good* in general also showing that the users experience a steep learning curve in using systems of that kind. Nevertheless, with regard to the SUS score the system is quite distant from being rated *excellent*. It should be a major focus of the pilot phase of the prototype to find out the causes of the usability issues and identify the improvements in user interface and platform features to be made before implementing the final software product.

User	Q0[1]	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS
T1[2]	2	4	2	4	1	4	1	4	3	3	2	75
T2[2]	3	4	2	3	3	3	2	4	3	4	2	65
T3[2]	2	4	1	3	3	3	3	5	1	3	3	67,5
T4[2]	5	4	2	4	2	4	2	5	2	4	1	80
T5[2]	5	4	2	4	1	5	2	5	2	4	1	85
T6[2]	4	3	2	4	1	5	2	5	2	4	1	82,5
T7	3	5	2	3	3	3	3	4	2	4	2	67,5
T8	3	4	2	3	2	3	2	4	2	4	3	67,5
T9	2	5	1	3	1	3	3	5	3	3	2	72,5
T10	1	3	3	5	1	4	3	4	2	3	3	67,5
T11	4	4	2	4	1	3	3	5	3	4	3	70
T12	4	4	3	4	2	4	2	5	3	3	2	70
T13	2	3	2	3	2	3	2	5	3	3	2	65
T14	2	3	3	4	3	3	2	3	3	3	3	55
Average	*3,0*	*3,9*	*2,1*	*3,6*	*1,9*	*3,6*	*2,3*	*4,5*	*2,4*	*3,5*	*2,1*	**70,7**
								Expert average[3]				**77,5**

Table 12.1: Ratings for each SUS statement and SUS scores for all test users.

[1] Proficiency self-assessment (1 minimum, 5 maximum).

[2] Test users that also took part in thinking-aloud tests.

[3] Average SUS scores of all test users with a proficiency self-assessment rating (Q0) of 4 or 5.

Part III

Discussion and Conclusions

Chapter 13

Results and Discussion

This chapter summarises the results of the work done, discusses the lessons learned and the significance of the achievements.

13.1 Results

The area of IVHM is growing strongly and requires efficient collaboration between buyers and suppliers of the corresponding technology. The analysis of the IVHM business context has shown that buyer-supplier relationships and thus their collaboration varies between joint development and simple COTS sourcing. However, even though IVHM buyers and suppliers in most cases work very closely together, complications in the collaboration such as limited or difficulties in the access to information on IVHM technology, low convenience in the interaction between the partners, or high costs in transaction and keeping up the relationships persist.

Market research showed that only services exist that are focused on sales and purchasing enabled by e-commerce. Exostar was identified as major service in the A&D area providing integration and collaboration services along the value chain. There is no platform dedicated to IVHM even though marketplaces of related areas, for instance aerospace, might offer IVHM related COTS products. A central platform facilitating the identification of suitable IVHM technology and the exchange of information with the corresponding suppliers at reasonable cost does not exist at all.

In this project an information intermediary was developed and implemented as prototype in order to tackle the mentioned challenges. At the beginning of the project a hypothesis that supports such a platform's feasibility was established combined with three platform objectives that deliver major benefits for the community. Following the designed research methodology the platform's technically feasibility and the its objectives were successfully assessed.

Therefore, at first the business context and its complications were analysed and documented representing the first deliverable of this project. Examining the features of general online marketplaces it was concluded that a platform comprising some of these features could alleviate or even solve many of the complications identified in the IVHM buyer-supplier interaction. The

prior analyses fed directly into the software requirements specification document of a prototype the information intermediary in form of the *IVHM B2B Platform*.

Magento was identified as the most suitable open source framework to serve as basis for the platform. Implementing the prototype Magento proved to be a very powerful and flexible system. With the implementation of the IVHM B2B Platform the main objective of the thesis was achieved. This way the online marketplace concept was successfully introduced to this completely new context constituting an innovation for the IVHM community.

The implemented prototype differs greatly from conventional marketplaces. Whereas conventional online marketplaces facilitate sales and purchasing, the IVHM platform serves as pure information intermediary to support information exchange and collaboration while using concepts and features from online marketplaces.

Considering the aim, the prototype proves the intermediary's technical feasibility and comprises the features that form the basis for the platform in order to deliver value for the community. Section 13.3 discusses to what extent the platform can deliver value and assesses the platform's objectives and benefits.

Concluding, the project aim was successfully achieved by accomplishing all project objectives and delivering the analysis and documentation of the business context, a concise software requirements specification document and a software prototype of the IVHM B2B platform.

13.2 Discussion of Applied Methodology

The research methodology applied in this project proved to be a successful approach to achieve the project's aim and objectives. Nevertheless, despite the tailoring to the project's specificity the general structure of the applied methodology is not restricted to it. It could be used in any other context where a software system potentially solves a problems or delivers benefits with the prototype approach being an effective way to assess a system's feasibility and benefits.

Best-practice methods identified in literature were used throughout all phases of the software development life cycle. Thus, the methodology can be used for any software prototype development project with a similar short-time-frame, single-developer setup. Depending on the context and the project configuration different methods for the various phases might be applied. For example, with a bigger developer team and more time a different development methodology might be used, or a more sophisticated software selection approach might have to be employed in order to take a more complex software selection decision. However, the phased framework of the methodology will remain the same.

For the software evaluation and selection phase two different methods were applied. Both identified *Magento* as the best solution considering the evaluation criteria and the software requirements. One major finding is that for the *Weighted Scoring Method* developing individual weights for each criterion given a large number of criteria does not deliver better results than just averaging the scores. Thus, the less cumbersome and more practical approach can be applied for a large criteria set and small rating bandwidth for the different criteria.

The *Analytical Hierarchy Process* on the other hand in accordance with literature was found to deliver much more reliable results at the drawback of filling evaluation matrices that grow polynomially with the number of alternatives and criteria. Concluding, for a reliable analysis up to a certain degree of decision complexity AHP should always be preferred over WSM.

To validate the system and retrieve user feedback user-acceptance tests with Thinking-Aloud tests and the System Usability Scale questionnaire were used. THA tests were confirmed to be an efficient approach to gain insights on the actual user interaction with the system providing valuable input for user interface improvements. The SUS questionnaire on the other hand is an effective way to assess the system's overall usability with the possibility to benchmark the results by means of an absolute scale.

Concluding, the applied methodology was the absolutely right approach for this project with the achievement of its aims and objectives proving its success.

13.3 Discussion of Platform Objectives and Benefits

At the beginning of the project three major platform objectives with associated benefits were hypothesised. The platform should improve the efficiency in terms of information exchange, serve the intensification of business relationships and collaboration beyond IVHM systems and facilitate the general business initiation between IVHM partners.

13.3.1 Efficiency Improvement in Information Exchange and Communication

Several aspects of the implemented platform prototype prove that the first objective is achieved. The platform centralises all information on state-of-the-art IVHM products and technologies including compatibility information provided by the IVHM suppliers. Thus, it streamlines the information retrieval process for these products from a buyer point of view. The platform solves the buyer-side complications of high transaction and search costs since the user can easily browse and perform specific searches on IVHM technology. It also offers the advantage that *all* capabilities of the suppliers represented on the platform including possible higher-quality new-technology substitutes are centralised and searchable in one single system.

The possibility to centrally comment and discuss technology and products improves the bidirectional communication while tackling the complication of low convenience in supplier-buyer interaction. At the same time this helps the suppliers to better understand their customers.

