Raphael Sommer, BSc

Motivation in an Agile, Educational, Free and Open Source Software Project

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Development and Business Management

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang Slany

Co-Supervisor

Dipl.-Ing. Annemarie Harzl, BSc

Institute of Software Technology

Graz, September 2016

AFFIDAVIT

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. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Graz, _____ _____

Date Signature

Abstract

Motivation is a key success factor in software projects. Performance, satisfaction and commitment to organizations are strongly linked with individuals' motivation. Motivation has a strong impact on the success of entire organizations. Catrobat is a Free and Open Source Software project using agile software development methods in an educational environment. Catrobat is influenced by several contexts like Free and Open Source Software, agile software development methods and education. In all this contexts motivation plays a crucial role. This thesis offers an insight into Catrobat members' attitudes and offers an overall picture of motivation in the Catrobat project. A survey was used to gather data from Catrobat members. The results of this survey are used for interpretations and suggestions. The outcomes of this thesis show many positive topics in Catrobat and also topics which offer potential for improvement.

Contents

List of Figures

List of Tables

1. Introduction

Nowadays, a huge amount of software projects are Free and Open Source (GitHub Team, 2016; SourceForge Team, 2016; Atlassian Team, 2016). Platforms like GitHub, SourceForge or Bitbucket are supporting such projects by providing a comfortable way for source code management. Impressively, GitHub is the largest Open Source community in the world (GitHub Team, 2016). "*GitHub is how people build software. With a community of more than 14 million people, developers can discover, use, and contribute to over 35 million projects using a powerful collaborative development workflow.*" (GitHub Team, 2016). This seems to be a huge amount, but it is still a challenge to keep these projects active and alive. Communities around these projects have their own dynamics and principles. Keeping members in such communities motivated and acquiring new community members are important and challenging tasks (Steinmacher, Wiese, et al., 2015).

Participants of Open Source projects can benefit from their participation in various ways. For example getting in contact with new tools, technologies and improving their skills are possible learning opportunities. Such participations can be valuable experiences for participants' future professional life. Therefore such projects offer special opportunities for the educational sector. Programs like Google Summer of Code or Facebook Open Academy support students to get in contact with Open Source projects (Steinmacher, Wiese, et al., 2015). Even direct cooperations between educational organizations and Open Source projects are arranged (Fagerholm et al., 2013; H. Ellis et al., 2011). Both parties can benefit from such cooperations. Educational organizations find great opportunities for learning and Open Source communities get in contact with interested people. Open Source communities need newcomers to ensure active and healthy communities in the long term.

In organizations, high retention rates are positive. Organizations have to deal with motivation and job satisfaction of their members to prevent them from leaving. One way to increase motivation and job satisfaction can be agile approaches. Agile software development methods have a stronger focus on people, than non agile approaches. A study done by Melnik and Maurer, 2006 reports higher job satisfaction rates in agile teams, than in non agile ones. Even the Agile Manifesto has a value defined for the importance of people: "*Individuals and interactions over processes and tools*" (Beck et al., 2001). Agile methods focus on individuals, their talents, their skills and interactions between them (Cockburn and Highsmith, 2001).

Technical skills are important and necessary for successful projects, but social skills are also crucial. The research field called *human aspects* is focusing on the people factor in working environments. Motivation is a human aspect and has tremendous impact on the success of projects, retention rates and performances. Motivation is influenced by motivational factors and demotivational factors. Depending on their presence, people can be motivated or demotivated. Demotivated people in organizations show less enthusiasm and the working atmosphere around them is rather poor (Frangos, 1998). Motivation strategies can be used to avoid and counteract demotivation in organizations (França and Fabio Q. B. da Silva, 2010).

1.1. Motivation

Catrobat is an agile, Free and Open Source Software (FOSS) project in an educational environment. These three contexts are influencing the Catrobat project. Investigating motivation in this hybrid project is an interesting task. There are many challenges leading such a huge project and keeping participants motivated. Several factors can have a demotivational impact on Catrobat members. For example coordination, management, teams or objectives are critical topics. Problems and difficulties arise with different stages of organizations (Shamir and Howell, 1999; Probst and Raisch, 2005). For example the leadership crisis is one possible crisis in the organizational life cycle. One person alone is not able to lead a certain amount of people. As a result, organizations have to establish structures for leadership and

coordination. Catrobat had a very dynamic evolution in the last years and is facing similar issues. Many people have joined Catrobat. In May 2016, there were around 150 people working on Catrobat. Also several sub-projects have been founded to implement further ideas.

This thesis is focusing on motivation in the Catrobat project. There are several objectives of this study. The primary objective is to capture the current motivational state in the project. With this knowledge is is possible to identify potentials for improvement. It is also possible to compare motivation in Catrobat with other Software Engineering (SE) settings.

1.2. Thesis Overview

This thesis is focusing on human motivation in the context of a hybrid SE project. This project is the Catrobat project which has its headquarter at Graz University of Technology. After this introduction a related work chapter is provided to give an overview of related scientific work, see Chapter 2.

Catrobat is a FOSS project. In Chapter 3 basic information about FOSS communities and corresponding issues like the onboarding process for newcomers are analyzed. In Chapter 4 an overview of Extreme Programming (XP) and Kanban is given which are relevant software development methods in the Catrobat project. Afterterwards retrospectives are described which are a possibility to establish continuous improvement.

In Chapter 5 the Catrobat project is described in general. Aspects like the evolution of Catrobat and its community structure are analyzed. Catrobat is influenced by several contexts, especially by *education*, *FOSS* and *agile*. How Catrobat is linked with this contexts is also described in this chapter. The Catrobat project represents a special setting of human motivation in SE. This setting is investigated in detail in this thesis.

Theoretical basics concerning human motivation are described in Chapter 6. This information is provided to give an understanding of human motivation. In this chapter a focus is also on motivation in SE. Topics like software engineers' characteristics, basic motivational factors and motivation in various SE settings is analyzed.

1. Introduction

Basics from previous chapters are important to understand the Catrobat project and how it is influenced from several contexts. A detailed problem description for this thesis is given in Chapter 7. In this chapter objectives of this thesis and the set of motivational factors is described which is used to gather data from Catrobat members. A survey is done within this thesis to collect data about factors in Catrobat which can have an impact on motivation. Chapter 8 describes the survey research methodology used to collect data within this thesis. There are also topics like sample, structure and objectives of the survey described.

Results of the survey are presented in Chapter 9. In this chapter general factors in Catrobat which can have an impact on motivation are presented. Survey results which directly addressed motivation in the Catrobat project are presented as well. Afterwards results of the survey are used for further investigations and interpretations. In this case the Herzberg Motivational Theory is applied to the Catrobat project.

At the end of this thesis, a conclusion with findings is provided, see Chapter 10. There are several possibilities for future work of this thesis. They are also described in this chapter. In the appendix of this thesis, results from literature reviews about motivation in SE are provided. A guide for retrospectives and the performed survey in German are also added to the appendix.

2. Related Work

In the last decades the field of human aspects in SE has gained notable attention (Lenberg, Feldt, and Wallgren, 2015; Hazzan and Dubinsky, 2010; Yanyan and Renzuo, 2008). Organizations are putting more focus in their daily business on human aspects (Lenberg, Feldt, and Wallgren, 2015; Hazzan and Dubinsky, 2010; Yanyan and Renzuo, 2008). A lot of related work is available concerning motivation in SE. The following related works offer a small snapshot of work done in the corresponding fields.

Human Factors Related Challenges in Software Engineering – An Industrial Perspective from Lenberg, Feldt, and Wallgren, 2015: This paper from 2015 analyzed human aspects in SE from an industrial perspective. Human factors have a remarkable impact on organizations' success. Agile software development methods are taking an important role to encourage the focus on individuals and groups beside technical aspects. For some people it is not easy to deal with agile methods, because there are still challenges. For example the necessity to interact increasingly with other people is one challenge. As a result social skills are crucial for working in agile environments. In this scientific work a qualitative research method was selected consisting of semi-structured interviews. Interviews were done with nine software engineers from Sweden working in agile environments. Several success factors were identified by these participants. For example *good relationship with customers* and *socialising communication* are highlighted by Lenberg, Feldt, and Wallgren, 2015 as important human factors in agile environments.

Does the XP environment meet the motivational needs of the software developer? An empirical study from S. Beecham et al., 2007: This study examined how XP covers motivational needs of software engineers. These researchers compared two different data sets. One data set consisted of data collected in

teams using non agile software development methods. This data set was gathered by semi-structured interviews. In this case nine software engineers were interviewed. The other data set was built with data collected by observing five XP teams. In some areas results of both data sets were similar. For example *progress* was a factor, which was identified as important for motivation in both environments. In non agile software development environments some critical factors were identified, which were not classified as critical in XP environments. For example *access to information* or *communication* are not critical in agile environments. Work is done differently by these two software development approaches. In XP teams work is done more collaboratively than in non agile teams. As a result, in agile teams it is not always easy to identify who has done which task and which work. This can lead to demotivated individuals, because they are not recognized for their work. This study shows also some disadvantages in XP environments. XP addresses not all needs of software engineers and can have also a demotivating effect on software engineers' attitude. Several positive factors in agile environments are highlighted by S. Beecham et al., 2007. For example *responsibility* and *autonomy* are typically satisfied in agile working environments.

Motivation in Software Engineering: A systematic literature review from Sarah Beecham et al., 2008: A systematic literature review in SE was published in 2008. The literature review had the objective to show the actual state in research concerning motivation in SE. For this study researchers collected huge amounts of papers, journals and other available material from 1980 to 2006. Researchers reviewed data from more than 2000 documents and selected 92 of them. They took these 92 documents and used them for the literature review. 86% were empirical based, 11% were from theoretical studies and 3% were from literature reviews. Most of this data was from Europe and the USA. One research question investigated typical characteristics of software engineers. Researchers filtered these documents after 16 personal characteristics. These 16 characteristics were identified by Sarah Beecham et al., 2008 as relevant. Characteristics mentioned in the highest number of scientific articles were *growth orientated*, *autonomous* and *introverted*. Another research question addressed motivational and demotivational factors in SE. Out of the 21 motivators identified in the literature review, the motivators *identify with the task*, *working with others* and *career path* where identified as the ones with the most occurrences in scientific articles. Demotivational

factors which had the highest number of scientific articles were *poor working environment*, *poor management* and *uncompetitive pay*. Results of this literature review are widely used and cited in papers concerning motivation in SE.

Motivation in software engineering: A systematic review update from A. Franca, Gouveia, et al., 2011: A few years later an update of the systematic literature review has been published in 2011. Researchers collected more than 6500 documents for this systematic review update. From this huge amount of data, only 53 relevant documents were chosen for data extraction and analysis. A. Franca, Gouveia, et al., 2011 used the same research questions as in the literature review from Sarah Beecham et al., 2008. Results were also similar. They found almost the same most relevant characteristics of software engineers: *autonomous*, *growth orientated* and *need for challenge*. Three new characteristics were appended: *competent in management*, *team worker* and *have fear of punishment*. Results of the research question concerning the most relevant motivational factors changed slightly to *working with others*, *career path* and *development needs addressed*. There were also eight new motivators appended. Thus the collection of motivators contained 29 factors. The ranking of demotivators changed to *stress*, *unfair reward system* and *task complexity*. One finding of this work was, that more awareness is spent on motivation in FOSS projects. The motivational factor *relationship with users* has been added to describe the relationship between developers and users of a FOSS community.

Toward an understanding of the motivation of open source software developers from Ye and Kishida, 2003: This research analyzed software engineers' motivation in FOSS projects. It highlights the importance of communities in FOSS projects. Communities are providing a platform for developers as well as for users. Authors of this paper underlined the importance of communities by an example FOSS project, the Gnu Image Manipulation Program (GIMP) project. This project develops software for image manipulation and processing on Linux machines. It was one of the first image processing tools for Linux. Founders of GIMP started building a community with a mailing list. The mailing list was used for communication in general and also for development purposes. In the GIMP community clear roles were used for handling participations. In 2003 there were four core members, 47 active developers and 111 peripheral developers. Peripheral developers' contributions had to be approved by core members or active developers.

It was always possible to move closer to the center of the community by committing source code. As a result, committing usable source code was the secret to earn status inside the community. Status was identified as one motivation behind participating in FOSS projects. Members with a higher status were rewarded with recognition and reputation. The resulting advice of these researchers was to focus not only on the software developed by communities, but to also put enough energy into creating and maintaining working FOSS communities.

How to involve students in FOSS projects from H. Ellis et al., 2011: The combination of FOSS projects and education seems to be beneficial. This paper is addressing this idea and offers a possible way to involve students in FOSS projects. Learning should be the main objective for students. FOSS projects are a great opportunity for students to gain experience in large software projects. In addition, companies like to see engagement in FOSS projects, thus it is an additional advantage for job applications as well. The researchers created a guide to handle student participations during a lecture. The guide consists of several steps. For example discovering possible FOSS projects and mentors is one step. In another step administrative tasks have to be carried out. Communication channels have to be clarified, the procedure for setting up the project environment has to be verified and suitable tasks for students have to be found. At the end of the lecture it is important to have the contributions at an acceptable level. Otherwise they would not get accepted by the community. This guidelines were proposed for building cooperations between educational environments and FOSS projects.

Increasing the Self-Efficacy of Newcomers to Open Source Software Projects from Steinmacher, Wiese, et al., 2015: This study is discussing the efficiency of newcomers in FOSS projects. This is a challenging problem in FOSS communities as well as in organizations. Typically, in active FOSS communities, interested volunteers join and begin to contribute. Without newcomers communities would not survive in the long term. Often newcomers drop out and make not a single contribution. Thus the onboarding process is important for the survival of communities. Therefore authors of this paper created a web portal to support newcomers in the first phase of their participation. The objective of this paper was to evaluate this web portal. The information hosted on this portal was for participations in FOSS projects like JabRef, LibreOffice or Vim. An action research was done with motivated

students. These students were split into two groups. One group was guided by the web portal with information, guides and tutorials. The other group was on their own during the experiment. At the beginning and at the end of participations, a questionnaire was filled out by all students. Students guided by the web portal showed higher self-efficacy, than students making their own way. However, some students had problems with the technical level of their tasks and the web portal could not help these students to get rid of such technical barriers. At the end, students and researchers assessed the web portal as a helpful and positive tool.

Several aspects of these works are used within this thesis. FOSS communities and the onboarding process is analyzed. Agile software development methods are analyzed and their influence on software engineers' motivation. The set of motivators identified by A. Franca, Gouveia, et al., 2011 is slightly adapted and used for the survey to gather data about motivation in the Catrobat project.

3. Free and Open Source Software

FOSS projects are very successful around the globe (Ye and Kishida, 2003; Wang, He, and J. Chen, 2005). Some very popular and successful FOSS projects include Apache, Linux, MySQL, Firefox, Tomcat, Moodle, OpenOffice and many more (Steinmacher, Chaves, et al., 2014).

The idea behind FOSS is changing the whole software industry enormously. With FOSS development models, economics changed for several stakeholders in the software market (Hars and Ou, 2002; Riehle, 2007). For example it is often not longer necessary to buy a license for using a piece of software. A lot of FOSS software is freely available on the market. This makes it harder for software manufacturers which are selling licenses for their software products.

This paragraph clarifies the use of FOSS within this thesis. There are several terms used for open source movements. Two large movements in this area are called *Free Software Foundation* and *Open Source Software* (Fitzgerald and Agerfalk, 2005; Raymond, 1999). To honor both movements and their ideas the hybrid term FOSS is used. FOSS is nowadays a widely used term for such projects. In the remaining part of this thesis, the term FOSS is used for Open Source Software and similar terms.

Catrobat is a FOSS project. In this chapter the focus is on FOSS communities. With a basic knowledge about FOSS, it is possible to become aware of pitfalls and hurdles in this type of software setting.

3.1. Roles

Usually, FOSS members are volunteers. If people come together and build a community, different roles become important. Interaction and influence depend on community members' roles. In this section roles in FOSS communities are described. These roles were identified by Ye and Kishida, 2003. Not all roles are present in every FOSS project. Members in a community can change their role with their activity in it.

Project leader: The project leader is often also the initiator of the project. The project leader starts with a vision. At the beginning he is often the only developer. If he finds interested people, he builds a community around his vision. The project leader gives the project direction, goals and visions.

Core member: Typically, core members have been in the project for a long time. They have also input in important decisions concerning the project. Core members are coordinating software development and corresponding processes. They are also guiding and supporting other members with their experience. If they have time, they are also developing software.

Active developer: This group is contributing in form of new features and bug fixes to the project. Often active developers are responsible for the most contributions to the codebase. Thus they are playing a very important role concerning progress in the whole community system.

Peripheral developer: Peripheral developers do the same tasks as active developers, but they are contributing more irregularly. The periods, in which they are active, are rather short. Sometimes they make only a single contribution to the project.

Bug fixer: Bug Fixers are spending most time on fixing discovered bugs. For bug fixing they need to understand only small parts of the source code.

Bug reporter: Bug reporters are discovering bugs. They do not fix found bugs, they only report them. Bug reporters can be people, who have never seen the source code at all.

Passive user: Passive users are using the software developed by the FOSS community. Usually, having numerous passive users is motivating for com-

munity members, because their software is used and honored by many people. Thus passive users are important for FOSS communities.

These roles can be more or less present in FOSS communities. Maybe namings of roles can differ, but activities behind these roles are often very similar. All members in FOSS communities have roles and together they are building a FOSS community. The structure of such communities is described in the following section.

3.2. Community Structure

After initiating and starting a FOSS project, often a community is built. Communities are an important success factor for FOSS projects (Ye and Kishida, 2003; Zhou and Mockus, 2015). With a working community, a platform for several stakeholders is provided. Such platforms offer possibilities for interaction between users, developers or other organizations.

Usually, hierarchies in FOSS communities are rather flat. However, influence plays an important role in FOSS communities. Figure 3.1 illustrates a typical community with roles already discussed in the previous section. Roles more to the center have more influence on the project. The project leader has the most influence on the project. In contrast, passive users have the least influence on the project. With significant contributions or an active participation, it is possible to move closer to the center of the community. Moreover, influence and status is typically increasing in the community. As a result role changes are possible as well. The community structure of traditional FOSS projects is illustrated in Figure 3.1 and can be compared with the community structure of the Catrobat project.

3.3. Onboarding

Acquiring newcomers is an important topic in FOSS projects, as well as in Catrobat. Many hurdles can occur which are also relevant for Catrobat. In this section a wide range of such hurdles is listed.

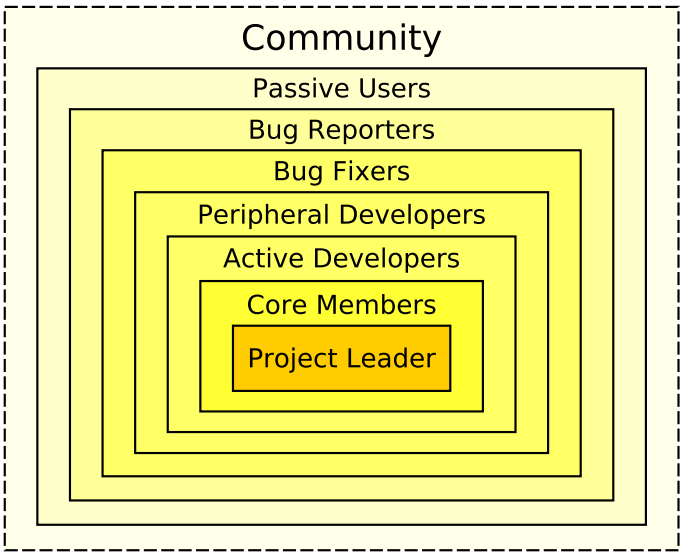

Figure 3.1.: Structure and roles in FOSS communities (Ye and Kishida, 2003)

Newcomers are essential for the long term survival of FOSS communities. For active communities manpower is needed. Passive users are potential developers. They are using the software provided by the community. Passive users have already made their first step into the community. With some motivation and initiative, they can start making contributions to the community. It is a crucial task for communities to ensure a motivating onboarding process. This process is identified as a critical challenge in FOSS communities (Ye and Kishida, 2003; Steinmacher, Wiese, et al., 2015; Steinmacher, Chaves, et al., 2014; Fagerholm et al., 2013). If onboarding is done in a poor way, it often leads to dropouts. If newcomers are left alone, they often drop out at an early stage of their participation.

Several challenges and hurdles can occur during the onboarding process of newcomers. There are various barriers, which can be categorized. Six categories of barriers are identified by Steinmacher, Chaves, et al., 2014. Examples for every category are shown in Figure 3.2. In the following paragraphs these six categories are described.

Newcomers' orientation: Newcomers are often faced with unfamiliar situations in the joining process. They need orientation to master such situations. One milestone for getting orientation is to make the first contribution to the community.

Newcomers' characteristics: Depending on characteristics and experiences of newcomers, the joining process can be easier or harder for them. How newcomers show their knowledge and how they integrate themselves into the community, is strongly depending on their personality.

Reception issues: This category represents barriers focusing on communication and interactions between newcomers and communities. A poor communication results typically in demotivation.

Cultural differences: Different cultural backgrounds can lead to misunderstandings and issues related to interaction and communication.

Documentation problems: This category comprises several pitfalls and barriers related to documentation. Topics for documentation are technical as well as social.

Technical hurdles: This category represents technical barriers and hurdles which can occur while making first steps. These first steps can be setting up the working environment or making a first contribution.

There are lots of possible hurdles in all these categories. FOSS communities can inspect their handling of newcomers and and set up countermeasures against common barriers. For this purpose barriers identified in Figure 3.2 can be used as a checklist.

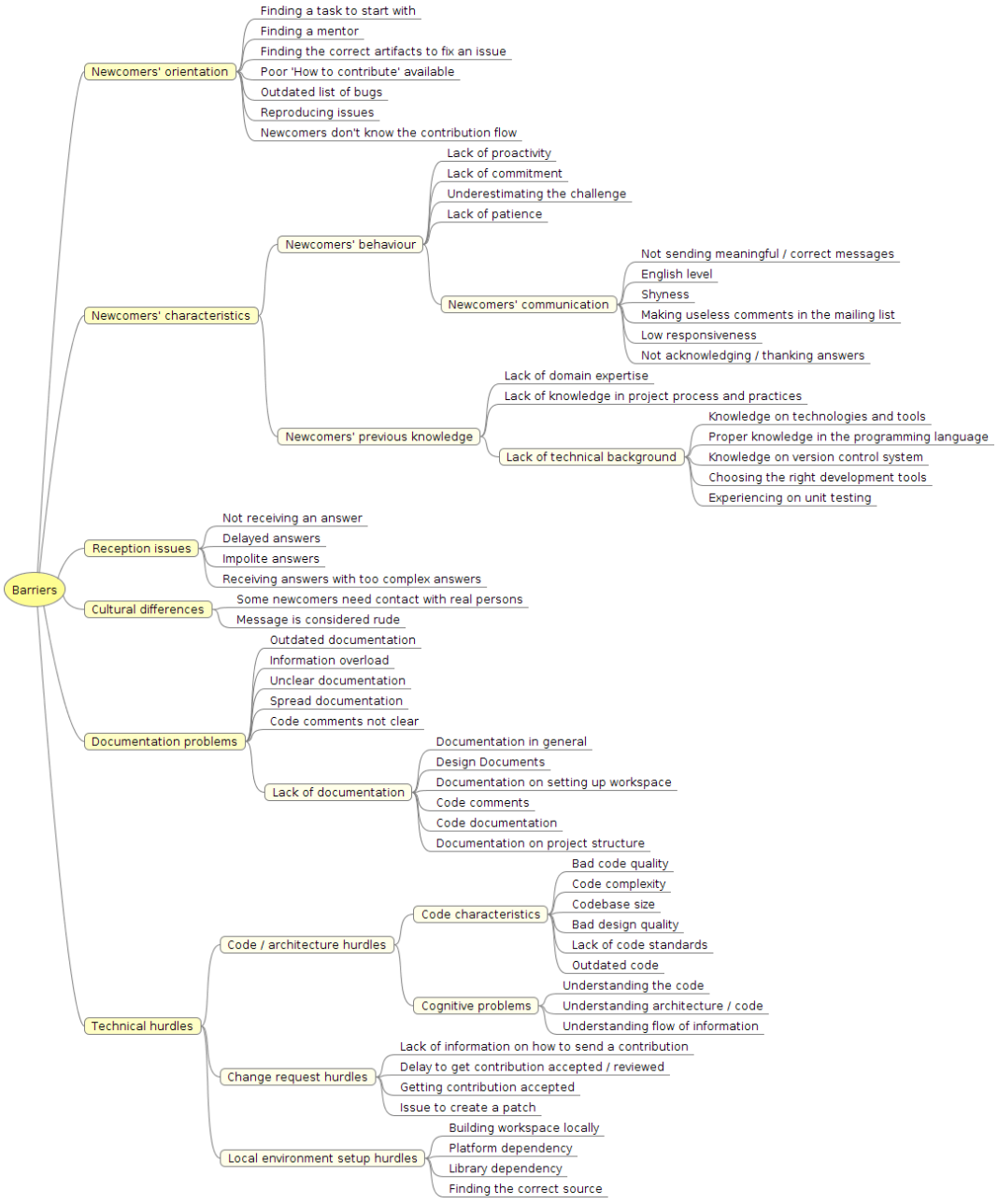

Figure 3.2.: Barriers for newcomers in FOSS projects (Steinmacher, Chaves, et al., 2014)

3.4. FOSS and Education

In the Catrobat project, FOSS and education are playing important roles. Both contexts can be present at the same time. Learning is an important factor in FOSS as well as in educational environments.

Communities need newcomers for their long term survival. Especially, computer science students are potential community members. Communities can benefit from integrating students into their projects. Students can benefit from their participation in FOSS projects as well. Students can improve social and technical skills. There are several programs which even support this idea with financial benefits (Steinmacher, Wiese, et al., 2015). For example Google Summer of Code or Facebook Open Academy are programs, which support student participations.

Another approach is to make cooperations between educational institutions and FOSS projects (H. Ellis et al., 2011). With such cooperations another way of integrating students into FOSS projects is possible. Integrating students into large real world software projects offers great learning opportunities for all involved stakeholders. In Table 3.1 possible benefits for such corporations are listed (H. J. C. Ellis et al., 2010; Gokhale, Smith, and McCartney, 2012; H. Ellis et al., 2011).

FOSS community	**Students**
Manpower, resources	Strengthen coding skills
Innovation, creativity of newcomers	Possibility to perform own ideas
Reflect onboarding process	Satisfy personal needs e.g. recognition
External feedback and opinions	Usage of professional tools
	Professional development workflows
	Earn money (Google Summer of Code)
	Meet new people

Table 3.1.: Benefits from cooperations between FOSS and education

4. Software Development

With the rise of agile software develop methods, a lot has changed in today's software development and corresponding processes (Bishop and Deokar, 2014; Abdullah and Abdelsatir, 2013; Cockburn and Highsmith, 2001). In most organizations the classical waterfall model is no longer relevant for software development. Agile methods became accepted in the software industry. Some examples for such agile software development methods are:

- Scrum
- Extreme Programming
- Kanban
- Scrumban
- Lean Software Development
- Feature Driven Development
- Crystal

Catrobat is using several practices in its software development process. This software development process is strongly influenced by agile software development methods. A poor realization of software development practices can lead to demotivated Catrobat members. In this chapter software development methods and corresponding practices are explained. Some of them are already used by sub-teams in Catrobat. Some of them can be introduced or adapted for the software development process in Catrobat. In the following sections XP and Kanban are described in more detail. Then retrospectives are explained as practice for improvement.

4.1. Extreme Programming

XP is the primary software development method in Catrobat. Thus its practices and values are described in this section. In the survey, done within this thesis, some of them are used to analyze the realization of XP in the Catrobat project.

XP is a software development method, which addresses many problems recognized in SE. It offers several practices to counteract problems concerning communication, economics and quality to achieve project success (Beck, 2004). Especially, quality is in the focus of XP.

Four variables are playing an important role in projects. These variables are *cost*, *time*, *quality* and *scope* (Beck, 2004; Chow and Cao, 2008). All four variables have dependencies among each other. For example if the scope is too large for a fixed deadline, quality can be affected tremendously in a negative way. In this case a solution would be to acquire further resources, but adding resources to an already late project will delay it further in the short term. If costs are fixed by contract it may be not possible to acquire further resources. In this case quality would be reduced to develop the software within the deadline. XP is very flexible and offers practices to avoid such problematic scenarios.

In XP it is important to pay attention to human aspects and business aspects as well. Four values are identified for working together in a social and effective way. *Communication* represents one value. In software projects a lot of communication is done in the daily business by software engineers, customers and management.

Another value in XP is *simplicity*. Especially for software engineers it is a fundamental principle. This is applicable for software architectures and source code.

The third value is *feedback*. In XP it is good practice to test written source code. Tests are delivering feedback in form of passing and failing tests. For software engineers passing tests are resulting in trust in their written software. Another form of tests is done by customers.

The fourth value is *courage*. *Courage* is necessary to make decisions. For example for an software engineer it needs courage to throw away a piece of code. The purpose for throwing code away can be poor quality or because it is no longer needed. These four values are building a solid basis in XP. In the following section XP roles and practices are described.

Roles and Practices

On-site customer, coach and manager: The on-site customer is playing an essential role in XP. For developers it is a great benefit to have a customer on-site (Beck, 2004). If there is anything to clarify, it is very effective to communicate directly with a customer. If the on-site customer is actually not needed, he can do his usual business or he can test the software. Another role in XP teams is the coach. Coaches help teams with communication and getting their work done. They are supporting engineers if they need any help. For example guiding newcomers on their way to get independent. Coaches are in direct contact with the management. The task of managers is to collect relevant data on the project. If the pace of a team is too slow or a timeline is in danger, then managers can react for example with more resources or crises meetings. Managers are also moderating planning games and are responsible for project outcomes.

Planning game: The planning game addresses different stakeholders. People from the business side are involved in this practice (Beck, 2004). The business side has to declare scope, priorities, dependencies and release deadlines. The business side has to interact with people from the technical side. Software developers are also involved into this practice. They do activities like cost estimations. One objective of this practice is to establish a dialog between these two groups. The result should be an agreement between them regarding the planning of upcoming tasks. After a successful planning game, both groups know what is the objective of the next iteration.

Small releases: This a common practice in most agile methods (Beck, 2004; Epping, 2011). Even at the beginning of projects it should be an intended goal to release after short periods. Of course a minimal running system has to be available, otherwise it does not make sense to present something to the customer. With short releases, testing by customers is possible at

an early stage. Thus it is possible to discover issues and bugs very early. Corresponding, the planning should be done only for the next one or two months and not too far into the future. Planning and releasing in small intervals should be preferred.

Metaphor: A metaphor is a simple description of the system at a high level of abstraction. It offers a simple, even maybe a naive view on the system. Metaphors offer the possibility for a better understanding of the problem and specific components. This can be helpful for all involved groups. Especially for the business side, the metaphor simplifies the problem and make it easier to deal with.

Simple design: It is a good practice to write code in a simple and easy way (Beck, 2004). In XP simple design is a measurement for high quality software. Software engineers should consider this practice in their daily work. An important aspect corresponding to simple design is to implement only functionality you need at the moment and not functionality which can matter later in future. It is better to ignore such speculative needs and implement them later if they are really needed, but actually the implementation should be as simple as possible.

Testing: One of the most fundamental practices in XP is testing. Non tested code can be seen as not existing (Beck, 2004). It is necessary to test code in XP projects. With every test, software engineers' trust is increasing in their code. Customers can write tests as well. Customers can test at a higher level of abstraction. For example they can make integration tests or test software on test systems. The combination of customers' and software engineers' tests is supporting the reliability of systems. Even if there are changes made in existing functionality, it is easy to check if other functionality is still working. For developers it is a good practice to write tests first and then write the corresponding functionality. Software engineers implement functionality to make these first written tests passing. This approach of writing tests first is called Test Driven Development (TDD).

Refactoring: As already mentioned above simplicity is a fundamental value in XP. Refactoring is strengthening the simple design practice. Refactoring is a practice to review code and to simplify it (Abdullah and Abdelsatir, 2013). Examples of possibilities for refactoring are *choosing better namings* or

making a better structure. As a result of refactoring, the whole code base is simpler and cleaner.

Pair programming: In XP code should be written in pairs. A pair is working in front of only one computer with one keyboard and one mouse (Abdullah and Abdelsatir, 2013). These two people are taking different roles. One person is coding, while the other one is thinking in a more strategic manner. After a period, these two people switch roles. After finishing a task, a partner switch can be done.

Collective Ownership and Coding Standard: The whole team is responsible for the whole code base (Beck, 2004). Therefore everyone has the right to change code, even the code has been written by another person. Every member uses the same coding standard, which is defined by the team.

Continuous integration: Another practice is to integrate written code after a short period. Often there are integration systems provided to integrate new code (Abdullah and Abdelsatir, 2013). Typically, tests are started automatically after integrating new code. As a result immediate feedback is provided. It is easy to check if the whole system works well and all test cases are passing.

XP offers many practices to achieve project success and positive attitudes in projects. In the following section another software development method, called Kanban is described.

4.2. Kanban

Several practices of Kanban are important in the Catrobat project. Some of them, which are actually not performed in Catrobat, offer a potential to influence the software development process in Catrobat in a positive way. Some of them are suggested as an improvement in the results chapter of this thesis.

Kanban is a process management system, which is used in several industries (Epping, 2011). Toyota, the Japanese car manufacturer, introduced the system to minimize waste in their production. The goal of this system was to

signalize events and specific situations. With these signals it was possible to react on certain situations immediately.

Kanban in SE is based on Lean Software Development values. Lean software development values identified by Epping, 2011 are listed below.

- Eliminate waste
- Amplify learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity in
- See the whole

The focus in Kanban is on the Kanban board and the tasks on this board. Kanban offers several practices for an effective software development process. In the following section relevant Kanban practices are described.

Practices

Visualize the workflow: One objective of Kanban is to ensure a constant workflow (Leopold and Kaltenecker, 2013). With a constant workflow, it is possible to satisfy customers. Kanban tries to make processes visible and observable. As a result, issues can be detected more easily. For this visualization a Kanban board is used, see Figure 4.1. With such a visualization of ongoing work, it is possible to see progress and additional information. On the Kanban board, tasks can be moved to different sections. Kanban boards are the main information medium for several stakeholders in SE projects.

Limit work in progress: Another Kanban principle is to limit work in progress (Ahmad, Markkula, and Oivo, 2013; Leopold and Kaltenecker, 2013). Only completed tasks have value for customers. A resulting objective is to keep work in progress low and to deliver tasks to customers constantly. The higher the work in progress is, the higher the throughput time. With a limit for work in progress, the throughput time should be kept low. Tasks are done sequentially instead of in parallel. Parallel work forces task switching which is time consuming and can dramatically increase the throughput time

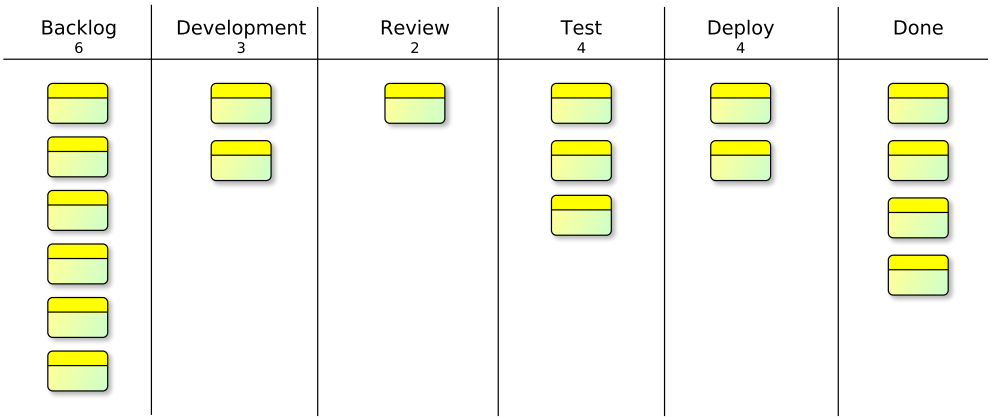

Figure 4.1.: Kanban board with several sections and limits (Epping, 2011)

(Ahmad, Markkula, and Oivo, 2013). In addition, it is more efficient to work on only one task at a time, because the focus is only on this task. As a result, the task is finished faster, than when one person is working on several tasks at the same time. Thus, the practice *limit work in progress* counteracts a long throughput time. A reliable throughput typically increases customers' trust. A trustful relationship between companies and customers are a valuable basis for collaboration.

Measure and manage flow: Kanban is focusing strongly on the workflow. Issues in the workflow are treated immediately (Leopold and Kaltenecker, 2013). Blockers can be examples for such issues. A constant workflow is desired. With a constant workflow, it is possible to plan timelines for upcoming tasks. With different sections onto the Kanban board, it is possible to measure and to monitor work in progress. It is also possible to observe in which sections possible bottlenecks are. In meetings like the daily standup meeting, the Kanban board can be used to discuss the progress. The purpose of daily standup meetings is to inform team members and to coordinate the actual workflow.

Make policies explicit: A set of rules is applied and should be made explicit

in teams using Kanban. These rules are defined by the team itself and are focusing on the workflow. For example these rules determine how and when tasks can be moved between sections on the Kanban board (Leopold and Kaltenecker, 2013). A structured and organized workflow is an outcome of such policies.

Improve collaboratively: Change plays an important role in Kanban. Existing processes are the initial point for change. Kanban tries to optimize processes in projects using Kanban (Ahmad, Markkula, and Oivo, 2013; Leopold and Kaltenecker, 2013). Individuals are only able to work as productive as the system permits. There is no desired final state of the system defined. There should be continuous improvement of processes for the actual context and the surrounding conditions. Retrospectives can support the process of improvement in teams using Kanban. Feedback on the current system can be given in retrospectives. In this practice one rule is to make decisions together.

Kanban is often combined with software development methods like XP or Scrum (Ahmad, Markkula, and Oivo, 2013). Kanban offers possibilities to detect potential issues at an early stage. Solutions for issues are developed by the team itself. The objective of Kanban is to ensure a constant workflow. Kanban gives the team strong power for decision making and improvement. Retrospectives can be used for continuous improvement. They are analyzed in the following section.

4.3. Retrospectives

Retrospectives offer a great possibility for continuous improvement. This practice can be applied to support the Kanban practice *improve collaboratively*. Retrospectives are offered to Catrobat members in the survey as possibility for improvement. Retrospectives and a few important aspects while performing retrospectives are described in this section.

In an educational context retrospectives offer great learning opportunities (Roach, 2011). Students doing retrospectives, can reflect their own perfor-

mance and behaviour. Afterwards they can develop possible improvement strategies to be more efficient and effective during their study.

Retrospectives are a best practice possibility for improvement in SE (Roach, 2011; Sertic, Marzic, and Kalafatic, 2007; Maham, 2008; Derby and Larsen, 2006). Retrospectives support teams to perform better and offer learning opportunities. In teams using agile software development methods, retrospectives are used to review team processes, technical processes and further possible topics around the team. Typical outcomes of retrospectives are concrete working agreements, which should be applied in the following iterations.

Five Steps for a Retrospective

The following steps are adapted from Derby and Larsen, 2006. These steps should be considered while preparing and performing retrospectives.

Setting the stage: The objective of this step is to inform people about the retrospective in general (Derby and Larsen, 2006). Information like timeboxing, objective or activities for the session are given. As a result retrospective participants start focusing on the current retrospective.

Gather data: The second step is to gather data from participants. At the beginning of this step it is typical to show facts about the last iteration (Derby and Larsen, 2006). For example completed tasks, defect rates or burndown charts. Then impressions, how participants have experienced the last iteration, are collected. Thus it is possible to get a feeling of how other people have experienced the last iteration. In this step, everyone should have the chance to give input. There can be various activities used for this step.

Generate insights: A deeper look on the gathered data is performed to generate insights. As a result positive and negative aspects of the last iteration can be identified (Derby and Larsen, 2006). Possible causes and effects of issues can be discussed in this step. A list of experiments or actions for improvement is also created in this step. For example brainstorming can be used for creating this list.

Set the stage	5%	6 minutes
Gather data	30-50%	40 minutes
Generate insights	20-30%	25 minutes
Decide what to do	15-20%	20 minutes
Close the retrospective	10%	12 minutes
Shuffle time	10-15%	17 minutes
Total	100%	120 minutes

Table 4.1.: Exemplary durations for a retrospective (Derby and Larsen, 2006)

Decide what to do: The team together can discuss the list from the previous step. Concrete plans for the implementation of these actions are also discussed and elaborated. Then the whole team chooses one or two actions for improvement. The commitment of most participants is necessary to introduce these actions in the next iteration. These actions serve to influence next iterations in a positive way. This results in working agreements for the next iterations. It is important to limit the number of working agreements to one or two for a single retrospective, because too many agreements could have a negative impact on the implementation of decided working agreements.

Closing the retrospective: In the last step of retrospectives, actions for the next iteration are repeated. The whole team tries to implement these actions for improvement in their daily business. Thanking all participants is a typical way to finish a retrospective (Derby and Larsen, 2006). The leader of the retrospective should capture gathered data and elaborated working agreements.

In Table 4.1 exemplary durations for all steps in a retrospective are shown. This is an example for a two hour retrospective adapted from Derby and Larsen, 2006. The shuffle time represents time for switching between the five steps. Derby and Larsen, 2006 prepared a lot of activities for all five steps. These activities can be used to plan and perform retrospectives. Changing the activity for generating insights is recommended, because this is the longest taking step in a retrospective. Activities can be switched with every retrospective. Two examples for a retrospective can be found in Appendix C.

Retrospective Leader

Anyone can lead a retrospective with some preparation. The retrospective leader should always prepare for a retrospective. Time for preparation will typically decrease with experience. The leader of a retrospective has several tasks before, during and after retrospectives. Some of them are described in the following paragraphs.

Before: For the preparation of retrospectives, the retrospective leader has to think about topics like *outcome of the iteration, environment and history of the team, goal of the retrospective, duration of the retrospective* and *activities using in the particular steps of the retrospective* (Derby and Larsen, 2006).

During: During performing the retrospective, the retrospective leader has to have an eye on *group dynamics, time management, managing activities* and *managing the overall process*. He is responsible for the effective and efficient performing of the retrospective (Derby and Larsen, 2006).

Afterwards: After the retrospective, the retrospective leader has to remind the team about their working agreements and improvements which they have elaborated. He is also responsible to capture the results of the retrospective. For example he can create a poster with results and place it in the working room of the team.

5. Catrobat

Catrobat is a FOSS project initiated by Wolfgang Slany at Graz University of Technology. Catrobat provides various services to offer young people a platform, where they can experiment with programming concepts. Programming with Catrobat is not like usual coding in form of using text editors and writing some keywords, it is much easier. Programming is done via visual bricks, a similar approach to LEGO bricks. It is possible to add bricks, build collections of them, play around and try out the project immediately. Maybe the most important use case for Catrobat is to offer young people a playful way to get in contact with basic programming concepts. With Catrobat, young people are able to build their own programs. The intuitive usability and the well designed bricks support a positive experience. Many programs are published on the web (Catrobat Developers Team, 2016d).

In this thesis, Catrobat is investigated in detail concerning motivation. A survey is used to gather data from Catrobat members. With this data investigations and interpretations about motivation and issues in the project can be made.

5.1. Evolution of Catrobat

The Catrobat project has experienced a very dynamic evolution. The number of people working on the project has increased enormously. This can lead to issues in organizations. Topics like leadership, communication or performance can become problematic. Thus it is important to get an understanding of the evolution of the whole Catrobat project. In this section, the evolution of Catrobat is described.