The *Product Basket* feature, i.e. the configuration of a specific combination of IVHM products and technology, offers the buyers the possibility to easily enquire information from the concerned suppliers for a specific setup. Receiving enquiries for certain IVHM system configurations also helps the suppliers to understand the requirements and the needs of their customers in general better.

The suppliers also benefit from a reduction in sales and marketing costs to inform buyers and keep up the corresponding relationships due to the fact that the information and interaction in

terms of IVHM technologies is centralised.

Summarising the arguments and platform features, the IVHM B2B platform provides efficiency improvements in information exchange and buyer-supplier communication and thus achieves objective 1.

13.3.2 Collaboration beyond IVHM

Considering the technical aspects and the implementation of the platform evidence fo the achievement of objective 2 is provided. The concept of IVHM itself is not constrained to vehicles and can be easily extended to any other area of complex technical assets while being interlinked with other systems. To serve this matter the platform supports products and technologies in the extended scope of IVHM. In fact, the platform is derived from a generic e-commerce platform and is implemented in a very generic way such that actually any kind of product or capability can be advertised on the platform on a B2B basis.

The technical implementation, in particular the data layer comprising the entity-attribute-value approach including an intuitive object-relational mapping, put no constraint whatsoever on what entities or items are dealt with on the platform. This means that the context of this platform could be easily extended to any other B2B area where the centralisation of information and of buyer-supplier communication delivers benefits. Thus, the suppliers are not constrained in broadening their advertised technology portfolio on the platform. In consultation with the platform operator the scope of platform portfolio can be easily extended to any desired area which in turn serves business relationships and collaboration beyond IVHM technology.

13.3.3 Facilitation of Business Initiation

The achievement of platform objective 3 of facilitating business initiation cannot be proven without collecting empirical data with the platform actually deployed and in use. At this point in time with the prototype only being ready for the pilot phase this data cannot be collected. Nevertheless, it is argued that with the first two benefits proven and the platform solving further complications in IVHM buyer-supplier-buyer relationships, the platform will also deliver the business-initiation benefit.

If buyers have a much simpler and continuous access to the available state-of-the-art technologies and are always kept up-to-date by means of the platform, if buyer-supplier communication and thus mutual understanding can be improved such that suppliers can better tailor their products to their customers, and if the collaboration on systems and thus business beyond IVHM is served, initiation of business is obviously facilitated by the platform.

Furthermore, as the platform is open to all registered buyers, it is easier for suppliers to broaden their customer base, which means the initiation of business with previous non-customers that are active on the platform. At the same time the transaction costs to initiate such new business relationships are reduced because the platform first of all provides buyers with the necessary information, for instance a platform search for products with a certain specification, and secondly

simplifies the active approach of these suppliers. Hence, even without empirically proving that platform objective 3 is achieved the argumentation's logical inference strongly corroborates its validity.

13.4 Discussion of IVHM B2B Platform Innovation

According to the previous sections the IVHM B2B platform achieves its objectives and associated benefits that were set out at the beginning of the project. This section discusses why and in what way the platform constitutes an innovation by comparing it Exostar services.

Exostar[1] as the leading provider of B2B collaboration and supply chain integration solutions for the A&D industry offers a variety of products throughout the extended value chain. Figure 13.1 shows the different Exostar products along the value chain such as *SourcePass* to streamline the sourcing process, *ForumPass* to enable secure collaboration and information sharing, and the *Supply Chain Platform (SCP)* offering supply chain planning and execution integration with different suppliers

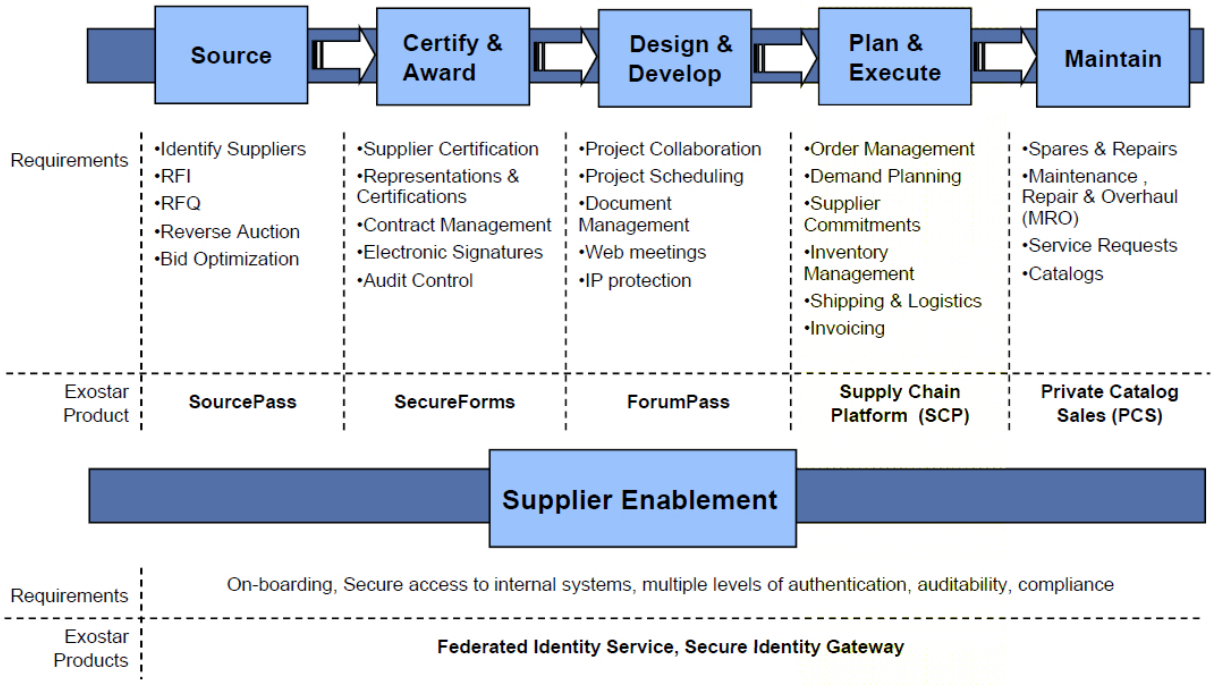

Figure 13.1: Exostar product portfolio throughout the value chain. (Source: adapted from Exostar [26])

Despite the vast portfolio of Exostar products and its reach in terms of buyers and suppliers in the A&D industry its focus to streamline and reduce costs of B2B interaction in a secure environment does not tackle the mentioned challenges imposed on the IVHM community. The *IVHM B2B Platform* fills this gap introducing the innovation to apply the online marketplace concept to the IVHM context.

[1] http://www.exostar.com, accessed 20.07.2011.

Two of Exostar's solutions as shown in figure 13.1 are tangent with the IVHM B2B Platform: SourcePass and ForumPass. *SourcePass*[2] is focused on gathering information from the suppliers on an RFx (e.g. Request For Information or Request For Proposal) basis and pricing topics. However, there is no possibility for buyers to browse, search and filter existing products and capabilities of suppliers in order to support the buyers' design processes.

Exostar's design and development product *ForumPass*[3] provides an integrated enterprise collaboration solution including information sharing and a concurrent working environment in a secure workspace employing Microsoft SharePoint. It offers full buyer-supplier interaction features including wikis and blogs as knowledge management repositories. The information sharing and interaction features fully cover a part of the IVHM B2B platform scope and much beyond, offering collaboration functionalities to manage all aspects throughout the project life cycle.