Wolfgang Slany started the project in February 2010 with a small group of students (Catrobat Developers Team, 2016a). In the following years 2011 and 2012 Catrobat was selected to be part of the Google Summer of Code program. In October 2012 Catrobat already consisted of several sub-projects: Live Wallpaper, Drone, Catroid, iOS, Web, Jenkins, Paintroid, Usability, Tutorial, Windows Phone and Arduino. In the following years Catrobat won prizes and awards. For example Catrobat won the Austrian National Innovation Award in March 2013. Catrobat was also promoted several times by Google. In addition Catrobat did several project cooperations and events with other organizations. For example a cooperation was built with Scratch, the University of Oxford and other partners to organize the Alice Game Jam 2015 (Alice Game Jam Partners, 2016).

At the moment, further sub-projects are under development. Some of them are Musicdroid, an HTML5/Javascript edition or Near Field Communication (NFC) for multiplayer coordination (Catrobat Developers Team, 2016a). In January 2016, a web platform has been launched to support teachers with the use of Catrobat in educational environments. The platform offers tutorials, documentation and exercises for the Android version. All these sub-projects are under the so called Catrobat umbrella project. In May 2016, more than 150 people have been part of the Catrobat community. Most Catrobat members are students at Graz University of Technology. They are doing their bachelor's thesis, master's thesis or master's projects within their participation in Catrobat.

The evolution of the Catrobat community has been very dynamic in the last years. With the growth of the Catrobat community, it has become quite a challenge to supervise and guide interested individuals. A further challenge is to keep Catrobat members motivated and productive during their whole participation.

5.2. Structure and Processes

Under the Catrobat umbrella project, several sub-teams are working on different ideas and technologies. Most of them are already listed the previous section. In May 2016, around 15 different sub-teams were working on

Catrobat. Some of them were sub-projects of sub-projects. There were sub-teams with only one member and sub-teams with more than ten members. For example the Windows Phone team had around 17 members. Six projects had more than nine members. These sub-projects are *Catroid, iOS, Jenkins, Usability, Windows Phone* and *Web*. Most sub-teams are lead by coordinators. Coordinators are doing organizational work and try to ensure a productive team. They are responsible for regular meetings in their team.

Several meetings are typical in Catrobat. *Regular team meetings* are usually held in weekly intervals. In this meetings individuals' progress, problems and next steps are discussed. The attendance to this meetings is mandatory. The *BiWeCo meeting* is for the project management and coordinators. It is held in a two week interval. In this meeting sub-teams' progress, problems and next steps are discussed and presented. In the following section the current Catrobat community is described.

5.3. Community

The structure of the Catrobat community is analyzed in this section. This offers an overview of roles in the Catrobat project. This is important to understand different roles and how they interact in Catrobat. A complete community exists around the Catrobat project. In this community several roles are present. In the following part, an overview of roles in the Catrobat community is given.

Project head: Wolfgang Slany is the project leader in the Catrobat community as well as the initiator of the project. He started the project in February 2010. After the start, he built a community around the project. Wolfgang Slany gives the project direction, objectives and visions. He also has the most influence on the project.

Project management: Annemarie Harzl is doing a lot of organizational and administrative work. She manages participation and students at Graz University of Technology. Accounts and permissions are typically granted by her.

Coordinators, senior members: They are involved in relevant decisions concerning the project. Thus they have large influence on Catrobat. This group is also coordinating, managing and supporting the project in various matters.

Active members: These members implement features and fix bugs in their daily work. In the Catrobat community, there is no role, which is only fixing bugs. Active members have the freedom to fix bugs or to implement new functionality. Most members of this group are students, thus they have varying amounts of time.

Passive users: Passive users are using services provided by the Catrobat community. Passive users can be for example teachers and young students in schools. They are reporting bugs in form of bug descriptions and ratings on Google Play. In May 2016, around 3200 ratings have been given on Google Play to Pocket Code, the Android version of Catrobat (Catrobat Team, 2016). In addition, 100.000 - 500.000 installations of Pocket Code are reported on Google Play at this time (Catrobat Team, 2016). As a result a lot of passive users around the globe are using services provided by Catrobat.

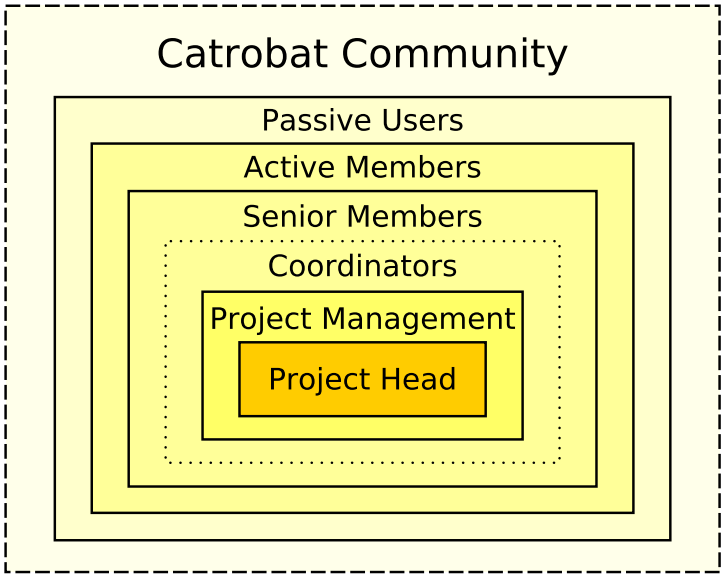

Figure 5.1.: Community structure and roles in Catrobat

In Figure 5.1 relevant roles of the Catrobat community are illustrated. The roles closer to the center have more influence on the project, than roles more outside. Some roles in the Catrobat project differ slightly from roles described by Ye and Kishida, 2003. Roles in Catrobat have different namings or are combined into one role. Most members of the inner circle of the Catrobat community have a strong connection to Graz University of Technology.

5.4. Software Development

Many Catrobat members produce a huge amount of source code in the Catrobat project. The software development process is influenced strongly by agile approaches. Catrobat has characteristics of agile software projects. In this section methods and practices used in the Catrobat project are described. A poor implementation of software development methods can also lead to issues in organizations. Thus it is important to investigate this topic.

In Catrobat, the primary software development method is XP. Some practices from Kanban are used and adapted. Some very important practices in Catrobat are *pair programming*, *refactoring*, *TDD*, *Kanban board* and *continuous integration*. These development methods and practices are described in detail in Chapter 4.

Catrobat members have different levels of experience in SE. Some of them are at an early stage of their studies. It is challenging to integrate such members and to increase their self-efficacy. For gaining a basic knowledge on writing simple and intuitive code, *"Clean Code: A Handbook of Agile Software Craftsmanship"* by Martin, 2008 is a basic reading for Catrobat newcomers. At the beginning of their participation, they have to attend tutorials. For example the Android tutorial or a testing tutorial. With these tutorials newcomers get an insight to how software is developed in Catrobat.

Kanban still offers potential to improve internal software development processes in Catrobat. For example *limit work in progress* and *improve collaboratively* are practices in Kanban, which can have a positive effect on processes in Catrobat. In Figure 5.2 the Kanban board of the sub-team

Catroid in the Catrobat project is illustrated. Many tasks are in in the sections *in development* and *ready for code review*. Such a high number of tasks in sections typically has a negative influence on the throughput time. The Kanban practice *limit work in progress* would have a positive influence on this task flow. A detailed description of the Kanban practice *limit work in progress* and other relevant Kanban practices can be found in Section 4.2.

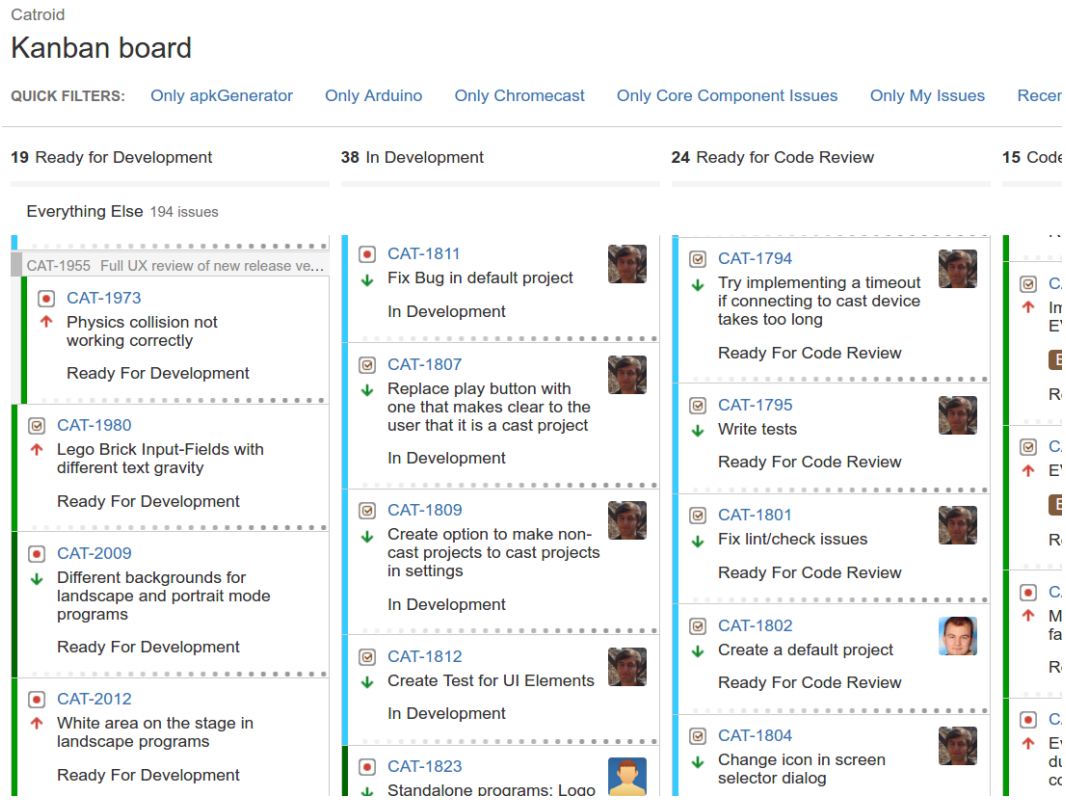

Figure 5.2.: Kanban board of a sub-project in Catrobat in May 2016

5.5. Catrobat and FOSS

Catrobat is a FOSS project. Thus it has characteristics of FOSS projects. In this section an overview is given on FOSS in Catrobat. In this thesis the topic

FOSS is analyzed in Catrobat and in traditional FOSS projects, especially concerning motivation.

Catrobat has already participated in the Google Summer of Code program several times. With this program the Catrobat community is getting in contact with possible contributors around the globe regularly. Several international contributors show regular interest in contributing to Catrobat. In May 2016, Catrobat had a mailing list with several international interested people.

Software developed by the Catrobat community is available under the GNU Affero General Public License (Catrobat Developers Team, 2016c). As a result, software developed by Catrobat can be modified and redistributed under the GNU Affero General Public License. The source code of most projects under the Catrobat umbrella project is hosted on GitHub (Catrobat Developers Team, 2016b). Thus it is visible to the public and always offers the opportunity for interested people to join Catrobat and its sub-projects. Interested people can gather general information about the project and first steps on the Catrobat developer website as well (Catrobat Developers Team, 2016a).

5.6. Catrobat and Education

Catrobat is strongly influenced by its educational context. Catrobat is a special mixture of several software settings. Education is one context. Therefore it is important to have a background knowledge about Catrobat and education.

The whole community around the project has strong relations to Graz University of Technology. The Infrastructure and most members are affiliated also to Graz University of Technology. Nearly the whole development and management is done there. Several lectures at the university are cooperating with Catrobat or even using it within lectures. For example such lectures are *Programming 0*, *Software Engineering and Knowledge Management* or *Mobile Applications*. Catrobat members should learn agile approaches, especially XP and testing. They can improve their programming skills and get in contact

with professional tools for software development. Catrobat members have also to deal with workflows like in a large FOSS project. In Catrobat all software is developed under common FOSS licences (Catrobat Developers Team, 2016c). Regular participations in the Google Summer of Code program are strengthening the FOSS context in Catrobat.

Another important field concerning education in the Catrobat project are passive users. Passive users are primary teachers and children. Catrobat has the objective to provide these users a platform to make their first experiences with programming. In schools Catrobat can be used to make computer science classes more exciting. In this context education also plays an important role.

Catrobat offers great learning opportunities at the university and also in the educational context of young children. The development of Catrobat and finally the usage of it is strongly related to education. Catrobat is making various contributions to the educational sector.

6. Human Motivation

The Catrobat project is examined in detail concerning motivation within this thesis. In this chapter general information about human motivation is provided.

6.1. Basics

Human beings are an important and crucial factor in organizations. The related research field of human beings in organizations is called human aspects. Human aspects focus on individuals and interactions between them. A human aspect with a strong impact on project success is human motivation. Human motivation determines individuals' behaviour in direction, duration and intensity (Comelli, Rosenstiel, and Nerdinger, 2014). It is an important aspect to set inner energy free for doing activities in a powerful way. Motivation counteracts exhaustion and tiredness.

Several factors with a motivational effect are identified (Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011; França and Fabio Q. B. da Silva, 2010; Asghar and Usman, 2013). These factors are called motivators. Motivators play an important role in the field of human motivation. Such motivators influence individuals' motivation. For example *recognition* is such a motivator. If people get recognized by others, it typically has a positive effect on their motivation.

Motivation has a strong impact on individuals' performance (P. C. Chen, Chern, and C. Y. Chen, 2012; A. Franca, Carneiro, and F. d. Silva, 2012; França and Fabio Q. B. da Silva, 2010; A. C. C. Franca, Araujo, and F. Q. B. da Silva, 2013; Asghar and Usman, 2013; S. Beecham et al., 2007). Performance is important at the individual level, the group level as well as at the

organizational level. Individuals are able to influence their performance with their behaviour. Human behaviour is influenced by four factors according to Comelli, Rosenstiel, and Nerdinger, 2014. These factors are visualized in Figure 6.1. One factor represents the permission. Individuals need the permission from their organizations or from their supervisors to do an activity. If individuals do not have permission, their behaviour is influenced and individuals will probably not do certain tasks.

Another factor represents skills and abilities of individuals. Usually, it is necessary to have a specific knowledge or experience to perform tasks. If these skills and abilities are insufficient, individuals need to gain more skills to be prepared for such tasks.

A further factor is represented by the actual situation. The situation can result in favorable or unfavorable conditions. Depending on these conditions, the behaviour is influenced positively or negatively.

The remaining factor represents volition. Volition is used in science as a term for the will of individuals. Volition is determined by inner values, motivation and other cognitive processes. These four factors also influence each other. For example if an individual gains more abilities, the motivation will increase to do more challenging tasks, where the gained abilities can be applied. These four factors influence human behaviour. There is a high potential for organizations to increase performance through motivating people. Organizations can use motivation strategies to motivate their staff (França and Fabio Q. B. da Silva, 2010).

For human motivation these four factors from above can be reduced to only two components. One resulting component is the situation, it consists of the factors *permission* and *situation*. If individuals are not motivated, then the situation can take a positive effect on individuals' motivation. The other component represents the individual, consisting of the factors *volition* and *abilities*. These two components together are determining individuals' motivation.

Often it is possible to shape situations which are motivating for individuals. For example working conditions can be used to trigger stimuli for motivation. Different stimuli can be used to shape such a motivating environment.

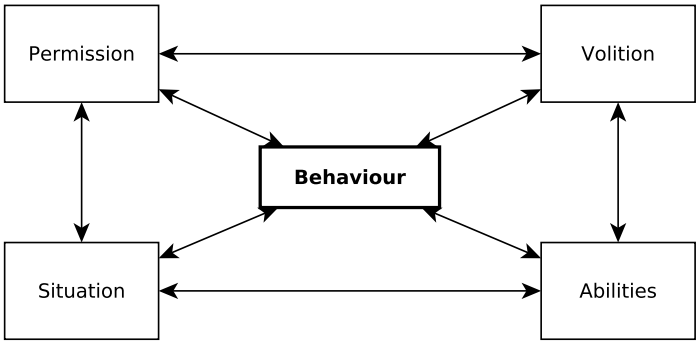

Figure 6.1.: Terms of human behaviour (Comelli, Rosenstiel, and Nerdinger, 2014)

6.1.1. Intrinsic and Extrinsic Motivation

Two different kinds of motivation are identified by Herzberg. They are called *intrinsic* and *extrinsic* motivation (Comelli, Rosenstiel, and Nerdinger, 2014; Helen Sharp et al., 2009; Wang, He, and J. Chen, 2005). In psychology both types are playing an important role. Motivators can be classified into intrinsic and extrinsic motivators. In Figure 6.2 examples for intrinsic and extrinsic motivators are illustrated.

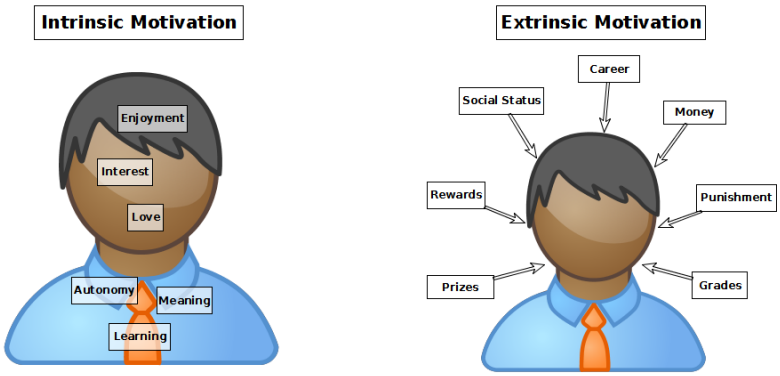

Figure 6.2.: Intrinsic and extrinsic motivators

Intrinsic motivation sets energy free and is triggered by internal stimuli. There are two relevant sources for intrinsic motivation (Comelli, Rosenstiel,

and Nerdinger, 2014). One source is having fun doing a certain activity. People like doing activities which result in fun, enthusiasm and satisfaction. The other source for intrinsic motivation is determined by moral and fundamental values in a person like honesty or fairness. These values and concepts are setting inner energy free to do also activities, which may not be a pleasure. Intrinsic motivators are for example *physical activity, contact with people, interest, sense* or *individual fulfilment*.

In contrast, extrinsic motivation is triggered by external stimuli. In scenarios with extrinsic motivation, a clear objective and corresponding advantages are given. With the achievement of this objective, clear advantages appear. Outcomes and possible consequences are in the focus. The activity itself is less important, possible outcomes are setting energy free. Some extrinsic motivators are *money, bonuses* or *social status*.

Typically, in working environments extrinsic and intrinsic motivators are present (Comelli, Rosenstiel, and Nerdinger, 2014). It is a mixture of both. Organizations have possibilities to provide motivating conditions to activate intrinsic or extrinsic motivation. Typically, organizations have more possibilities to trigger extrinsic motivators.

Research in the field of human motivation has introduced several models and theories to describe and to measure human motivation. In the next section these theories are listed.

6.1.2. Theories of Human Motivation

There are several theories of human motivation available in classic psychology (Tracy Hall et al., 2009; Rehman et al., 2011; Steinberga and Smite, 2011; A. Franca, Carneiro, and F. d. Silva, 2012; César et al., 2012; Farias Junior et al., 2012; Helen Sharp et al., 2009). In the field of SE numerous classical motivation theories have been applied. The most relevant theories in the field of SE are (Tracy Hall et al., 2009):

- Job Characteristics Theory
- Herzberg Motivational Theory
- Maslow's Hierarchy of Needs

- McClelland's Theory
- Goal-setting Theory
- Expectancy Theory
- Equity Theory
- Stimulus Response Theory

Despite these classical theories, a new model was introduced by Helen Sharp et al., 2009. This new model is called Motivators, Outcomes, Characteristics, Context (MOCC) model. This model is described in Section 6.2.4. In the following section the popular Herzberg Motivational Theory is described in detail. The Herzberg Motivational Theory and the MOCC model are used in the results chapter of this thesis for investigations.

Herzberg Motivational Theory

One of the most popular theories of human motivation is the Herzberg Motivational Theory. Herzberg did research on factors which lead to extreme satisfaction or extreme dissatisfaction (Comelli, Rosenstiel, and Nerdinger, 2014). In Herzberg's results, factors tended to be satisfying or dissatisfying. Herzberg recognized two different dimensions. These dimensions are satisfaction and dissatisfaction. In Figure 6.3 several factors are listed for both dimensions. As a result factors can be classified as satisfiers or dissatisfiers (Tracy Hall et al., 2009; Farias Junior et al., 2012; Li, Tan, Teo, et al., 2006; Hars and Ou, 2002; França and Fabio Q. B. da Silva, 2010).

He identified factors, which are frequently leading to extreme satisfaction and motivation. They are called intrinsic factors, motivators or satisfiers. These factors are primarily relevant for the satisfaction dimension. If such satisfiers are present, individuals tend to be motivated. Interestingly, if they are not present, they do not cause much dissatisfaction. Three satisfiers with the most impact on motivation and satisfaction are *achievement*, *recognition* and *work itself* (Comelli, Rosenstiel, and Nerdinger, 2014).

The second type of factors are called extrinsic factors, hygiene factors or dissatisfiers. If these factors are given, dissatisfaction will be avoided. They are not stimulating motivation. In contrast, if these factors are not given, individuals tend to be dissatisfied. The most relevant dissatisfiers are *company*

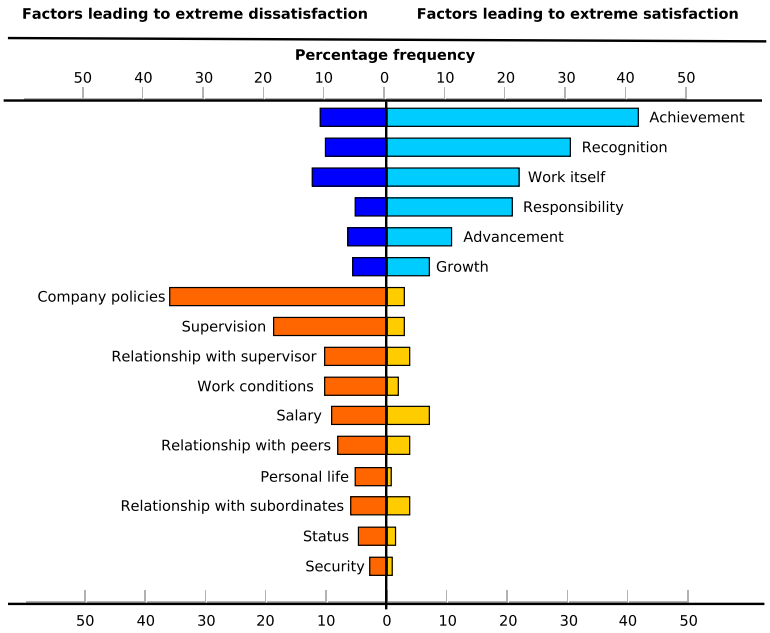

Figure 6.3.: Satisfiers and dissatisfiers identified by Herzberg (Comelli, Rosenstiel, and Nerdinger, 2014)

policies, *supervision* and *work conditions* (Comelli, Rosenstiel, and Nerdinger, 2014). Interestingly, *salary* is identified as a dissatisfier. Poor *salary* is resulting in demotivation and dissatisfaction.

All factors in Figure 6.3 are classified as satisfiers or dissatisfiers, but they all have characteristics of both dimensions. For example *achievement* has a higher importance in the satisfaction dimension. *Achievement* is leading to dissatisfaction four times less, than it leads to satisfaction. Thus *achievement* is classified as a satisfier in the Herzberg Theory.

Satisfiers and dissatisfiers determine individuals' performance. If satisfiers are not given, individuals will not be motivated. If dissatisfiers are not present, individuals tend to be dissatisfied. Possible outcomes of dissatisfaction are performing only a minimum or leaving the organization. Organizations should spend attention on satisfiers and dissatisfiers to ensure productive and satisfied individuals.

6.1.3. Sources of Human Motivation

Every human being is different and has different likings, interests and experiences. Motivation is influenced by the context of individuals. Several sources of human motivation are identified and visualized in Figure 6.4 (Comelli, Rosenstiel, and Nerdinger, 2014). All these sources overlap and influence each other. The sources *ego*, *groups* and *organizations* are described in the following subsections. In these subsections possible pitfalls and hurdles which can influence motivation are highlighted.

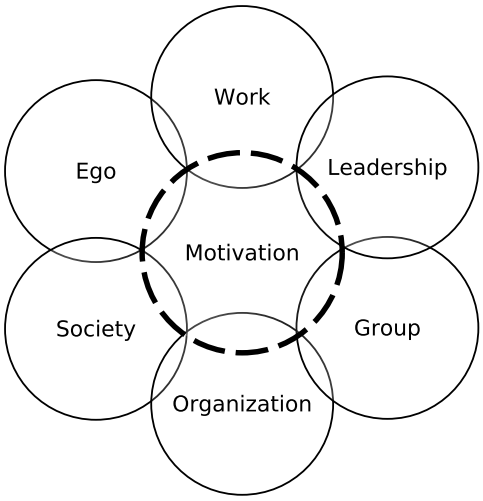

Figure 6.4.: Sources of human motivation (Comelli, Rosenstiel, and Nerdinger, 2014)

Ego

Maybe the most important source of human motivation is ego. Intrinsic motivation has a high motivational effect. Ego plays an important role for intrinsic motivation (Comelli, Rosenstiel, and Nerdinger, 2014). With a high level of intrinsic motivation, the will is strong to do activities, independently of outside forces. For example software developers typically have fun while developing software. They are interested in the field and like corresponding

activities (Wang, He, and J. Chen, 2005; H. Sharp and T. Hall, 2009; Hertel, Niedner, and Herrmann, 2003).

However, sometimes it is necessary to do something, which is not fun at all. Especially in organizations, unpopular activities also have to be done. For example writing a detailed documentation can be such an activity. In this case motivation can be very low, but it has to be done by somebody. Human beings have the possibility to influence their own behaviour in a certain way. In this case volition becomes important (Li, Tan, Xu, et al., 2011). Volition or will is the process of making decisions and setting corresponding actions. Without being motivated, everyone has the possibility to influence the own volition and the corresponding behaviour. In professional life, volition is often needed to force oneself to do something, which has to be done. Being motivated and having fun has another level of quality, than only to force oneself doing something.

The source of human motivation ego has a very strong impact on motivation. If motivation has its source in oneself, it is typically a very strong motivation.

Groups

Another source of human motivation is within groups. Several needs can be satisfied within groups. Human beings want to be part of groups (Comelli, Rosenstiel, and Nerdinger, 2014). Nowadays, a lot of projects are large and challenging. A huge amount of work has to be done. In this case groups are needed to master such a huge amount of work. Therefore groups are shaped and assigned to work. Large software projects are mastered in teams. The software development process is highly collaborative. For all team members it is necessary to work together in an interactive, effective and efficient way (A. Franca, Carneiro, and F. d. Silva, 2012; P. C. Chen, Chern, and C. Y. Chen, 2012). With groups it is possible to satisfy motivational factors like *security, social interaction, recognition* and further motivators. Groups have their own dynamics. These dynamics are influencing the behaviour of group members. It is a challenge to shape a group and make its members working together efficiently and effectively.

Motivation in groups is a significant aspect, but also a very complex one (Comelli, Rosenstiel, and Nerdinger, 2014; P. C. Chen, Chern, and C. Y. Chen, 2012; França and Fabio Q. B. da Silva, 2010; Farias Junior et al., 2012; O. Melo, Santana, and Kon, 2012). One possibility to motivate groups, is to declare group objectives (Comelli, Rosenstiel, and Nerdinger, 2014). It is a good practice to create group objectives with all group members together. Building and having group objectives is supporting group cohesion (Acuña, Gómez, and Juristo, 2009). Communication is also influencing group cohesion (Comelli, Rosenstiel, and Nerdinger, 2014). For example face to face or informal communication has a positive effect on group cohesion (Steinberga and Smite, 2011). The higher the group cohesion is, the higher the likeliness of each group member to stay within said group. Good group cohesion is an important aspect for group performance, but not a guarantee.

As already mentioned, communication is an important aspect in groups and also a very important motivational factor (Yanyan and Renzuo, 2008; Lenberg, Feldt, and Wallgren, 2015; Comelli, Rosenstiel, and Nerdinger, 2014; Sarah Beecham et al., 2008). For effective and efficient communication group size is crucial. If teams are too large, communication between all group members is nearly impossible. Especially, regular face to face communication is not possible in large teams.

The behaviour of individuals in meetings is very different. Some individuals like to take a leading role in discussions and to be active. Others tend to be silent listeners. With larger groups the number of such silent listeners increases, which is undesired. Therefore meetings with smaller group sizes are recommended.

Organizations

Organizations have possibilities to activate extrinsic and intrinsic motivators of their staff. For example one of the most frequent extrinsic motivators in organizations is money.

A very important task for organizations is to ensure good working conditions. Working in a positive working atmosphere supports activating

motivation (César et al., 2012; França and Fabio Q. B. da Silva, 2010; Asghar and Usman, 2013). In organizations with a good working atmosphere, mistakes are not cause for punishment. In error cultures where individuals get punished for making errors, people tend to be dissatisfied. For a positive working atmosphere such negative error cultures with punishments should be avoided. It is better to see errors as a possibility for learning.

Another source of human motivation identified by Comelli, Rosenstiel, and Nerdinger, 2014 is leadership or supervision. Supervision is one possibility to influence motivation of individuals in organizations. For example Herzberg identified *supervision* and *relationship with supervisor* as important extrinsic motivational factors (Tracy Hall et al., 2009; Comelli, Rosenstiel, and Nerdinger, 2014). A supervisor is able to influence the behaviour of individuals by giving them objectives and instructions. For example objective agreements are a possibility for supervisors to motivate people. Another possibility for supervisors is to give recognition in form of appreciation or constructive feedback (Steinberga and Smite, 2011; A. Franca, Carneiro, and F. d. Silva, 2012). This is a very strong motivator, but often rarely applied by supervisors.

Organizations and supervisors have a strong impact on motivation. Organizations can react with countermeasures against demotivation. For example such countermeasures can be appreciating performance, shaping interesting tasks, giving responsibility or offering a career path.

6.1.4. Job Satisfaction

For individuals it is important to be satisfied with their daily work and their job. Everyone has different expectations of a job. For example *salary*, *colleagues* or *career path*. If these individual expectations are almost fulfilled, job satisfaction will be a result (Comelli, Rosenstiel, and Nerdinger, 2014). If there is a strong mismatch between expectations and real conditions, typically dissatisfaction is the consequence. However, dissatisfied individuals are able to accept dissatisfaction for a certain time. This can result in behaviour changes like reducing the claims, working less hard or in extreme cases leaving an organization. Thus job satisfaction is an important aspect

in organizations (P. C. Chen, Chern, and C. Y. Chen, 2012; Said and Munap, 2010; Helen Sharp et al., 2009; Thatcher, Liu, and Stepina, 2002).

Job satisfaction is strongly linked to human motivation. Both are increasing continuously together. Especially, the Herzberg Motivational Theory, described in Section 6.1.2, combines motivation and job satisfaction (House and Wigdor, 1967; Tietjen and Myers, 1998). Typically, the level of motivation and job satisfaction is very similar to each other.

6.1.5. Big Five Personality Traits

The Big Five personality traits describe the personality of individuals, see Figure 6.5. Individuals' characteristics are important for understanding their motivation. As already described, a strong source of human motivation is oneself. Thus personality characteristics are crucial. Questionnaires like the Ten Item Personality Inventory (TIPI) are built upon these traits (Gosling, Rentfrow, and Jr., 2003). Depending on results and corresponding scores of the questionnaire, these five traits can be determined and used for describing individuals' characteristics. In the following subsections these five traits are described. For every trait, characteristics are appended to get a better feeling for high and low scores.

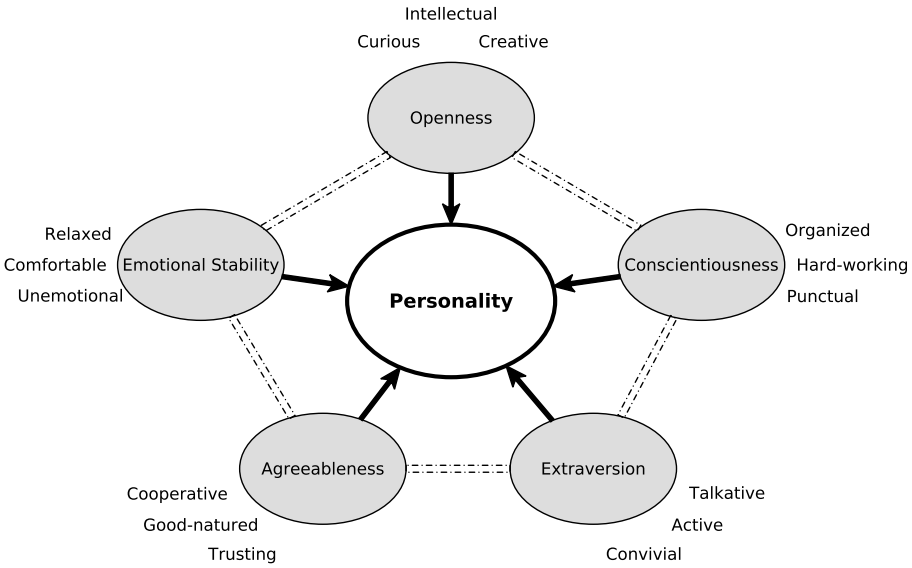

Figure 6.5.: Big five personality traits (Gosling, Rentfrow, and Jr., 2003)

Extraversion

Extraversion stands for interpersonal interaction and the general activity of individuals. In Table 6.1 characteristics of high and low scores of extraversion are shown.

High scores	**Low scores**
Feel comfortable around people	Stay in the background
Start conversations	Have a little to say
Like being the center of attention	Like to stay in the background
Energetic	Reserved

Table 6.1.: Extraversion characteristics (Goldberg et al., 2006; Goldberg, n.d.; Judge et al., 1999)

High scores	Low scores
Interested in people	Feel little concern for others
Sympathize with others' feelings	Insult others
Have a soft heart	Not interested in others
Take time for others	

Table 6.2.: Agreeableness characteristics (Goldberg et al., 2006; Goldberg, n.d.; Judge et al., 1999)

High scores	Low scores
Always prepared	Leave belongings around
Pay attention to details	Make a mess of things
Get chores done right away	Avoid doing duties
Like order	Chaotic
Follow a schedule	

Table 6.3.: Conscientiousness characteristics (Goldberg et al., 2006; Goldberg, n.d.; Judge et al., 1999)

Agreeableness

Agreeableness describes how individuals get along with others. In Table 6.2 characteristics of high and low scores of agreeableness are shown.

Conscientiousness

Conscientiousness represents self-control and organization of individuals. In Table 6.3 characteristics of high and low scores of conscientiousness are shown.

Emotional Stability

Emotional stability describes how individuals react on their emotions. In Table 6.4 characteristics of high and low scores of emotional stability are shown.

High scores	Low scores
Relaxed	Get stressed out easily
Comfortable	Worry about things
Unemotional	Many mood changes
	Get irritated easily

Table 6.4.: Emotional stability characteristics (Goldberg et al., 2006; Goldberg, n.d.; Judge et al., 1999)

High scores	Low scores
Have a vivid imagination	Have a poor imagination
Creative	Uncreative
Curious	Uncurious
Original	Conventional

Table 6.5.: Openness characteristics (Goldberg et al., 2006; Goldberg, n.d.; Judge et al., 1999)

Openness

Openness represents the willingness to new experiences and adventures. In Table 6.5 characteristics of high and low scores of openness are shown.

The TIPI questionnaire is part of the survey done within this thesis. With this questionnaire, it is possible to become aware of personalities in the Catrobat project.

6.2. Software Engineering

In this section a special focus is on human motivation in SE. A lot of research has already been done in this field. Important findings in literature are described in this chapter. Several of them are used in the survey or in the results chapter of this thesis to interpret survey results.

6.2.1. Human Motivation and Software Engineering

A common goal in SE projects is to deliver high quality software in time (Cockburn and Highsmith, 2001; Van Kelle et al., 2015; Lenberg, Feldt, and Wallgren, 2015). For achieving success within SE organizations technical aspects and competences are needed. These aspects represent only one part in the value-added chain. Another part are human aspects. With the rise of agile software development methods, the awareness for social and human aspects has become more important (Cockburn and Highsmith, 2001).

For organizations it is valuable to invest efforts in human aspects. Several social success factors have been identified for example *leadership, communication, project size* or *the relationship between seniors and juniors* (Van Kelle et al., 2015). For example the last one has a remarkable impact on juniors' attitudes. If seniors are willing to support juniors, juniors will have a high self-efficacy.

A lot of software projects fail and are aborted (Van Kelle et al., 2015; Chow and Cao, 2008). Several factors for project failure in software projects are caused by human behaviour. For example such factors can be *bad customer relationship, lack of teamwork, organizational culture is too traditional* or *a lack of motivation* (Van Kelle et al., 2015). The last factor *motivation* is in the focus of this chapter.

Motivation is a key success factor in SE (Cockburn and Highsmith, 2001; Van Kelle et al., 2015; Lenberg, Feldt, and Wallgren, 2015). In the Agile Manifesto the importance of motivation is mentioned by Beck et al., 2001. Motivated individuals take actions to be productive and to make progress. As a result, motivated individuals have a tremendous impact on the success of projects as well as of organizations. Typically, software engineers are motivated by

their work itself. In this case they have a high intrinsic motivation, because their interest in the field is very high (Wang, He, and J. Chen, 2005; H. Sharp and T. Hall, 2009; Hertel, Niedner, and Herrmann, 2003).

Engineers often reach a kind of flow during their work (Csikszentmihalyi and LeFevre, 1989). It is typically a good feeling to be in a flow. During such a flow people can focus on the activity and to master a challenge. This has a motivational effect and people tend to be very productive during such a flow. Specific conditions have to be given to experience such a flow. For example a task with appropriate difficulty and a working environment without interruptions is needed. The difficulty of tasks has to be challenging, but not too difficult. A balance between abilities of the software engineer and the difficulty of the task is necessary, see Figure 6.6 (Comelli, Rosenstiel, and Nerdinger, 2014). If the ratio between challenge and abilities is not given, stress, boredom or concern can be consequences. In contrast, if these conditions are given, then the engineer has good chances to experience a flow phenomenon. Inside such a flow, time flies by. For the engineer it appears to be one unity with the task and all other things become nonrelevant.

Figure 6.6.: Flow phenomenon depending on challenge and abilities (Comelli, Rosenstiel, and Nerdinger, 2014)

In Catrobat there is a team room available for meetings and working to-

	Western countries	**Malaysia**
1	Growth oriented	Growth oriented
2	Introverted	Achievement oriented
3	Autonomous	Introverted
4	Need for organizational stability	Needs a competent supervisor
5	Technically competent	Technically competent

Table 6.6.: Characteristics of software engineers: Western countries versus Malaysia (Sarah Beecham et al., 2008; Rehman et al., 2011)

gether. If a lot of Catrobat members are using it at the same time, this can lead to noise and interruptions. In this case the flow experience can be interrupted. As a result rooms with many interactions can be a hurdle for flow experiences.

Another factor is influencing motivation of software engineers. This factor is the personality of software engineers. In the following section personality characteristics are investigated.

6.2.2. Characteristics of Software Engineers

Software engineers have typical characteristics (Sarah Beecham et al., 2008). In western countries most relevant characteristics of software engineers are *growth oriented* and *introverted*. Further relevant characteristics of software engineers are *autonomous, need for stability* and *technically competent*. A complete list of characteristics and needs of software engineers is illustrated in Figure 6.7. These characteristics were identified by Helen Sharp et al., 2009. Also motivators have been identified for all these characteristics by Helen Sharp et al., 2009. Depending on these characteristics it is possible to describe motivational effects of motivators. For example if a person has a strong need for feedback, then the motivator feedback will have a strong impact on the person's motivation.

There are also some differences among cultures. For example in Malaysia these characteristics differ slightly from western countries, see Table 6.6 (Rehman et al., 2011).

Software engineers' characterics	
Software engineers are:	And have a need for/to:
Growth-oriented	Variety
Introverted	Challenge
Autonomous	Identify with group
Achievement-oriented	Competent supervisors
Technically-competent	Feedback
Marketable	Contribute
Creative	Stability

Figure 6.7.: Software engineers' characteristics (Helen Sharp et al., 2009)

Depending on individuals' characteristics, motivators have more or less impact on motivation. A focus on different motivators and demotivators in SE is given in the following section.

6.2.3. Motivators and Demotivators in Software Engineering

There are various results in research concerning motivation in SE depending on software development method, culture, personalities and context. In this section a short overview of motivators in SE is given. In Table 6.7 most motivators and demotivators are identified by Sarah Beecham et al., 2008 and A. Franca, Gouveia, et al., 2011.

In Section 7.3 all relevant motivators for this thesis are listed and described shortly. An exemplary demotivator is *poor quality software*. Producing poor quality software is a strong demotivator in SE. Coding can be seen as art for software engineers. They can create elegant solutions. If they have insufficient time, a lack of resources or inexperienced staff, this can result in poor quality. For software engineers it is demotivating to produce such a piece of software. Working on poor code is also very demotivating and time

	Motivators	Demotivators
1	Trust/Respect	Poor working environment
2	Working conditions	Poor communication
3	Autonomy	No opportunity for relationships
4	Equity	Unrealistic goals
5	Learning	Lack of promotion
6	Management/Coordination	Poor quality software
7	Work life balance	Lack of influence
8	Career path	Unfair reward system
9	Team quality	Uninteresting work
10	Sense of belonging	Inequity
11	Making a contribution	Stress
12	Creativity/Innovation	Task Complexity is too difficult
13	Successful organization	Task Complexity is too easy
14	Development needs addressed	Organizational overhead
15	Sufficient resources	
16	Technical challenging work	
17	Empowerment/Responsibility	
18	Employee participation	
19	Identify with the task	
20	Meet new people	
21	Variety of work	
22	Grading	
23	Project success	
24	Feedback	
25	Recognition	
26	Status/Reputation	
27	Non-financial benefits	
28	Professionalism	
29	Relationship with users	
30	Financial benefits	

Table 6.7.: Relevant motivators and demotivators (Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011)

consuming. Quality is identified as a success factor for projects by Chow and Cao, 2008. Poor quality software can result in project failure.

6.2.4. Motivators Outcomes Characteristics Context Model

This model is described in detail, because it is applied to the Catrobat project in the results chapter of this thesis. With the collected data from the survey and some general data about Catrobat, this model is applied to Catrobat and gives an overview of the whole Catrobat from several perspectives, especially motivation.

Several theories from classic psychology are applied in the research field of motivation in SE (A. Franca, Gouveia, et al., 2011). Beside these classical motivation theories, a special one was created and introduced, the MOCC model (Helen Sharp et al., 2009). Its objective is to model the world of software engineers. It is built upon the results of the systematic literature review in SE done by Sarah Beecham et al., 2008. As a result, the MOCC model has many predefined variables to describe SE environments. These variables are adapted from the systematic literature review in SE. An extensive investigation can be done with these predefined variables. The MOCC model consists of the following components: *motivators*, *outcomes*, *characteristics* and *context*. The naming of the MOCC model is the acronym of these four components. They are visualized in Figure 6.8.

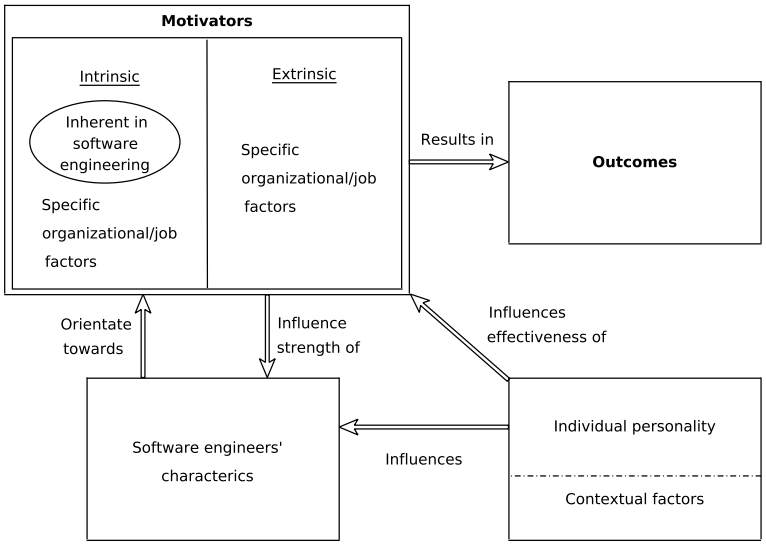

Figure 6.8.: Overview of the Motivators, Outcomes, Characteristics, Context model (Helen Sharp et al., 2009)

The characteristics component represents software engineers' characteristics. These characteristics are already described in Section 6.2.2. For example software engineers can be *growth oriented*, *introverted* or *autonomous* and have needs like *variety*, *challenge* or *feedback*. These personality characteristics are

represented by the software engineers' characteristics component.

The characteristics component is influenced by other factors. For describing these factors a further component is added. This second component is the context component. This component consists of two groups of factors. The first group of factors represents the individual personality of every involved individual. This personality can differ strongly from generalized personalities. The second group of factors represents contextual factors. For example these factors can be geographical locations, type of organizations, nationality, cultures, job roles and many more (Helen Sharp et al., 2009). The characteristics and the context component are illustrated together in Figure 6.9.

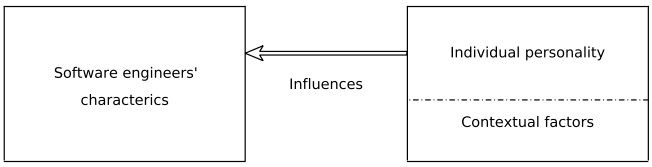

Figure 6.9.: Context component influences the characteristics component (Helen Sharp et al., 2009)

The motivators component is split into intrinsic and extrinsic motivators. These two types of motivation are already described in Section 6.1.1. A special group of motivators is identified, which is inherent in SE (Helen Sharp et al., 2009). For example some inherent motivators are *challenging, problem-solving* or *experimental*. The motivators component represents intrinsic and extrinsic motivators, see Figure 6.10. The motivators component and the characteristics component influence each other. The context component also has influence on the motivators component. Thus motivators can be directly influenced by individual personalities and contexts.

The outcomes component represents and describes outcomes of the system which is analyzed. Possible outcomes are *retention rates, productivity, progress* or *project delivery time* (Helen Sharp et al., 2009). If quantities are used in the outcome component, then it is possible to compare these quantities at different points in time.

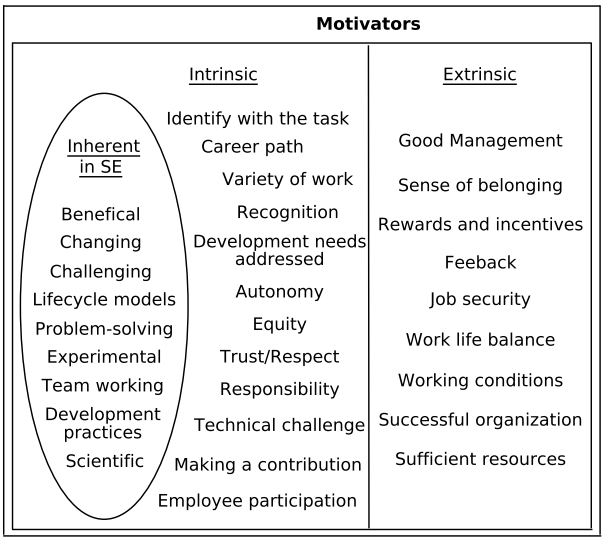

Figure 6.10.: Intrinsic and extrinsic motivators in the motivators component (Helen Sharp et al., 2009)

There are some limitations coming with this model. For example the reliance on literature used in the literature review is one limitation (Helen Sharp et al., 2009).

The MOCC model is a framework to analyze motivation in SE settings. In the following chapter various SE settings are analyzed concerning motivation.

6.2.5. Motivation in Various Software Engineering Settings

Various settings are possible for developing software. These settings depend on different software development methods, geographical locations, type of organizations and many more. Combinations of these settings are possible. In this section various settings for SE are analyzed. Catrobat is a mixture of several of them. For example some relevant settings for Catrobat are *agile*, *education* and *FOSS*. In the results chapter of this thesis, a comparison between motivators in Catrobat and these settings is done.

An overview of settings which are analyzed in this section is given below. Some exemplary motivators for each setting are listed in every corresponding section.

- SE in companies
- SE in agile teams
- SE in FOSS projects
- SE in educational environments

Companies

In this section motivation in companies is analyzed. Relevant motivators in Brazilian companies and motivators in a company from Cambridge called *Red Gate* are described. The first company setting represents motivation in multiple Brazilian companies. This data was collected via questionnaires. The most important motivators identified by França and Fabio Q. B. da Silva, 2010 are *working with people*, *work life balance*, *problem solving* and *meaningful products*.

The second company setting represents motivators identified by Sach, H. Sharp, and Petre, 2011. These motivators were gathered from semi-structured interviews with 13 professional software engineers from a software company in Cambridge. The most important motivational factors in this setting are *work that is useful*, *producing good software*, *problem solving* and *collaborating*.

The most relevant motivators in these two settings are listed below.

- Collaborating
- Work life balance
- Work that is useful
- Working with people
- Producing good software
- Meaningful products
- Problem solving

Results in both company settings indicate the importance of working with others. This is identified as very motivating. Corresponding motivators are *working with people* and *collaborating*. Thus, these software engineers like to be part of a group, which is identified as a source of motivation. Other important motivators in these settings are *work that is useful* and *meaningful products*. In this case a meaning behind work is motivating.

Agile

In environments using agile software development methods there is a special focus on people (Beck et al., 2001; Cockburn and Highsmith, 2001; O. Melo, Santana, and Kon, 2012; Van Kelle et al., 2015; Melnik and Maurer, 2006; Chow and Cao, 2008). In the Agile Manifesto it is defined as a core value: *"Individuals and interactions over processes and tools"* (Beck et al., 2001). Motivation is also an important aspect in agile processes. Again the Agile Manifesto itself addresses motivation directly with *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."* (Beck et al., 2001). The creators of the manifesto honor the importance of motivated individuals and created a principle for this aspect. As a consequence, motivated individuals are crucial for project success in agile environments.

A study done by Melnik and Maurer, 2006 compared job satisfaction in agile and in non agile teams. Results from a non agile perspective are rather poor. Not only do software engineers exhibit poorer job satisfaction, roles like managers also exhibit poorer job satisfaction. Further results of this study are listed below (Melnik and Maurer, 2006). The statements describe job satisfaction in non agile environments compared with agile environments.

- 8 times as many *very dissatisfied* individuals
- 3.5 times as many *somewhat dissatisfied* individuals
- Almost three times as many indifferent *neither satisfied nor dissatisfied* individuals
- Almost twice fewer *somewhat satisfied* individuals
- Three times fewer *very satisfied* individuals

Examples for relevant motivators in agile environments are (O. Melo, Santana, and Kon, 2012; Šteinberga and Šmite, 2011):

- Feeling of progress
- Technical challenging work
- Development needs addressed
- Employee participation
- Identify with the task
- Project success
- Team quality

Agile software development methods focus on people and interactions between them. A further positive effect of agile teams is a higher job satisfaction, than in non agile organizations (Melnik and Maurer, 2006). With the increasing number of organizations using agile software development methods, it is important to have social skills for teamwork and collaboration. This may be a challenge for people switching from non agile to agile environments.

FOSS

In FOSS community systems it is necessary to have a reliable and powerful community (Ye and Kishida, 2003; Hars and Ou, 2002). Accordingly, motivation is an important topic in FOSS projects. For the existence of FOSS communities, it is essential to have active members. A related task is to keep participating members motivated and prevent them from leaving. Another related task is to engage new members in FOSS projects. Many newcomers drop out at an early stage of their participation (Steinmacher, Chaves, et al., 2014; Steinmacher, Wiese, et al., 2015; Zhou and Mockus, 2015; Ye and Kishida, 2003). Several hurdles for newcomers are described in Section 3.3.

A motivating and supportive onboarding process is required to avoid such dropouts.

FOSS raises several interesting questions like *why are community members working for free* or *what motivates individuals to join a FOSS project*. Intrinsic as well as extrinsic factors are relevant motivators for software engineers to join FOSS projects (Ye and Kishida, 2003; Hars and Ou, 2002). Especially intrinsic motivation is crucial for staying in the long term.

A strong intrinsic motivator is *learning* (Ye and Kishida, 2003; H. Ellis et al., 2011). Human beings have a natural desire to develop skills and abilities in topics they like. Software engineers are interested in computers and software. Thus they want to become better in the field. In FOSS projects, there is often a large existing codebase. Reading source code of others is a great learning opportunity to become a better engineer.

Another benefit from gaining more abilities is becoming a more important member in the community. As a consequence, more influence in decision making is gained. Aside from intrinsic factors, extrinsic factors can be satisfied as well. Positive *feedback* or *status / reputation* can be examples of such extrinsic factors (Raymond, 1999; Ye and Kishida, 2003). For individuals' *career path*, FOSS participations can be an advantage (Hars and Ou, 2002; H. Ellis et al., 2011). This can lead to better job opportunities and higher salaries. Relevant FOSS motivators are listed below (Wang, He, and J. Chen, 2005; Ye and Kishida, 2003):

- Learning
- Trust / respect
- Status / reputation
- Employee participation
- Team quality
- Career path
- Recognition

Participation in FOSS projects can have several advantages. A chain of motivators can be satisfied with an active FOSS participation.

Education

One scenario for motivation in SE is within an educational environment. Research in this special setting is focusing primarily on SE students and universities. There are courses at universities which handle software development and its processes. Even courses with virtual teams from different geographical locations are held (Bosnić et al., 2011). Some courses build cooperations with FOSS projects (H. Ellis et al., 2011; Villarrubia and Kim, 2015; Jaccheri and Osterlie, 2007; H. J. C. Ellis et al., 2010). In the following paragraph, motivators and demotivators in educational environments are described.

Starting with the classical extrinsic motivator in education *grading* (Bosnić et al., 2011). This motivator is very present and has a large influence on students' behaviour. Usually, students want to have good grades. They have to take actions and adjust their behaviour to achieve that. A further important motivator is *learning*. *Learning* is an intrinsic motivator and maybe the most creditable one in educational environments. The higher the interest of a student in a certain field is, the stronger is the motivator *learning*. It is important for students to study a field in which they are interested in. Examples for motivators in educational software projects are listed below.

- Grading
- Learning
- Team quality
- Project success
- Meet new people
- Technical challenging work

In educational environments, several demotivators are identified as well. For example *lack of time* can be demotivating in educational environments (Bosnić et al., 2011). Thus a good self-organization and time management is important during study. Possible demotivators in educational environments are listed below.

- Stress
- Task complexity
- Poor supervision

- Poor communication
- Unfair reward system
- Producing poor quality software
- Bad relationship with colleagues

Especially, in educational contexts there are several challenges. Students experience issues such as technical deficit, poor team communication or lack of information. Often students have to invest more and more time to get their tasks done, but time is often scarce. This can be very challenging and frustrating for students. *Learning* and *grading* seem to be the most present motivators in educational contexts (Bosnić et al., 2011).

7. Problem

In this chapter the problem of this thesis is verbalized and discussed. Therefore challenges and success factors in the Catrobat project are described. Afterwards objectives of this thesis are listed. In the last section of this chapter, the motivator set and the demotivator set for the survey done within this research are listed.

7.1. Catrobat

The Catrobat project is a mixture of different SE settings. It has characteristics of *agile, FOSS* and *education*. Catrobat is a hybrid software project which is a combination of these three categories. A FOSS project with a very strong educational background using agile software development approaches is rare. Therefore it offers special opportunities for research, because such a special combination is a rare phenomenon.

Many people have to be managed and guided. Sub-teams are led by coordinators who try to get the best out of their members. Management and coordination are success factors in such a project. Another success factor is communication and information exchange. People have a natural need for information. Thus it is important to pass on information to all Catrobat members in the project. Onboarding of newcomers is a further success factor. Most Catrobat members are students at Graz University of Technology. After they have finished their current course, they often leave the project. As a result, the retention rate is low and a constant exchange of Catrobat members is a consequence. Thus the onboarding process in such a project is crucial. For newcomers it is important to have a supportive environment. It is not easy to get started in such a large project. Especially the enormous

codebase can be very challenging. It is important to support newcomers and to keep them motivated during their participation.

7.2. Objectives

This research tries to capture the actual state of the Catrobat project concerning motivation. An objective is to identify potential issues and demotivating factors in Catrobat. Then strategies and possible improvements can be evaluated to perform. Such strategies can encourage Catrobat members to stay in the project for a longer time.

General data about Catrobat is also collected. For example data about demography, communication, meetings, teamwork or coordination is collected too. With this data it is possible to further analyze single areas of interest in the Catrobat project.

With data about motivators in the Catrobat project, it is also possible to compare present motivators in Catrobat with other settings in SE. In literature there are various settings reported, which can be used for comparison. Such comparisons are done in the results chapter of this thesis.

Another objective is to categorize the Catrobat project by motivators into the following categories *agile, FOSS, education* and *company*. For this categorization relevant motivators for every setting are taken and compared with their presence in the Catrobat project. Then motivators which have a rather low presence in Catrobat and are important in a specific category can be used for improvement. This categorization is done in the results chapter of this thesis to learn from other settings.

With data from the survey it is possible to apply the Herzberg theory and the MOCC model to the Catrobat project. These models describe the motivational state of the project. With the Herzberg Theory and the MOCC model applied to Catrobat, possible actions for improvement can be discussed.

7.3. Set of Motivators and Demotivators

In this section a set of motivators and a set of demotivators are listed and described briefly. These motivators and demotivators are used in the survey done within this thesis to measure motivation in the Catrobat project. Most of them were identified by Sarah Beecham et al., 2008 and A. Franca, Gouveia, et al., 2011.

Motivators

Learning: Learning and mastery is a strong motivator. Human beings are learning their whole life. Beginning at school up to the professional life, learning and improving skills is a general need.

Autonomy: Autonomy is the freedom for self-governing or self-directing. It is possible to do things in an own way, independently of others.

Grading: Especially in educational environments grading is a motivator.

Creativity / Innovation: Depending on industry, organization, role and job, it is easier or harder to bring in own creativity or innovation.

Work life balance: Life should not only consist of work. For example too much work can lead to burnouts. A balance between work and leisure time should be aspired.

Identify with task: Doing tasks which make no sense for the overall goals can be demotivating for individuals. In addition, doing tasks which do not match with expertise or qualification can also be demotivating.

Making a contribution: Being part of something larger than oneself can be very motivating. For example contributing to a large project.

Variety of work: Variety differs strongly depending on industry, organization, role and job. It is demotivating to always do the same activity. Avoiding monotonous work should be aspired.

Technically challenging work: For motivation a too low level of difficulty is boring and demotivating. In contrast, challenges have rather a motivating effect.

Development needs addressed: Doing work for meaningful purposes or objectives is very important. If people see meaning in their work, they have an enormous potential for motivation.

Meet new people: Human beings are very social and like to be in groups. For individuals it can be motivating to meet new people.

Team quality: Work is often done in groups. In this case cooperation is needed. If there is a good team quality, team members tend to be motivated.

Employee participation: Employee participation is the will of employees to take action in various matters. For example employees help to improve internal processes in organizations.

Empowerment / Responsibility: Responsibility and empowerment are motivating for people. In groups and organizations this motivator becomes relevant.

Feedback: Receiving feedback is often very important for people. If individuals never receive feedback, they do not know how they are performing and how others see them. Constructive feedback is usually motivating.

Trust / Respect: Trust and respect are also important motivating factors in groups and organizations.

Status / Reputation: In groups and organizations everyone has a certain status or reputation. Individuals with a high status have a relative high social position within groups.

Recognition: If someone is doing a good job, it is very motivating to be recognized by others. Some examples for recognition are appreciation, promotion or money (Gardazi et al., 2009).

Professionalism: Doing tasks in a professional way is motivating. In contrast, unprofessional behaviour of individuals can be demotivating. If most members of a team are acting unprofessionally, then a demotivated team can be a result.

Sense of belonging: Human beings are social. They want to feel a sense of belonging to groups or organizations.

Appropriate working conditions: Appropriate working conditions are very important to feel comfortable and to be productive.

Working in a successful organization: Working in a successful and popular organization is typically motivating.

Good management and coordination: Supervision and a good relationship to supervisors have a strong impact on motivation. Management has also a strong impact on organization policies. These policies are also relevant for motivation.

Career path: Most people want to be promoted during their professional life. Good perspectives for promotions are motivating.

Financial benefits: Financial benefits are often used as a possibility to motivate people. Financial benefits tend to influence motivation more in the short term.

Non-financial benefits: Benefits in non-financial form are also motivating. Training opportunities can be such a non-financial benefit.

Project / Product success: Successful projects or products have a strong motivational effect on individuals. Usually, they have a high job security and they can be proud of their outcomes.

Relationship with Users / Customers: Getting feedback and appreciation from users or customers is motivating. Good relationships between different stakeholders also have positive effects.

Equity: In groups and organizations equity is important. Especially, in topics like salary, promotion or benefits it is necessary to treat individuals equally.

Sufficient Resources: Enough resources in form of manpower, hardware or other aspects influence motivation. A lack of specific resources can lead to stress and demotivation.

Demotivators

Poor working environment: The working environment has a large impact on individuals' motivation in teams and organizations.

Poor communication: This demotivator is present if individuals are provided with only little information.

No opportunity for relationships: In some working environments there are only rare opportunities to build relationships. Missing interpersonal communication strengthens this demotivator.

Unrealistic goals: If management or coordinators set unattainable goals this demotivator is present.

Lack of promotion: Missing opportunities for promotion are the cause for this demotivator.

Unfair reward system: This demotivator describes inequity in grading and payment.

Poor quality software: In SE quality is an important factor. If quality is poor, then software engineers tend to be demotivated.

Lack of influence: Missing influence in projects and organizations is demotivating.

Uninteresting work: Monotonous and unchallenging work can cause this demotivator.

Inequity: Inequity can be present in projects and organizations. Promotions and money are possible sources of inequity.

Stress: A high workload or unrealistic timelines can result in stress.

Task Complexity is too difficult: Too difficult tasks can be demotivating for individuals.

Task Complexity is too easy: Also too easy tasks can lead to demotivation.

Organizational overhead: In organizations and projects administrative work has to be done. If there is much administrative overhead, then this demotivator can be present.

8. Research Methodology

An empirical quantitative research method was chosen for this research (Bässler, 2014). A survey was created to gather relevant data about the Catrobat project. The survey consists primarily of quantitative questions. Some qualitative open ended questions are used to enable participants to give qualitative answers as well.

This research is an applied one, it focuses on the Catrobat project. It collects data and generates knowledge about motivation, satisfaction and attitudes in the Catrobat project. Afterwards this knowledge is used to measure and describe the current motivational state in the project. Then potentials for improvement are detected. The following research procedure has been used, which is adapted from Bässler, 2014.

- Examination of the problem
- Theoretical preparation
- Creation of the survey
- Execution of the survey
- Preparation and processing of the results
- Interpretation of the results
- Conclusion and future work

Survey

In May 2016, the Catrobat community consisted of around 150 people. A survey was chosen in order to reach most of them and to get a meaningful sample. The survey was built with LimeSurvey, see Figure 8.1 (LimeSurvey Project Team / Carsten Schmitz, 2016). LimeSurvey is a FOSS survey tool. It offers great functionalities for creating, executing and visualizing surveys.

For example sending invitations or reminders can be done with a single click. A LimeSurvey instance was hosted on an university server to ensure data privacy.

The survey consists of 64 questions. Open ended, nominal, ordinal, Likert and quantitative questions are used within this survey. Depending on these question types, different diagram types are chosen for visualization. For example nominal questions are shown in pie charts. Satisfaction factors, motivators and demotivators are treated in Likert scale questions.

For the relative ordering of them, the mean or the sum of the Likert scores is built (H. N. Boone and D. A. Boone, 2012; Bertram, 2007). Likert scores for satisfaction factors are *very good (5), good (4), average (3), rather poor (2), poor (1)*. Scores for motivators are *motivates me strongly (4), motivates me (3), motivates me a little (2), no impact on motivation (1)*. Demotivators have the scores *demotivates me strongly (4), demotivates me (3), demotivates me a little (2), no impact on motivation (1)*. One more likert scale is used for motivators' and demotivators' presence in Catrobat. The following scores are used for this scale: *completely present (4), rather present (3), rather not present (2), not present (1)*.

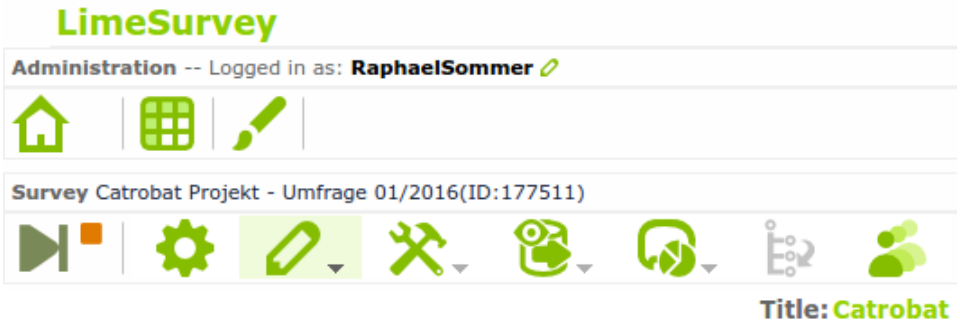

Figure 8.1.: LimeSurvey used for managing the survey (LimeSurvey Project Team / Carsten Schmitz, 2016)

Objectives

The main objective of this survey is to collect data about the Catrobat project. For example, likes, dislikes, image, teamwork, communication, personalities

and more. Data about satisfaction, motivation and demotivation is collected as well. There are various possibilities to use the data gathered by this survey. For example the Herzberg Motivational Theory is applied to the Catrobat project in the results chapter of this thesis. Further possible investigations can also be found in the results chapter or in the future work section. The results of the survey are analyzed and discussed in Chapter 9.

This survey collects data about Catrobat for a certain point in time. Thus this survey can be repeated to assess the evolution of Catrobat.

Sample

The survey started in February 2016 and was open until the end of March 2016. In this period of time all active Catrobat members at the Graz University of Technology were invited to participate in this survey. 118 students or alumni from Graz University of Technology received an invitation email. Translators and other external Catrobat members were not invited, because they are rarely contributing and are not as integrated as the Catrobat members at Graz university of Technology. 66 participants completed the entire survey. They represent 55.9% out of all 118 invited participants. 37 responses were only partially completed, they were not examined for the results of the survey. The remaining 15 invited participants have never started the survey. All results and charts were created with the 66 completed responses.

Structure

The survey consists of several parts. Every part tries to collect some specific data. In the following section these parts are described briefly.

Demographic: In this part, demographic information about Catrobat members was gathered. With this information it is later possible to analyze the results in various ways. For example it is possible to build sub samples out of the whole sample and compare them with other sub samples. Correlations between a certain sub sample and specific results are possible.

Ten Item Personality Inventory Questionnaire: This questionnaire is called TIPI. It results in a measure of the Big Five personality dimensions *extraversion, agreeableness, conscientiousness, emotional stability* and *openness* (Gosling, Rentfrow, and Jr., 2003). The dominance of each dimension can be compared to standardized norms. This questionnaire is popular in research. With this questionnaire it is possible to identify different types of individuals, who are participating in the Catrobat project. The TIPI results offer a basis to compare personality characteristics with typical software engineers' characteristics.

Motivation in Software Engineering: For collecting data concerning motivation in SE, motivators and demotivators from the systematic literature review update are used (A. Franca, Gouveia, et al., 2011). This part has two objectives. One objective is to collect general data about the most motivating factors. The second objective is to collect data concerning motivation and demotivation in the Catrobat project. With this knowledge potentials for improvement can be detected and strategies for improvements can be developed.

Job Satisfaction: This part is similar to the motivation one. Job satisfaction questions are adapted from Al-Rubaish et al., 2011. They created and published a job satisfaction questionnaire for academic organizations. This questionnaire has been adapted for the Catrobat survey. Topics like *working conditions, communication, supervision* and *teamwork* are examined within this satisfaction part. Again, it is interesting to investigate, which areas are satisfying and which are causing dissatisfaction.

Catrobat Specific Questions: The goal of this part is to collect concrete feedback about general attitudes, practices and tools used in the Catrobat project. As a result practices and tools can be examined on the basis of their popularity. There are also several open ended questions to give participants the chance to answer in their own words and thoughts. Results of these part are also very interesting for potential improvements in the Catrobat project.

Feedback: If participants have anything further to say, they can do it in this part. There is also the possibility to give feedback concerning the survey itself.

9. Results

In this chapter survey results are presented and interpreted. Most questions are visualized in form of charts. 66 Catrobat members filled out this survey. Therefore in most figures there are 66 responses visible.

Two different sections are used for presenting survey results. In Section 9.1 general survey results are presented. This thesis has a special focus on motivation in SE, thus a separate section is created for motivation results, see Section 9.2. Meaningful and frequently given comments are appended to questions' results.

Then results concerning motivation in the Catrobat project are further investigated and interpreted in Section 9.3. Findings of the survey and suggestions are elaborated on in Section 9.4.

9.1. Survey Results

In this section basic results of the survey are shown. In this section, subsections are used to group questions about the same topic together.

9.1.1. Demographic

What is your nationality?

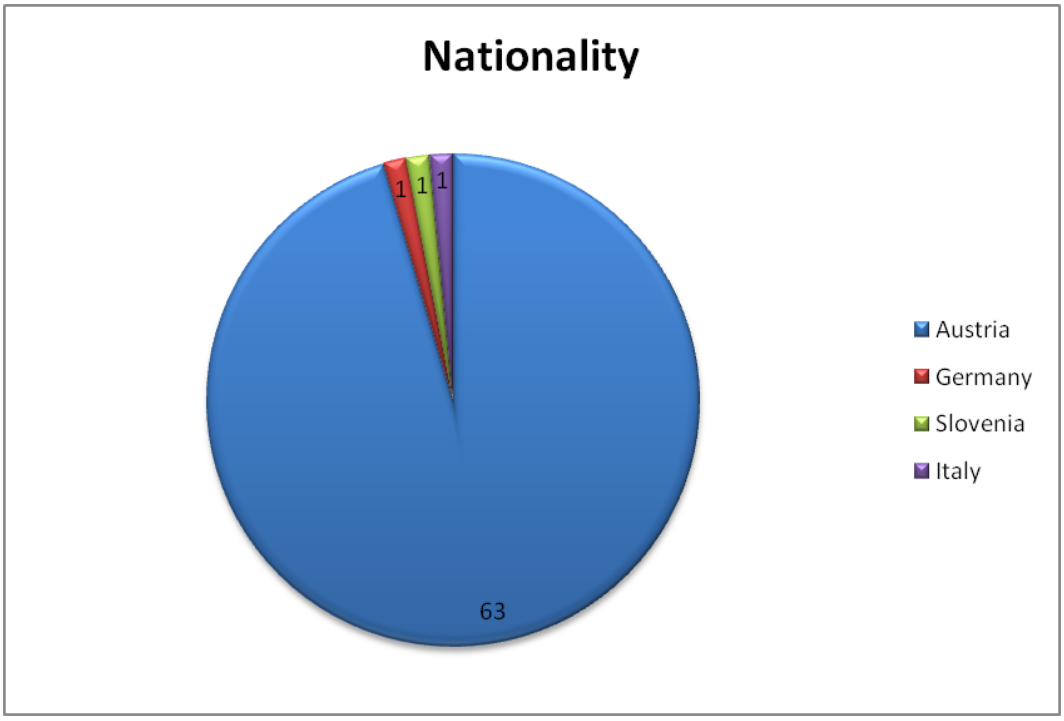

Figure 9.1.: What is your nationality?

Most people working on Catrobat are from *Austria*. Only three out of all 66 respondents are from another country.

How old are you?

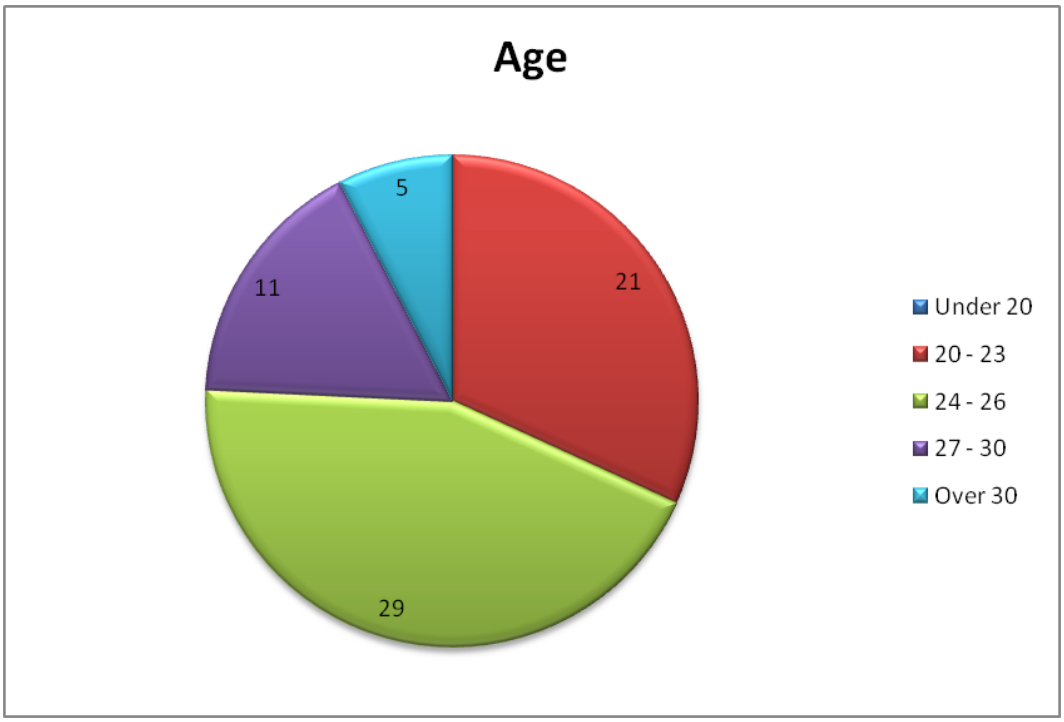

Figure 9.2.: How old are you?

No participant is *under 20*. Most participants are in the range from *24 - 26*. Five participants are *over 30*. They have a potential to share valuable experiences with Catrobat members. For example they have already worked in professional environments, thus they could improve professionalism in Catrobat.

What is your gender?

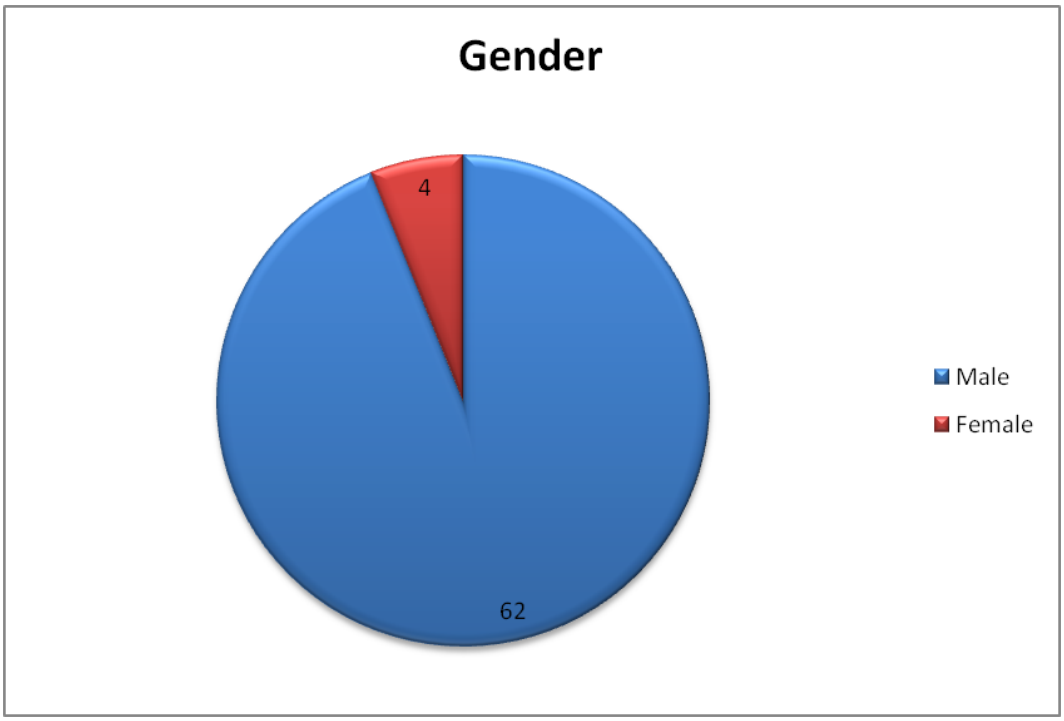

Figure 9.3.: What is your gender?

The gender in the Catrobat project has a typical ratio for a FOSS project. Only four participants are *female*. The remaining 62 participants are *male*, who represent 94%. This ratio is similar to the ration in typical FOSS projects. In comparison 25% of workers in IT are women and 2% of FOSS participants are women (Kuechler, Gilbertson, and Jensen, 2012).

What is your highest education?

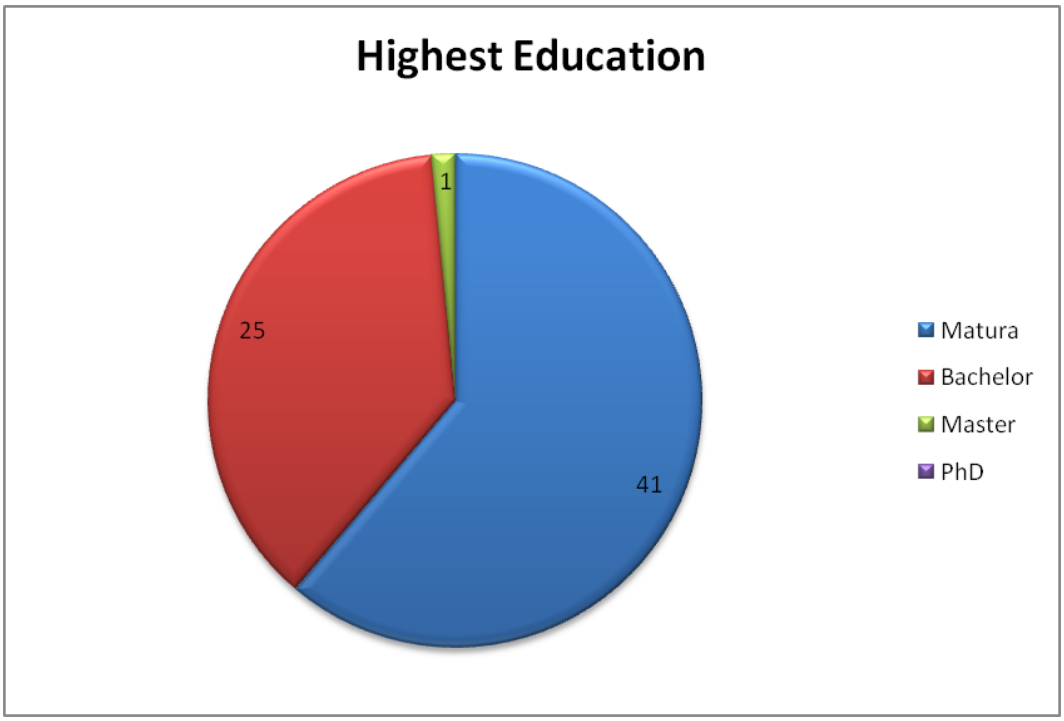

Figure 9.4.: What is your highest education?

Most participants have *Matura* as highest education. The *Matura* is the general qualification for university entrance in Austria. 25 participants have finished their bachelor's degree. One participant has already finished his master's degree. The master degree is the highest educational level of all 66 participants.

At which institution are you studying?

All 66 participants are studying or have studied at Graz University of Technology.

What is your field of study?

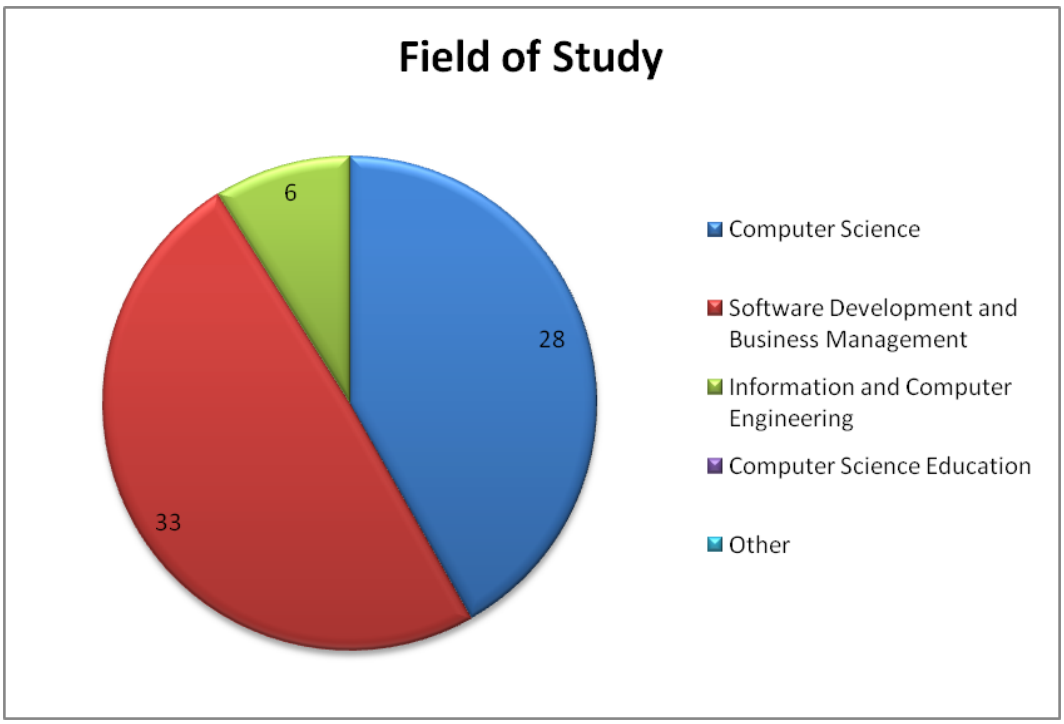

Figure 9.5.: What is your field of study?

Three fields of study are dominant. The most attended study by Catrobat members is *Software Development and Business Management*. 33 participants are studying *Software Development and Business Management*. The other half of the 66 are *Computer Science* and *Information and Computer Engineering* students.

How much work experience do you have in the IT industry ?

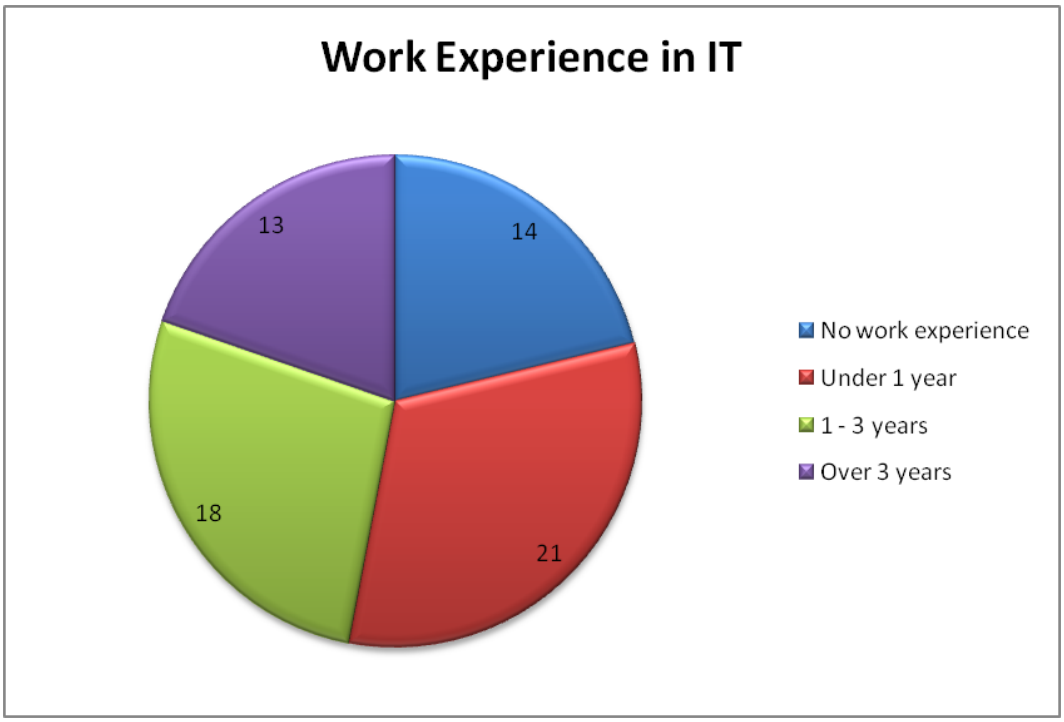

Figure 9.6.: How much work experience do you have in the IT industry?

Results concerning *work experience* are very interesting. All possible answers have a similar number of votes. 14 participants have *no work experience* in the IT industry. 13 participants have work experience *over 3 years*. Remaining participants have work experience *under 1 year* or *1 - 3 years*. Many participants already have experience in the IT industry. This is a interesting information for the Catrobat project. These Catrobat members can share their experiences in the project.