ForumPass is a tool for OEMs to interact with suppliers that have already been selected and are already integrated in the design and development process. Nevertheless, this is not necessarily the objective of the IVHM B2B Platform where buyers want to investigate and interact with potentially new suppliers of state-of-the-art IVHM technology.

The IVHM B2B Platform compared to the Exostar solutions provides a central system where IVHM products and technologies are centralised including simple search and filtering capabilities. It also offers buyers to interact with all suppliers in one central platform. The platform thus fills the gap of getting an overview of and locating certain technology before and during the design process.

Buyers also have the possibility to retrieve further information from the suppliers before fully integrating them in their for instance Exostar collaboration environment. Suppliers on the other hand are provide with the opportunity to centrally advertise their capability portfolio, interact with potential buyers and possibly broaden their customer base. Thus, the IVHM B2B Platform constitutes a 'meet-up' service for IVHM technology that is not provided by Exostar but tackles the challenges mentioned in the IVHM community.

The IVHM B2B platform can thus be seen as complementary to Exostar where the suppliers and technologies are identified in order to be integrated in the Exostar value chain services including pricing, supplier auditing and certification checks, design collaboration and supply chain execution.

Concluding, the IVHM B2B Platform provides an innovation that solves the identified issues in the IVHM buyer-supplier interaction while supplementing services such as Exostar for full collaboration and system integration.

13.5 Platform Success Factors and Barriers of Adoption

In order for the IVHM B2B Platform to be successful and economically viable several factors including the prototype's current limitations need to be considered. In particular the strategic

[2] http://www.exostar.com/products-SourcePass.aspx, accessed 20.07.2011.
[3] http://www.exostar.com/productsForumPass.aspx?ekmenselc580fa7b_30_386_btnlink, accessed 20.07.2011.

implications of buyer-supplier relationships deserve special attention concerning a successful adoption of the platform.

13.5.1 Data Integration

Entering data into the system is a manual process for the prototype. That means a supplier needs to access the administrator backend of the system to enter new IVHM products or technologies. This is an extremely cumbersome and inefficient process that needs to be automated. If this is not the case, the manual entry costs are likely to be higher than the actual delivered benefits for the suppliers. Thus, during the pilot phase import software customised to the suppliers' data sources need to be written in order to automate the transfer of IVHM product and technology data to the platform databases.

In the long-term and considering a potential final production version of the platform the aim should be to directly integrate the suppliers' product data sheet management software, their ERP system or even their design tools with the platform such as offered by Exostar. This way the platform can be continuously and consistently kept up-to-date with the platform automatically and directly pulling IVHM data from the suppliers' systems.

13.5.2 Critical Mass

The second factor is concerning the number of available suppliers and IVHM items on the platform. The platform needs to cover a *critical mass* of products and suppliers in order to be attractive for buyers. Without covering this critical mass the platform cannot fully deliver its benefits since buyers would still have to use the conventional way of technology investigation and supplier interaction for most of their needs.

Nevertheless, the critical-mass argument has a strong link to the factor introduced in the beginning. Providing an easy and automated way to import and transfer product and technology data is essential to acquire this critical mass of IVHM items as well as suppliers on the platform.

13.5.3 Buyer and Supplier Strategies and Confidentiality

Factor three was discovered throughout the study of the business context analysing the relationship between IVHM suppliers and OEMs that implement IVHM technology. Their strategies in the corresponding business relationships contradict each other. OEMs want to broaden their supply base in order to increase the competition between their suppliers and reduce costs but at the same time they do not want to share suppliers of state-of-the-art technology with other OEMs. However, these special suppliers want to broaden their customer base and reduce their dependency on certain OEMs.

The platform serves both ways depending on the perspective. It serves the competition between the suppliers on the platform because it makes their products easily comparable but also offers them the possibility to extend their customer base because all OEMs active on the platform have

access to their portfolio. This really helps smaller suppliers with certified high-quality technology that do not have established relationships with big OEMs. At the same time this also means that potentially other suppliers have access to the competitor's technology and product portfolio on the platform.

For obvious reasons suppliers are really sensitive with sharing all their capabilities and technologies with competitors especially with the platform facilitating the access to this information. The balance between sharing enough information to acquire customers, and not sharing too much information not to threaten one's own competitive advantages needs to be found.

The issue will also strongly depend on the environment where the platform is deployed and its initial partners or users. The mentioned implications of sourcing strategy and information sharing do not impose any issues on the platform as long as it is exclusively used by the partners involved in the Cranfield IVHM Centre. The partners have set agreements in terms of information sharing and confidentiality such a platform would go in line with. Thus no issues for the pilot phase are expected.

13.5.4 Other Barriers of Adoption for the Platform Pilot

The automation of data import can be seen as major barrier of adoption during the pilot phase. Also the platform benefits cannot be fully leveraged during the initial phase since the number of suppliers and IVHM items will be limited.

Considering the introduction of a new software platform the barriers of adoption for IVHM B2B platform can be discussed with regards to a similar context in literature. Loukis et al. [57] studied the barriers of adoption of online marketplaces for large enterprises in a case study for the strongly IVHM related aerospace sector. They identified major barriers of organisational and technological nature, mainly due to inconsistent and complex processes, rules and regulations, IT systems and lacking trust to unknown suppliers. The authors intent was to leverage existing B2B marketplaces to reduce purchasing costs and inventory, and lower marketing costs while increasing sales, respectively. However, this does not apply for the IVHM B2B platform since it only serves as information intermediary without supporting purchasing and sales transactions.

Thus, due to the special type of the IVHM B2B platform the barriers discussed by Loukis et al. [57] are rather low. The technological barrier except for the automated data import is not given to that large extent. The system is hosted by an operator and during the pilot no integration with other IT systems is intended.

Both buyers and suppliers can access the externally hosted system simply via web browser and since it is only used as information exchange non of the parties has to perform major organisational or procedural changes. Only the right areas in the organisations have to be granted access, which would be marketing, sales and R&D for suppliers, and R&D and purchasing for the buyers. There is also no issue of lacking trust in (new) suppliers as long as the system is used in the closed environment of the Cranfield IVHM Centre. With a potential 'going public' of the platform its operator needs to set clear standards in line with the OEM's requirements for supplier certifications in order for a supplier to be granted membership to the platform.

13.6 Business Model and Next Steps

With the platform now being proven feasible from a technical and qualitative economic benefit point of view the pilot phase will serve as basis for a platform business case and cost-benefit analysis where the benefits and costs need to be quantified precisely.

Throughout this cost-benefit analysis a business model, for instance based on a membership fee, needs to be designed in order to make the IVHM B2B Platform also commercially viable in the long-term. There the closed policy needs to be rethought in particular with regard to gaining the critical mass in order to really deliver a benefit for the suppliers and buyers which they would be ready to pay for. Assuming the Cranfield IVHM Centre as operator of the platform together with its partners a sophisticated rights and permission scheme needs to be elaborated in order to find the balance for the strategic as well as confidentiality implications mentioned. There should be no barrier for the platform being opened to the public IVHM community if appropriate platform terms and regulations are set in place and the system provides an absolutely secure environment with proper identity, rights and permission management.

Chapter 14

Conclusions and Future Work

This project created a prototype of an information intermediary in form of the *IVHM B2B Platform* for the IVHM community in order to further facilitate the collaboration between buyers and suppliers of IVHM technology. The platform constitutes an innovation in complementing existing services for information exchange an collaboration such as Exostar with a 'meet-up' service between buyers and suppliers of IVHM technology.