9.1.2. Ten Item Personality Inventory Questionnaire

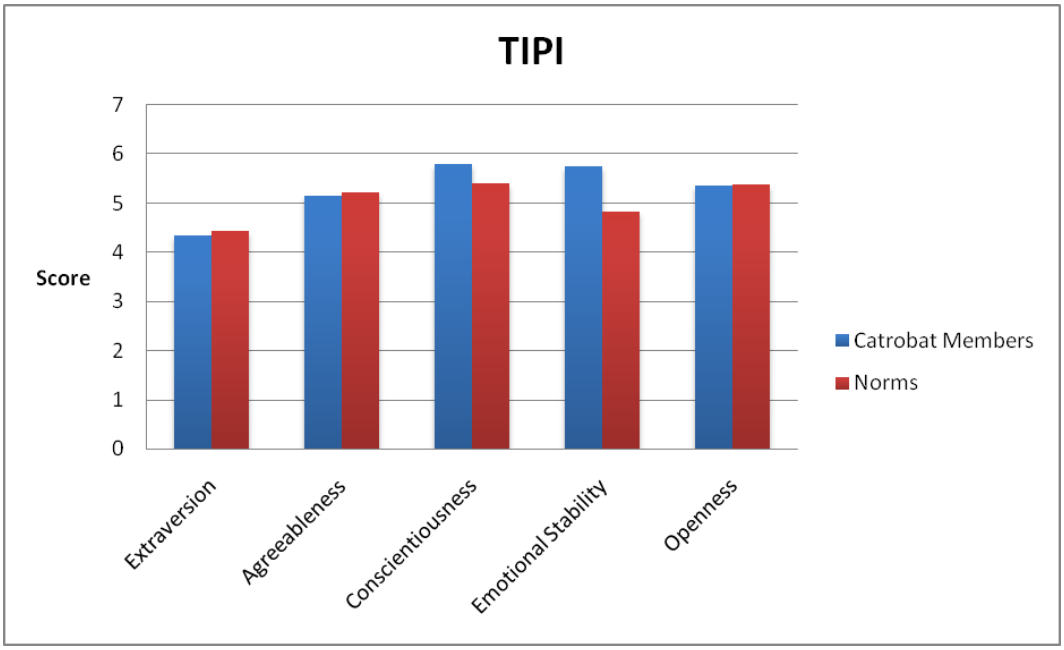

Figure 9.7.: Ten Item Personality Inventory questionnaire

The five personality dimensions have already been described in Section 6.1.5. The TIPI results, in form of the Big Five personality dimensions, are compared with norms from Gosling, Rentfrow, and Jr., 2003. These norms are published on the website of the TIPI questionnaire and represent average values for many people. Three dimensions in the Catrobat project are a very similar to the norms. These very similar dimensions are *extraversion*, *agreeableness* and *openness*.

Extraversion: Extraversion is a little bit lower than the norms. Typical characteristics for lower values in this dimension are *stay in the background, have a little to say* or *being reserved*.

Agreeableness: Agreeableness is a little bit lower than the norms. Characteristics for low values in this personality dimension are *feel little concern for others, insult others* or *not interested in others*.

Conscientiousness: Conscientiousness is higher than the norms. Examples for higher values in this personality dimension are *always prepared, pay attention to details, like order* or *follow a schedule*.

Emotional Stability: Emotional Stability is much higher than the norms. Catrobat members seem to be *relaxed, unemotional* or *comfortable*.

Openness: Openness is a little bit lower than the norms. Examples for low values of this personality dimension are *have a poor imagination, uncreative, uncurious* or *conventional*.

9.1.3. Agile

How well do you know the following software development methods?

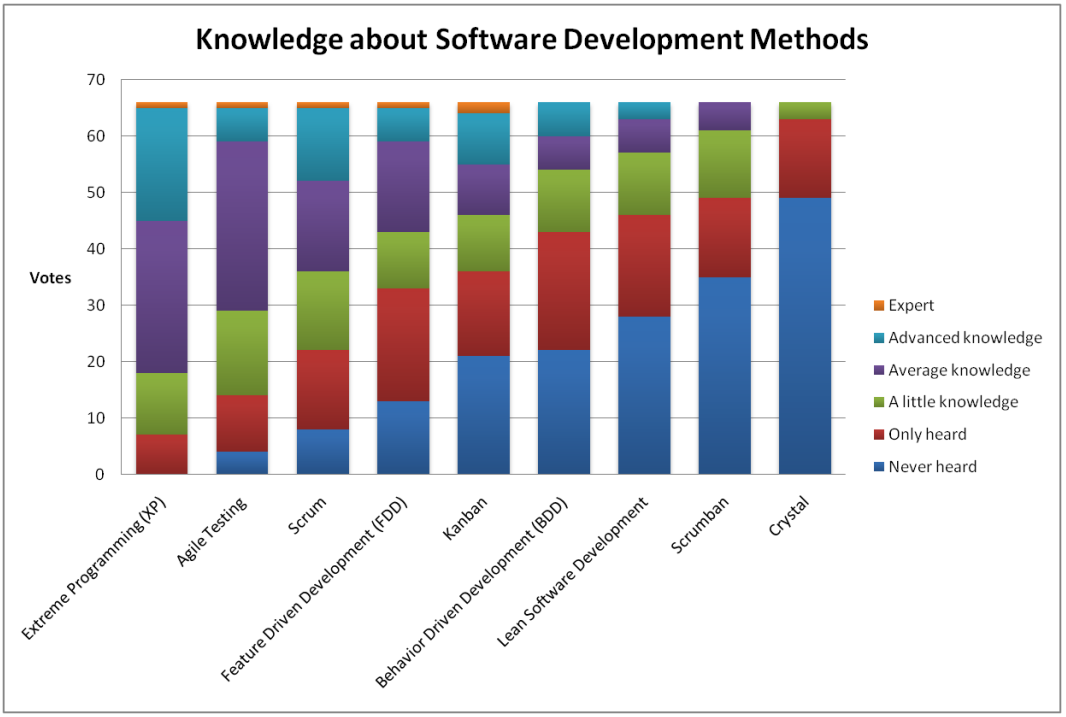

Figure 9.8.: How well do you know the following software development methods?

In Figure 9.8 software development methods are sorted by participants' knowledge about these methods..

The most known software development method is *XP*. *XP* is the subject of a course at Graz University of Technology, this can be the reason for the large popularity of *XP*. *Agile Testing* and *Scrum* are the second and the third most known software development methods. *Kanban* is in the middle of this ranking.

The least known software development methods are *Scrumban* and *Crystal*.

How much experience do you have with the following software development methods?

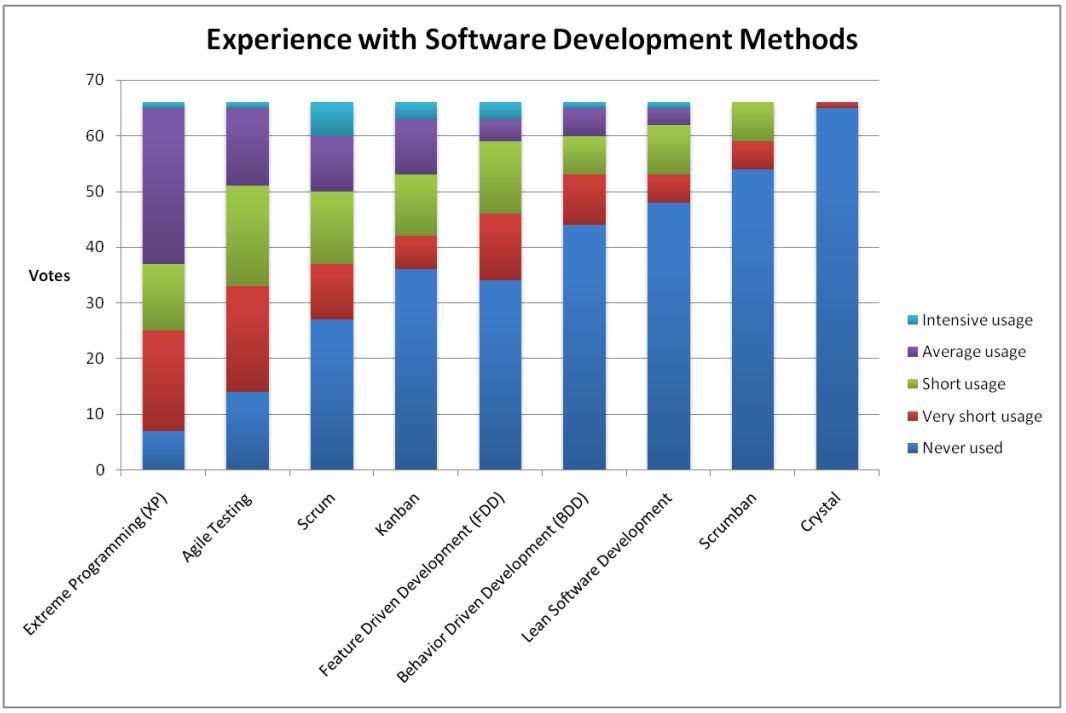

Figure 9.9.: How much experience do you have with the following software development methods?

Software development methods in Figure 9.9 are sorted by participants' experience with them. These results are very similar to the previous question concerning knowledge about software development methods. Participants have the most knowledge in software development methods, which they have used and have experience with.

The software development method with the highest values for participants' experience is *XP*. *Agile Testing* and *Scrum* are the second and the third software development methods with the most participants' experience.

Participants have the least experience with the software development methods *Scrumban* and *Crystal*.

9.1.4. Catrobat General

In this subsection results of questions concerning Catrobat in general are presented. For example some question topics are *role, team size, experience in Catrobat* and open ended questions concerning Catrobat.

What is your current role in Catrobat?

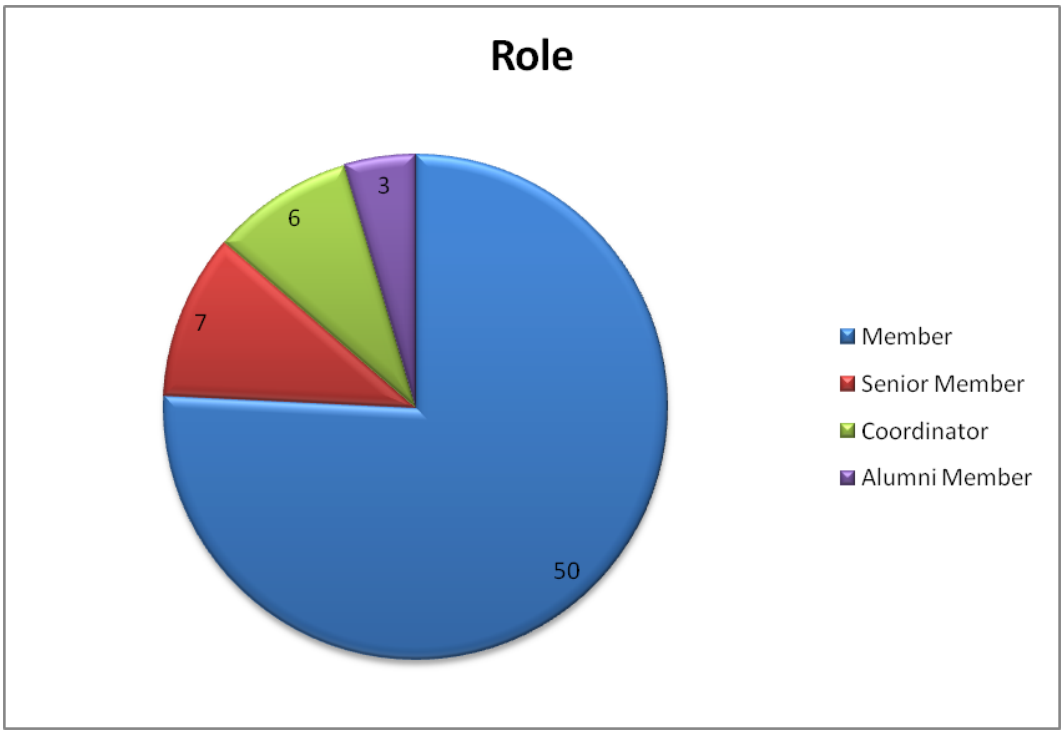

Figure 9.10.: What is your current role in Catrobat?

Most participants, who filled out this survey, are *active members*. Seven *senior members* and six *coordinators* have also participated in this survey. Three participants are *alumni senior members*. A further description of these roles and the whole Catrobat community can be found in Section 5.3.

What is the size of your team?

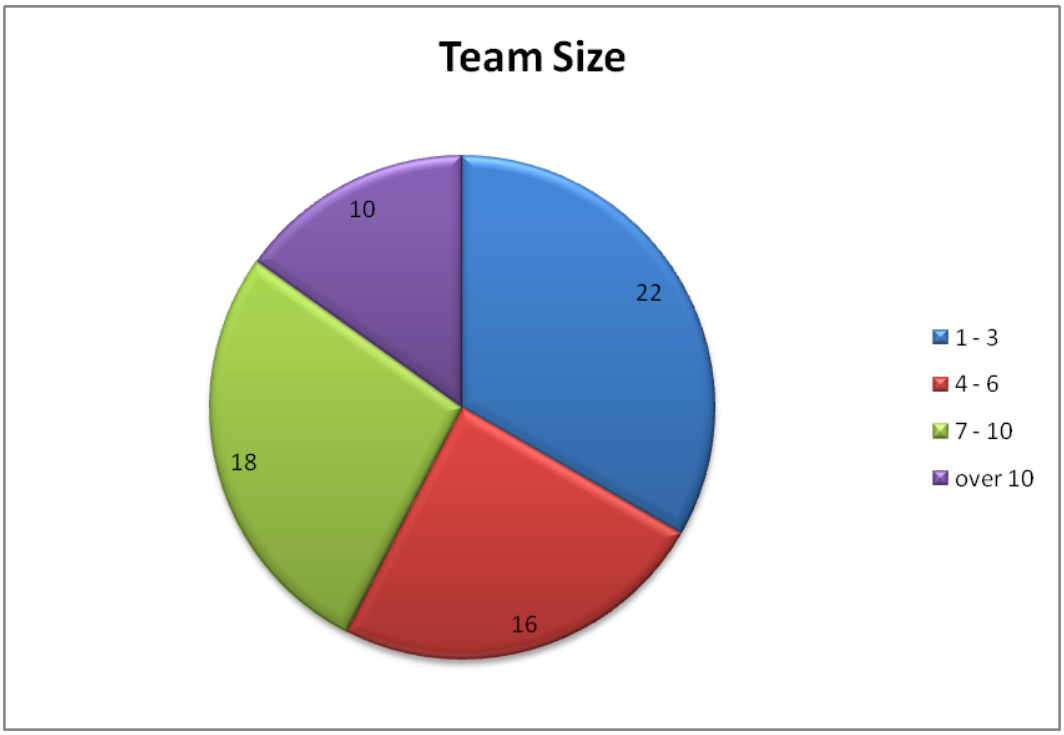

Figure 9.11.: What is the size of your team?

There are many teams under the Catrobat umbrella project. They strongly differ in size. There are several small teams consisting of *1-3* members. One third of all participants is in such a small team. The second largest group of participants are in teams with *7-10* participants. 10 participants are in a team with *over 10* members.

When have you joined Catrobat?

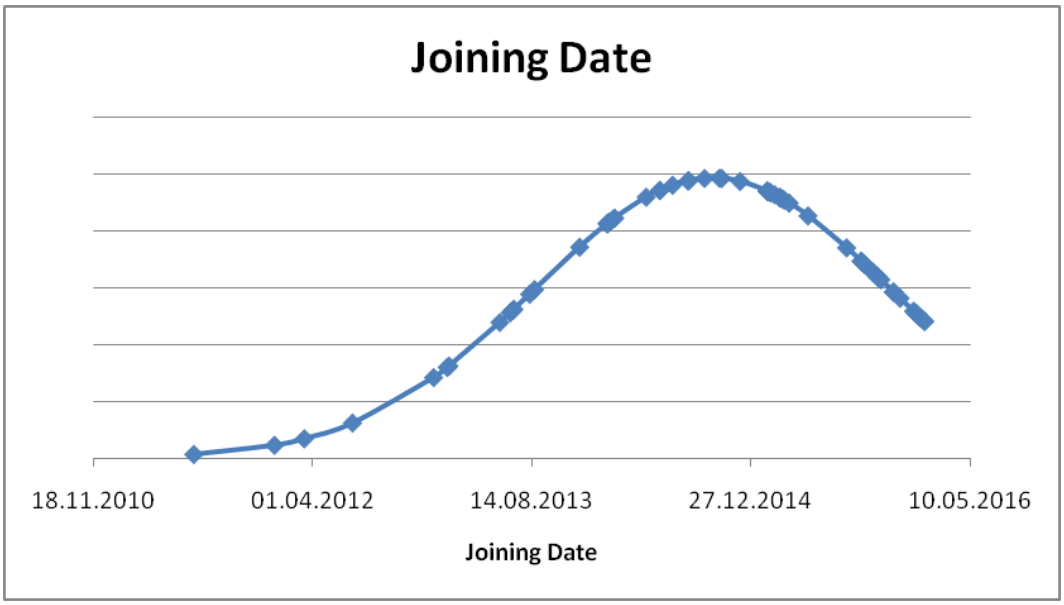

Figure 9.12.: Normal distribution of *When have you joined Catrobat?*

Some members joined Catrobat a long time ago. The participant, who joined the longest time ago, joined Catrobat on *04.07.2011*. This participant has been active in the Catrobat community for around five years.

As the median indicates, half of all survey participants have joined Catrobat before *02.02.2015*. The other half has joined Catrobat after February 2015. The range is *1667 days*. This means the period of time between the first participant and the last participant, who has joined Catrobat, is around *4.5 years*.

Maximum	26.01.2016	Standard deviation	404 days	Quartile 1	03.02.2014
Minimum	04.07.2011	Mode	01.02.2015	Median	02.02.2015
Range	1667 days	Mean	29.09.2014	Quartile 3	09.09.2015

Table 9.1.: Catrobat joining statistics

Maximum	1600	Standard deviation	410	Quartile 1	100
Minimum	5	Mode	80	Median	255
Range	1595	Mean	411	Quartile 3	547

Table 9.2.: Worked hours on Catrobat statistics

The mean is *29.09.2014*, this indicates an average participant has been contributing to Catrobat for around 1.5 years.

How many hours have you worked on Catrobat yet?

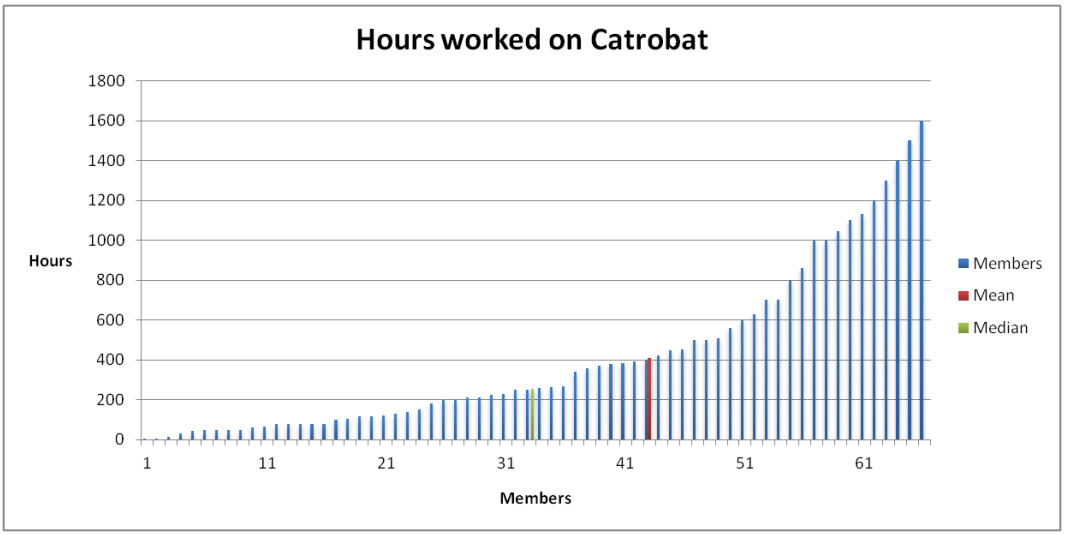

Figure 9.13.: How many hours have you worked on Catrobat yet?

The participant who has spent the most hours on Catrobat has already worked *1600 hours* on Catrobat. In contrast, one participant has only spent *five hours* on Catrobat.

As the median indicates, half of all participants have already worked on Catrobat for *255 hours* or more. The other half has spent less than *255 hours* working on the project.

The mean is *411 hours*. This indicates an average contribution duration of *411 hours*.

What is your current attitude in Catrobat?

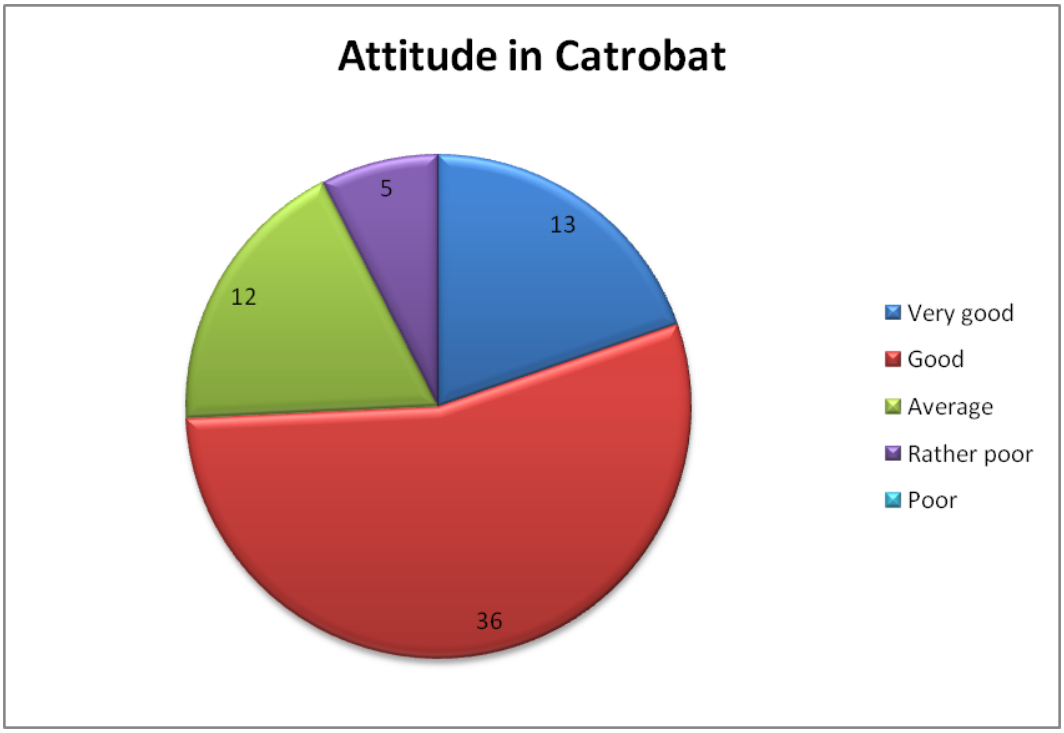

Figure 9.14.: What is your current attitude in Catrobat?

The result of this question is positive. Around three quarters voted *very good* or *good*. Only a few participants voted *average* or *rather poor*. No one voted his current attitude as *poor*.

Comments: Several participants like the idea behind Catrobat and being a part of it. The huge number of users is impressive and motivates Catrobat members to invest time in the Catrobat project.

There are also some negative comments. The non productive attitude of other members slows progress down. Many comments indicate missing

motivation and time during their participation. They recognized even own time management issues as well.

How do you think is the overall atmosphere in Catrobat?

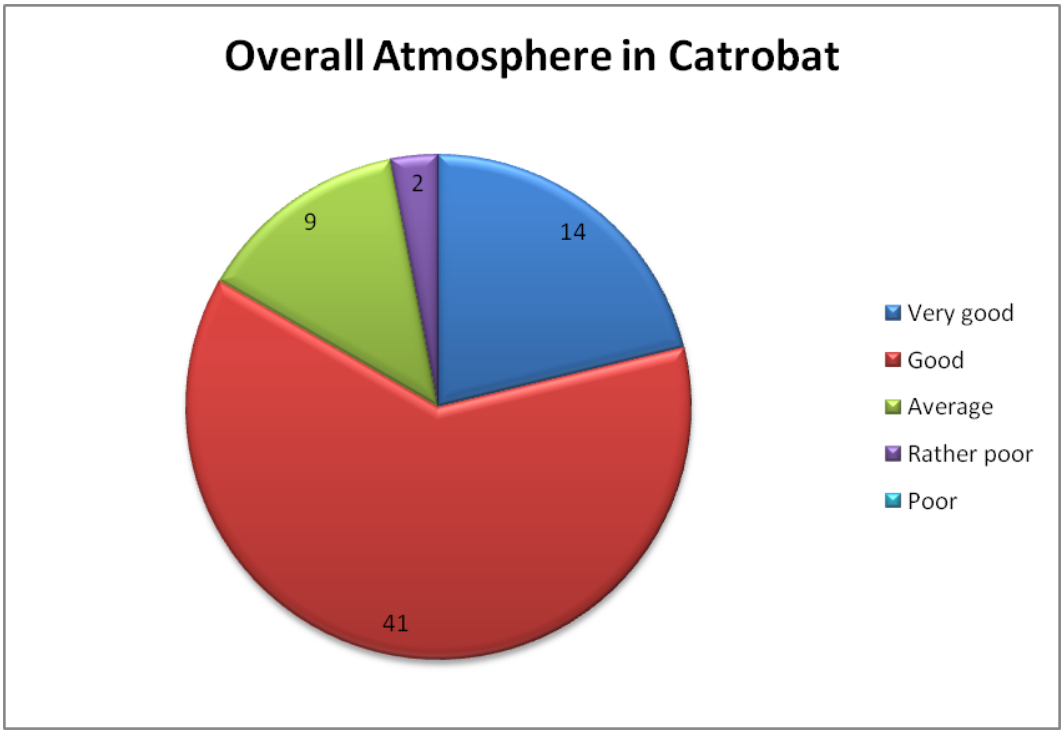

Figure 9.15.: How do you think is the overall atmosphere in Catrobat?

This result is very similar to the previous question's result, even a little bit more positive. Only nine participants voted *average* and two voted *rather poor*. All other participants voted the atmosphere in Catrobat as *very good* or *good*.

Comments: Most comments are positive too. They describe a friendly and cooperative atmosphere within the Catrobat project. Some participants describe again negative attitudes and missing progress.

What image has Catrobat in your colleagues' opinion?

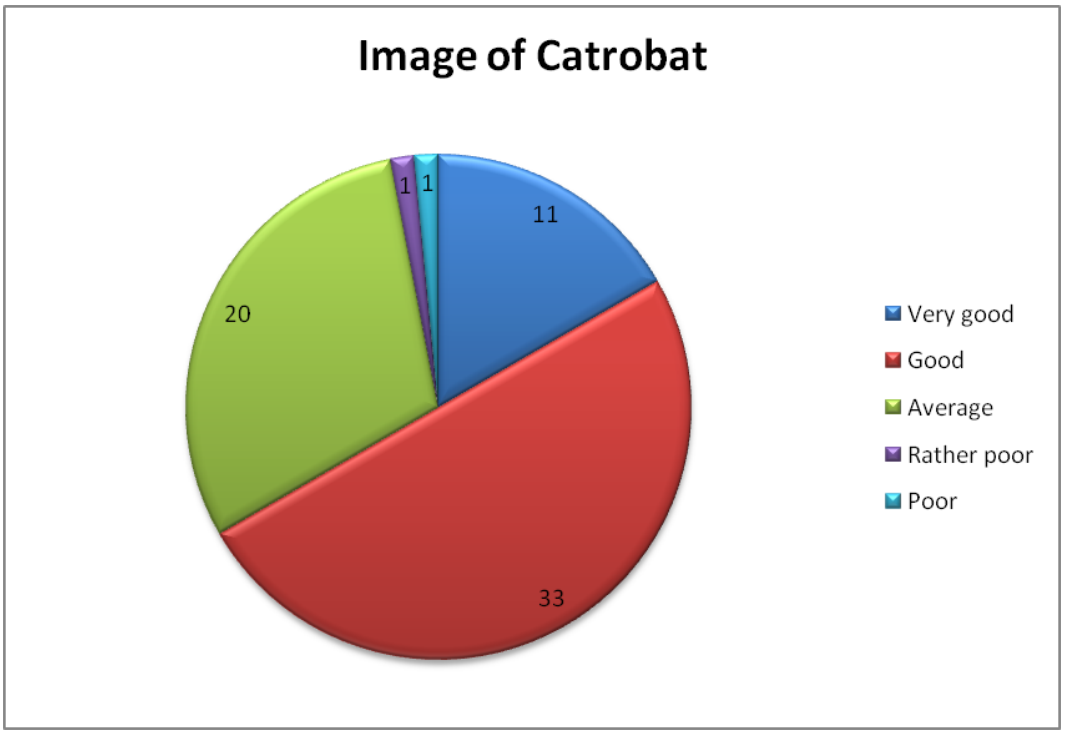

Figure 9.16.: What image has Catrobat in your colleagues' opinion?

Answers concerning this question tend to be positive. Several participants voted *very good* or *good*. In comparison to previous results *average* has a higher frequency. *Average* is voted by 20 participants.

Comments: Several comments report a lack of knowledge about Catrobat. Catrobat is not known by many colleagues at all. In this case there is still potential to promote Catrobat, especially at the university.

Other participants report a good image in their colleagues' opinion. Some participants report a misunderstanding of Catrobat's purpose and audience.

What do you like most about Catrobat?

All 66 participants answered this question. Comments are very diverse. However, some aspects are identified by several participants.

Comments: Good teamwork and a positive working atmosphere are identified as positive in Catrobat. Another aspect, which is liked by participants in Catrobat, is freedom. They can act autonomously concerning time management, field of interest and the work itself.

Learning and collecting experiences are also seen as very positive in Catrobat. Especially, gaining experience in a working environment, which is similar to real word environments in companies, is liked. Participants like to collect such experiences and to learn about software development methods and their practices.

Other participants like the idea behind the Catrobat project and being part of such a large FOSS project. The project infrastructure is also highlighted as very positive in Catrobat.

What do you dislike most about Catrobat?

A lot of different answers are given to this question. Again, this was an open ended question, which was obligatory. A lot of different aspects are handled in participants' responses. The most frequently reported aspects are listed below.

Comments: Communication is not working quite well in Catrobat. Many participants dislike information flow and communication between different teams. Internet Relay Chat (IRC) is also identified as an impractical communication tool.

Missing motivation and discipline of other Catrobat members are seen as problematic. Sometimes Catrobat members do not invest enough energy to be productive and to push the overall progress of the project.

The entrance into the project is seen as difficult by several participants. The complex codebase in combination with sometimes poor code quality, makes

it very hard for newcomers to get started. Some participants want more documentation.

Some participants would also like to get more information about the whole vision and objectives of the Catrobat project.

Other participants see problems in the infrastructure, especially with the test system.

What can be improved to make Catrobat more attractive to other students?

Comments: Catrobat is not known by many people at Graz University of Technology. This is reported by several students. Participants suggest to do more promotion of the Catrobat project at university. A few possibilities for promotion are also suggested by them. For example more promotion in programming lectures would be a possibility. Therefore fascinating demo projects would be helpful. Especially, demo projects with hardware like Arduino or drone would be impressive. Some participants also like the idea to build more lectures around the Catrobat project.

Other participants suggest to invest more into the onboarding process for newcomers. For example more tutorials at the beginning would be helpful. Speakers corners to spread knowledge about different topics would also support this approach. In addition, more documentation and even comments in source code would be helpful, according to the opinion of one participant.

Communication and information exchange is also a topic, which is seen as problematic by several participants. Some participants do not have information surrounding the project. For example they do not know about events and cooperations with other organizations.

A few participants suggest to use Scrum and its practices. Especially, for deadlines Scrum would be a helpful approach. One participant thinks, that a release or information timeline for all projects under the Catrobat umbrella project would be a good idea for information propagation.

Others would like to see coordinators, who demand more from their members. In their opinion coordinators should control and encourage members in their team to be more productive. Others would find it nice to get more feedback from their coordinator and even to make objective agreements.

Two participants would like the possibility to earn money during their participation in the Catrobat project.

Catrobat General Satisfaction

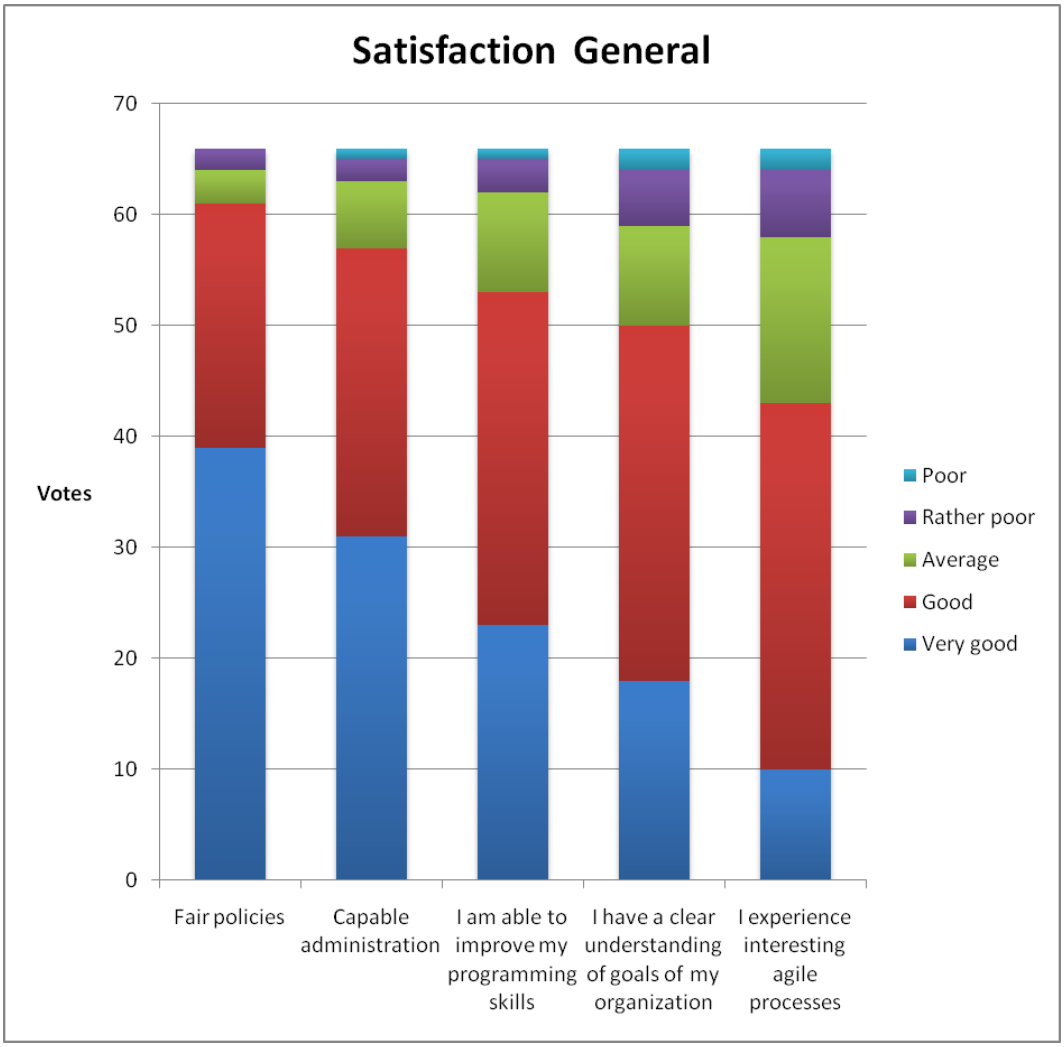

Figure 9.17.: Catrobat general satisfaction

These job satisfaction factors are sorted by their sum of Likert scale scores. Good votings like *very good* and *good* are dominating all factors. The best rated factor is *fair policies*. The factor *I experience interesting agile processes* has the poorest rating in this category.

9.1.5. Communication

Communication is a social success factor in agile projects (Van Kelle et al., 2015). Poor communication can be a reason for project failure. Especially, for management and coordinators, it is important to communicate efficiently. Results concerning communication in the Catrobat project are presented in this subsection.

How would you rate communication and information exchange in Catrobat?

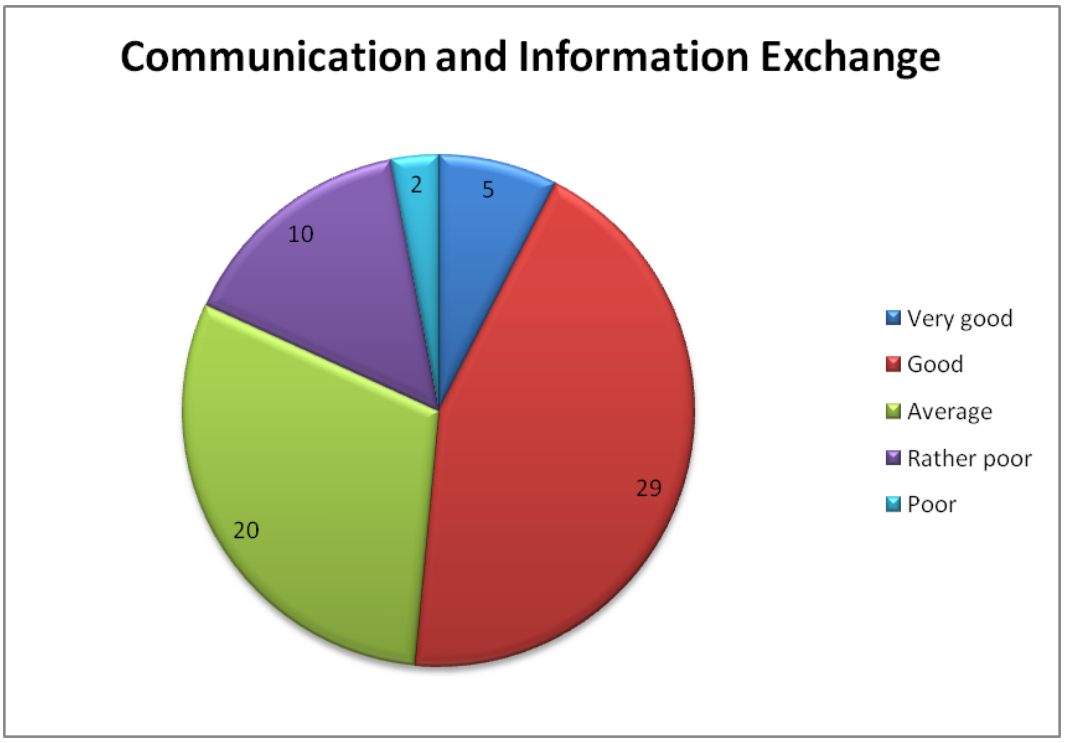

Figure 9.18.: How would you rate communication and information exchange in Catrobat?

Answers to this question indicate a mixed perception of communication in the Catrobat project. Nearly half of all participants rate communication

and information exchange as *middle*, *rather poor* or *poor*. Communication with other teams, IRC or communication inside teams are handled in own questions. Answers to these topics are presented in detail in other questions' results.

Comments: There are several communication methods and tools identified in Catrobat. One identified possibility is communication via pull requests. One participant sees communication between teams as primarily taking place via pull requests.

Another comment highlights the importance of coordinators. If teams have coordinators with good communication skills, communication with other teams works well. In contrast, if coordinators have less communication skills, communication with other teams is rather poor. A similar comment describes communication between coordinators as often going well, but communication between members of different teams as typically poor.

How would you rate IRC as communication tool in Catrobat?

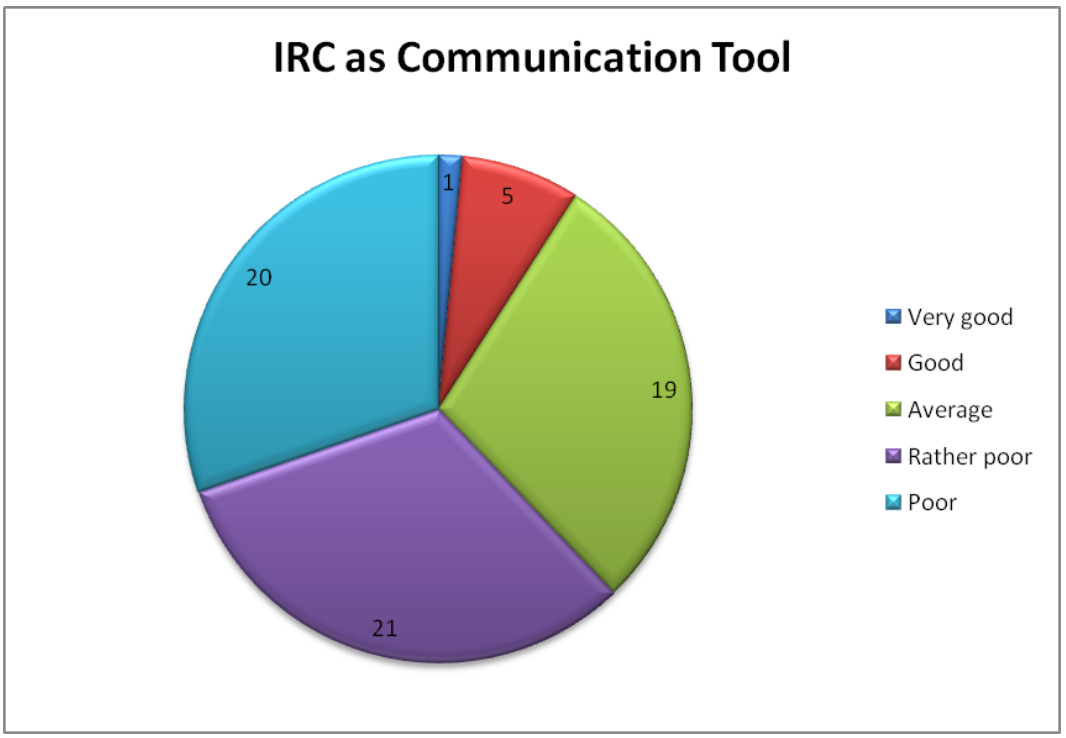

Figure 9.19.: How would you rate IRC as communication tool in Catrobat?

IRC was introduced to provide the whole Catrobat project with a primarily communication tool. At the moment IRC is used only infrequently. Answers to this question indicate a strong dissatisfaction with IRC. Only one participant rates IRC as *very good* and five rate it as *good*. Most participants see issues and disadvantages using IRC as communication tool.

Comments: Some participants have problems with setting up IRC and also while using it. Other participants think, that IRC is out of date. As a result several participants are thinking, that other communication tools like Skype, Slack or HipChat would be better communication tools, than IRC .

A few participants rate IRC as okay, but the irregular usage of it is demotivating. Missing discipline can be the reason for such a poor usage of IRC, is the opinion of one participant. In this case a switch from IRC to another

communication tool would not have a significant effect, if discipline would be still missing.

How would you rate communication in your team?

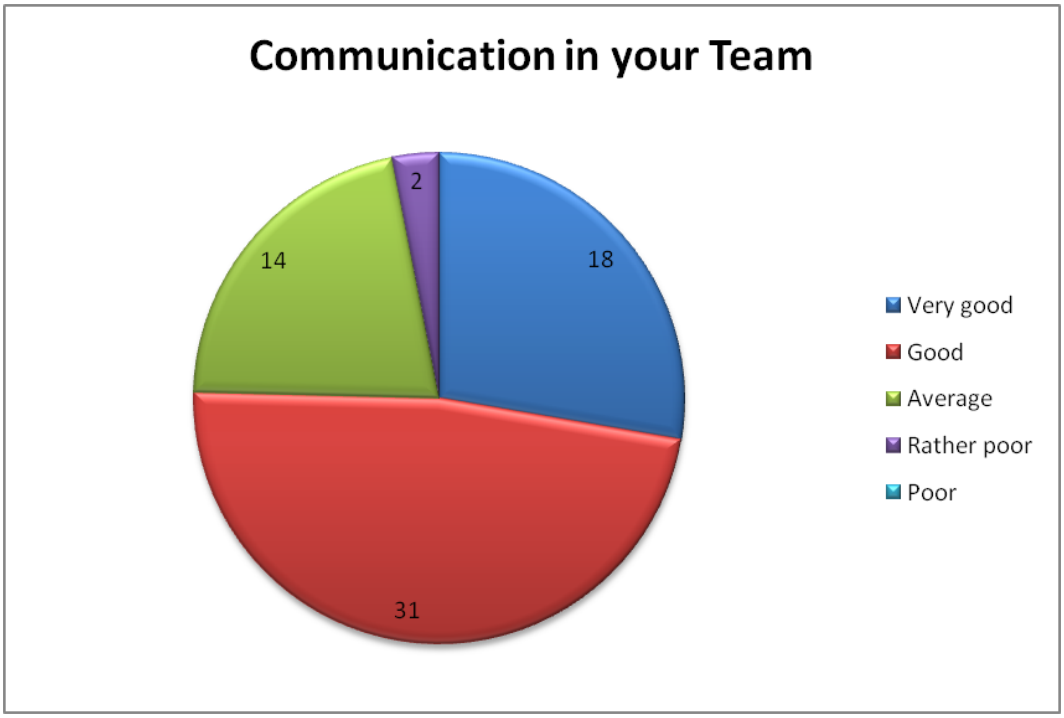

Figure 9.20.: How would you rate communication in your team?