The business context analysis showed that the IVHM community collaborates in various business scenarios in between co-development of IVHM technology and simple sourcing of COTS parts. Throughout the collaboration the buyers face the major complications of having high transaction and search costs related to IVHM technology, access to only limited suppliers and no transparency of the potential supply base. Suppliers face complications of high costs in sales and marketing to keep up business relationships, difficulties to broaden the customer base and improve the understanding of their customers.

To tackle these challenges the concept of an online marketplace was successfully adapted to this new context of IVHM. Extracting the features of existing marketplaces and mapping them on to corresponding issues in the IVHM community a concise software requirements specification was documented as basis for a prototype implementation of the IVHM B2B Platform.

The requirements served as basis for the prototype-based feasibility study that was conducted. With a set of requirements-inspired evaluation criteria six software solution approaches for the prototype including custom development were evaluated. Therefore the Weighted Scoring Method and the Analytical Hierarchy Process were employed. Even though both recommended the same solution, AHP was concluded to deliver much more reliable results with the disadvantage of requiring more effort.

The open source e-commerce platform *Magento* was selected as most appropriate basis for the IVHM B2B Platform. During the implementation phase Magento proved to be a very powerful and flexible framework and customising it a fully functional IVHM B2B Platform prototype was implemented. Employing user-acceptance test in terms of Thinking-Aloud tests and the System Usability Scale as usability questionnaire the system prototype was validated with the client.

The UATs showed that the system is intuitive and easy to learn comprising advantages for users that have more experience with e-commerce websites such as Amazon or Ebay.

The prototype itself not only proves the technical feasibility of the platform but considering the features offered also achieves the objectives and benefits of the platform in the area of information exchange, supplier-buyer communication, intensification of collaboration beyond IVHM and general facilitation of business initiation between IVHM buyers and suppliers. The deliverables produced, the analysis and documentation of the business context, the concise software requirements specification and the functional prototype, serve as basis for the next phase of the project.

14.1 Limitations and Future Work

The implemented system prototype is fully functional and will be deployed as pilot at the IVHM Centre at Cranfield University. Nevertheless, as defined from the beginning the prototype is limited with regards to software security and reliability and can only be deployed inside a secure IT environment such as the IVHM Centre. In addition to that, due to lack of access to supplier data the prototype does not provide automated import functions which would simplify the setup of the prototype's databases with suppliers and their IVHM products and technologies.

The next step for the platform is to be rolled out as a pilot inside the IVHM Centre. Incorporating the implementation of data import functions the prototype will then be evaluated from a technical, performance and economic point of view. The initial platform requirements are reevaluated and refined during the pilot phase including a potential enhancement of the platform's feature set.

A business model for the platform taking all strategic implications in buyer-supplier and supplier-supplier relationships into account needs to be designed in order to calculate the business case. The business case considerations require a quantitative evaluation of the platform objectives and benefits, that have only been assessed qualitatively throughout this project.

With the post-pilot refinement of the requirements, the business model and the estimated costs of the implementation of a potential final platform product a cost-benefit analysis will decide the next steps for the IVHM B2B Platform prototype and if it will be refurbished or reimplemented for final production use.

References

[1] Aaseng, G. B. Blueprint for an integrated vehicle health management system. In *Proceedings of the 20th Digital Avionics Systems Conference*, volume 1, pages 3C1/1–3C1/11, Daytona Beach, FL, USA, 2001. IEEE.

[2] Baines, T. S., Braganza, A., Kingston, J., Lockett, H., Martinez, V., Michele, P., Tranfield, D., Walton, I., and Wilson, H. State-of-the-art in product service-systems. In *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pages 1543–1552. Sage Publications, 2007.

[3] Bangor, A., Kortum, P., and Miller, J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies (JUS)*, 4(3):114–123, 2009.

[4] Banks, J. and Merenich, J. Cost Benefit Analysis for Asset Health Management Technology. In *Proceedings of the 2007 Annual Reliability and Maintainability Symposium - RAMS'07*, pages 95–100, Orlando, FL, USA, 2007. IEEE. ISBN 0780397665.

[5] Benedettini, O., Baines, T. S., Lightfoot, H. W., and Greenough, R. M. State of the art in integrated vehicle health management. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, volume 223. SAGE Publications, 2009.

[6] Bertolino, A. and Marchetti, E. A brief essay on software testing. In Thayer, R. H. and Christensen, M. J., editors, *Software Engineering - The Development Process*. Wiley-IEEE Computer Society Press, Hoboken, NJ, USA, 3rd edition, 2005. ISBN 978-0-471-68417-6.

[7] Beynon-Davies, P. *E-Business*. Palgrave Macmillan, Basingstoke, Hampshire, UK, 2004. ISBN 140391348X.

[8] Bird, G., Christensen, M., Lutz, D., and Scandura, P. A. Use of integrated vehicle health management in the field of commercial aviation. In *Poceedings of the 1st International Forum on System Health Engineering and Management in Aerospace - NASA ISHEM Foum 2005*, Napa Valley, CA, USA, 2005.

[9] Boehm, B. W. Software engineering: R&D trends and defense needs. *Research directions in software technology*, 1979.

[10] Brooke, J. SUS - A "quick and dirty" usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A., and McClelland, I. L., editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis, London, 1996.

[11] Business Wire. More Double-Digit Growth Ahead For Online Retail In US And Western Europe, accessed May 15th, 2011. URL `http://www.businesswire.com/news/home/20110228005297/en/Double-Digit-Growth-Online-Retail-Western-Europe`.

[12] Chaffey, D. *E-business and e-commerce management: strategy, implementation and practice*. FT Prentice Hall, Essex, UK, 3rd edition, 2007. ISBN 9780273719601.

[13] Choi, T., Ellram, L., and Koka, B. Supplier-supplier relationships and their implications for buyer-supplier relationships. *IEEE Transactions on Engineering Management*, 49(2): 119–130, 2002.

[14] Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. Non-Functional Requirements in Software Engineering. In Borgida, A., Chaudhri, V., Giorgini, P., and Yu, E., editors, *Conceptual Modeling Foundations and Applications*, volume 5 of *Lecture Notes in Computer Science*, chapter 19, pages 363–379. Kluwer Academic Publishers, 2000. ISBN 0792386663.

[15] Clason, D. L. and Dormody, T. J. Analyzing Data Measured by Individual Likert-Type Items. *Journal of Agricultural Education*, 35(4):31–35, 1993.

[16] Copeland, L. Extreme Programming, accessed June 17th, 2011. URL `http://www.computerworld.com/s/article/66192/Extreme_Programming`.

[17] Craig, R. D. and Jaskiel, S. P. *Systematic Software Testing*. Archtech House Publishers, London, 2002. ISBN 1580535089.

[18] Cranfield IVHM Centre. About Us, accessed April 7th, 2011. URL `http://www.cranfield.ac.uk/ivhm/aboutus/index.html`.

[19] Davis, A. M. *Software Requirements: Objects, Functions and States*. Prentice-Hall, Upper Saddle River, NJ, USA, 1993. ISBN 013805763X.

[20] Denne, M. and Cleland-Huang, J. *Software by Numbers: Low-Risk, High-Return Development*. Sun Microsystems Press, Prentice Hall, Santa Clara, CA, USA, 2003. ISBN 0131407287.