Communication inside teams is mostly seen as positive. Only 14 participants rated it as *average* and two rated it as *rather poor*. In comparison to the overall communication and information exchange results, this is a relatively positive result.

Comments: Two aspects are reported within this question. One aspect is the usage of other communication tools like Whatsapp or Skype. Some teams are using these communication tools instead of IRC. Catrobat members using these tools report a good communication in their teams.

The second aspect is the inactivity of team members. This inactivity can be caused by study stress. If several members are inactive, there is no progress to discuss with other team members.

How would you rate regular team meetings?

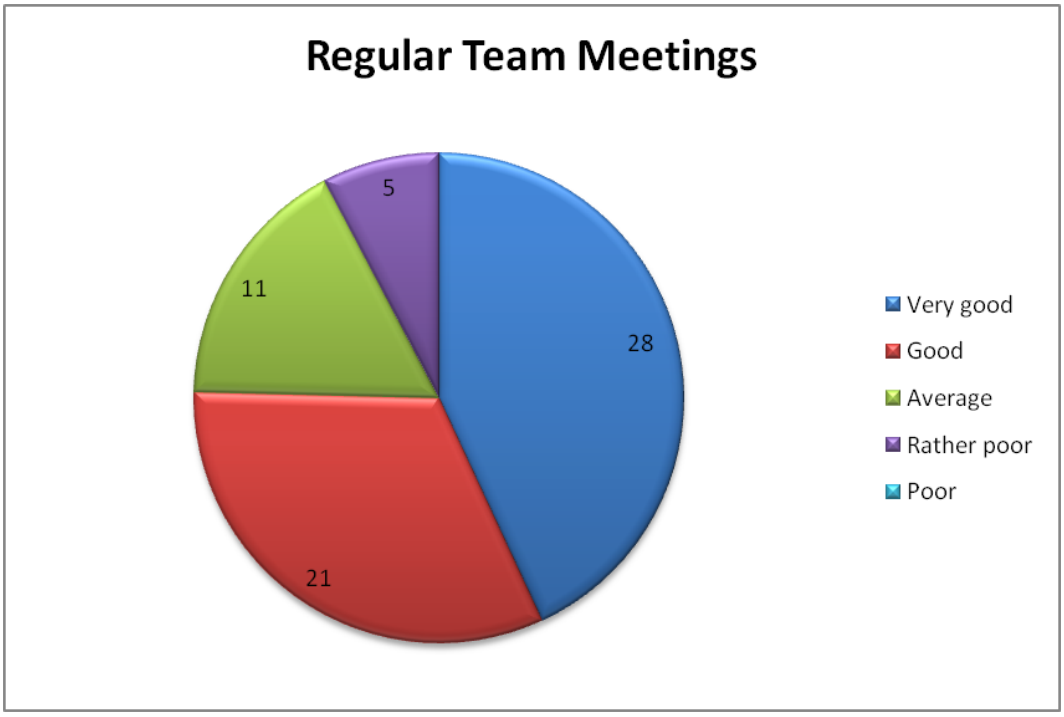

Figure 9.21.: How would you rate regular team meetings?

Regular team meetings are rated as very positive. 28 participants rated it as *very good*. Most comments are positive too.

Comments: Participants describe an essential information exchange during team meetings. Important topics are handled within these meetings. The moderation and execution of team meetings is done by coordinators. Coordinators are doing this moderation very well.

Some Catrobat members are not regularly attending team meetings. This can be a consequence of small team sizes or inactive team members. One

participant reports a too long duration of team meetings. In this case detailed discussions should be had later.

What should be improved concerning regular team meetings?

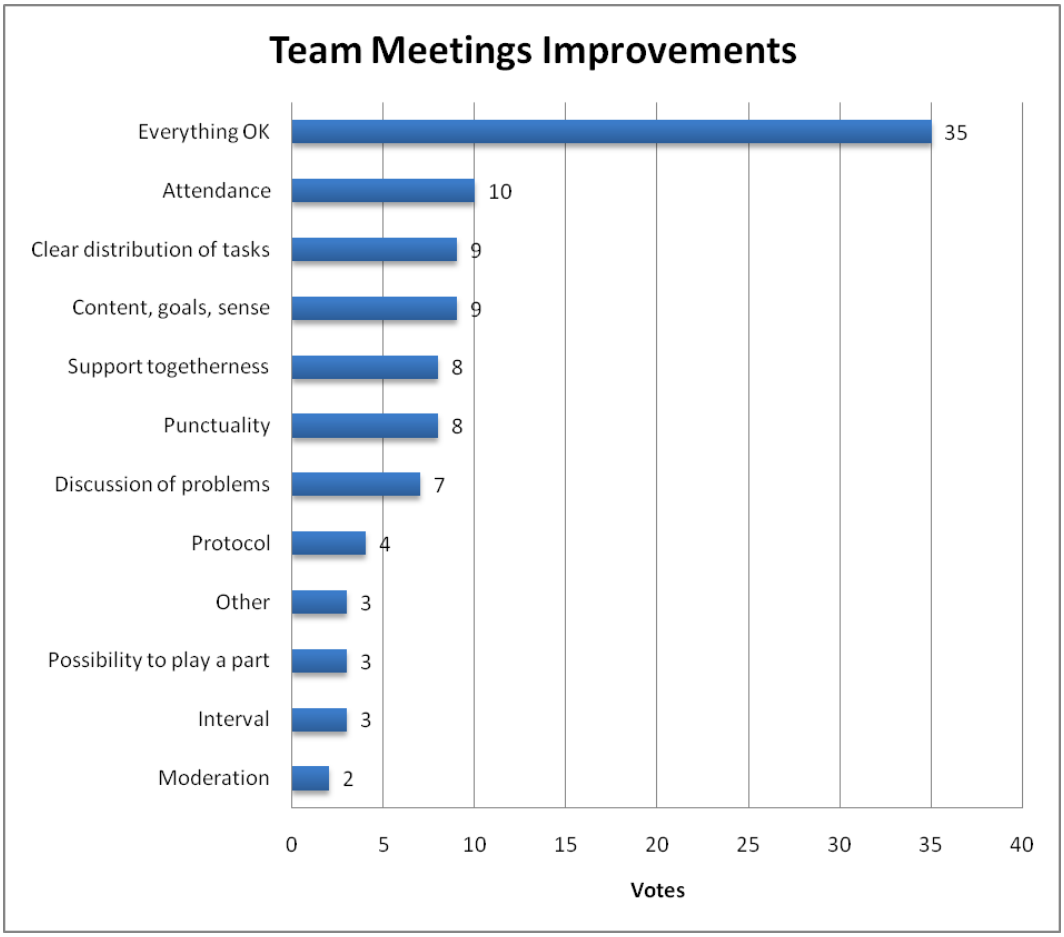

Figure 9.22.: What should be improved concerning regular team meetings?

More than half of all 66 participants voted *everything ok*. This confirms the positive result of the previous question.

Some factors for improvement have also a couple of votes. With votes from

seven like *discussion of problems* up to ten like *attendance*, there is still a potential to think about possible improvements.

Other factors like *moderation* or *interval* have only a few votes and do not indicate a need for action.

How would you rate BiWeCo meetings?

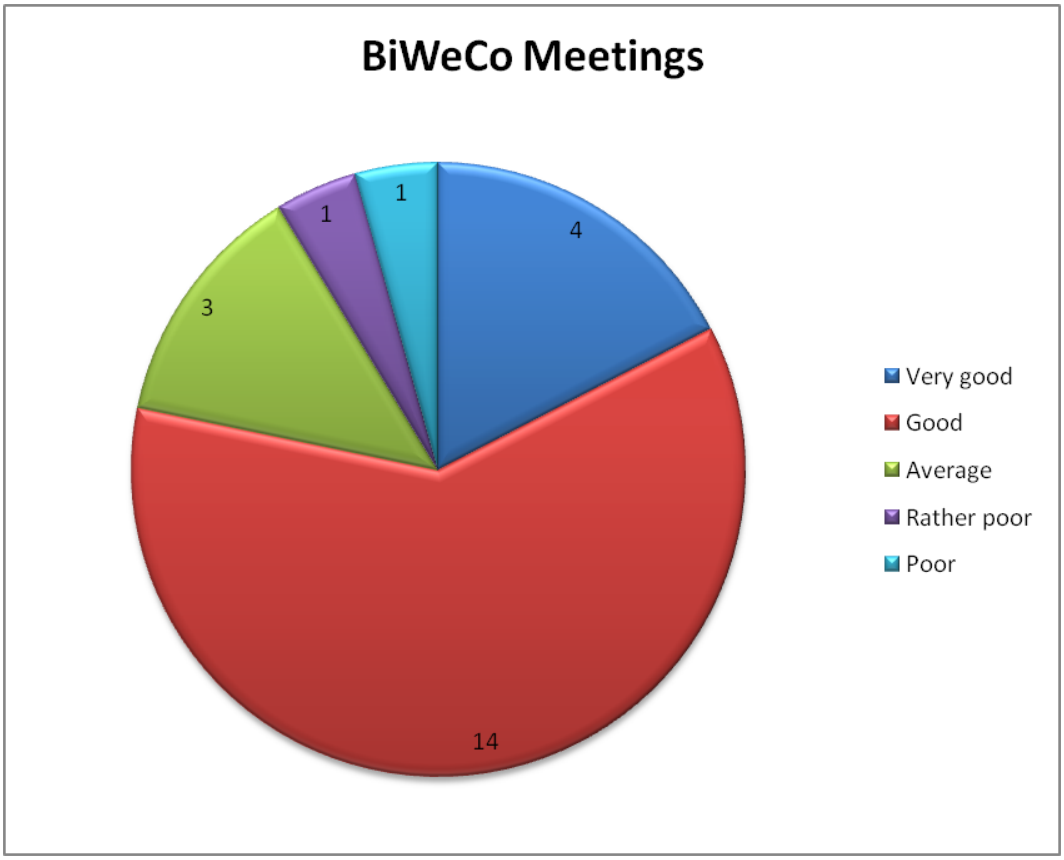

Figure 9.23.: How would you rate BiWeCo meetings?

This question is answered by 23 participants, because they attended a BiWeCo meeting at least once. These meetings are also seen as very positive. Most participants rated them as *good*.

Comments: Participants see BiWeCo meetings as a good possibility to get an overview of the progress in the whole Catrobat project. Progress of all sub-projects is reported briefly. Interactions and information exchange with other teams is a further advantage of this meeting. However, one negative comment still indicates lacking information exchange between teams in BiWeCo meetings.

What should be improved concerning BiWeCo meetings?

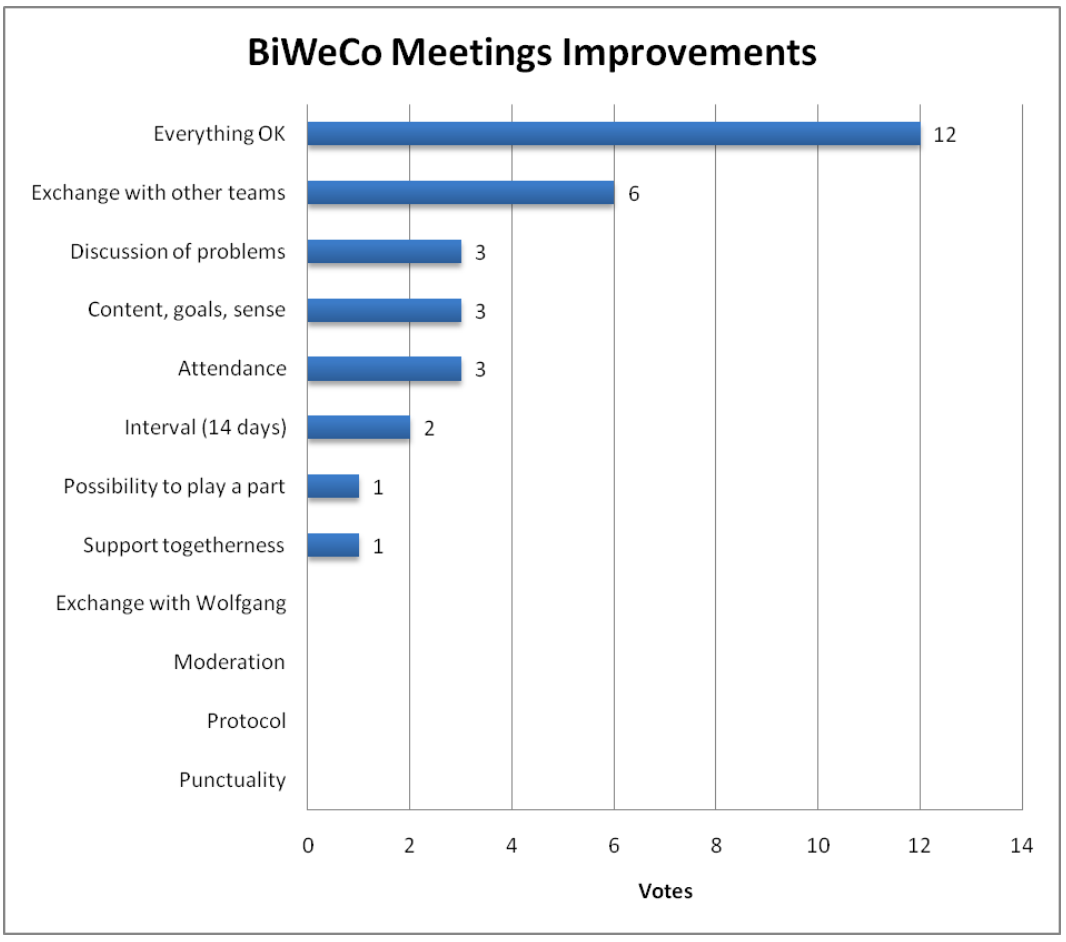

Figure 9.24.: What should be improved concerning BiWeCo meetings?

More than half of all 23 participants voted *everything ok* concerning BiWeCo meetings. Some improvement factors even have zero votes, for example *moderation* or *protocol*.

The improvement factor with the most votes is *exchange with other teams*. As already reported a lack of communication between teams in Catrobat exists.

How would you rate Planning Games?

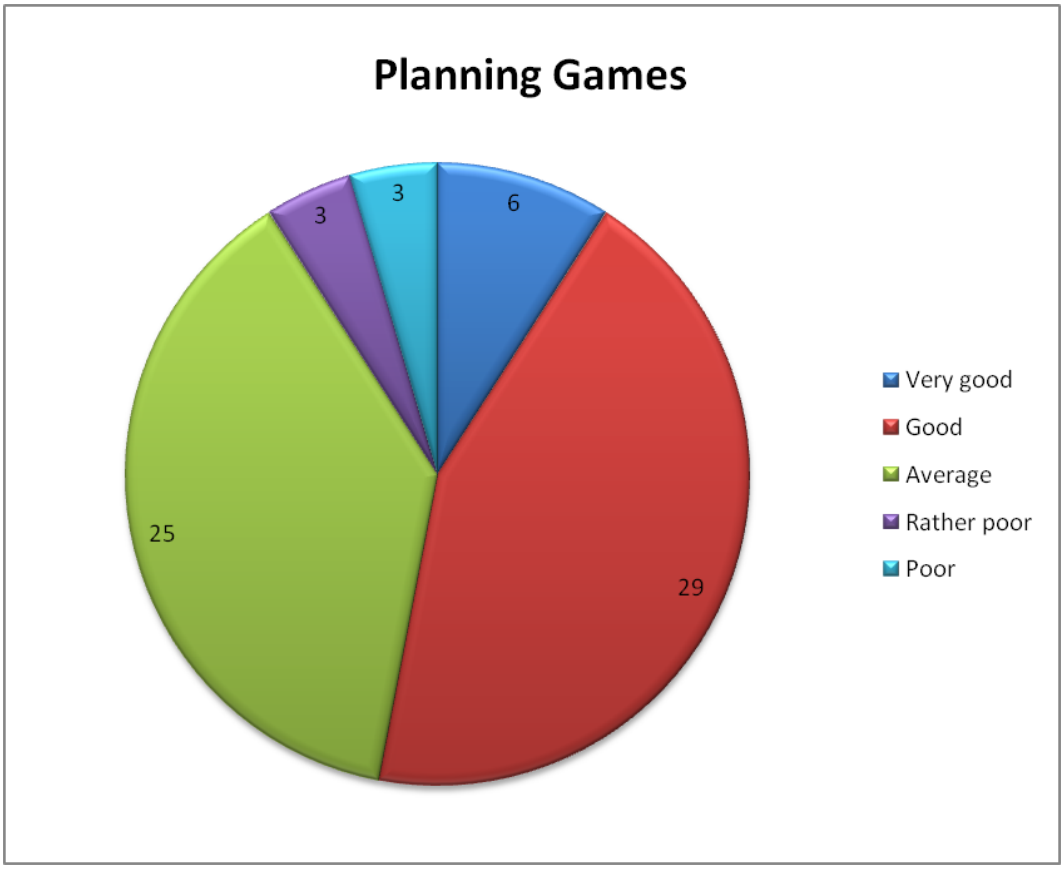

Figure 9.25.: How would you rate Planning Games?

More than half of all participants rated Planning Games as *very good* or *good*.

Comments: Several teams are not doing Planning Games or only do them irregularly. Some participants have not attended to a Planning Game yet. In this case the result of this question should be considered carefully.

Communication Satisfaction

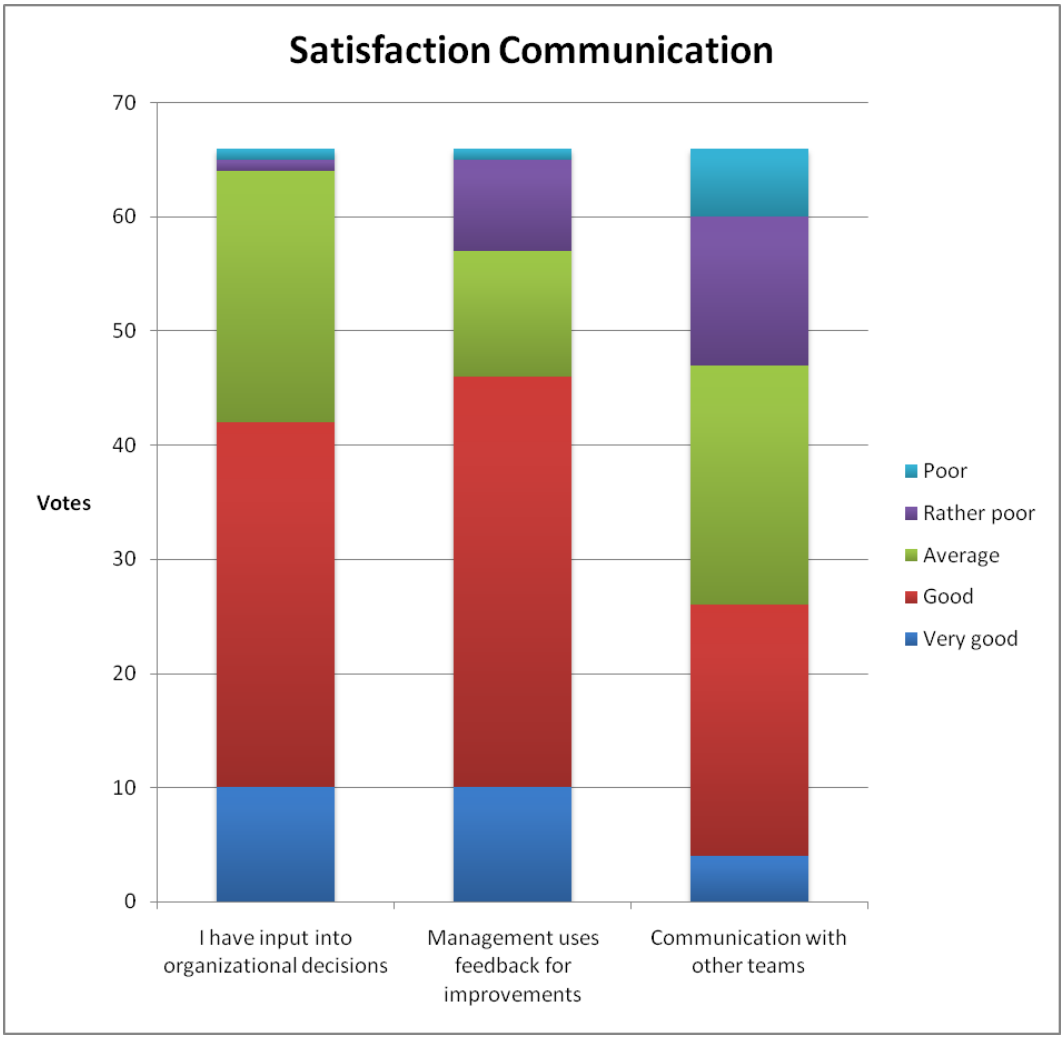

Figure 9.26.: Communication satisfaction

In this satisfaction question *communication with other teams* attracts attention. As already reported in previous questions this is identified as an issue. The other two satisfaction factors also have rather poor ratings.

9.1.6. Teamwork and Work

SE is often done in teams. In Catrobat teamwork is an important aspect, because there are several teams under the Catrobat umbrella project. In this section results concerning teamwork and work itself are presented.

How would you rate teamwork in your team?

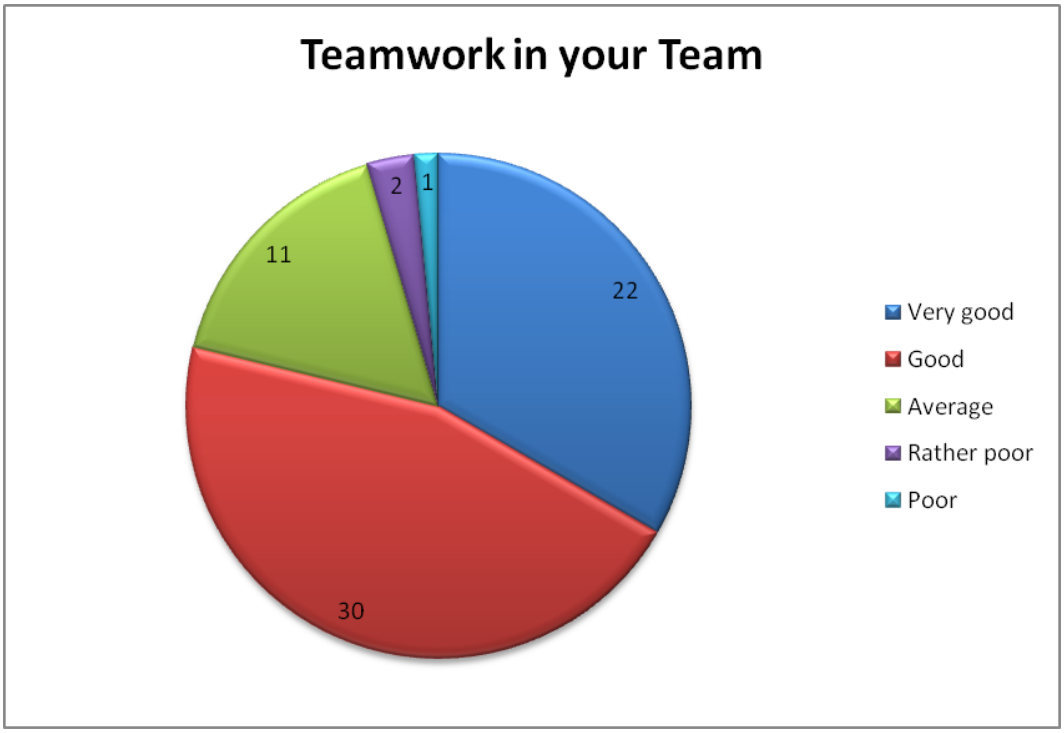

Figure 9.27.: How would you rate teamwork in your team?

Most participants voted teamwork in teams as positive. *Very good* is voted by 22 and *good* by 30 participants.

Comments: However, some comments also indicate negative teamwork. They describe teamwork as: *not existing*, *it has already been better* or *it depends strongly on qualification and motivation of other members*.

How would you rate teamwork in the whole Catrobat project?

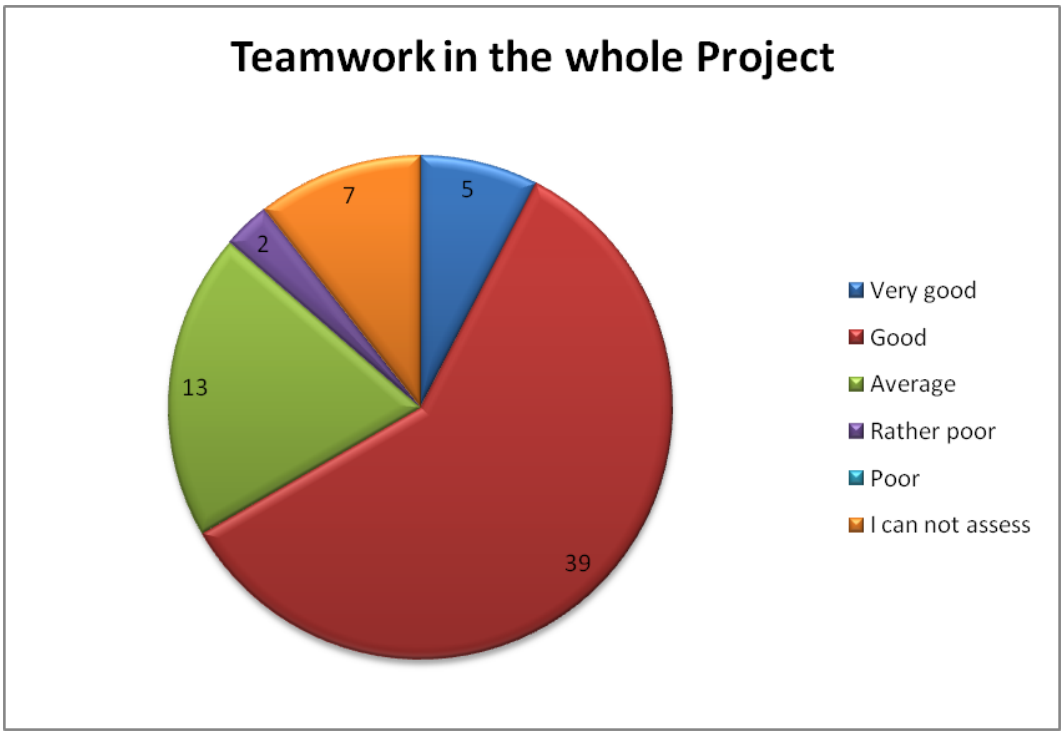

Figure 9.28.: How would you rate teamwork in the whole Catrobat project?

Teamwork in the whole Catrobat project is not as positive as teamwork in teams. Five participants voted *very good* and 39 voted *good*. Also seven participants voted *I can not assess*, maybe because they have joined Catrobat recently.

Comments: Comments concerning this question indicate a missing communication. Participants think, that Catrobat members have to show more initiative to influence teamwork in a positive way.

Teamwork Satisfaction

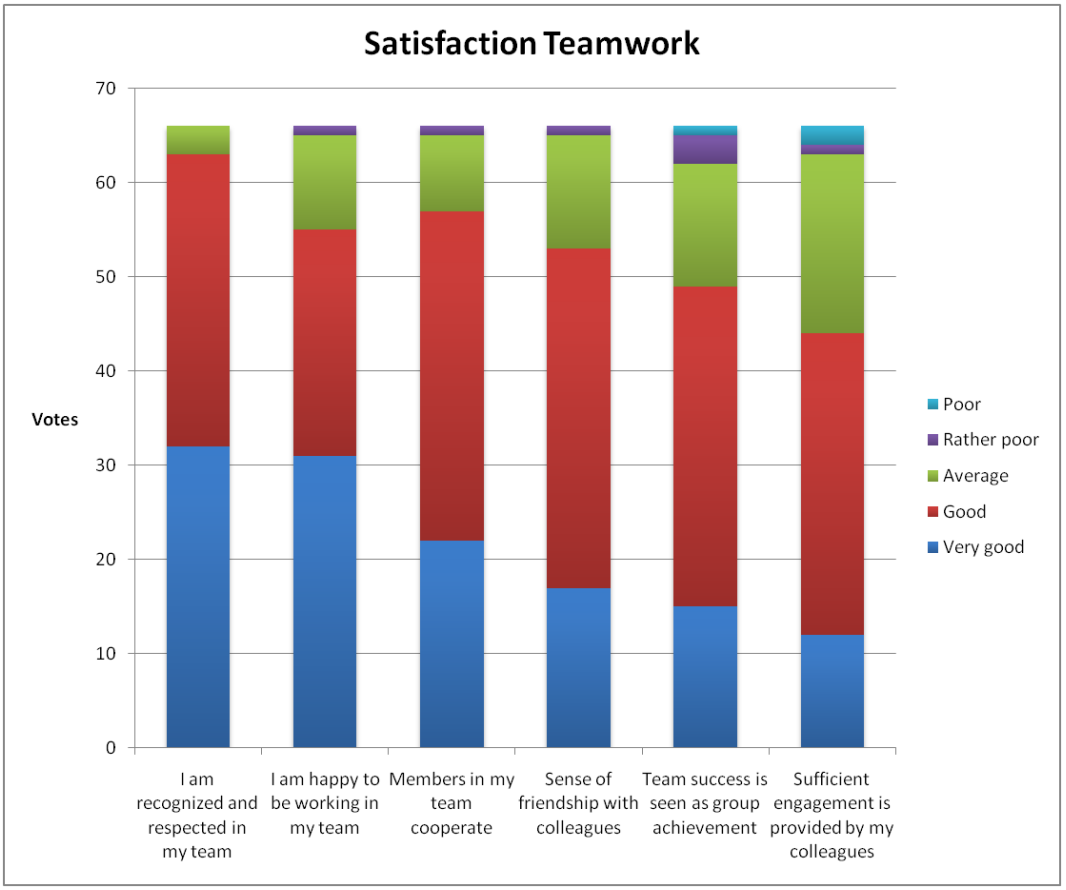

Figure 9.29.: Teamwork satisfaction

These job satisfaction factors are sorted by their sum of Likert scale scores. The satisfaction factor with the best rating is *I am recognized and respected in my team*. In contrast, *sufficient engagement is provided by my colleagues* has the poorest rating. The other factors lie between these extremes and tend to be rather positive.

Work Satisfaction

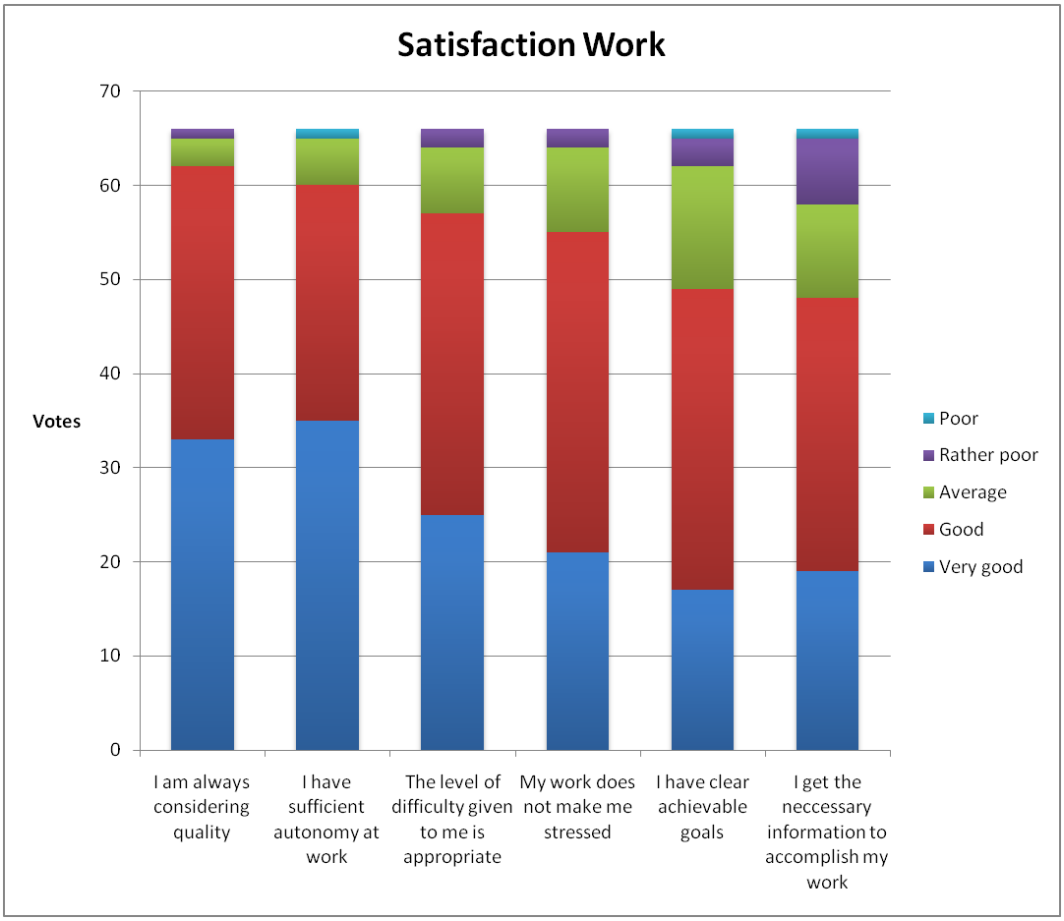

Figure 9.30.: Work satisfaction

The satisfaction factor *I am always considering quality* has the best rating. In contrast, *I have clear achievable goals* and *I get the necessary information to accomplish my work* have the poorest ratings.

9.1.7. Coordination

Coordinators have an important role in the Catrobat project. Most subprojects have an own coordinator. In this section results concerning coordination are presented.

How would you rate coordination in your team?

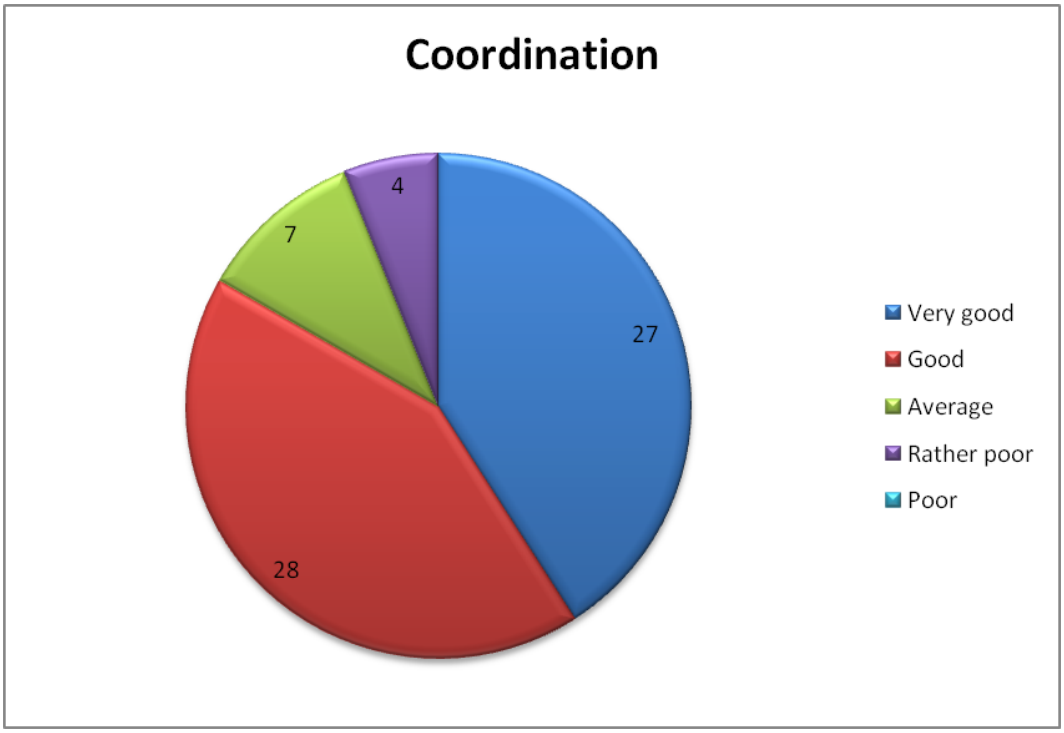

Figure 9.31.: How would you rate coordination in your team?

Ratings for coordination are positive. 27 participants voted coordination as *very good* and 28 voted it as *good*. No one voted coordination as *poor*. This result indicates satisfaction concerning coordination.

Comments: Nearly all comments concerning this question are positive too. Coordinators are supporting newcomers. Some participants report a good

availability of coordinators. One participant noted, that his coordinator is also available on the weekend.

What should be improved concerning coordination in your team?

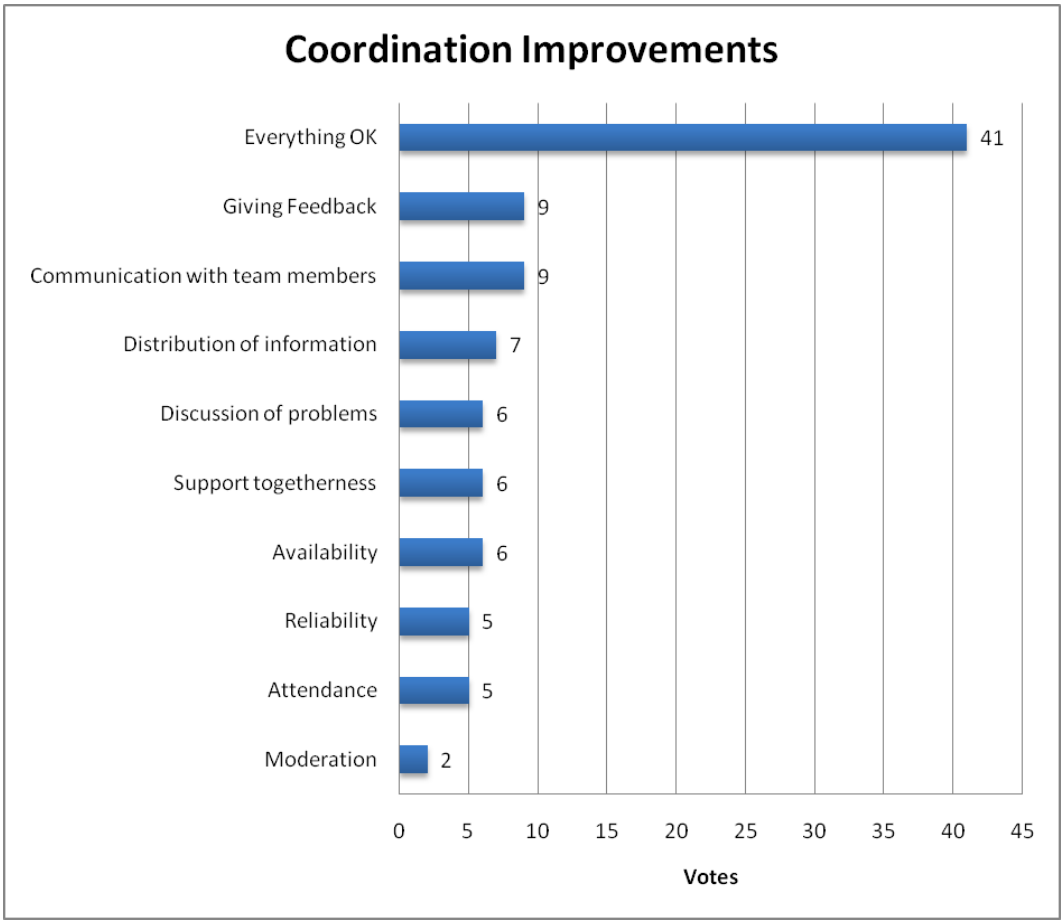

Figure 9.32.: What should be improved concerning coordination in your team?

Coordination is rated as very positive. The result of this question strengthens this positive attitude concerning coordination. 41 participants voted *everything ok*.

Some possible improvement factors with the most votes are *giving feedback*, *communication with team members* and *distribution of information*. These factors indicate a potential for improvement in this already positive topic.

Comments: Regular feedback and objective agreements would be motivating.

Coordination Satisfaction

These job satisfaction factors are sorted by their sum of Likert scale scores. The most positive rated factor is *my coordinator treats members fairly*. *I receive regular feedback on my performance* is the one with the poorest rating.

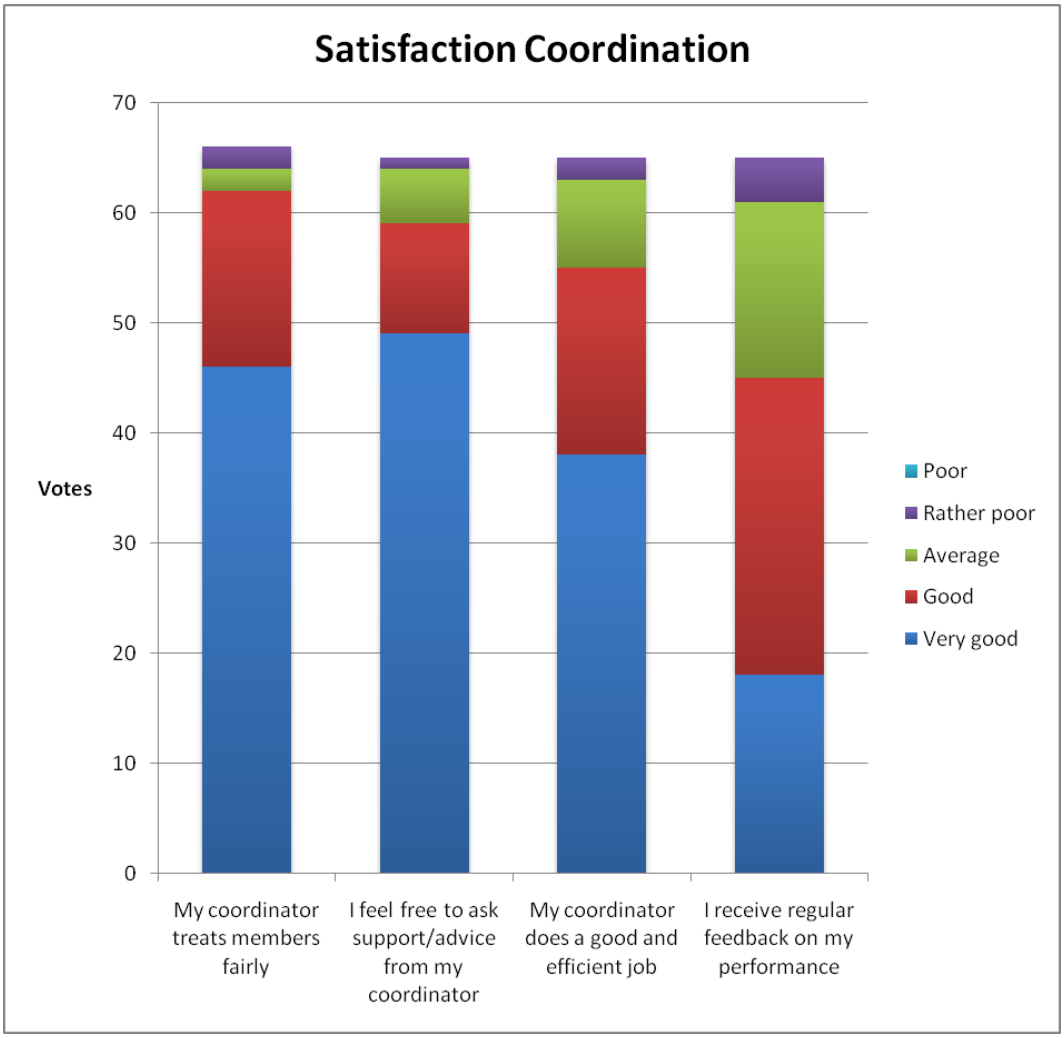

Figure 9.33.: Coordination satisfaction

9.1.8. Practices and Processes

In the Catrobat project several practices from XP and Kanban are used. These practices are described in Chapter 4. Results concerning some practices and processes in the Catrobat project are presented in this section.