[21] Dinu, V. and Nadkarni, P. Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. *International Journal of Medical Informatics*, 76(11-12): 769–79, 2007.

[22] Dubey, A. and Wagle, D. Delivering software as a service. *McKinsey Quarterly*, 6(May): 1–12, 2007.

[23] Duening, T. N. and Click, R. L. *Essentials of Business Process Outsourcing*, volume 2. Wiley, 2005. ISBN 0471709875.

[24] EC21. About Us, accessed May 15th, 2011. URL http://www.ec21.com/html/ec/AU/AU_Overview.html.

[25] Eliott, S. *Electronic commerce: B2C strategies and models*, volume 12. John Wiley & Sons, Chichester, UK, 2002. ISBN 0471487058.

[26] Exostar. Plan & Execute - Supply Chain Platform - Overview and Demonstration, (accessed 20.07.2011), 2008. URL http://www.exostar.com/DocumentViewer.aspx?id=2026.

[27] Exostar. Homepage, accessed July 20th, 2011. URL http://www.exostar.com.

[28] Feather, M. S., Uckun, S., and Hicks, K. A. Technology Maturation of Integrated System Health Management. In *Proceedings of the 6th Conference on Human/Robotic Technology and the Vision for Space Exploration, Space Technology & Applications International Forum (STAIF-2008)*, pages 827–838, Albuquerque, NM, USA, 2008.

[29] Financial Times. Tesco checks in as fourth biggest online retailer, (accessed 04.04.2011), 2006. URL http://www.ft.com/cms/s/0/933f9680-eb8a-11da-823e-0000779e2340.html.

[30] Fowler, M. GUI Architectures, (accessed 18.05.2011), 2006. URL http://www.martinfowler.com/eaaDev/uiArchs.html.

[31] Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., and Stafford, R. *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, MA, USA, 2002. ISBN 0-321-12742-0.

[32] Fox, J. J. and Glass, B. J. Impact of integrated vehicle health management (IVHM) technologies on ground operations for reusable launch vehicles (RLVs) and spacecrafts. In *Proceedings of the 2000 IEEE Aerospace Conference*, pages 179–186, Big Sky, MT, USA, 2000.

[33] Franch, X. and Carvallo, J. P. A quality-model-based approach for describing and evaluating software packages. In *Proceedings of the IEEE Joint International Conference on Requirements Engineering*, pages 104–111, 2002.

[34] Franch, X. and Carvallo, J. P. Using quality models in software package selection. *IEEE Software*, 20(1):34–41, 2003. ISSN 07407459.

[35] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns CD: Elements of reusable object-oriented software*, volume 47 of *Addison-Wesley Professional Computing Series*. Addison-Wesley Longman, Chicago, IL, USA, 1998. ISBN 0201309521.

[36] Gora, W. and Mann, E. *Handbuch Electronic Commerce: Kompendium zum elektronischen Handel*. Springer-Verlag, Berlin/Heidelberg/New York, 2nd edition, 2001.

[37] Graham, G. and Ahmed, P. Buyer-supplier management in the aerospace value chain. *Integrated Manufacturing Systems*, 11(7):462–468, 2000. ISSN 0957-6061.

[38] Gustafson, D. *Schaum's Outlines: Software Engineering.* McGraw-Hill, 2002. ISBN 0071377948.

[39] Haig, M. *The B2B e-commerce handbook.* Kogan Page Ltd, London, 2001. ISBN 0749435771.

[40] Hess, A. J., Calvello, G., and Dabney, T. PHM a key enabler for the JSF autonomic logistics support concept. In *Proceedings of the 2004 IEEE Aerospace Conference*, pages 3543–3550, Big Sky, MT, USA, 2004. ISBN 0780381556.

[41] Holzinger, A. Usability engineering methods for software developers. *Communications of the ACM*, 48(1):71–74, 2005. ISSN 00010782.

[42] Hsia, P., Kung, D., and Sell, C. Software requirements and acceptance testing. *Annals of Software Engineering*, 3(1):291–317, 1997.

[43] IEEE Computer Society. IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998*, 1998.

[44] IEEE Standards Board. IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, 1990.

[45] Illik, J. A. *Electronic Commerce. Grundlagen und Technik für die Erschließung elektronischer Märkte.* Oldenbourg Verlag, 2002.

[46] Jadhav, A. and Sonar, R. Evaluating and selecting software packages: A review. *Information and Software Technology*, 51(3):555–563, 2009. ISSN 09505849.

[47] Jadhav, A. S. and Sonar, R. M. Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software*, 2011. ISSN 01641212.

[48] Jalote, P. *A Concise Introduction to Software Engineering.* Undergraduate Topics in Computer Science. Springer-Verlag, London, 2008. ISBN 9781848003019.

[49] Jamaludin, A., Hashim, R. A., and Yahya, Y. Developing Malaysian Micro Entrepreneur Through E-Business. In *Proceedings of the 2nd International Conference on Business and Economic Research (2nd ICBER 2011)*, pages 1527–1540, Langkawi Kedah, Malaysia, 2011.

[50] Juristo, N. Verification and Validation: Current and Best Practice. Technical Report RTO-TR-IST-027, NATO Research and Technology Organisation, 2005.

[51] Juristo, N., Moreno, A. M., and Vegas, S. Reviewing 25 Years of Testing Technique Experiments. *Empirical Software Engineering*, 9(1/2):7–44, 2004. ISSN 13823256.

[52] Kaner, C., Falk, J. L., and Nguyen, H. Q. *Testing Computer Software.* John Wiley & Sons, New York, 2nd edition, 1999. ISBN 0471358460.

[53] Kaplan, J. M. SaaS: Friend Or Foe? *Business Communications Review*, 37(6):49–53, 2007.

[54] Koufteros, X., Edwincheng, T., and Lai, K.-H. 'Black-box' and 'gray-box' supplier integration in product development: Antecedents, consequences and the moderating role of firm size. *Journal of Operations Management*, 25(4):847–870, 2007. ISSN 02726963.

[55] Kulak, D. and Guiney, E. *Use Cases: Requirements in Context*. Addison-Wesley Professional, 2nd edition, 2003. ISBN 9780321154989.

[56] Lianghua, X., Liqing, R., Mei, Z., Lixin, W., and Qiang, M. Research on open system architecture for equipment health management based on OSA-CBM. In *Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2010)*, Guilin, China, 2010.

[57] Loukis, E., Spinellis, D., and Katsigiannis, A. Barriers to the Adoption of B2B e-Marketplaces by Large Enterprises: Lessons Learned From the Hellenic Aerospace Industry. *Information Systems Management*, 28(2):130–146, 2011. ISSN 1058-0530.

[58] Lowe, M., Tokuoka, S., Dubay, K., Gereffi, G., Palilla, M., Pracht, M., Roberts, M., Stimson, R., and Wochele, C. U. S. Manufacture of Rail Vehicles for Intercity Passenger Rail and Urban Transit. Technical report, Center on Globalization, Governance & Competitiveness, Duke University, Durham, NC, USA, 2010.

[59] Magento Inc. Magento Wiki, accessed April 7th, 2011. URL `http://www.magentocommerce.com/wiki/`.

[60] Maiden, N. A. M. and Ncube, C. Acquiring COTS software selection requirements. *IEEE Software*, 15(2):46–56, 1998.

[61] Melillo, L. Use ROI analysis for business software investments, accessed March 12th, 2011. URL `http://office.microsoft.com/en-us/access-help/use-roi-analysis-for-business-software-investments-HA001199974.aspx`.

[62] Microsoft MSDN. Deployment Patterns - Layered Application, accessed March 12th, 2011. URL `http://msdn.microsoft.com/en-us/library/ff650258.aspx`.

[63] Microsoft MSDN. Model-View-Controller, accessed March 12th, 2011. URL `http://msdn.microsoft.com/en-us/library/ff649643.aspx`.

[64] Miller, R. B. Response time in man-computer conversational transactions. *Proceedings of the AFIPS Fall Joint Computer Conference*, 33:267–277, 1968.

[65] Myers, G. J., Thomas, T. M., and Sandler, C. *The Art of Software Testing*, volume 1. John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2004. ISBN 0471469122.

[66] Nadkarni, P. M., Marenco, L., Chen, R., Skoufos, E., Shepherd, G., and Miller, P. Organization of heterogeneous scientific data using the EAV/CR representation. *Journal of the American Medical Informatics Association (JAMIA)*, 6(6):478–93, 1999. ISSN 1067-5027.

[67] Naik, K. and Tripathy, P. *Software Testing and Quality Assurance - Theory and Practice*, volume 1. John Wiley & Sons, Hoboken, NJ, USA, 2008. ISBN 9780470382844.

[68] Nikoukaran, J. and Paul, R. J. Software selection for simulation in manufacturing: a review. *Simulation Practice and Theory*, 7, 1999.

[69] Nock, C. *Data Access Patterns: Database Interactions in Object-Oriented Applications*. Addison-Wesley, Boston, MA, USA, 2003. ISBN 0-13-140157-2.

[70] Ohnemus, K. R. and Biers, D. W. Retrospective Versus Concurrent Thinking-Out-Loud in Usability Testing. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, pages 1127–1131, Santa Monica, CA, USA, 1993.

[71] Pal, K. and Campbell, J. A. An Application of Rule-Based and Case-Based Reasoning within a Single Legal Knowledge-Based System. *The Data Base For Advances In Information Systems*, 28(4):48–63, 1997.

[72] Perry, W. *Effective methods for software testing*. John Wiley & Sons, New York, 3rd edition, 2006. ISBN 9780764598371.

[73] Pezzè, M. and Young, M. *Software Testing and Analysis - Process , Principles and Techniques*. John Wiley & Sons, Hoboken, NJ, USA, 2008.

[74] Porter, M. E. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, 1985. ISBN 0029250900.

[75] Pressman, R. S. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, London, 5th edition, 2000. ISBN 0077096770.

[76] Rayport, J. and Jaworsky, B. *Introduction to e-commerce*. McGraw-Hill Higher Education, New York, 2002. ISBN 0072510242.

[77] Ritter, T. and Walter, A. Matching high-tech and high-touch in supplier-customer relationships. *European Journal of Marketing*, 40(3/4):292–310, 2006. ISSN 0309-0566.

[78] Rosenberg, D. and Kendall, S. *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*. Addison-Wesley, 2001. ISBN 0201730391.

[79] Runeson, P. A survey of unit testing practices. *IEEE Software*, 23(4):22–29, 2006. ISSN 07407459.

[80] Runeson, P., Andersson, C., Thelin, T., Andrews, A., and Berling, T. What Do We Know about Defect Detection Methods? *IEEE Software*, 23(3):82–90, 2006.

[81] Saaty, T. How to make a decision: The analytic hierarchy process. *European Journal Of Operational Research*, 48(1):9–26, 1990. ISSN 03772217.

[82] Saaty, T. L. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1):83, 2008. ISSN 17531446.

[83] Sabbah, D. The future of software delivery. Technical report, IBM, 2007.

[84] Sarkis, J. Evaluating and selecting e-commerce software and communication systems for a supply chain. *European Journal Of Operational Research*, 159(2):318–329, 2004. ISSN 03772217.

[85] Scandura, P. A. Integrated vehicle health management as a system engineering discipline. In *Proceedings of the 24th Digital Avionics Systems Conference (DASC 2005)*, pages 7.D.1–1–10, 2005. ISBN 0-7803-9307-4.

[86] Segev, A., Porra, J., and Roldan, M. Internet-based EDI strategy. *Decision Support Systems*, 21(3):157–170, 1997. ISSN 01679236.

[87] Sledgianowski, D., Tafti, M. H. A., and Kierstead, J. SME ERP system sourcing strategies: a case study. *Industrial Management Data Systems*, 108(4):421–436, 2008. ISSN 02635577.

[88] Sommerville, I. *Software Engineering*. Pearson Education, Essex, UK, 8th edition, 2007. ISBN 9780321313799.

[89] Standish Group. CHAOS Summary 2009, (accessed 05.04.2011). Technical report, The Standish Group International, Inc., 2009. URL `http://www.standishgroup.com/newsroom/chaos_2009.php`.

[90] Tewoldeberhan, T. W., Verbraeck, A., and Hlupic, V. Implementing a discrete event software selection methodology for supporting decision making at Accenture. *Journal of the Operational Research Society*, 61(10):1446–1458, 2009.

[91] The Economist. Another digital gold rush. *The Economist Print Edition*, 399(8733):78–81, 2011.

[92] Timmers, P. *Electronic Commerce: Strategies and Models for Business-to-Business Trading*. John Wiley & Sons, Chichester, UK, 1999.

[93] Tozzi, J. Bloomberg Businessweek - Top Online Marketplaces, (accessed 20.04.2011), 2008. URL `http://images.businessweek.com/ss/08/11/1107_ecommerce/index.htm`.

[94] Tullis, T. S. and Stetson, J. N. A Comparison of Questionnaires for Assessing Website Usability. In *Proceedings of the Usability Professionals Association (UPA) 2004 Conference*, Minneapolis, MN, USA, 2004.

[95] Turban, E. and King, D. *Introduction to E-Commerce*. Pearson Education, Prentice Hall, Upper Saddle River, NJ, USA, 2003.

[96] Turban, E., Lee, J., King, D., and Chung, H. M. *Electronic commerce: a managerial perspective*. Prentice Hall, New Jersey, 2000. ISBN 0139752854.

[97] US Census Bureau. Annual Retail Trade Report. Technical report, US Census Bureau, 2009. URL `http://www.census.gov/retail/`.

[98] US Department of Commerce. The U.S. Jet Transport Industry - Competition, Regulation, and Global Market Factors Affecting U.S. Producers. Technical report, US Department of Commerce, Washington, DC, USA, 2005.

[99] US Department of Defense. Technology Readiness Assessment (TRA) Deskbook. Technical report, US Department of Defense, 2009.

[100] Van Solingen, R. Measuring the ROI of software process improvement. *IEEE Software*, 21(3):32–38, 2004. ISSN 07407459.

[101] Williams, Z. Benefits of IVHM: an analytical approach. In *Proceeding of the 2006 IEEE Aerospace Conference*, Big Sky, MT, USA, 2006.

[102] Zend Technologies Ltd. Zend Framework, accessed May 12th, 2011. URL `http://framework.zend.com/`.

[103] Zhu, H. *Software design methodology.* Butterworth-Heinemann, Elsevier, Oxford, UK, 2005. ISBN 978-0-7506-6075-4.

Appendix A

Software Evaluation

In this chapter the scores for the considered software packages for the different evaluation methods are elaborated.