How would you rate your start into Catrobat?

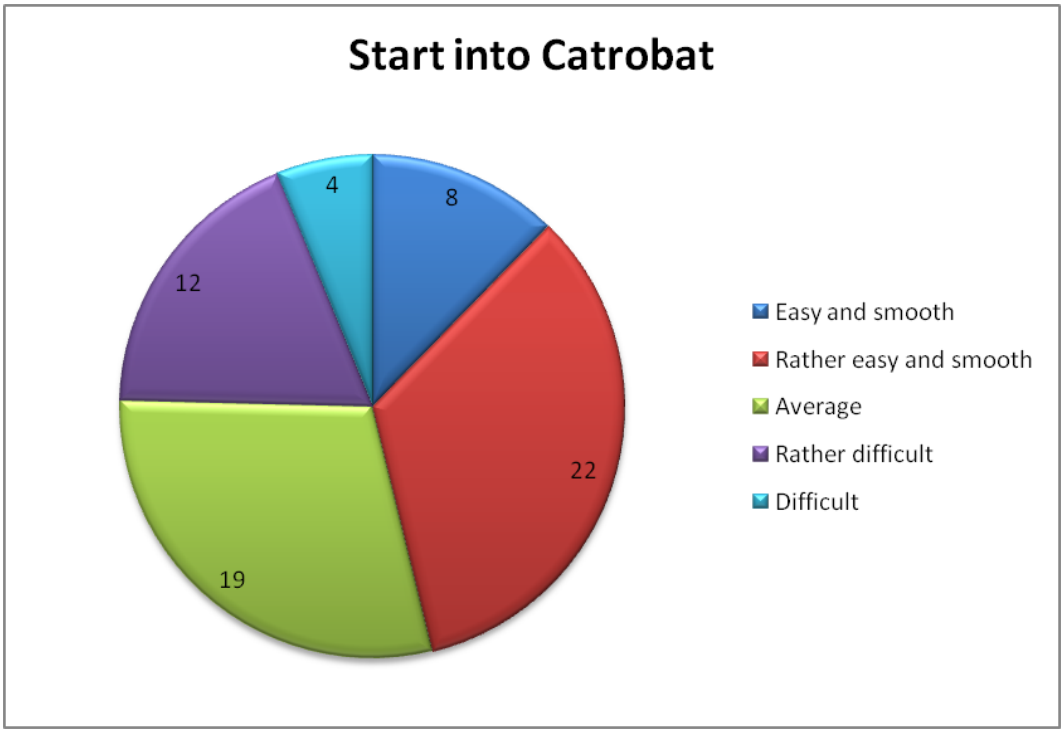

Figure 9.34.: How would you rate your start into Catrobat?

A little bit less than half of all 66 participants rated their start as *very good* or *good*. 35 participants had not really a good start into Catrobat. These 35 participants rated their start as *average*, *rather difficult* or *difficult*.

Comments: Comments indicate issues with documentation and the large codebase. Several participants report the large codebase as very challenging,

but in several cases they received support from other Catrobat members. However, some participants still report a lack of support from their team colleagues. They also report a poor communication with coordinators, senior members and other members.

Several participants like the tutorials at the beginning of their participation, but sometimes these tutorials are held too late. Another participant reports missing information about the project in general, after he has joined.

Practices and Processes Quantity

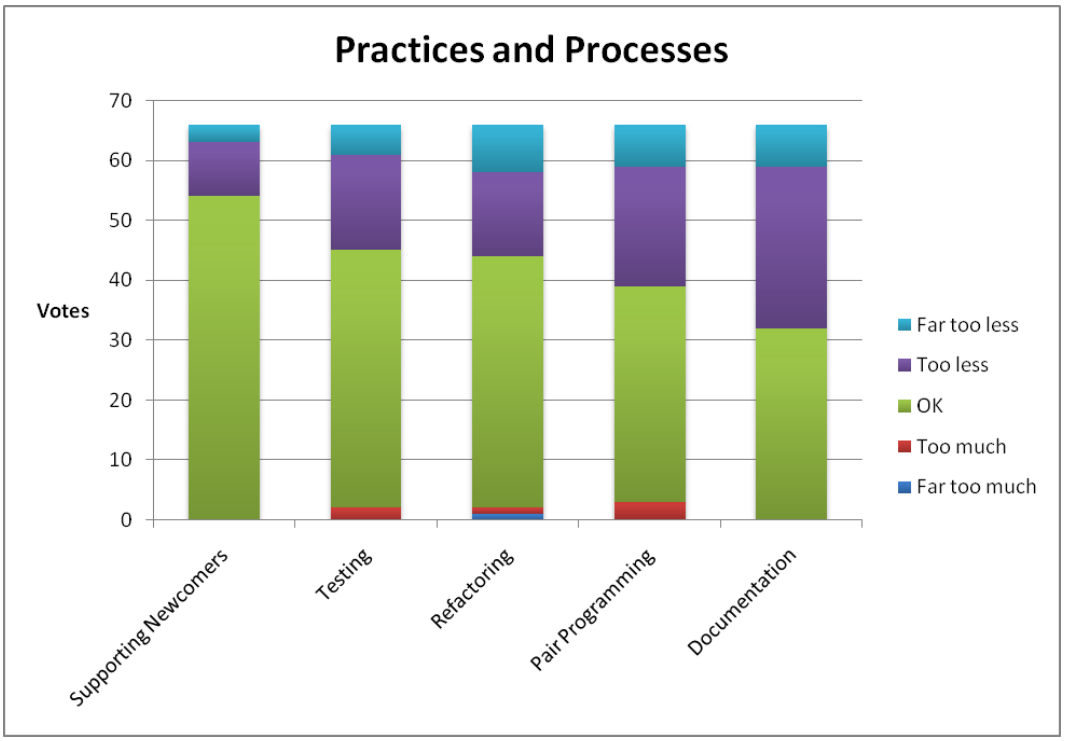

Figure 9.35.: Practices and processes quantity

Participants had to rate the extent of the practices and processes in Figure 9.35. Most practices and processes have many *OK* votes, which indicates an overall positive result. *Supporting newcomers* has the best rating. *Pair*

programming and *documentation* have many votes for *too less* and *much too less*. These two factors offer potentials for improvement.

It seems, that Catrobat members are supporting newcomers, however many newcomers had a rather poor start into Catrobat.

Comments: One participant commented, that starting all tests and fixing them should be done more often. Several other comments highlight missing documentation. Especially, for complex source code, it is later very hard to change this code without documentation.

Do you have already contributed to another FOSS project?

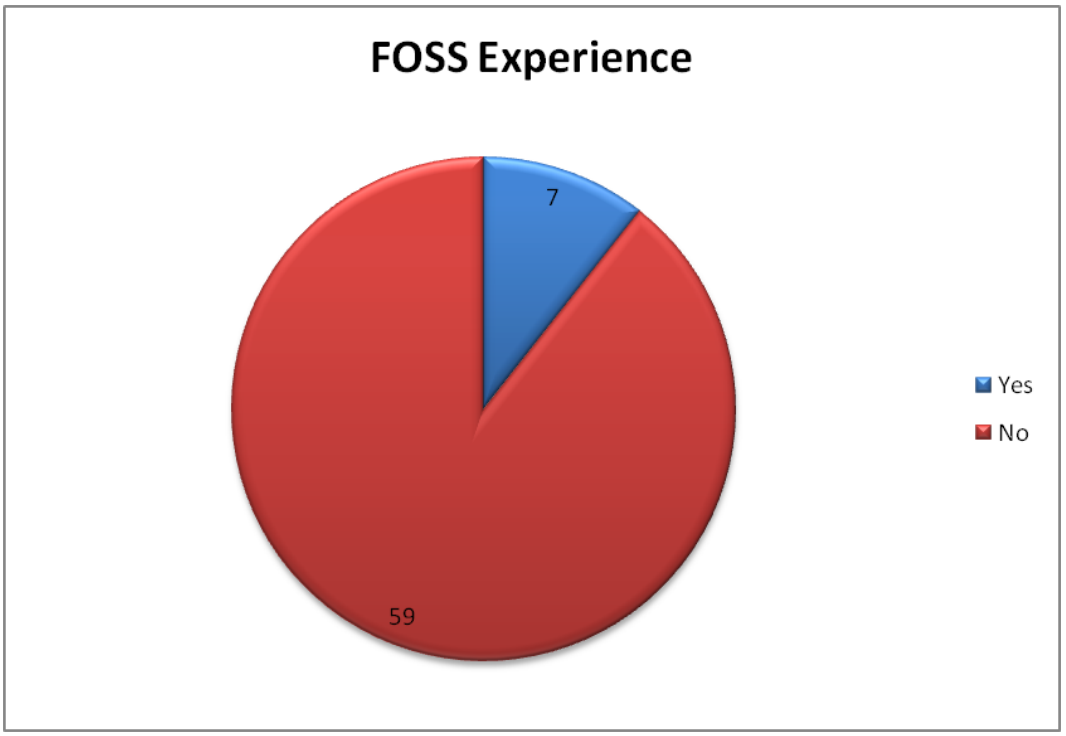

Figure 9.36.: Do you have already contributed to another FOSS project?

Seven participants have already contributed to at least one other FOSS project. One participant has even contributed to three other FOSS projects.

Maximum	3	Standard deviation	0.73	Quartile 1	1
Minimum	1	Mode	1	Median	1
Range	2	Mean	1.57	Quartile 3	2

Table 9.3.: Contributing to other FOSS projects statistics

What can you take with you and what have you learned during your participation in Catrobat?

This question addresses education within the Catrobat project. 27 participants answered this optional question concerning learning in Catrobat.

Comments: Several participants collected valuable experiences and a lot of practice for their further professional life. Several participants learned a lot about software development methods and corresponding practices like XP, TDD or Kanban.

Ten participants reported great learning opportunities concerning tools and technologies like Android, Jira, Confluence, Github and many more.

Not only technical skills are highlighted by participants, also social skills like teamwork. In addition, they liked meeting new people. Sometimes real friendships are the result.

Other participants like to participate in a FOSS project. They learned a lot about self-organization. One participant got ideas for own projects during his participation.

9.1.9. Improvement

Collecting data for improvement is a major objective of this thesis. In this subsection questions concerning possible improvements are presented.

What should be improved in Catrobat?

This question was an optional one. 23 participants answered this question concerning general improvements in the Catrobat project.

Comments: The most frequent answer is *communication and information exchange*. This is already identified as an issue and should be improved. Several participants see an issue especially with *communication between teams*. Two participants want more information about visions and objectives of the Catrobat project in general.

Other participants think, that coordinators should demand more from their members. One participant thinks, that leadership trainings would be a good idea to strengthen coordinators' skills. Another participant suggests trainings for technical topics like Git, Android or Jira.

Further thoughts of participants concerning improvements are: *more pair programming*, *more documentation*, *using deadlines*, *a working test environment*, *teambuilding event* or *more promotion at university*.

What should be improved with the handling of newcomers?

25 participants answered this optional question. This question is focusing on the onboarding process in the Catrobat project.

Comments: The most frequent answer is *pair programming with a senior member or an experienced member*. Especially, at the beginning this would be very helpful. One participant thinks, that newcomers should spend a certain amount of time with an experienced member to get started. Two other participants think, that mentoring systems are positive, but only if mentors have enough time for their mentee.

Another improvement suggested by several participants are tutorials or trainings about various topics. For example participants recommend topics like TDD, XP, test system, Android, Catrobat in general or *how to create my first brick*. This kind of trainings is already suggested by participants in the previous question. Another participant suggests, that every team should have its own tutorials for newcomers.

Guidelines for communication and interactions between teams are improvement ideas mentioned by three participants. For example how coding teams should interact with the UX team should be described in a guide.

Further improvements identified by participants are: *more documentation, more general support for newcomers* or *an introduction of all team members*.

Would you work further on Catrobat after finishing your current thesis or project?

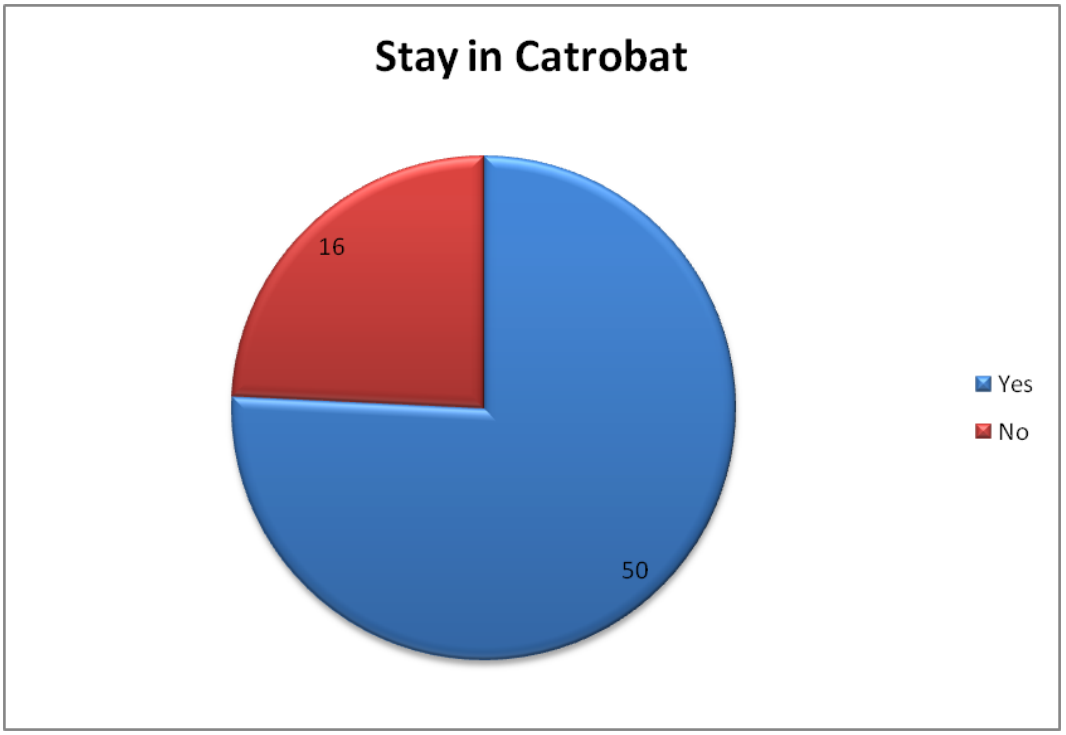

Figure 9.37.: Would you work further on Catrobat after finishing your current thesis or project?

50 participants would work further on Catrobat. Around one quarter would actually not work further on Catrobat after finishing their current course. Several reasons for a *no* are identified in the next question.

What has to be improved to change your decision to yes?

Ten participants out of these 16, who voted *no* for working further on Catrobat, filled out this question.

Comments: Most of these participants will leave Catrobat, because they do not have enough time or have other plans in their study or professional life.

Three participants would change their decision, if they had the possibility to earn money with their participation in the Catrobat project.

Another participant would not change his decision, because he thinks, that working with students is too hard.

Improvement Ideas

In this question five ideas for improvement are proposed. Results concerning these ideas are sorted by participants' popularity. The idea *retrospectives* has the best rating. Most ratings for *retrospectives* are *great idea* and *good idea*. There are no votes as *poor idea* for *retrospectives*. The idea with the second best result is *regular meetings with Wolfgang*. *Test for newcomers* is seen as the poorest idea.

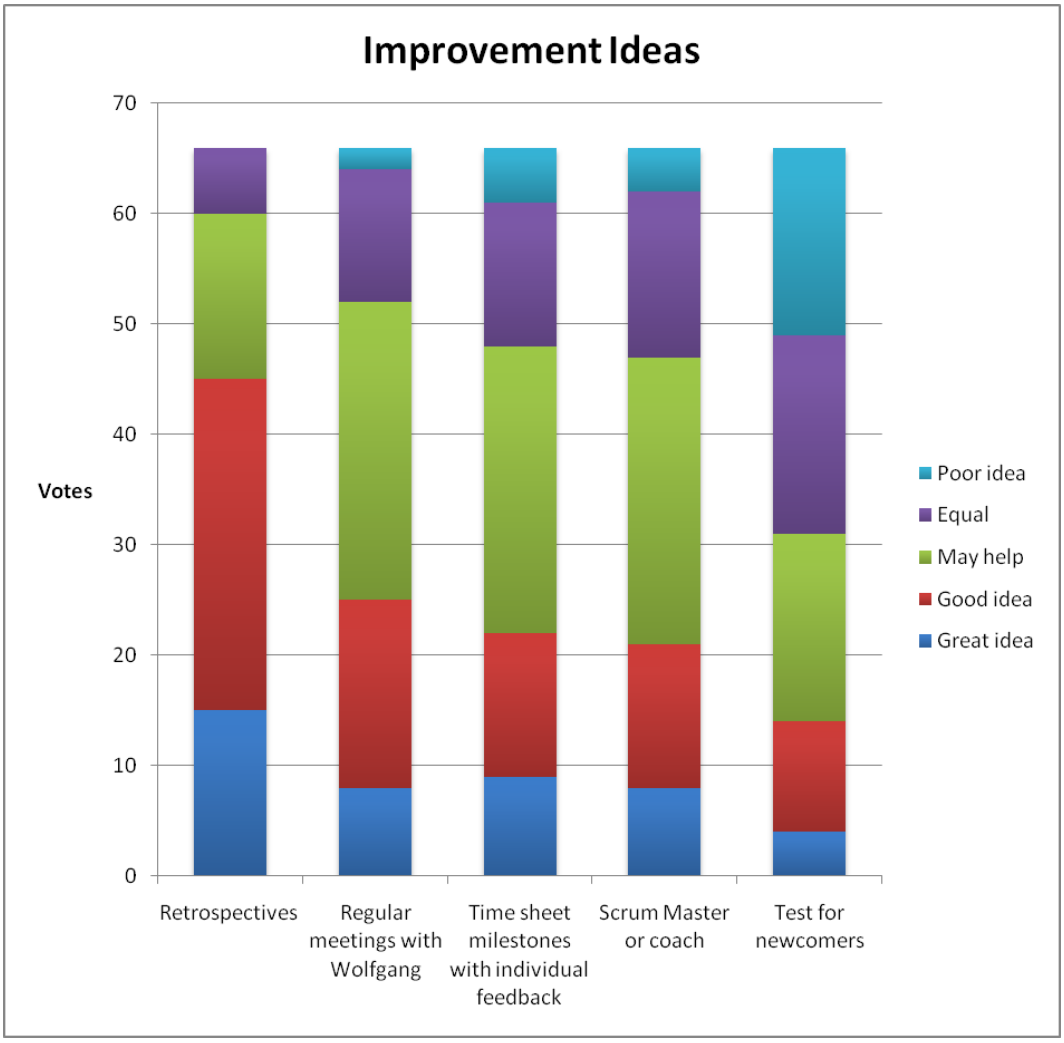

Figure 9.38.: Improvement ideas

Do you have an idea to improve Catrobat?

This question offered participants the possibility to bring in their own ideas. 15 participants answered this optional question. Answers to this question are very diverse. Thoughts and answers are brought together in the comments section below.

Comments: Skills and abilities should be considered more during participation in the Catrobat project. At the end of the participation there should be a small examination for topics like TDD or XP.

Coordinators should demand more from their team members. In this case they should be stricter in supervising their members. If the overall performance of a member was not good or a member has not learned important skills, then a stricter grading should be enforced. Giving more feedback in general would also be helpful.

A better communication and information exchange is desired by one participant. An own Catrobat newsgroup would be valuable for the Catrobat project, is the opinion of another participant.

Further ideas from participants are *code reviews, obligate refactoring, a working test system, yearly Catrobat community team event* or *a paid core team*.

9.1.10. Feedback

Do you have anything to say about Catrobat?

Eleven participants answered this optional question.

Comments: Most of them answered this question and honored Catrobat as a great project. Further answers are *nice people in the project, making valuable experiences, being part of Catrobat* and *the idea behind FOSS*.

Do you think this survey can expose some potentials for improving Catrobat?

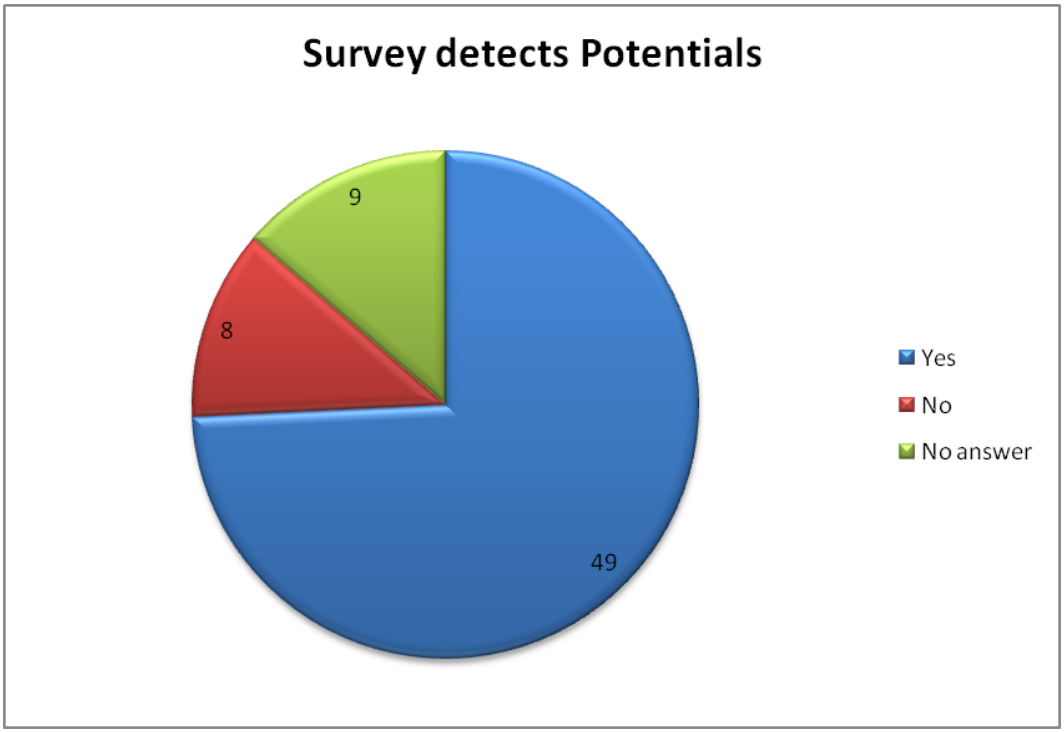

Figure 9.39.: Do you think this survey can expose some potentials for improving Catrobat?

Most participants rated this survey as useful for further improvements in the Catrobat project. 50 participants voted *yes*. Eight participants voted *no*. Nine participants did not answer this question.

9.2. Survey Results concerning Motivation

In this section results concerning motivation are presented. First survey results concerning motivation are analyzed. Then survey results concerning demotivation are presented as well.

9.2.1. Motivation

30 motivators are used in in this survey. The first question offers a snapshot of what are the most motivating factors in general. The second question shows the presence of motivators in the Catrobat project. Then these results are combined and illustrated together. At the end of this subsection, comments to motivators are presented as well.

How motivating are the following motivators for you?

The result of this question illustrates the general importance of motivators. Motivators are sorted by their general motivational effect in descending order. The first 15 motivators are illustrated in Figure 9.40. The other half is illustrated in Figure 9.41.

Motivators with the best ratings are *trust / respect*, *team quality*, *appropriate working conditions* and *project success*. Participants rated these motivators as the motivators with the highest motivational effect. *Motivates me strongly* and *motivates me* are dominating votes for these motivators.

Feedback and *making a contribution* have no votes as *no impact on motivation*. Interestingly, these motivators are only in the middle of the overall sorted motivators.

At the top of the second half there are *equity*, *employee participation*, *professionalism* and *financial benefits* present as the most motivating factors.

Motivators with the poorest ratings of all 30 motivators are *meet new people*, *grading* and *good relationship with users / customers*. Participants see these motivators as the least motivating ones.

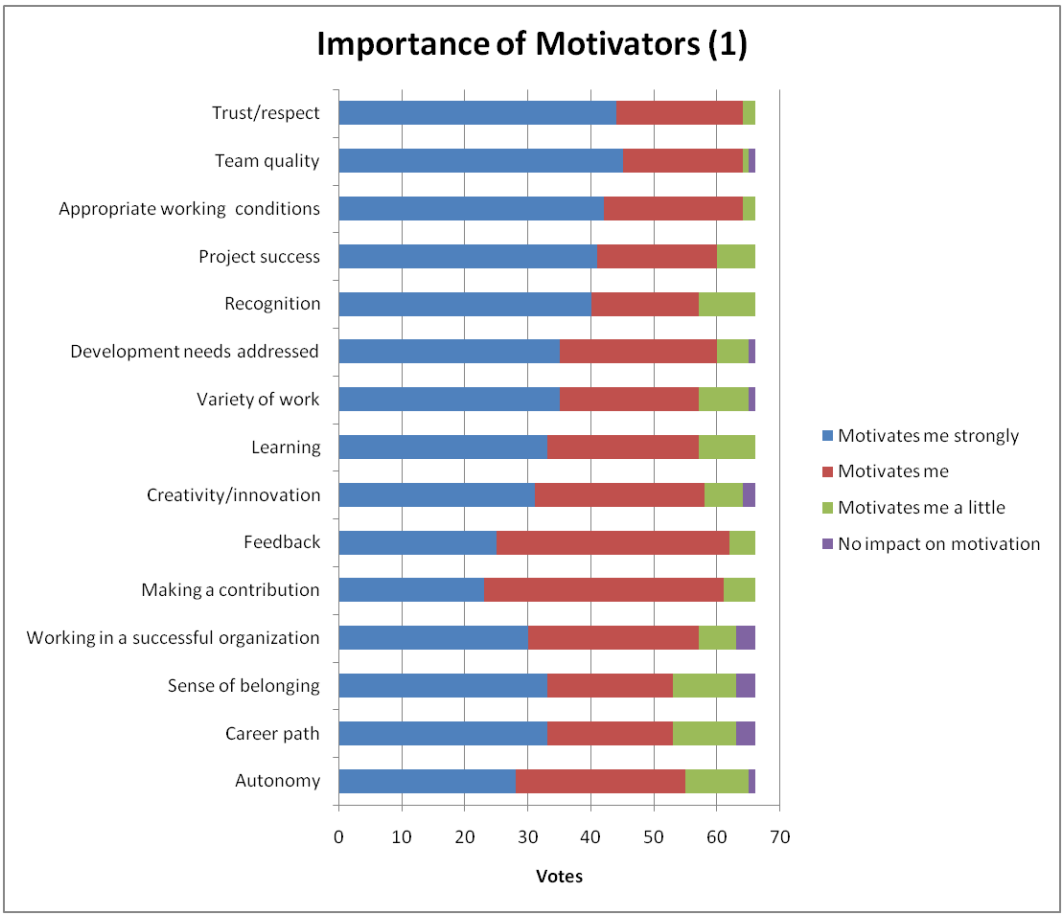

Figure 9.40.: Importance of motivators (1)

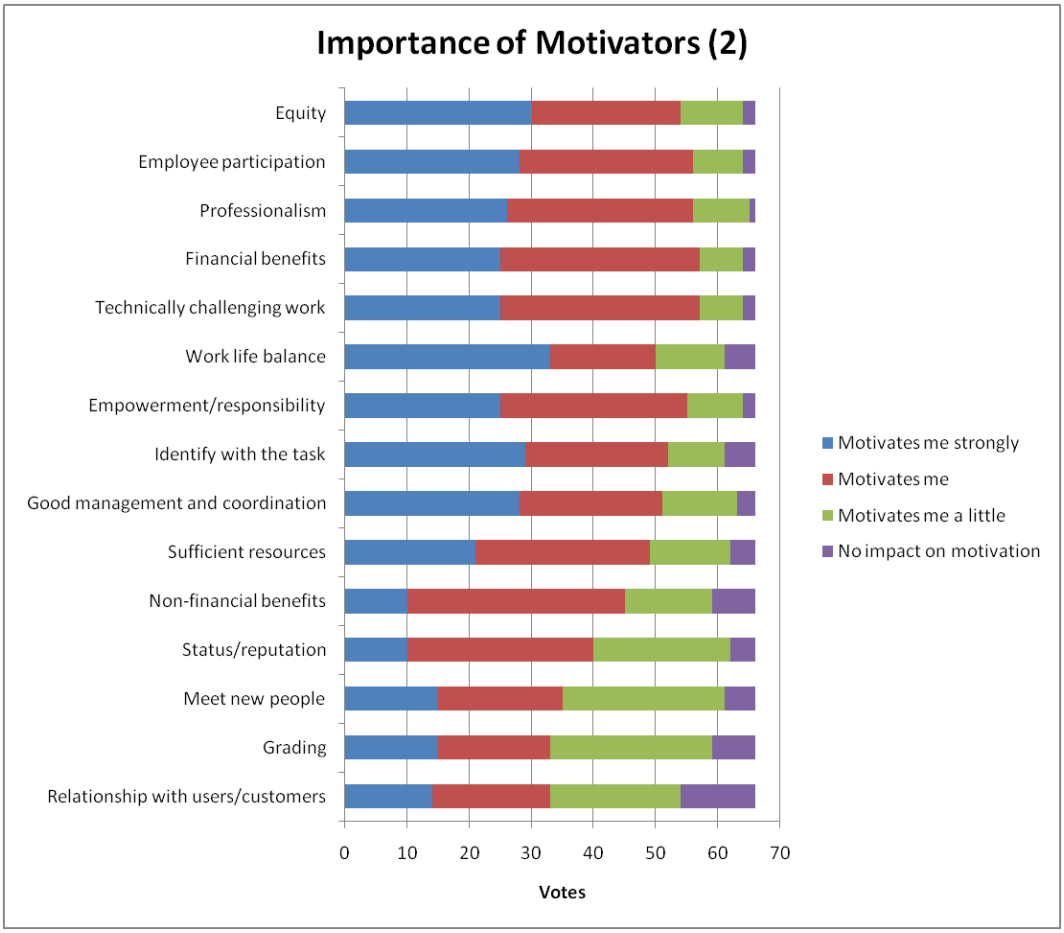

Figure 9.41.: Importance of motivators (2)

How present are the following motivators for you in Catrobat?

Results of this question show the presence of motivators in the Catrobat project. Participants had the possibility to rate these motivators as *completely present*, *rather present*, *rather not present* or *not present*. Motivators are also sorted descending by their presence in the Catrobat project. Results are split up into Figure 9.42 and Figure 9.43.

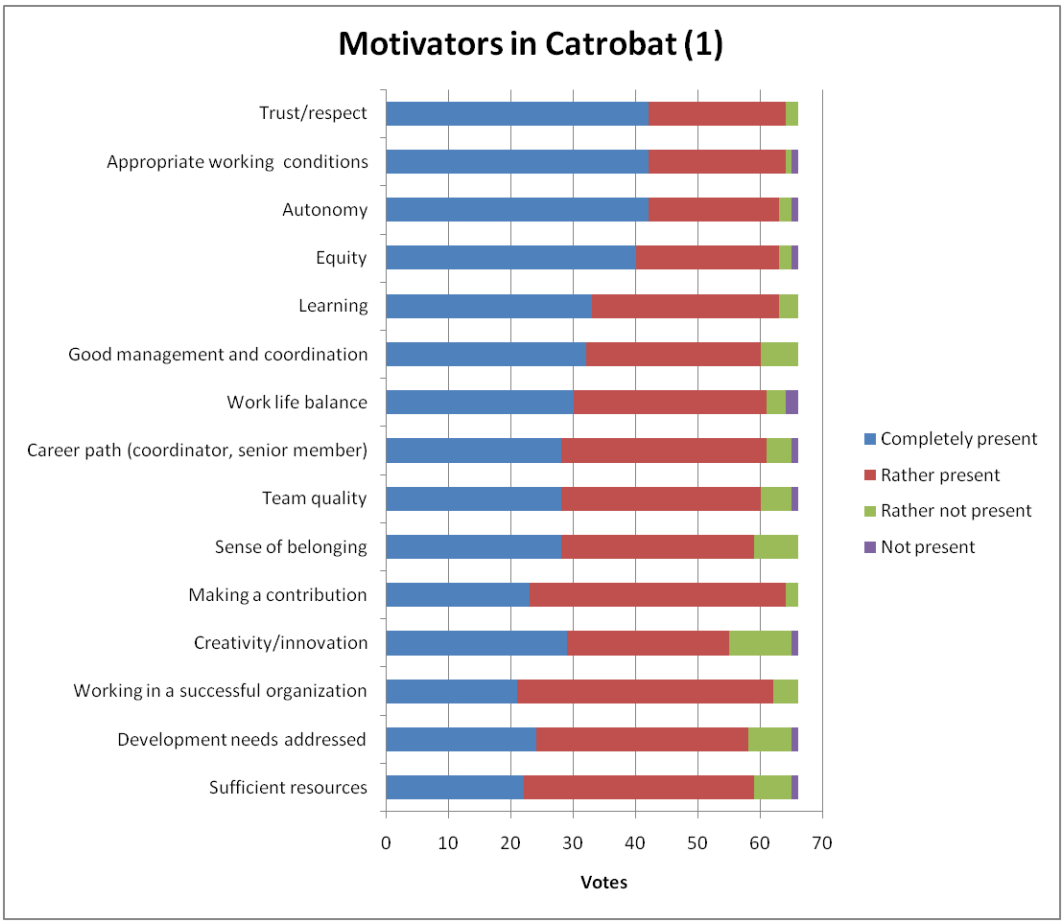

Figure 9.42.: Motivators in Catrobat (1)

Motivators with the highest presence in Catrobat are *trust / respect*, *appropriate working conditions*, *autonomy* and *equity*. These motivators have also the most

votes as *completely present* and *rather present*. The fifth one with the highest presence in Catrobat is *learning*.

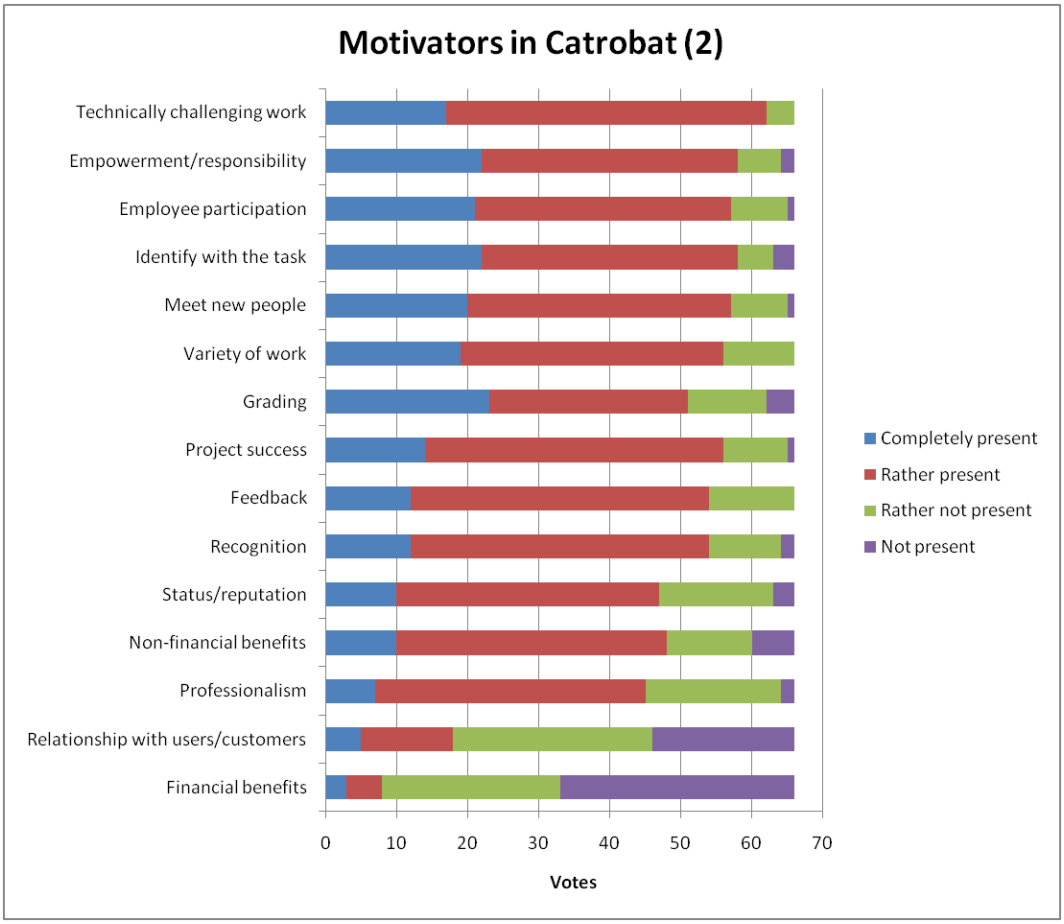

Figure 9.43.: Motivators in Catrobat (2)

Motivators with the least presence in Catrobat are *professionalism*, *relationship with users / customers* and *financial benefits*. Especially *relationship with users / customers* and *financial benefits* have a lot of votes as *not present* in Catrobat.

Importance of motivators versus motivators in Catrobat

In this subsection results of motivators' general motivational effect and their presence in Catrobat are compared. Motivators in Figure 9.44 and Figure 9.45 are sorted by their general motivational effect. The dominance in these figures is the sum of the Likert scale scores. Therefore the following weights are used for answers' options: *motivates me strongly (4), motivates me (3), motivates me a little (2), no impact on motivation (1)* and *completely present (4), rather present (3), rather not present (2), not present (1)*. The sum of these weighted answer options is representing the dominance.

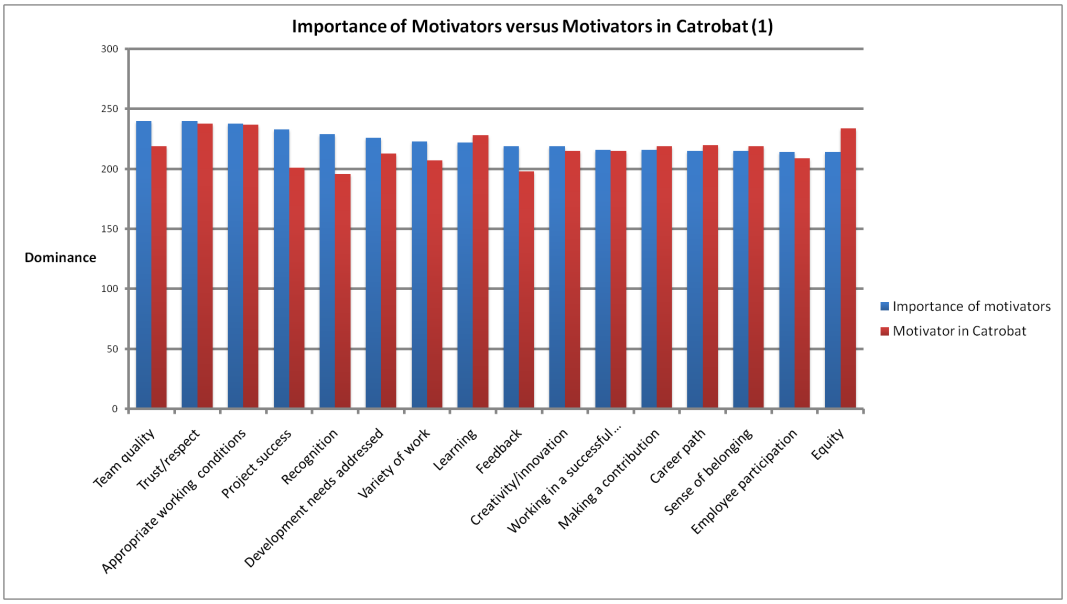

Figure 9.44.: Importance of motivators versus motivators in Catrobat (1)

In the first half, both types are similar to each other. Some of them have a small gap like *team quality*, *feedback* or *equity*.

In the second half, more motivators have gaps. Some of them with higher gaps are *professionalism*, *financial benefits*, *meet new people* or *grading*.

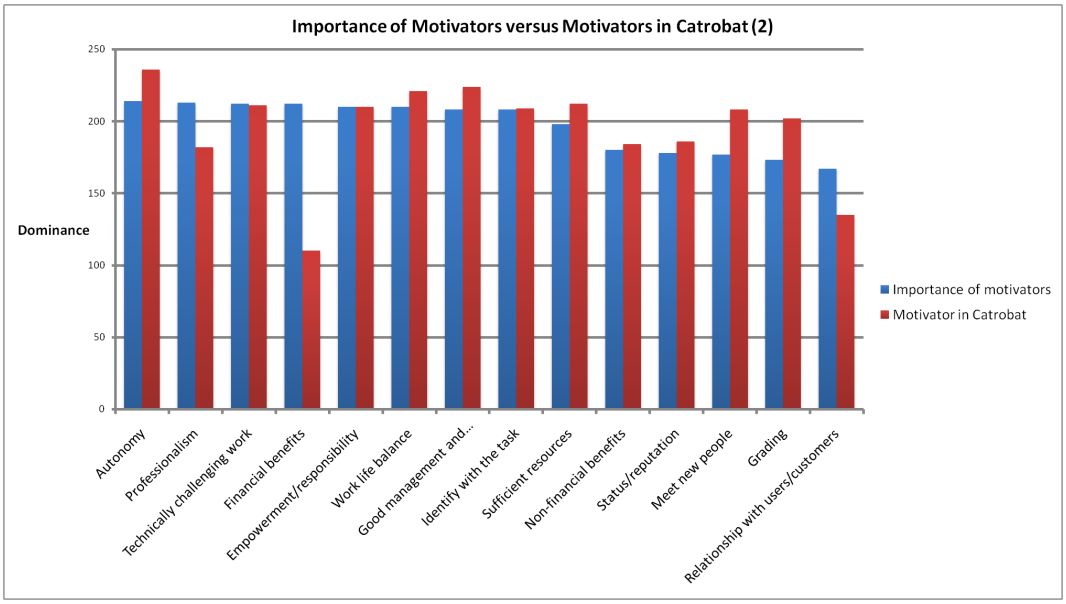

Figure 9.45.: Importance of motivators versus motivators in Catrobat (2)

What encourages you to expend extra energy to Catrobat?