A.1 Weighted Scoring Method

Magento		
Code	**Score**[1]	**Comment**
F0	5	Sophisticated user management and multiple administrators possible.
F1	2	Not possible for users, rudimentary for admins.
F2	5	Advanced CMS.
F3	4	Shopping Cart and Wishlist feature.
F4	4	Slightly advanced catalogue and category management.
F5	5	Standard feature.
F6	5	Excellent system to define and manage products, add own attributes and cateorise them.
F7	5	Fully integrated tagging system.
F8	4	Good search system.
F9	4	Good graphical reporting features.
Q0	5	Excellent implementation of MVC, separation of design, logic and data, Event/Observer, clean code and excellent documentation.
Q1	5	Very reliable with no crashes or display issues.
Q2	5	Secure setup even offering SSL.
Q3	5	State-of-the-art separation between data, logic and presentation, object-relational mapping, best practice implementation of design patterns.
U0	4	Advanced error reporting for user and admins.
U1	5	Simple and intuitive.
V0	5	Widely deployed professionally, enterprise edition is available.
C0	3	With given functionality and source code quality rather fair. Risk of underestimating software complexity.

Table A.1: Analysis of evaluation criteria for Magento.

[1] Very poor corresponds to 1, very good corresponds to 5.

OSCommerce		
Code	**Score**[1]	**Comment**
F0	3	Multiple admins possible but very basic user management.
F1	1	Neither possible for users nor admins.
F2	1	No CMS.
F3	3	Shopping cart feature.
F4	3	Basic system catalogue management and categories.
F5	1	No feature.
F6	3	Basic product management, no definition of own product attributes.
F7	1	No feature.
F8	4	Good search system.
F9	3	Basic reporting of product views and customer activitiy.
Q0	3	Mix of design, logic and data access, otherwise clean and intuitive code with good documentation.
Q1	5	Very reliable with no crashes or display issues.
Q2	5	Secure setup offering SSL.
Q3	3	Well implemented but outdated.
U0	3	Basic error reporting for user and admins.
U1	5	Simple and intuitive.
V0	5	Widely deployed professionally.
C0	3	Simpler software architecture but less features implemented.

Table A.2: Analysis of evaluation criteria for OSCommerce.

[1] Very poor corresponds to 1, very good corresponds to 5.

Zen Cart		
Code	**Score[1]**	**Comment**
F0	2	Only one administrator possible, very poor user management.
F1	1	No access rights and permission system.
F2	1	No CMS.
F3	3	Shopping cart feature.
F4	3	Basic system catalogue management and categories.
F5	1	No feature.
F6	3	Basic product management, no definition of own product attributes
F7	1	No feature.
F8	4	Good search system.
F9	3	Basic reporting of product views and customer activitiy.
Q0	3	Mix of design, logic and data access, otherwise clean and intuitive code with good documentation.
Q1	5	Very reliable with no crashes or display issues.
Q2	5	Secure setup offering SSL.
Q3	3	Well implemented but outdated.
U0	3	Basic error reporting for user and admins.
U1	5	Simple and intuitive.
V0	4	Branched from OSCommerce, not as used.
C0	3	Simpler software architecture but less features implemented.

Table A.3: Analysis of evaluation criteria for Zen Cart .

[1] Very poor corresponds to 1, very good corresponds to 5.

PHPB2B		
Code	**Score**[1]	**Comment**
F0	5	Advanced user management and multiple admins possible.
F1	5	Advanced permission management for users and admins.
F2	3	Basic CMS.
F3	4	Shopping cart and favorites feature.
F4	5	Advanced catalogue management (categorisation by industry and region etc.).
F5	1	No feature.
F6	3	Basic product management, no definition of own product attributes
F7	1	No feature.
F8	4	Good search system.
F9	3	Basic reporting of product views and customer activitiy.
Q0	1	Complex and counter-intuitive structure of source code, “includes” vastly distributed with hard-coded and difficult to maintainable links, partially counter-intuitive naming, “chaotic” code, no documentation at all.
Q1	1	Regular crashes, layout problems, display errors, very buggy.
Q2	2	Very buggy, thus questionable security and no SSL.
Q3	1	Very poor implementation without proper software architecture.
U0	2	Buggy and unreliable error reporting.
U1	2	Often counterintuitive structures and grouping of categories and layout.
V0	3	Websites states several references.
C0	2	Counterintuitive and complex structure and poor code quality, but partially good functionality.

Table A.4: Analysis of evaluation criteria for PHPB2B.

[1] Very poor corresponds to 1, very good corresponds to 5.

A.2 Analytical Hierarchy Process

	F0	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**Q0**	**Q1**	**Q2**	**Q3**	**U0**	**U1**	**V0**	**C0**	EV^1	PV^2
F0	1,0	1,0	1,0	0,5	0,3	2,0	0,3	2,0	1,0	2,0	0,2	1,0	2,0	0,5	3,0	1,0	2,0	3,0	0,17	5%
F1	1,0	1,0	3,0	2,0	1,0	3,0	0,3	3,0	1,0	4,0	0,3	3,0	5,0	1,0	4,0	3,0	5,0	1,0	0,23	7%
F2	1,0	0,3	1,0	1,0	0,3	1,0	0,3	1,0	1,0	2,0	0,3	1,0	2,0	0,3	1,0	1,0	2,0	0,3	0,10	3%
F3	2,0	0,5	1,0	1,0	0,2	1,0	0,2	1,0	1,0	2,0	0,2	1,0	3,0	0,3	1,0	1,0	2,0	0,3	0,10	3%
F4	3,0	1,0	3,0	5,0	1,0	3,0	1,0	3,0	3,0	5,0	1,0	5,0	7,0	1,0	5,0	5,0	9,0	1,0	0,35	10%
F5	0,5	0,3	1,0	1,0	0,3	1,0	0,1	0,1	1,0	1,0	0,1	0,3	1,0	0,2	2,0	0,3	1,0	0,1	0,06	2%
F6	3,0	3,0	3,0	5,0	1,0	7,0	1,0	8,0	8,0	8,0	0,5	5,0	7,0	0,5	6,0	2,0	6,0	1,0	0,40	12%
F7	0,5	0,3	1,0	1,0	0,3	7,0	0,1	1,0	1,0	1,0	0,1	1,0	2,0	0,1	2,0	0,3	3,0	0,1	0,10	3%
F8	1,0	1,0	1,0	1,0	0,3	1,0	0,1	1,0	1,0	1,0	0,2	1,0	2,0	0,2	5,0	2,0	5,0	0,3	0,12	4%
F9	0,5	0,3	0,5	0,5	0,2	1,0	0,1	1,0	1,0	1,0	0,1	0,5	1,0	0,1	1,0	0,5	1,0	0,1	0,06	2%
Q0	5,0	4,0	3,0	5,0	1,0	9,0	2,0	9,0	6,0	9,0	1,0	5,0	9,0	3,0	7,0	5,0	9,0	1,0	0,54	16%
Q1	1,0	0,3	1,0	1,0	0,2	3,0	0,2	1,0	1,0	2,0	0,2	1,0	1,0	0,3	1,0	2,0	3,0	0,3	0,10	3%
Q2	0,5	0,2	0,5	0,3	0,1	1,0	0,1	0,5	0,5	1,0	0,1	1,0	1,0	0,1	3,0	0,3	1,0	0,1	0,05	2%
Q3	2,0	1,0	3,0	4,0	1,0	5,0	2,0	7,0	5,0	7,0	0,3	4,0	9,0	1,0	5,0	2,0	5,0	0,5	0,34	10%
U0	0,3	0,3	1,0	1,0	0,2	0,5	0,2	0,5	0,2	1,0	0,1	1,0	0,3	0,2	1,0	0,3	1,0	0,1	0,05	2%
U1	1,0	0,3	1,0	1,0	0,2	3,0	0,5	3,0	0,5	2,0	0,2	0,5	4,0	0,5	3,0	1,0	5,0	0,3	0,13	4%
V0	0,5	0,2	0,5	0,5	0,1	1,0	0,2	0,3	0,2	1,0	0,1	0,3	1,0	0,2	1,0	0,2	1,0	0,1	0,05	1%
C0	0,3	1,0	4,0	3,0	1,0	8,0	1,0	7,0	3,0	8,0	1,0	4,0	9,0	2,0	7,0	3,0	9,0	1,0	0,37	11%
																λ_{max}	19,8	$\sum$	3,33	100%

Table A.5: AHP pairwise criteria comparison matrix.