Comments: 13 participants answered this question. The dominating answer of these participants is *making a contribution*. They also like the vision and the idea behind Catrobat. They want to be a part of this large FOSS project.

Other participants like the people in the Catrobat project. As a result they are expending extra energy to Catrobat. A similar reason is to get positive feedback from other Catrobat members.

9.2.2. Demotivation

14 demotivators are examined in this subsection. The structure of these results is similar to the one in the motivators subsection. First the general demotivational effect of demotivators is illustrated. Then the presence of these demotivators is examined in the Catrobat project. Afterwards these results are combined and visualized together.

How demotivating are the following demotivators for you?

This question shows the general importance of demotivators and their general demotivational effect. Results in Figure 9.46 are sorted by their demotivational effect in descending order.

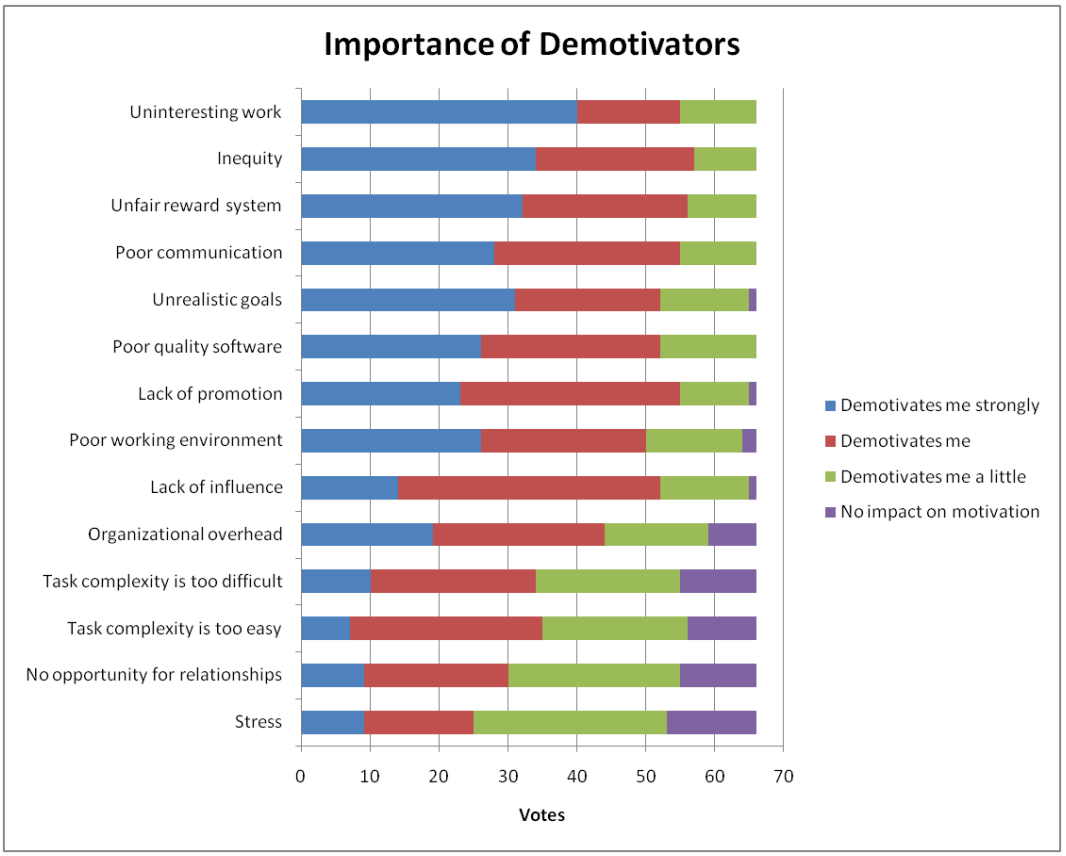

Figure 9.46.: Importance of demotivators

The most demotivating factors, identified by participants, are *uninteresting work*, *inequity*, *unfair reward system* and *poor communication*. These four demotivators have a lot of votes as *demotivates me strongly* and *demotivates me* and even no vote as *no impact on motivation*.

Demotivators with the least impact on motivation are *task complexity is too easy*, *no opportunity for relationships* and *stress*.

How present are the following demotivators for you in Catrobat?

Results of this question show demotivators' presence in Catrobat. Demotivators in Figure 9.47 are sorted by their presence in Catrobat descending.

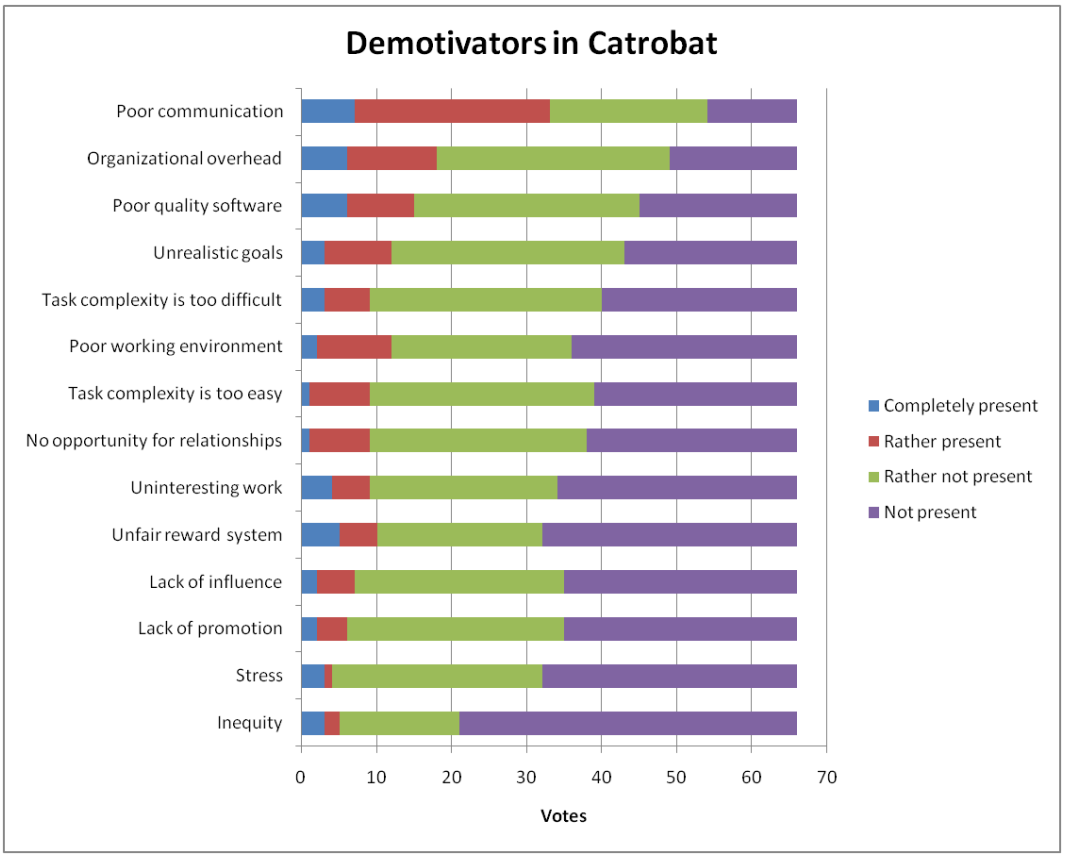

Figure 9.47.: Demotivators in Catrobat

Demotivators with the highest presence in Catrobat are *poor communication*, *organizational overhead*, *poor quality software* and *unrealistic goals*.

In contrast, demotivators which are less present in Catrobat, are *lack of influence*, *lack of promotion*, *stress* and *inequity*.

Importance of demotivators versus demotivators in Catrobat

Results of demotivators' general demotivational effect and their presence in Catrobat are opposed in Figure 9.48. The dominance in these figures is the sum of the Likert scale scores. Therefore the following weights are used for answers' options: *demotivates me strongly (4), demotivates me (3), demotivates me a little (2), no impact on motivation (1)* and *completely present (4), rather present (3), rather not present (2), not present (1)*. The sum of these weighted answer options is representing the dominance.

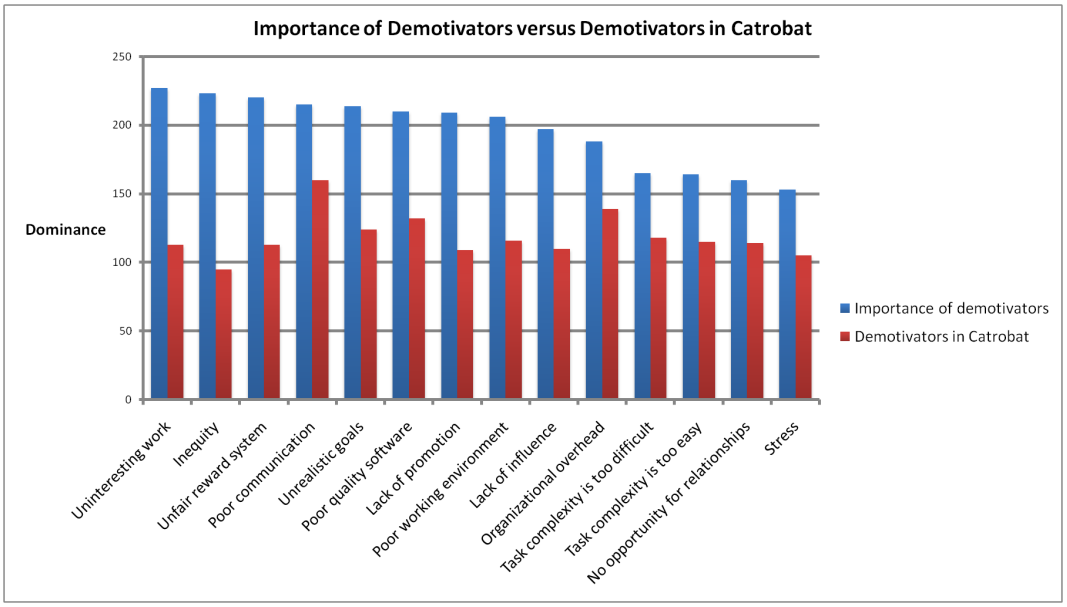

Figure 9.48.: Importance of demotivators versus demotivators in Catrobat

Most demotivators have a very low dominance concerning their presence in Catrobat. For example *uninteresting work* or *inequity* have a very low presence in Catrobat. *Poor communication* is the fourth highest demotivator in general and has the highest dominance of all demotivators in Catrobat. This demotivator indicates a potential for improvement.

What really saps your energy working in Catrobat?

Comments: Participants reported several reasons which sap their energy in Catrobat. *Missing code quality* and *complex code* are identified as problematic by several participants.

Other participants reported issues concerning *testing*. For example they see the jenkins instance and a non working testing infrastructure as sapping their energy. One participant reported *test fixing of older functionality* as demotivating.

Further participants reported *missing teamwork* and *too less encouragement by other Catrobat members* as demotivating.

One participant does not like the implementation of software development methods and practices in Catrobat. In his opinion, *XP, TDD* and *continuous integration* are done poorly in Catrobat.

9.3. Interpretation of Motivation Results

In this section results concerning motivation and demotivation are interpreted and used for further examination. First the top ten motivators in the Catrobat project are analyzed and interpreted. These motivators have the highest presence in the Catrobat project. Then the top five demotivators in Catrobat are examined. Motivators in the Catrobat project are also compared with motivators in other SE settings. Afterwards the Herzberg theory and the MOCC model are applied to the Catrobat project. At the end of this section, a categorization of Catrobat is done via motivators of different SE settings.

9.3.1. Top Ten Motivators in Catrobat

In this subsection the top ten motivators in the Catrobat project are listed and interpreted. These motivators represent one third of the entire set and have the highest presence in Catrobat. The interpretation is done via survey results, survey comments, collected experiences and derived assumptions.

1. Trust / Respect: This is the most present motivator in the Catrobat project. Project members have the feeling to be recognized as human beings with valuable skills and opinions. Catrobat members are friendly to each other. The contact between Catrobat members with different roles seems to be good. For example coordinators trust their members to get the job done and respect them. This is a solid basis for working together in a positive atmosphere. As a result people tend to feel comfortable, which can have a positive effect on teamwork.

2. Working conditions: Catrobat offers great working conditions. A lot of infrastructure and existing processes are provided by the Catrobat community. For example the team room, the test systems or tutorials for newcomers. Relationships between colleagues seem to be positive. Another advantage for catrobat members is the possibility to shape their own working conditions. For example they can select a project under the Catrobat umbrella project in which they are interested. Further examples are provided under the motivator *autonomy* below.

3. Autonomy: Catrobat members have a lot of freedom during their participation. At the beginning of their participation, they can select a sub-project in which they are interested or they can even realize their own ideas within the Catrobat project. During their participation they have good working conditions, see motivator *working conditions* with the second highest presence in Catrobat. Catrobat members can act almost autonomously while implementing their tasks. Further aspects concerning autonomy are described under *work life balance*, which has the rank seven.

4. Equity: Catrobat members have the feeling to be treated fairly. Coordinators and management treat people fairly. Equity is a motivator, which is not causing issues in the Catrobat project.

5. Learning: Learning is one of the most relevant motivators in FOSS projects and also in education (Ye and Kishida, 2003; Bosnić et al., 2011). Learning is very present in the Catrobat project. Catrobat members can learn a lot during their participation. They can improve their skills under social and technical aspects. Most Catrobat members are working within teams, thus they learn how to act in teams. They work on a large project and collect valuable experiences for their professional life. For example Catrobat members become familiar with professional tools and services like GitHub, Sourcetree, Android Studio, Jenkins and many more.

6. Management / Coordination: Management and Coordination is also a very present motivator in the Catrobat project. As already mentioned equity, trust and respect are provided by coordinators and management while they interact with Catrobat members. Coordinators play an important role in sub-teams. They are moderate and organize team meetings, distribute work, offer support and do administrational work. Coordinators and management are doing a good job in Catrobat.

7. Work life balance: Work life balance is supported by the Catrobat policies. Catrobat members can act autonomously. They can plan and make their own time management depending on their workload. If there is a lack of time during the semester, they can reduce their work on Catrobat. If they have more time, they can work as much as they can. It is important to attend to team meetings, but much work could be done from home.

8. Career path: Career path can be seen from two perspectives. One is the career path within the Catrobat project. With activity, performance and knowledge, it is possible to become a coordinator or a senior member in Catrobat. Another perspective is the professional life. Being active in such a large project can be positive for making social connections which can later be beneficial. The contribution to a FOSS project is a further advantage in CVs.

9. Team quality: Most work in Catrobat is done in teams. Therefore good teamwork is required. With this good ranking of team quality such a teamwork seems to be given. Catrobat members work together well in their teams. Teamwork across teams is not as positive as teamwork within single teams.

10. Sense of belonging: Catrobat has a certain presence at university, which is relevant for the motivator *sense of belonging*. Catrobat members feel a sense of belonging to their team and to Catrobat. In this case team cohesion seems to be good. Meetings and the team room support this motivator. Catrobat members have good relationships with their colleagues and feel they are a part of their team and the whole Catrobat project.

9.3.2. Top Five Demotivators in Catrobat

In this subsection the five most present demotivators in the Catrobat project are listed and interpreted. They represent one third of the demotivator set. The interpretation is done via survey results, survey comments, collected experiences and derived assumptions.

1. Poor communication: Communication is identified as problematic. Communication in teams seems to work quite well. In contrast, communication between teams is seen as poor. Communication concerning general information like news, objectives and visions sometimes does not satisfy Catrobat members' need for information. IRC as communication tool is missing its intention. IRC is used sparsely and only by a few members.

2. Organizational overhead: This is the demotivator with the second highest presence in the Catrobat project. Catrobat is influenced strongly by the

educational context. As a result, some organizational overhead is necessary. For example grading has to be done in this context. Catrobat members from Graz University of Technology have to keep detailed records of the time they spend on the project. Maybe this is one reason for the high rank of this demotivator.

3. Poor quality software: In several sub-teams of the Catrobat project, there is a large codebase. Most source code is created by students. Catrobat members with different experience levels are writing source code. Some Catrobat members are at an early stage of their studies. They do not have much experience in developing software. Another aspect concerning the codebase is the fast exchange of members. Most Catrobat members are students at Graz University of Technology. If they have a grade for their course, they leave the project. As a result, many people leave the project and others join. This aspect can also have a large impact on the evolution of the codebase.

4. Unrealistic goals: This demotivator has the fourth highest presence in the Catrobat project. Maybe sometimes the scope for releases is too large or there are not enough resources available to accomplish given goals. Too little time for a given goal can result in reduced quality. If Catrobat members reduce quality it has also a negative effect on motivation. *Poor quality software* is identified as the third most present demotivator in Catrobat. If teams are not able to accomplish given goals, this demotivator can have a high presence. Short term switches of goals can also be critical for teams. Consequences of short term switches can be stress and teams may have to discard already implemented functionality. As a result, short term switches can lead to demotivation.

5. Task complexity is too difficult: For several participants tasks have a too high complexity. This can be caused by the large codebase or by inexperienced Catrobat members. Coordinators have the possibility to assign easier tasks to less experienced members.

9.3.3. Herzberg Theory Applied to Catrobat

In this subsection, the Herzberg Motivational Theory is applied to Catrobat. This theory has already been described in Section 6.1.2. With this application Catrobat can be described from a motivational view. The Herzberg theory applied to Catrobat can be used to investigate satisfaction and dissatisfaction in Catrobat. Factors with a negative impact on Catrobat can be evaluated and considered for improvement. For this purpose survey results and other elaborated results are used. All dissatisfiers and satisfiers are interpreted and assessed as having a *positive*, *average* or *negative* impact on Catrobat. At the end an overview is provided.

Satisfiers

Achievement: Catrobat members achieve a lot. Many services and applications are are already published and some are not. Several sub-teams are working on non published applications. As a result, they probably do not have as much satisfaction in their achievement as teams which have already published their applications. They do not see their results in the public and they do not get feedback from users. As a consequence, the satisfier *achievement* is assessed as having an *average* impact on satisfaction in Catrobat.

Recognition: *Recognition* is the motivator with the fifth highest general motivational effect. In contrast, *recognition* has a relative low presence in Catrobat. In this ranking it has the rank 25. Catrobat members get recognition from others, but not enough to become motivated. Maybe coordinators do not honor jobs enough which are done by their team members. In addition Catrobat members want more feedback, which underlines the need for recognition. The satisfaction factor *I am recognized and respected in my team* has a positive rating. Thus this satisfier's is assessed as having an *average* influence on satisfaction in Catrobat.

Work itself: *Creativity / innovation* is a relatively present motivator in Catrobat. *Technical challenging work* and *variety of work* are in the second half of the ranking. In this case there are several motivators which are not completely

present in Catrobat. As a result, the satisfier *work itself* is assessed as having an *average* influence on Catrobat.

Responsibility: Catrobat members have responsibility regarding their job. If they are coordinators or senior members they have further responsibilities. The motivator *empowerment / responsibility* is present in Catrobat. This satisfier is assessed as having a *positive* impact on Catrobat..

Advancement: The motivator *career path* has the seventh highest presence in Catrobat. Catrobat members have possibilities to become more important members in the community. They have the chance to become coordinators, but there is at most one coordinator for a sub-team. Thus possibilities for becoming a coordinator are rather rare. Becoming a senior member is another possibility to become a more important member. However, there are no clear policies to become a senior member or a coordinator. The satisfier *advancement* is assessed as having an *average* impact on satisfaction in Catrobat.

Growth: Learning and gaining new skills is seen as very present in Catrobat. Catrobat members like this aspect within their participation. They experience a similar project environment as in companies with large projects. Thus this satisfier is having a *positive* effect on satisfaction in Catrobat.

Dissatisfiers

Company policies: Policies in Catrobat enable members freedom from several perspectives. As already mentioned, Catrobat members have a lot of freedom concerning time management. Good management and coordination is seen as present in Catrobat. The corresponding motivator has the sixth highest presence in Catrobat. Thus management is assessed as positive in general. There are also some present demotivators in Catrobat, which have a negative effect on this dissatisfier. For example these demotivators are *poor communication*, *organizational overhead* and *unrealistic goals*. They can be influenced by policies and management. These three demotivators have already been described and are three out of the four most present demotivators in Catrobat. As a result, this dissatisfier is assessed as having an *average* influence on Catrobat.

Supervision: Typical supervisors in Catrobat are coordinators. In the survey they are assessed as very positive. Coordination in teams is rated as *very good* and *good* from 83% of all survey participants. The motivator concerning good coordination has the sixth highest presence in Catrobat. Two third of all survey participants have no suggestions for improvement concerning coordination. They like the job, which is done by coordinators. Coordinators are treating members fairly and they are open to questions from their team members. Catrobat members also get technical support from their coordinators. Giving more feedback is identified as a factor, which can be improved and considered by coordinators. All in all *supervision* can be assessed as having a *positive* impact on Catrobat.

Interpersonal relationships: Dissatisfiers concerning relationships are combined in the dissatisfier *interpersonal relationships*, because gathered data does not provide detailed information for relationships between different roles. The demotivator *no opportunity for relationships* has the eighth rank in Catrobat out of all 14 demotivators. This demotivator has a rather low presence in Catrobat. Some participants reported, that they are spending extra energy on Catrobat, because of their relationships to Catrobat members. Two factors concerning teamwork are assessed as positive which are *sense of friendship with colleagues* and *I am happy working in my team*. Thus this dissatisfier can be seen as having a *positive* impact on Catrobat

Work conditions: *Working conditions* is the motivator with the second highest presence in Catrobat. As already described, Catrobat offers much infrastructure at Graz University of Technology. For example a team room with a coffee machine, hardware and test systems are provided. There are also professional tools used like Jira, Confluence and Github, which have a positive effect on *work conditions*. As a result, this dissatisfier can be assessed as having a *positive* impact on Catrobat.

Salary: Typical participations in Catrobat are not compensated in form of money, because Catrobat is a FOSS project in an educational context. Most members are students and get grades for their participation. The motivator *financial benefits* has the lowest presence in Catrobat. However, there are some possibilities in Catrobat to earn money. One possibility is the Google Summer of Code program. With the exception of a few possibilities, this dissatisfier has a *negative* impact on Catrobat.

Personal life: Catrobat offers policies, which support a good work life balance. Catrobat members have a high level of education. In this case they have good job opportunities. However, there is only little data collected concerning Catrobat members' personal life. Thus this dissatisfier is assessed as having an *average* influence on satisfaction in Catrobat.

Status: This dissatisfier is having a rather *negative* effect on Catrobat, because the motivator *status / reputation* has a relative low presence in Catrobat. It has the 26th rank of all 30 motivators. Maybe status is not important for Catrobat members, because most of them leave Catrobat when they have a grade.

Security: Catrobat is seen as a successful organization by most Catrobat members. This is a result from the survey. Catrobat is tending to grow and to become a larger organization. There is no need to have fear of being fired. Catrobat members at Graz University of Technology get a grade for their participation. If they are active and show initiative, they can get a good grade. Under this aspects *security* is having a *positive* impact on Catrobat.

Resulting Overview

In Figure 9.49 Herzberg's results concerning satisfaction and dissatisfaction are illustrated. Percentages leading to extreme satisfaction and dissatisfaction are also from Herzberg's results. In addition, the assessed impact on Catrobat is added for every satisfier and dissatisfier. This figure represents a satisfactional overview of the Catrobat project.

Satisfiers: Most satisfiers are assessed as having an *average* impact on satisfaction in Catrobat. Thus not all satisfiers have a *positive* impact on Catrobat. *Recognition* is identified by Herzberg as a satisfier, which can frequently lead to extreme satisfaction. This factor has a low rank in Catrobat. Performance and contributions seem to be recognised only a little. Especially, management and coordinators have possibilities to give recognition and appreciation. *Growth* and *responsibility* are having a *positive* impact on Catrobat. Satisfiers in Catrobat tend to be positive, but there is still a potential to improve satisfaction in Catrobat.

Dissatisfiers: Most dissatisfiers are assessed as having a *positive* or *average* impact on Catrobat. *Salary* and *status* are assessed as having a *negative* impact on Catrobat. These two dissatisfiers can be seen as potential for improvement. In an educational environment *salary* is rather unusual. In addition, *company policies* is assessed as having an *average* effect on Catrobat. There are also some notable factors which can lead to dissatisfaction. Apart from these dissatisfiers, most of them are not causing dissatisfaction in Catrobat.

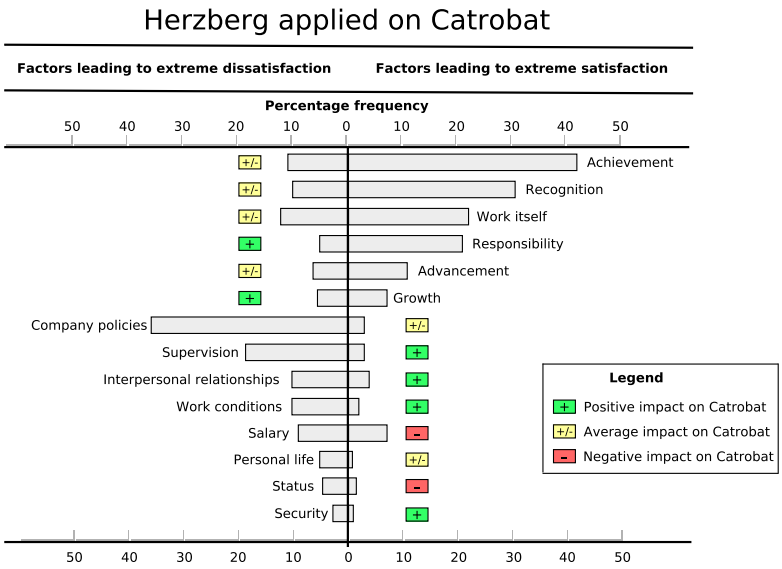

Figure 9.49.: Herzberg theory applied to Catrobat

In the Catrobat environment, dissatisfaction is mostly avoided, because most dissatisfiers are not having a *negative* impact on Catrobat. Most satisfiers are also present to a certain degree in Catrobat, but not as much as dissatisfiers. As a result, Catrobat members tend to be motivated, satisfied and committed to the organization. However, several satisfiers are only assessed as having an *average* impact on satisfaction in Catrobat. There is a potential to motivate Catrobat members more, than it is actually the case. Catrobat offers an environment, which is offering a general satisfaction, but only limited conditions for high motivation.

A limitation of this approach is the subjective classification as having a *positive, average* and *negative* impact on Catrobat. The subjective classification of satisfiers and dissatisfiers in Catrobat is done via survey results, experiences and interpretations.

9.3.4. MOCC Model Applied to Catrobat

In this subsection the MOCC model is applied to the Catrobat project. For this purpose general data on Catrobat and survey results are used. In this subsection the four components of the MOCC model are analyzed within the Catrobat context. At the end an overall overview of the MOCC model applied to Catrobat is given. This overview is visualized in Figure 9.50.

Context Component

This component represents the individual personality of every Catrobat member. In addition it describes the context of the Catrobat project. Four factors are used to describe the context of the Catrobat project. These factors are *FOSS, agile, university* and *Austria*. Catrobat is a FOSS project. Its software is licensed under the GNU Affero General Public License (Catrobat Developers Team, 2016c). Catrobat offers interested people the possibility to join the community. Most contribution and management are done at Graz University of Technology. The characteristics component is influenced by this component.

Characteristics Component

In this application of the MOCC model, the characteristics component contains some basic characteristics of software engineers as identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011; Helen Sharp et al., 2009. For example *growth-oriented, autonomous* and *technically-competent* are mentioned. This component also contains survey results concerning personality traits in the Catrobat project. For this purpose a small chart with the Big Five personality traits is added in Figure 9.50. In this chart, dashed

lines represent norms of personality traits, which are identified by Gosling, Rentfrow, and Jr., 2003. The yellow area is representing Catrobat members' personality traits, which are gathered in the survey.

Motivators Component

This component represents 30 motivators. These motivators are split up into intrinsic and extrinsic motivators. All these motivators are sorted by their presence in Catrobat. Next to every motivator is an orange and a blue value. The orange value is the rank of the corresponding motivator's presence in Catrobat. The blue value is the rank of the corresponding motivator's general motivational effect. These ranks represent the relative position of motivators in descending order. For all motivators these two values are taken from survey results. For example *professionalism* has *28* as orange value. This indicates *professionalism* has the 28th highest presence in Catrobat of all 30 motivators. *Professionalism* has *18* as blue value. This indicates *professionalism* is the motivator with the 18th highest general motivational effect of all 30 motivators. All 30 motivators with their ranks are representing the motivators component.

Outcomes Component

In this component outcomes of the whole Catrobat project are listed. For example *educational contributions* at Graz University of Technology and in school classes is one outcome. Other outcomes are *awards, cooperations with famous organizations* or the large Catrobat community. The Catrobat project has a lot of outcomes. Some of them are illustrated in this component.

Resulting Overview

All these four components together build the MOCC model applied to Catrobat. The context component influences the strength of the characteristics component and also has direct impact on the motivators component. Depending on this characteristics component, the motivators component

is orientated towards this characteristics component. The motivators component with all 30 motivators and ranks also has an effect on the characteristics component. Catrobat members working within this context, have these characteristics and experience these motivational factors during their participation. These three components together are resulting in various outcomes. Some outcomes are represented in the outcomes component.

All components and their relations are illustrated in Figure 9.50. This application of the MOCC model represents the whole Catrobat project from a motivational view. This overview can be used by management to initiate investigations and improvements. It also offers the possibility to become aware of various factors which are influencing the outcomes of the Catrobat project. Measureable outcomes would be beneficial to get facts out of this applied model. It would be helpful to monitor retention rates, newcomer rates, completed story points in an iteration or worked time in an iteration on sub-project level. These are possible future works of this thesis.

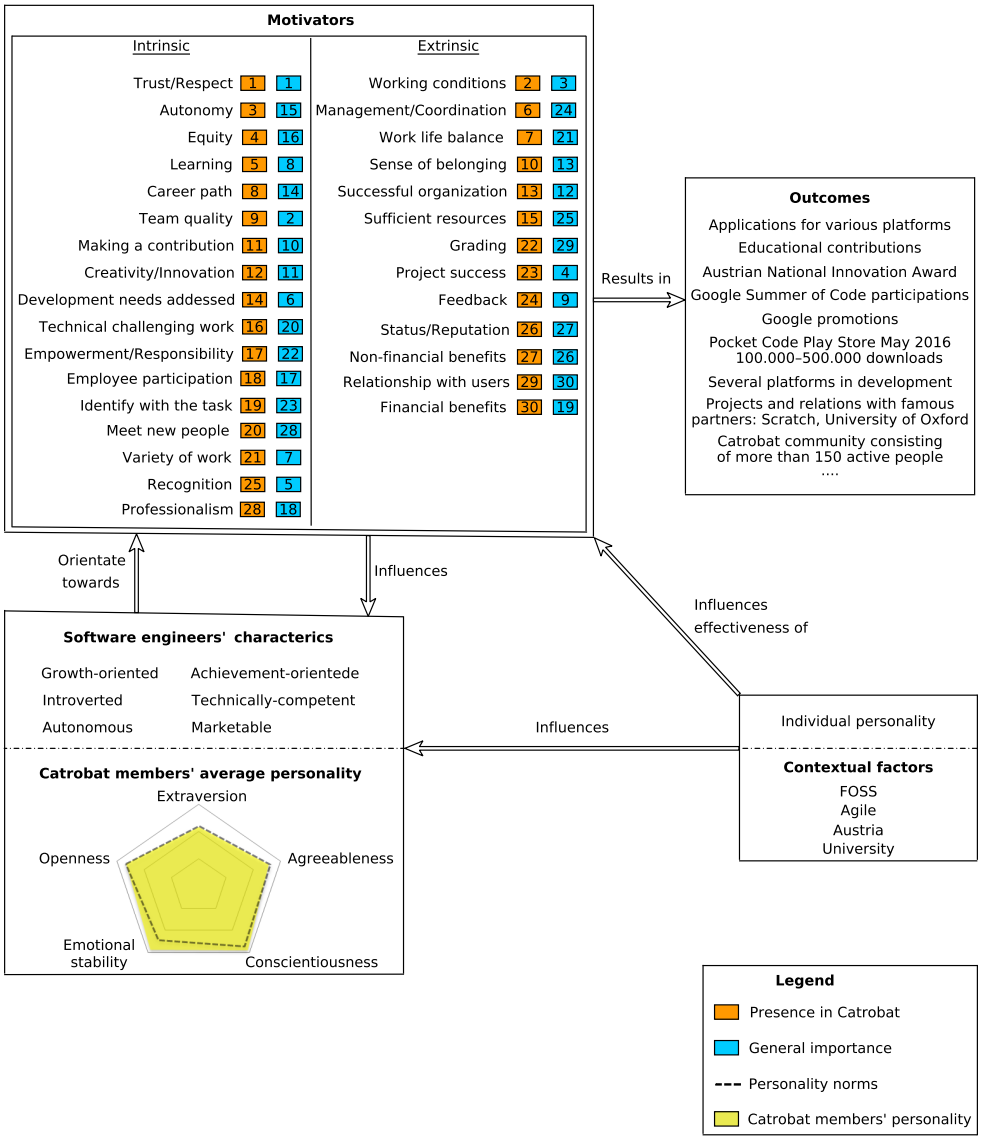

Figure 9.50.: MOCC model applied to Catrobat

9.3.5. Comparison with Various Settings

In this subsection motivators in Catrobat are compared with motivators in different settings. For example such settings are *importance in literature, agile teams, student motivation, brazilian companies, a company from Cambridge* or *FOSS projects*. These various settings are compared with the motivators in the Catrobat project.

In some settings, the naming of motivators is very similar to the naming within this thesis. In this case the naming of this thesis is used. The motivator naming mapping can be found in Appendix B.1. In some rankings, motivators have the same rank. In this case a quantity value is added to make motivators with the same rank visible.

Motivators in Catrobat versus Motivators Motivational Effect versus Motivators in Literature

Rankings: In Table 9.4 three rankings of motivators are compared. The first column shows the presence of motivators in the Catrobat project. The second one illustrates the general motivational effect. These two rankings are taken from survey results. The third column are combined results from literature reviews on motivation in SE (Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011). This ranking is done by the number of relevant documents in the literature. The quantity for relevant documents is appended to every motivator. The most relevant motivators in literature are *employee participation, identify with the task* and *career path*. Motivators with the least relevance are *fun, ideology, non-financial benefits* and *penalty policies*.

Comparison: While comparing survey results ordered by their presence in Catrobat with their general motivational effect, some motivators have similar rankings, but there are motivators with very different rankings as well. The two motivators with the highest presence in Catrobat are *trust / respect* and *working conditions*. These two motivators are the first and the third motivator ranked by their general motivational effect. In this case two very important motivators are very present in the Catrobat project. This can have a positive effect on the Catrobat members' overall motivation.

Rank	**Catrobat Motivators (Survey)**	**Motivators in General (Survey)**	**Literature (Beecham)**
1	Trust/Respect	Trust/Respect	Employee participation (26)
2	Working conditions	Team quality	Identify with the task (26)
3	Autonomy	Working conditions	Career path (24)
4	Equity	Project success	Management/Coordination (20)
5	Learning	Recognition	Development needs addressed (20)
6	Management/Coordination	Development needs addressed	Variety of work (19)
7	Work life balance	Variety of work	Rewards and incentives (17)
8	Career path	Learning	Autonomy (16)
9	Team quality	Feedback	Recognition (15)
10	Sense of belonging	Making a contribution	Sense of belonging (15)
11	Making a contribution	Creativity/Innovation	Technical challenging work (15)
12	Creativity/Innovation	Successful organization	Feedback (13)
13	Successful organization	Sense of belonging	Job security (12)
14	Development needs addressed	Career path	Trust/Respect (10)
15	Sufficient resources	Autonomy	Work life balance (9)
16	Technical challenging work	Equity	Empowerment/Responsibility (9)
17	Empowerment/Responsibility	Employee participation	Making a contribution (9)
18	Employee participation	Professionalism	Working conditions (6)
19	Identify with the task	Financial benefits	Equity (5)
20	Meet new people	Technical challenging work	Team quality (4)
21	Variety of work	Work life balance	Creativity/Innovation (4)
22	Grading	Empowerment/Responsibility	Successful organization (3)
23	Project success	Identify with the task	Professionalism (2)
24	Feedback	Management/Coordination	Relationship with users (2)
25	Recognition	Sufficient resources	Sufficient resources (2)
26	Status/Reputation	Non-financial benefits	Fun (1)
27	Non-financial benefits	Status/Reputation	Ideology (1)
28	Professionalism	Meet new people	Non-financial benefits (1)
29	Relationship with users	Grading	Penalty policies (1)
30	Financial benefits	Relationship with users	

Table 9.4.: Motivation in Catrobat compared with other settings (1)

There are also some motivators which have much higher ranks ordered by their presence in Catrobat, than ordered by their general motivational effect. These motivators are for example *autonomy*, *management / coordination*, *work life balance* or *sufficient resources*.

Some motivators with much lower ranks ordered by their presence in Catrobat, than ordered by their general motivational effect are *team quality*, *development needs addressed*, *variety of work*, *project success*, *feedback*, *recognition*, *professionalism* and *financial benefits*. This indicates a potential for improvement with a relative high motivational effect. If these motivators with a high motivational effect are present, then they can have a large influence on the overall motivation of Catrobat members.

Motivators in Catrobat versus Motivation in Agile Teams versus Student Motivation

Rankings: In Table 9.5 the presence of motivators in the Catrobat project is compared with motivation in agile teams and with motivation in a software development course. The second column represents motivators in agile teams. This ranking of motivators is published by O. Melo, Santana, and Kon, 2012. Several motivators in this ranking have the same quantity. The quantity represents the identification of motivators in interviews with employees in the IT industry. Thus the quantity is added to every motivator in this column. In agile teams the most motivating factors are *feeling of progress, technical challenging work, development needs addressed* and *team quality*. The third column represents student motivation in a software project at university which is identified by Bosnić et al., 2011. In their software project, students have to develop software with students from other countries. During this course information concerning motivation has been collected via polls and questionnaires. *Grading* and *learning* are the most motivating factors.

Comparison: *Learning* has a high rank in Catrobat and also in student settings. *Autonomy* is very present in Catrobat. In contrast, in agile teams autonomy is ranked much lower. In Catrobat *technical challenging work* has the rank 16, in agile teams it has the second rank. *Grading* is the most important motivator in the student course. In contrast, in Catrobat *grading* has no high relevance at all. *Project success* has the rank 23 in Catrobat. In agile teams the motivator *project success* is ranked higher.

Rank	**Catrobat Motivators (Survey)**	**Agile Teams (Melo)**	**Student Motivation (Bosnic)**
1	Trust/Respect	Feeling of progress (8)	Grading
2	Working conditions	Technical challenging work (7)	Learning
3	Autonomy	Development needs addressed (7)	Technical challenging work
4	Equity	Employee participation (5)	Team quality
5	Learning	Identify with the task (3)	Meet new people
6	Management/Coordination	Team quality (3)	Project success
7	Work life balance	Project success (3)	Empowerment/Responsibility
8	Career path	Autonomy (2)	Relationship with community members
9	Team quality	Equity (2)	
10	Sense of belonging	Work life balance (2)	
11	Making a contribution	Development practices (2)	
12	Creativity/Innovation	Management/Coordination (2)	
13	Successful organization	Feedback (2)	
14	Development needs addressed	Experiment (2)	
15	Sufficient resources	Elimination of waste (2)	
16	Technical challenging work	Problem solving (2)	
17	Empowerment/Responsibility		
18	Employee participation		
19	Identify with the task		
20	Meet new people		
21	Variety of work		
22	Grading		
23	Project success		
24	Feedback		
25	Recognition		
26	Status/Reputation		
27	Non-financial benefits		
28	Professionalism		
29	Relationship with users		
30	Financial benefits		

Table 9.5.: Motivation in Catrobat compared with other settings (2)

Motivators in Catrobat versus Motivation in Companies

Rankings: In Table 9.6 motivators in Catrobat are compared with motivation in Brazilian companies and with motivation in a company from Cambridge called *Red Gate*. The second column illustrates motivating factors from multiple Brazilian companies. This data is collected via questionnaires done by França and Fabio Q. B. da Silva, 2010. The most important motivators identified by França and Fabio Q. B. da Silva, 2010 are *working with people, work life balance, problem solving* and *meaningful products*. The third column represents the ranking of motivational factors identified by Sach, H. Sharp, and Petre, 2011. This result is from semi-structured interviews with 13 professional software engineers from a software company in Cambridge.

Rank	**Catrobat Motivators (Survey)**	**Brazilian Companies (França)**	***Red Gate* Company (França)**
1	Trust/Respect	Working with people	Work that is useful (7)
2	Working conditions	Work life balance	Producing good software (5)
3	Autonomy	Problem solving	Problem solving (5)
4	Equity	Meaningful products	Collaborating (4)
5	Learning	Successful organization	Variety of work (4)
6	Management/Coordination	Creativity/Innovation	Learning (3)
7	Work life balance	Empowerment/Responsibility	Identify with the Task (2)
8	Career path	Technical challenging work	People (2)
9	Team quality	Development practices	Building (1)
10	Sense of belonging	Experiment	Ownership (1)
11	Making a contribution	Autonomy	Result like vision (1)
12	Creativity/Innovation	Learning	Career path (1)
13	Successful organization	Decision making	Challenge (1)
14	Development needs addressed	Feedback	
15	Sufficient resources	Participating into entire product lifecycle	
16	Technical challenging work	Identify with the task	
17	Empowerment/Responsibility	Career path	
18	Employee participation	Variety of work	
19	Identify with the task	Rewards and incentives	
20	Meet new people		
21	Variety of work		
22	Grading		
23	Project success		
24	Feedback		
25	Recognition		
26	Status/Reputation		
27	Non-financial benefits		
28	Professionalism		
29	Relationship with users		
30	Financial benefits		

Table 9.6.: Motivation in Catrobat compared with other settings (3)

The number of software engineers who identified corresponding motivators as relevant is added as quantity. The most important motivating factors in this setting are *work that is useful, producing good software, problem solving* and *collaborating*. Results of both company settings showcase the importance of people and a meaning behind work.

Comparison: *Autonomy* is ranked much higher in Catrobat, than in Brazilian companies. *Learning* and *career path* are also ranked higher in Catrobat, than in the two company settings. The motivator *successful organization* has a lower rank in Catrobat, than in brazilian companies.

Rank	**Catrobat Motivators (Survey)**	**FOSS Motivators (Fei-Rong)**	**FOSS Motivators (Yunwen Ye)**
1	Trust/Respect	Learning	Learning
2	Working conditions	Share knowledge and skills	Status/reputation
3	Autonomy	Idea behind FOSS	Trust/respect
4	Equity	Improve FOSS products	Employee participation
5	Learning	Problem solving	Teamwork
6	Management/Coordination	Get FOSS community support	Recognition
7	Work life balance	Career path	
8	Career path	Status/reputation	
9	Team quality	Money	
10	Sense of belonging		
11	Making a contribution		
12	Creativity/Innovation		
13	Successful organization		
14	Development needs addressed		
15	Sufficient resources		
16	Technical challenging work		
17	Empowerment/Responsibility		
18	Employee participation		
19	Identify with the task		
20	Meet new people		
21	Variety of work		
22	Grading		
23	Project success		
24	Feedback		
25	Recognition		
26	Status/Reputation		
27	Non-financial benefits		
28	Professionalism		
29	Relationship with users		
30	Financial benefits		

Table 9.7.: Motivation in Catrobat compared with other settings (4)

Motivators in Catrobat versus Motivation in FOSS Communities

Rankings: In Table 9.7 motivators in Catrobat are compared with motivators in FOSS projects. The second column illustrates motivators in FOSS projects identified by Wang, He, and J. Chen, 2005. The most relevant motivators in this column are *learning*, *share knowledge and skills* and *idea behind FOSS*. In the third column relevant motivational factors in FOSS projects are illustrated, which are identified by Ye and Kishida, 2003. The most relevant factors in this FOSS setting are *learning*, *status / reputation* and *trust / respect*.