[1] Priority vector (PV) as normalised EV.

[2] Eigenvector (EV) of maximum eigenvalue λ_{max}.

F0	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	4,00	6,00	0,33	0,43	28%
OSCommerce	0,25	1	3,00	0,17	0,15	10%
Zen Cart	0,17	0,33	1	0,13	0,07	5%
PHPB2B	3,00	6,00	8,00	1	0,89	58%
$\lambda_{max}/\sum$				λ_{max}	4,14	100%
F1	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	3,00	5,00	0,14	0,24	18%
OSCommerce	0,33	1	1,00	0,13	0,09	6%
Zen Cart	0,20	1,00	1	0,11	0,08	6%
PHPB2B	7,00	8,00	9,00	1	0,96	70%
$\lambda_{max}/\sum$				λ_{max}	4,20	100%
F2	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	8,00	8,00	5,00	0,95	67%
OSCommerce	0,13	1	1,00	0,33	0,10	7%
Zen Cart	0,13	1,00	1	0,33	0,10	7%
PHPB2B	0,20	3,00	3,00	1	0,26	18%
$\lambda_{max}/\sum$				λ_{max}	4,05	100%
F3	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	2,00	2,00	1,00	0,63	33%
OSCommerce	0,50	1	1,00	0,50	0,32	17%
Zen Cart	0,50	1,00	1	0,50	0,32	17%
PHPB2B	1,00	2,00	2,00	1	0,63	33%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
F4	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	3,00	3,00	0,33	0,43	26%
OSCommerce	0,33	1	1,00	0,25	0,17	11%
Zen Cart	0,33	1,00	1	0,25	0,17	11%
PHPB2B	3,00	4,00	4,00	1	0,87	53%
$\lambda_{max}/\sum$				λ_{max}	4,08	100%
F5	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	9,00	9,00	9,00	0,98	75%
OSCommerce	0,11	1	1,00	1,00	0,11	8%
Zen Cart	0,11	1,00	1	1,00	0,11	8%
PHPB2B	0,11	1,00	1,00	1	0,11	8%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
F6	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	6,00	6,00	6,00	0,96	66%
OSCommerce	0,17	1	1,00	0,50	0,13	9%
Zen Cart	0,17	1,00	1	0,50	0,13	9%
PHPB2B	0,17	2,00	2,00	1	0,22	16%
$\lambda_{max}/\sum$				λ_{max}	4,06	100%
F7	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	9,00	1,00	1,00	0,80	44%
OSCommerce	0,11	1	1,00	1,00	0,24	13%
Zen Cart	1,00	1,00	1	1,00	0,39	21%
PHPB2B	1,00	1,00	1,00	1	0,39	21%
$\lambda_{max}/\sum$				λ_{max}	4,66	100%

Table A.6: AHP matrix, maximum eigenvector and priority vector resulting from pairwise software package comparison for each criterion. (1/3)

F8	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	1	1	1	0,500	25%
OSCommerce	1	1	1	1	0,500	25%
Zen Cart	1	1	1	1	0,500	25%
PHPB2B	1	1	1	1	0,500	25%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
F9	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	3,00	3,00	3,00	0,86	49%
OSCommerce	0,33	1	1,00	2,00	0,34	19%
Zen Cart	0,33	1,00	1	2,00	0,34	19%
PHPB2B	0,33	0,50	0,50	1	0,20	12%
$\lambda_{max}/\sum$				λ_{max}	4,06	100%
Q0	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	3,00	3,00	8,00	0,89	55%
OSCommerce	0,33	1	1,00	3,00	0,31	19%
Zen Cart	0,33	1,00	1	3,00	0,31	19%
PHPB2B	0,13	0,33	0,33	1	0,11	7%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
Q1	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	1,00	1,00	9,00	0,58	32%
OSCommerce	1,00	1	1,00	9,00	0,58	32%
Zen Cart	1,00	1,00	1	9,00	0,58	32%
PHPB2B	0,11	0,11	0,11	1	0,06	4%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
Q2	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	1	1	4	0,571	31%
OSCommerce	1	1	1	4	0,571	31%
Zen Cart	1	1	1	4	0,571	31%
PHPB2B	0,25	0,25	0,25	1	0,143	8%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%
Q3	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	7,00	7,00	9,00	0,97	70%
OSCommerce	0,14	1	1,00	4,00	0,18	13%
Zen Cart	0,14	1,00	1	4,00	0,18	13%
PHPB2B	0,11	0,25	0,25	1	0,06	4%
$\lambda_{max}/\sum$				λ_{max}	4,17	100%
U0	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	3,00	3,00	6,00	0,86	52%
OSCommerce	0,33	1	1,00	5,00	0,35	21%
Zen Cart	0,33	1,00	1	5,00	0,35	21%
PHPB2B	0,17	0,20	0,20	1	0,09	6%
$\lambda_{max}/\sum$				λ_{max}	4,11	100%
U1	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	1,00	1,00	6,00	0,57	32%
OSCommerce	1,00	1	1,00	6,00	0,57	32%
Zen Cart	1,00	1,00	1	6,00	0,57	32%
PHPB2B	0,17	0,17	0,17	1	0,10	5%
$\lambda_{max}/\sum$				λ_{max}	4,00	100%

Table A.7: AHP matrix, maximum eigenvector and priority vector resulting from pairwise software package comparison for each criterion. (2/3)

V1	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	4,00	5,00	9,00	0,90	59%
OSCommerce	0,25	1	4,00	7,00	0,40	26%
Zen Cart	0,20	0,25	1	4,00	0,16	11%
PHPB2B	0,11	0,14	0,25	1	0,06	4%
$\lambda_{max}/\sum$				λ_{max}	4,25	100%
C0	*Magento*	*OSCommerce*	*Zen Cart*	*PHPB2B*	*Eigenvector*	*Priority*
Magento	1	2,00	2,00	3,00	0,78	42%
OSCommerce	0,50	1	1,00	2,00	0,42	23%
Zen Cart	0,50	1,00	1	2,00	0,42	23%
PHPB2B	0,33	0,50	0,50	1	0,22	12%
$\lambda_{max}/\sum$				λ_{max}	4,01	100%

Table A.8: AHP matrix, maximum eigenvector and priority vector resulting from pairwise software package comparison for each criterion. (3/3)

Appendix B

Usability Questionnaire

System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Figure B.1: System Usability Scale (SUS) questionnaire developed by John Brooke in 1986 for DEC Ltd. (Source: Brooke [10]).