Comparison: *Learning* has a high rank in Catrobat, but it is ranked higher in FOSS settings. The motivator *status / reputation* has a similar rank in Catrobat, as in one FOSS setting.

Various settings have been compared with the Catrobat setting in this subsection. There are some similarities, but also differences. These tables offer a detailed overview of motivators in different settings. They are used in the following subsection to try to categorize the Catrobat project.

9.3.6. Categorization of Catrobat

In this subsection motivators with a high relevance in agile, educational, FOSS and company settings are used to categorize the Catrobat project. For the categorization, the presence of motivators in the Catrobat project is used. Several motivators for every category are listed in Table 9.8. The rank in Catrobat is added to every motivator. Motivators for corresponding settings are identified by O. Melo, Santana, and Kon, 2012; Bosnić et al., 2011; Wang, He, and J. Chen, 2005; Ye and Kishida, 2003; França and Fabio Q. B. da Silva, 2010. Then means for these categories are calculated. They are used to describe the matching between Catrobat and a category.

With the mean values of these categories, it is possible to categorize the Catrobat project into *agile, company, FOSS* and *education*. Two different cases for categorization are chosen. In one case only the three most relevant motivators of each category are used for categorization. In the other case the eight most relevant motivators are used for categorization. Two categories have only seven relevant motivators, because in corresponding literature only this amount is listed. In this case the mean is calculated by seven relevant motivators in corresponding settings. The result of both cases is illustrated in Figure 9.51. The resulting ranking of these cases is shown in Table 9.9. In this table, calculated means of corresponding motivators are added. The lower the mean of a category is, the better is the matching with Catrobat.

Agile: The agile category has the poorest matching with Catrobat in the first categorization using only three motivators. In this case the mean has the value *16* and indicates a poor matching with Catrobat. In contrast, in the case using eight motivators, agile has the best matching with Catrobat as the mean value *13.25* indicates. With this diverse result, it is not easy to interpret the matching of agile with Catrobat. *Team quality, autonomy* and

Rank	**Agile** (Rank in Catrobat)	**Company** (Rank in Catrobat)
1	Technical challenging work (16)	Team quality (9)
2	Development needs addressed (14)	Meet new people (20)
3	Employee participation (18)	Work life balance (7)
4	Identify with the task (19)	Technical challenging work (16)
5	Team quality (9)	Development needs addressed (14)
6	Project success (23)	Variety of work (21)
7	Autonomy (3)	Successful organization (13)
8	Equity (4)	Creativity/Innovation (12)
	FOSS (Rank in Catrobat)	**Education** (Rank in Catrobat)
1	Learning (5)	Grading (22)
2	Status/Reputation (26)	Learning (5)
3	Trust/Respect (1)	Technical challenging work (16)
4	Employee participation (18)	Team quality (9)
5	Team quality (9)	Meet new people (20)
6	Career Path (8)	Project success (23)
7	Financial benefits (30)	Empowerment/Responsibility (17)

Table 9.8.: Relevant motivators used for categorization of Catrobat (O. Melo, Santana, and Kon, 2012; Bosnić et al., 2011; Wang, He, and J. Chen, 2005; Ye and Kishida, 2003; França and Fabio Q. B. da Silva, 2010)

equity are motivators with a very high presence in Catrobat and also in an agile setting as identified by O. Melo, Santana, and Kon, 2012. These motivators with a focus on social interactions are given in both settings. They are responsible for the best matching category in the second case.

Education: In the first case education is the category with the third best matching. In the second case it has the weakest matching with Catrobat as the mean value *16* indicates. This is a really unexpected result. Catrobat has its headquarter at Graz University of Technology. Thus management, infrastructure and most manpower is there. However, the education category has a poor matching in this categorization via motivators. Motivators used for this category with a very low presence in Catrobat are *grading*, *meet new people*, *project success* or *responsibility*. The low rank of *grading* is surprising. A major fraction of all Catrobat members are students and they are getting a grade for their participation. Thus the low rank of grading is unexpected.

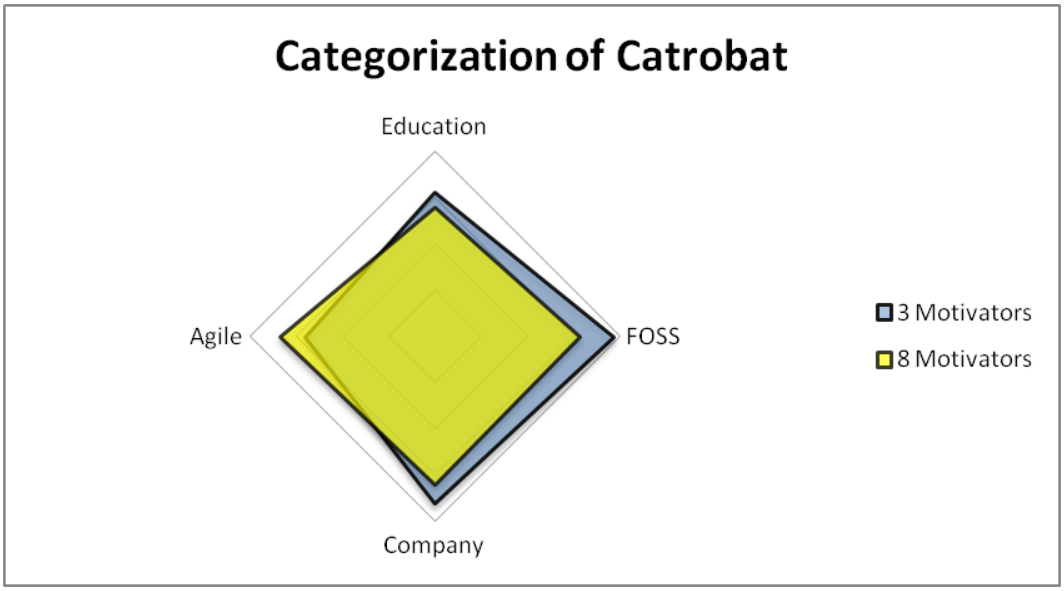

Figure 9.51.: Categorization of Catrobat regarding motivation

#	Case 1 (Mean of 3 Motivators)	Case 2 (Mean of 8 Motivators)
1	FOSS (10.67)	Agile (13.25)
2	Company (12)	FOSS (13.86)
3	Education (14.33)	Company (14)
4	Agile (16)	Education (16)

Table 9.9.: Ranking of categorization of Catrobat

In contrast, the motivator *learning* has a high rank in Catrobat and in educational settings. Catrobat members are learning a lot during their participation and they experience it as motivating.

FOSS: In both cases of this categorization, FOSS has a good matching with Catrobat. In one case it has the best matching and in the other one it has the second best matching. Catrobat claims, that it is a FOSS project (Catrobat Developers Team, 2016a). This categorization regarding motivation of the Catrobat project, confirms this statement. There are several relevant motivators in a FOSS setting, which also have a high rank in the Catrobat project. These motivators are *learning*, *trust / respect*, *team quality* and *career*

path. The motivator *status / reputation* is relevant in FOSS settings (Ye and Kishida, 2003; Riehle, 2007), but in Catrobat this motivator has a very low rank. *Status / reputation* seems to be a less present motive for participations in Catrobat. For people who are already in the community, *status / reputation* is not a present motivator. In contrast, in typical FOSS communities, *status / reputation* is a primary reason for participation and motivation in such communities (Ye and Kishida, 2003). This is a remarkable difference between Catrobat and typical FOSS projects.

Company: In Catrobat there are also motivators present which are relevant in company settings. The category company has in one case the second best matching and in the other case the third best matching with Catrobat. Relevant motivators in company settings also have a high presence in Catrobat. For example *work life balance, team work, creativity / innovation* and *successful organization* are relevant in both settings. *Work life balance* and *creativity / innovation* are motivators, which are also very present in Catrobat. Catrobat members can schedule their time working for Catrobat. They also have several possibilities to be creative and innovative. If Catrobat members have a good idea, they can discuss it with their coordinator or with Wolfgang Slany. If the idea is promising, they have good chances to get the permission to realize this idea. The motivator *successful organization* is relevant in both settings. Catrobat members evaluate Catrobat as an successful organization, thus this motivator has a relatively high rank in Catrobat.

Catrobat has characteristics of all categories. Some categories are matching better and some are matching less. Motivators like *meet new people, project success, grading, status / reputation* and *financial benefits* offer a potential for improvement. They have a relatively low rank in Catrobat, but are identified as relevant in the settings used for categorization.

However, this categorization has to be regarded carefully. Motivators used within this categorization, are identified by different teams of researchers. The motivator set used by them differs. Only motivators which are within the set of this thesis and are also in corresponding sets of motivators, are used for this categorization. In several settings, some motivators are not presented or are not investigated at all. This is one limitation of this categorization. Another limitation is the relative ranking of motivators in

the Catrobat project. Motivators can have a certain presence, but they can have a relatively low rank.

9.4. Findings and Suggestions

In this section findings of the survey are analyzed and presented. Afterwards suggestions which offer approaches for improvement in the Catrobat project are described.

9.4.1. Survey Findings

Catrobat pros: Participants report several positive factors. For example they highlight *teamwork*, *autonomy*, *learning*, *infrastructure* or *organization* as positive.

Catrobat cons: There are also some factors, which are not liked by participants. For example some factors are *progress*, *onboarding*, *motivation of others* or *documentation*.

Motivators: Some motivators are very present in Catrobat and also have a very high motivational effect like *trust / respect* and *working conditions*. There are also motivators with a high presence in Catrobat, but with a less motivational effect. For example *autonomy*, *equity* and *management / coordination* are such motivators. These motivators with a low presence in Catrobat and a higher motivational effect are also identified like *financial benefits*, *feedback* and *recognition*. Motivators with a low presence in Catrobat, but with a relatively high motivational effect, offer potentials for improvement. Thus strategies can be developed and implemented to increase important motivators' presence.

Demotivators: There are also some demotivators present in Catrobat. Demotivators with a relative high presence in Catrobat are *poor communication*, *organizational overhead* or *poor quality software*. *Poor quality software* is the motivator with the third highest presence in Catrobat. Quality plays an important role in XP. There is a mismatch between what Catrobat members notice and what is important in XP. A possible cause for poor quality software in Catrobat can be inexperienced Catrobat members.

Work experience: A large part of all Catrobat members have work experience in the IT industry. This potential can be used to make Catrobat more

professional. They can share their knowledge and skills. For example the software development process can be adapted.

Communication: Communication and information exchange across teams has been identified as an issue. In this case communication between teams should be restructured and considered for improvement.

IRC: IRC is used only rarely by Catrobat members. The usability of it is uncomfortable, in the opinion of several members. Sub-teams are using their own communication tools. For example they are using Skype, Slack or WhatsApp. This can be the reason why communication in teams is working well and across teams it is not. The usage of different communication tools has a negative effect on communication across teams.

Team meetings: Regular team meetings are very positive and useful for team members. Some possibilities for improvement are suggested by participants. For example attendance or distribution of tasks can be improved.

BiWeCo meetings: BiWeCo meetings are also assessed as positive, but a little bit less than regular team meetings. In this meeting an overview of all sub-teams is given. If coordinators present their actual state in a useful way, all attending members have an overview of the whole project. This information is not always propagated to all team members who are not attending BiWeCo meetings. Thus they are not really noticing progress in other teams.

Coordination: Coordination is assessed as very positive. Coordinators are doing a good job. They have a good availability and are supporting newcomers. Some potentials for improvement concerning coordination are *giving feedback, communication with team members* or *distribution of information*. Especially, *giving feedback* is identified as a possibility for improvement in several questions.

Onboarding: Survey results concerning onboarding are rather poor. For many Catrobat members, the start into the project has been difficult. This can be caused by several factors. For example a missing documentation and a large codebase are identified as hurdles. Some members report missing general information about Catrobat at the beginning. Tutorials at the beginning are assessed as positive.

Improvement: Out of five given ideas for improvement, *retrospectives* are seen as the best idea by participants. As a consequence, a guide for retrospectives is created and appended in the Appendix. This guide can be used to conduct retrospectives within sub-teams. Retrospectives in general are described in Section 4.3.

9.4.2. Suggestions

In this section suggestions concerning different topics are presented. Some of them can be adapted or implemented to improve motivation in the Catrobat project.

Coordination:

- Information distribution to all team members: For example information from BiWeCo meetings.
- Enforce pair programming within teams
- Stricter supervision of team members
- Demand quality instead of quantity from team members
- Enforce feedback during participation from coordinator to team members: For example after every 150 hours working on the project.
- Make objective agreements between coordinators and team members
- Create documentation if necessary
- Improve communication to other teams
- Hand out a certificate for competent coordinators

Communication:

- Replace IRC with another communication tool
- Enforce teams to use this new standard communication tool
- Info newsletter for all Catrobat members for important general information
- Make info meetings to report relevant achievements and further plans
- Communicate middle and long term objectives of Catrobat
- Introduce a shared public timeline for all sub-teams
- Introduce a Catrobat newsgroup
- Enforce all leaving members to give feedback

- More information about user feedback and feedback on Google Play

Onboarding:

- Introduction / get to know meeting for newcomers
- Assign competent mentor to newcomers
- Restructure *getting started* guide
- More mandatory basic tutorials: For example about agile, XP or GitHub
- Team-specific tutorials for every sub-team
- Inform potential newcomers about required skills
- Enforce promotion in FOSS scene

Knowledge:

- Introduce speakers corners to share knowledge
- Restructure documents in Confluence
- Enforce comments in source code for critical sections
- Abstract developer documentation for Catrobat implementation

Improvement:

- Improve collaboratively
- Introduce retrospectives in all sub-teams
- Limit work in progress
- Coach or Scrum Master to ensure high quality in meetings
- Impart basic skills like XP or Git
- Yearly Catrobat event for promotion or team building
- More promotion at Graz University of Technology
- Implement a more modern usability interface for Catrobat applications
- More possibilities for financial compensation

10. Conclusion and Future Work

This thesis addresses motivational factors and corresponding pitfalls in a hybrid software project. A survey has been done to gain an insight into Catrobat members' attitudes and to get an overall picture of motivation in the Catrobat project. Motivation in the Catrobat team is okay, but findings of this thesis offer various potentials for improvement and further investigations.

Several topics are done very well in Catrobat. For example coordination and team meetings are assessed as very positive. Motivators with the highest presence in Catrobat project are *trust / respect*, *appropriate working conditions*, *autonomy*, *equity*, *learning* and *management / coordination*. Motivators with the least presence in Catrobat are *professionalism*, *financial benefits* and *relationship with users / customers*. The low scores of these motivators are not really surprising. Especially, *professionalism* and *financial benefits* are not present, because Catrobat is a FOSS project in an educational context and most Catrobat members are students who are contributing to the project during their study.

The most present demotivators in the Catrobat project are *poor communication*, *organizational overhead*, *poor quality software*, *unrealistic goals* and *task complexity is too difficult*. These demotivators offer a potential to be further investigated.

There is a potential to increase motivation in the Catrobat project. As the application of the Herzberg theory indicates, the Catrobat project offers conditions to avoid dissatisfaction, but there is still a potential to increase Catrobat members' motivation.

There are also some general aspects which offer potentials for improvement. Communication, information exchange and IRC are such aspects. Onboarding of newcomers is also seen as problematic. Many Catrobat members

report a difficult start into the project. For these aspects strategies can be developed to improve them.

Catrobat members have worked on the project for an average of 411 hours. They would like to do more pair programming and would like to have more documentation, than they have at the moment.

Retrospectives are seen as the most promising improvement of the suggested ideas. Therefore a guide for retrospectives with exemplary retrospectives is created. Maybe it is possible to enable continuous improvement with retrospectives in the whole Catrobat project.

Catrobat is a mixture of agile, FOSS and education. Catrobat can not be categorized completely in one of these categories. It is a hybrid project and all these categories have a strong impact on it. The category education has a surprisingly poor matching regarding motivation with Catrobat.

This thesis collected a huge amount of data. This data can be further interpreted and investigated. In the following section some possibilities for future work are described.

Future Work

Various possibilities for future work can be done with the data collected from the survey. For example having a closer look at topics like coordination, communication or onboarding. These topics can be investigated further aided by interviews or short surveys.

A lot of suggestions are listed at the end of this thesis. They offer a potential for further investigations and evaluations. If they seem promising, they can be discussed and strategies can be shaped to implement them.

The same survey can be used to gather data at other points in time. It is also possible to take only a part of the survey, because some participants recommended to shorten the survey. Repeating the motivation part of the survey can be used to observe the evolution of motivation in the Catrobat project.

Measureable outcomes should also be monitored in the project. For example newcomer rates, completed story points in an iteration or worked time on a sub-project in an iteration could be measured by management. It would also be helpful to use the MOCC model as a tool for monitoring in the project.

Introducing retrospectives to all sub-teams in the Catrobat project is a further possibility for future work. After using retrospectives for a certain time, they can be assessed by Catrobat members. They offer a great chance for organizing the process for improvement within teams. Teams can improve their processes by themselves via retrospectives.

With this huge amount of data, there are various ways to do further work. Improvement should be a continuous process. In such a large project, it is crucial to pay attention to the human factor and to continuous improvement.

Appendix

Bibliography

Abdullah, E. and E. T. B. Abdelsatir (2013). "Extreme programming applied in a large-scale distributed system." In: *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, pp. 442–446. DOI: 10.1109/ICCEEE.2013.6633979 (cit. on pp. 17, 20, 21).

Acuña, Silvia T., Marta Gómez, and Natalia Juristo (2009). "How Do Personality, Team Processes and Task Characteristics Relate to Job Satisfaction and Software Quality?" In: *Inf. Softw. Technol.* 51.3, pp. 627–639. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2008.08.006. URL: http://dx.doi.org/10.1016/j.infsof.2008.08.006 (cit. on p. 44).

Ahmad, M. O., J. Markkula, and M. Oivo (2013). "Kanban in software development: A systematic literature review." In: *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, pp. 9–16. DOI: 10.1109/SEAA.2013.28 (cit. on pp. 22–24).

Alice Game Jam Partners (2016). *Alice Game Jam: Design a game or tell a story*. Alice Game Jam Partners. URL: http://www.alicegamejam.com/ (cit. on p. 29).

Asghar, I. and M. Usman (2013). "Motivational and De-motivational Factors for Software Engineers: An Empirical Investigation." In: *Frontiers of Information Technology (FIT), 2013 11th International Conference on*, pp. 66–71. DOI: 10.1109/FIT.2013.20 (cit. on pp. 36, 45).

Atlassian Team (2016). *Bitbucket: Code, Manage, Collaborate*. Atlassian. Sydney, Australia. URL: https://bitbucket.org/ (cit. on p. 1).

Bässler, R. (2014). *Quantitative Forschungsmethoden: ein Leitfaden zur Planung und Durchführung quantitativer empirischer Forschungsarbeiten*. Reihe: Wissenschaftliches Arbeiten / Reihe: Wissenschaftliches Arbeiten. RB Research- & Consulting-Verlag. ISBN: 9783950266016 (cit. on p. 71).

Beck, Kent (2004). *Extreme Programming – Das Manifest*. München: Addison-Wesley. ISBN: 3827321395 (cit. on pp. 18–21).

Beck, Kent et al. (2001). *Manifesto for Agile Software Development*. URL: http://www.agilemanifesto.org/ (cit. on pp. 2, 50, 59).

Beecham, Sarah et al. (2008). "Motivation in Software Engineering: A systematic literature review." In: *Information and Software Technology* 50.9–10, pp. 860–878. ISSN: 0950-5849. DOI: http://dx.doi.org/10.1016/j.infsof.2007.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0950584907001097 (cit. on pp. 6, 7, 36, 44, 52–55, 66, 146, 150, 180–183).

Beecham, S. et al. (2007). "Does the XP environment meet the motivational needs of the software developer? An empirical study." In: *Agile Conference (AGILE), 2007*, pp. 37–49. DOI: 10.1109/AGILE.2007.22 (cit. on pp. 5, 6, 36).

Bertram, Dane (2007). "Likert scales." In: *Retrieved November* 2, p. 2013 (cit. on p. 72).

Bishop, D. and A. Deokar (2014). "Toward an Understanding of Preference for Agile Software Development Methods from a Personality Theory Perspective." In: *2014 47th Hawaii International Conference on System Sciences*, pp. 4749–4758. DOI: 10.1109/HICSS.2014.583 (cit. on p. 17).

Boone, Harry N and Deborah A Boone (2012). "Analyzing likert data." In: *Journal of extension* 50.2, pp. 1–5 (cit. on p. 72).

Bosnić, Ivana et al. (2011). "Student Motivation in Distributed Software Development Projects." In: *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*. CTGDSD '11. Waikiki, Honolulu, HI, USA: ACM, pp. 31–35. ISBN: 978-1-4503-0590-7. DOI: 10.1145/1984665.1984672. URL: http://doi.acm.org/10.1145/1984665.1984672 (cit. on pp. 62, 63, 138, 152, 156, 157, 185).

Catrobat Developers Team (2016a). *Catrobat: Free educational apps for children and teenagers.* International Catrobat Association. Graz, Austria. URL: http://developer.catrobat.org/ (cit. on pp. 29, 34, 158).

Catrobat Developers Team (2016b). *Catrobat Github*. International Catrobat Association. Graz, Austria. URL: https://github.com/Catrobat (cit. on p. 34).

Catrobat Developers Team (2016c). *Catrobat License*. International Catrobat Association. Graz, Austria. URL: http://developer.catrobat.org/licenses (cit. on pp. 34, 35, 146).

Catrobat Developers Team (2016d). *Catrobat Web: Platform for sharing programs.* International Catrobat Association. Graz, Austria. URL: https://share.catrob.at/ (cit. on p. 28).

Catrobat Team (2016). *Pocket Code: Programmiere Apps*. International Catrobat Association. Graz, Austria. URL: https://play.google.com/store/apps/details?id=org.catrobat.catroid&hl=de (cit. on p. 31).

César, A. et al. (2012). "Towards an explanatory theory of motivation in software engineering: A qualitative case study of a government organization." In: *Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on*, pp. 72–81. DOI: 10.1049/ic.2012.0010 (cit. on pp. 39, 45).

Chen, P. C., C. C. Chern, and C. Y. Chen (2012). "Software Project Team Characteristics and Team Performance: Team Motivation as a Moderator." In: *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*. Vol. 1, pp. 565–570. DOI: 10.1109/APSEC.2012.152 (cit. on pp. 36, 43, 44, 46).

Chow, Tsun and Dac-Buu Cao (2008). "A Survey Study of Critical Success Factors in Agile Software Projects." In: *J. Syst. Softw.* 81.6, pp. 961–971. ISSN: 0164-1212. DOI: 10.1016/j.jss.2007.08.020. URL: http://dx.doi.org/10.1016/j.jss.2007.08.020 (cit. on pp. 18, 50, 54, 59).

Cockburn, A. and J. Highsmith (2001). "Agile software development, the people factor." In: *Computer* 34.11, pp. 131–133. ISSN: 0018-9162. DOI: 10.1109/2.963450 (cit. on pp. 2, 17, 50, 59).

Comelli, G., L. Rosenstiel, and F.W. Nerdinger (2014). *Führung durch Motivation: Mitarbeiter für die Ziele des Unternehmens gewinnen*. Management Competence. Vahlen. ISBN: 9783800648405. URL: https://books.google.at/books?id=1%5C_6aBQAAQBAJ (cit. on pp. 36–45, 51).

Csikszentmihalyi, Mihaly and Judith LeFevre (1989). "Optimal experience in work and leisure." In: *Journal of personality and social psychology* 56.5, p. 815 (cit. on p. 51).

Derby, Esther and Diana Larsen (2006). *Agile retrospectives : Making good teams great*. Raleigh, North Carolina, Dallas, Texas: The Pragmatic Bookshelf. ISBN: 0-9776166-4-9 (cit. on pp. 25–27, 187, 189).

Ellis, H. J. C. et al. (2010). "Panel 2014; Teaching students to participate in Open Source Software projects." In: *Frontiers in Education Conference (FIE), 2010 IEEE*. DOI: 10.1109/FIE.2010.5673437 (cit. on pp. 16, 62).

Ellis, H.J.C. et al. (2011). "How to involve students in FOSS projects." In: *Frontiers in Education Conference (FIE), 2011*, T1H1–T1H6. DOI: 10.1109/FIE.2011.6142994 (cit. on pp. 1, 8, 16, 61, 62).

Epping, Thomas (2011). *Kanban für die Softwareentwicklung*. Informatik im Fokus. Heidelberg: Springer. ISBN: 978-3-642-22594-9. DOI: 10.1007/978-3-642-22595-6 (cit. on pp. 19, 21–23).

Fagerholm, F. et al. (2013). "Onboarding in Open Source Software Projects: A Preliminary Analysis." In: *2013 IEEE 8th International Conference on Global Software Engineering Workshops*, pp. 5–10. DOI: 10.1109/ICGSEW.2013.8 (cit. on pp. 1, 13).

Farias Junior, I.H. de et al. (2012). "Motivational Factors for Distributed Software Development Teams." In: *Global Software Engineering Workshops (ICGSEW), 2012 IEEE Seventh International Conference on*, pp. 49–54. DOI: 10.1109/ICGSEW.2012.17 (cit. on pp. 39, 40, 44).

Fitzgerald, B. and P. J. Agerfalk (2005). "The Mysteries of Open Source Software: Black and White and Red All Over?" In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 196a–196a. DOI: 10.1109/HICSS.2005.609 (cit. on p. 10).

Franca, A. C. C., A. C. M. L. de Araujo, and F. Q. B. da Silva (2013). "Motivation of software engineers: A qualitative case study of a research and development organisation." In: *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on*, pp. 9–16. DOI: 10.1109/CHASE.2013.6614726 (cit. on p. 36).

França, A. César C. and Fabio Q. B. da Silva (2010). "Designing Motivation Strategies for Software Engineering Teams: An Empirical Study." In: *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*. CHASE '10. Cape Town, South Africa: ACM, pp. 84–91. ISBN: 978-1-60558-966-4. DOI: 10.1145/1833310.1833324. URL: http://doi.acm.org/10.1145/1833310.1833324 (cit. on pp. 2, 36, 37, 40, 44, 45, 58, 153, 156, 157, 185).

Franca, A.C.C., D.E.S. Carneiro, and F.Q.B. da Silva (2012). "Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Small Software Company." In: *Software Engineering (SBES), 2012 26th Brazilian Symposium on*, pp. 61–70. DOI: 10.1109/SBES.2012.28 (cit. on pp. 36, 39, 43, 45).

Franca, A.C.C., T.B. Gouveia, et al. (2011). "Motivation in software engineering: A systematic review update." In: *Evaluation Assessment in Software*

Engineering (EASE 2011), 15th Annual Conference on, pp. 154–163. DOI: 10.1049/ic.2011.0019 (cit. on pp. 7, 9, 36, 53–55, 66, 74, 146, 150, 180–183).

Frangos, S.A. (1998). "Motivated humans for reliable software products." In: *Microprocessors and Microsystems* 21.10. Achieving Quality, Safety, pp. 605–610. ISSN: 0141-9331. DOI: http://dx.doi.org/10.1016/S0141-9331(98)00062-3. URL: http://www.sciencedirect.com/science/article/pii/S0141933198000623 (cit. on p. 2).

Gardazi, S. U. et al. (2009). "Motivation in software architecture and software project management." In: *Emerging Technologies, 2009. ICET 2009. International Conference on*, pp. 403–409. DOI: 10.1109/ICET.2009.5353138 (cit. on p. 67).

GitHub Team (2016). *GitHub: How people build software*. GitHub, Inc. San Francisco. URL: https://github.com/ (cit. on p. 1).

Gokhale, S. S., T. Smith, and R. McCartney (2012). "Integrating Open Source Software into software engineering curriculum: Challenges in selecting projects." In: *Software Engineering Education based on Real-World Experiences (EduRex), 2012 First International Workshop on*, pp. 9–12. DOI: 10.1109/EduRex.2012.6225697 (cit. on p. 16).

Goldberg, Lewis R. "A broad-bandwidth, public-domain, personality inventory measuring the lower-level facets of several five-factor models." In: *Personality Psychology in Europe, Vol. 7*. Ed. by Mervielde et al. Tilburg University Press. URL: http://ipip.ori.org/ipip/newBroadbandText.htm (cit. on pp. 47–49).

Goldberg, Lewis R. et al. (2006). "The international personality item pool and the future of public-domain personality measures." In: *Journal of Research in Personality* 40.1. Proceedings of the 2005 Meeting of the Association of Research in PersonalityAssociation of Research in Personality, pp. 84–96. ISSN: 0092-6566. DOI: http://dx.doi.org/10.1016/j.jrp.2005.08.007. URL: http://www.sciencedirect.com/science/article/pii/S0092656605000553 (cit. on pp. 47–49).

Gosling, Samuel D, Peter J Rentfrow, and William B Swann Jr. (2003). "A very brief measure of the Big-Five personality domains." In: *Journal of Research in Personality* 37.6, pp. 504–528. ISSN: 0092-6566. DOI: http://dx.doi.org/10.1016/S0092-6566(03)00046-1. URL: http://www.sciencedirect.com/science/article/pii/S0092656603000461 (cit. on pp. 46, 47, 74, 82, 147).

Hall, Tracy et al. (2009). "A Systematic Review of Theory Use in Studies Investigating the Motivations of Software Engineers." In: *ACM Trans. Softw. Eng. Methodol.* 18.3, 10:1–10:29. ISSN: 1049-331X. DOI: 10.1145/1525880.1525883. URL: http://doi.acm.org/10.1145/1525880.1525883 (cit. on pp. 39, 40, 45, 180, 184).

Hars, Alexander and Shaosong Ou (2002). "Working for Free? Motivations for Participating in Open-Source Projects." In: *Int. J. Electron. Commerce* 6.3, pp. 25–39. ISSN: 1086-4415. URL: http://dl.acm.org/citation.cfm?id=1286960.1286963 (cit. on pp. 10, 40, 60, 61).

Hazzan, Orit and Yael Dubinsky (2010). "The 2Nd Workshop on Human Aspects of Software Engineering (HAoSE2010)." In: *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*. OOPSLA '10. Reno/Tahoe, Nevada, USA: ACM, pp. 271–272. ISBN: 978-1-4503-0240-1. DOI: 10.1145/1869542.1869608. URL: http://doi.acm.org/10.1145/1869542.1869608 (cit. on p. 5).

Hertel, Guido, Sven Niedner, and Stefanie Herrmann (2003). "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel." In: *Research Policy* 32.7. Open Source Software Development, pp. 1159–1177. ISSN: 0048-7333. DOI: http://dx.doi.org/10.1016/S0048-7333(03)00047-7. URL: http://www.sciencedirect.com/science/article/pii/S0048733303000477 (cit. on pp. 43, 51).

House, Robert J. and Lawrence A. Wigdor (1967). "HERZBERG'S DUAL-FACTOR THEORY OF JOB SATISFACTION AND MOTIVATION: A REVIEW OF THE EVIDENCE AND A CRITICISM." In: *Personnel Psychology* 20.4, pp. 369–390. ISSN: 1744-6570. DOI: 10.1111/j.1744-6570.1967.tb02440.x. URL: http://dx.doi.org/10.1111/j.1744-6570.1967.tb02440.x (cit. on p. 46).

Jaccheri, L. and T. Osterlie (2007). "Open Source Software: A Source of Possibilities for Software Engineering Education and Empirical Software Engineering." In: *Emerging Trends in FLOSS Research and Development, 2007. FLOSS '07. First International Workshop on*, pp. 5–5. DOI: 10.1109/FLOSS.2007.12 (cit. on p. 62).

Judge, Timothy A. et al. (1999). "THE BIG FIVE PERSONALITY TRAITS, GENERAL MENTAL ABILITY, AND CAREER SUCCESS ACROSS THE LIFE SPAN." In: *Personnel Psychology* 52.3, pp. 621–652. ISSN: 1744-6570.

DOI: 10.1111/j.1744-6570.1999.tb00174.x. URL: http://dx.doi.org/10.1111/j.1744-6570.1999.tb00174.x (cit. on pp. 47–49).

Kuechler, Victor, Claire Gilbertson, and Carlos Jensen (2012). "Gender differences in early free and open source software joining process." In: *IFIP International Conference on Open Source Systems*. Springer, pp. 78–93 (cit. on p. 78).

Lenberg, P., R. Feldt, and L.G. Wallgren (2015). "Human Factors Related Challenges in Software Engineering – An Industrial Perspective." In: *Cooperative and Human Aspects of Software Engineering (CHASE), 2015 IEEE/ACM 8th International Workshop on*, pp. 43–49. DOI: 10.1109/CHASE.2015.13 (cit. on pp. 5, 44, 50).

Leopold, K. and S. Kaltenecker (2013). *Kanban in der IT: Eine Kultur der kontinuierlichen Verbesserung schaffen*. Hanser Fachbuchverlag. ISBN: 9783446438262. URL: https://books.google.at/books?id=pdZUnwEACAAJ (cit. on pp. 22–24).

Li, Yan, Chuan-Hoo Tan, Hock-Hai Teo, et al. (2006). "Motivating Open Source Software Developers: Influence of Transformational and Transactional Leaderships." In: *Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges &Amp; the Future*. SIGMIS CPR '06. Claremont, California, USA: ACM, pp. 34–43. ISBN: 1-59593-349-2. DOI: 10.1145/1125170.1125182. URL: http://doi.acm.org/10.1145/1125170.1125182 (cit. on p. 40).

Li, Yan, Chuan-Hoo Tan, Heng Xu, et al. (2011). "Open Source Software Adoption: Motivations of Adopters and Amotivations of Non-adopters." In: *SIGMIS Database* 42.2, pp. 76–94. ISSN: 0095-0033. DOI: 10.1145/1989098.1989103. URL: http://doi.acm.org/10.1145/1989098.1989103 (cit. on p. 43).

LimeSurvey Project Team / Carsten Schmitz (2016). *LimeSurvey: An Open Source survey tool*. LimeSurvey Project. Hamburg, Germany. URL: http://www.limesurvey.org (cit. on pp. 71, 72).

Maham, M. (2008). "Planning and Facilitating Release Retrospectives." In: *Agile, 2008. AGILE '08. Conference*, pp. 176–180. DOI: 10.1109/Agile.2008.60 (cit. on p. 25).

Martin, Robert C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR. ISBN: 0132350882, 9780132350884 (cit. on p. 32).

Melnik, Grigori and Frank Maurer (2006). "Comparative Analysis of Job Satisfaction in Agile and Non-agile Software Development Teams." In: *Proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering*. XP'06. Oulu, Finland: Springer-Verlag, pp. 32–42. ISBN: 3-540-35094-2, 978-3-540-35094-1. DOI: 10.1007/11774129_4. URL: http://dx.doi.org/10.1007/11774129_4 (cit. on pp. 2, 59, 60).

O. Melo, C. de, C. Santana, and F. Kon (2012). "Developers Motivation in Agile Teams." In: *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, pp. 376–383. DOI: 10.1109/SEAA.2012.45 (cit. on pp. 44, 59, 60, 152, 156, 157, 185).

Probst, Gilbert and Sebastian Raisch (2005). "Organizational crisis: The logic of failure." In: *The Academy of Management Executive* 19.1, pp. 90–105 (cit. on p. 2).

Raymond, Eric S. (1999). *The Cathedral and the Bazaar*. Ed. by Tim O'Reilly. 1st. Sebastopol, CA, USA: O'Reilly & Associates, Inc. ISBN: 1565927249 (cit. on pp. 10, 61).

Rehman, M. et al. (2011). "Motivation in software engineering amp; personal characteristics of software engineers." In: *National Postgraduate Conference (NPC), 2011*, pp. 1–5. DOI: 10.1109/NatPC.2011.6136312 (cit. on pp. 39, 52).

Riehle, Dirk (2007). "The Economic Motivation of Open Source Software: Stakeholder Perspectives." In: *Computer* 40.4, pp. 25–32. ISSN: 0018-9162. DOI: 10.1109/MC.2007.147. URL: http://dx.doi.org/10.1109/MC.2007.147 (cit. on pp. 10, 159).

Roach, S. (2011). "Retrospectives in a software engineering project course: Getting students to get the most from a project experience." In: *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE T)*, pp. 467–471. DOI: 10.1109/CSEET.2011.5876126 (cit. on pp. 24, 25).

Al-Rubaish, Abdullah. et al. (2011). "Academic job satisfaction questionnaire: Construction and validation in Saudi Arabia." In: *Journal of Family and Community Medicine* 18.1, pp. 1–7. DOI: 10.4103/1319-1683.78630 (cit. on p. 74).

Sach, R., H. Sharp, and M. Petre (2011). "Software Engineers' Perceptions of Factors in Motivation: The Work, People, Obstacles." In: *2011 International Symposium on Empirical Software Engineering and Measurement*, pp. 368–371. DOI: 10.1109/ESEM.2011.50 (cit. on pp. 58, 153, 185).

Said, N. A. and R. Munap (2010). "Job characteristics and job satisfaction: A relationship study on supervisors performance." In: *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, pp. 714–719. DOI: 10.1109/ICMIT.2010.5492732 (cit. on p. 46).

Sertic, H., K. Marzic, and Z. Kalafatic (2007). "A Project Retrospectives Method in Telecom Software Development." In: *2007 9th International Conference on Telecommunications*, pp. 109–114. DOI: 10.1109/CONTEL.2007.381858 (cit. on p. 25).

Shamir, Boas and Jane M Howell (1999). "Organizational and contextual influences on the emergence and effectiveness of charismatic leadership." In: *The Leadership Quarterly* 10.2, pp. 257–283 (cit. on p. 2).

Sharp, H. and T. Hall (2009). "An initial investigation of software practitioners' motivation." In: *Cooperative and Human Aspects on Software Engineering, 2009. CHASE '09. ICSE Workshop on*, pp. 84–91. DOI: 10.1109/CHASE.2009.5071418 (cit. on pp. 43, 51).

Sharp, Helen et al. (2009). "Models of motivation in software engineering." In: *Information and Software Technology* 51.1. Special Section - Most Cited Articles in 2002 and Regular Research Papers, pp. 219–233. ISSN: 0950-5849. DOI: http://dx.doi.org/10.1016/j.infsof.2008.05.009. URL: http://www.sciencedirect.com/science/article/pii/S0950584908000827 (cit. on pp. 38–40, 46, 52, 53, 55–57, 146).

SourceForge Team (2016). *SourceForge: Download, Develop and Publish Open Source software for free*. Slashdot Media. URL: https://sourceforge.net/ (cit. on p. 1).

Steinberga, L. and D. Smite (2011). "Towards Understanding of Software Engineer Motivation in Globally Distributed Projects." In: *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*, pp. 117–119. DOI: 10.1109/ICGSE-W.2011.31 (cit. on pp. 39, 44, 45).

Šteinberga, Liva and Darja Šmite (2011). "Towards a contemporary understanding of motivation in distributed software projects: solution proposal." In: *Datorzintne un informcijas tehnologijas*, p. 15 (cit. on p. 60).

Steinmacher, I., A. P. Chaves, et al. (2014). "Preliminary Empirical Identification of Barriers Faced by Newcomers to Open Source Software Projects." In: *Software Engineering (SBES), 2014 Brazilian Symposium on*, pp. 51–60. DOI: 10.1109/SBES.2014.9 (cit. on pp. 10, 13, 15, 60).

Steinmacher, I., I. Wiese, et al. (2015). "Increasing the Self-Efficacy of Newcomers to Open Source Software Projects." In: *Software Engineering*

(SBES), 2015 29th Brazilian Symposium on, pp. 160–169. DOI: 10.1109/SBES.2015.10 (cit. on pp. 1, 8, 13, 16, 60).

Thatcher, Jason Bennett, Yongmei Liu, and Lee P. Stepina (2002). "The Role of the Work Itself: An Empirical Examination of Intrinsic Motivation's Influence on IT Workers Attitudes and Intentions." In: *Proceedings of the 2002 ACM SIGCPR Conference on Computer Personnel Research*. SIGCPR '02. Kristiansand, Norway: ACM, pp. 25–33. ISBN: 1-58113-466-5. DOI: 10.1145/512360.512365. URL: http://doi.acm.org/10.1145/512360.512365 (cit. on p. 46).

Tietjen, Mark A. and Robert M. Myers (1998). "Motivation and job satisfaction." In: *Management Decision* 36.4, pp. 226–231. DOI: 10.1108/00251749810211027 (cit. on p. 46).

Van Kelle, E. et al. (2015). "An Empirical Study into Social Success Factors for Agile Software Development." In: *Cooperative and Human Aspects of Software Engineering (CHASE), 2015 IEEE/ACM 8th International Workshop on*, pp. 77–80. DOI: 10.1109/CHASE.2015.24 (cit. on pp. 50, 59, 97).

Villarrubia, A. and Hyunju Kim (2015). "Building a community system to teach collaborative software development." In: *Computer Science Education (ICCSE), 2015 10th International Conference on*, pp. 829–833. DOI: 10.1109/ICCSE.2015.7250360 (cit. on p. 62).

Wang, Fei-Rong, Dan He, and Jin Chen (2005). "Motivations of Individuals and Firms Participating in Open Source Community." In: *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*. Vol. 1, pp. 309–314. DOI: 10.1109/ICMLC.2005.1526964 (cit. on pp. 10, 38, 43, 51, 61, 155–157, 185).

Yanyan, Z. and X. Renzuo (2008). "The Basic Research of Human Factor Analysis Based on Knowledge in Software Engineering." In: *Computer Science and Software Engineering, 2008 International Conference on*. Vol. 5, pp. 1302–1305. DOI: 10.1109/CSSE.2008.219 (cit. on pp. 5, 44).

Ye, Yunwen and K. Kishida (2003). "Toward an understanding of the motivation of open source software developers." In: *Software Engineering, 2003. Proceedings. 25th International Conference on*, pp. 419–429. DOI: 10.1109/ICSE.2003.1201220 (cit. on pp. 7, 10–13, 32, 60, 61, 138, 155–157, 159, 185).

Zhou, M. and A. Mockus (2015). "Who Will Stay in the FLOSS Community? Modeling Participant x2019;s Initial Behavior." In: *IEEE Transactions on*

Software Engineering 41.1, pp. 82–99. ISSN: 0098-5589. DOI: 10.1109/TSE.2014.2349496 (cit. on pp. 12, 60).

Appendix A.

Results of Relevant Literature

In this section, results of literature reviews in the field of motivation in SE are shown. These results are illustrated in tables. This data is elaborated from numerous of research documents in the field of motivation in SE. The relevance is measured by the number of relevant documents in literature. The literature review done by Sarah Beecham et al., 2008, analyzed relevant documents from 1980 to 2006. The literature review update done by A. Franca, Gouveia, et al., 2011, examined the time span from 2006 to 2010. These results are taken together in Tables A.1, A.2, A.3, A.4, A.5. Table A.6 illustrates used motivation theories in the field of SE (Tracy Hall et al., 2009).

	Motivator	1980-2006	2006-2010	Sum
M1	Rewards and incentives	14	3	17
M2	Development needs addressed	11	9	20
M3	Variety of work	14	5	19
M4	Career path	15	9	24
M5	Empowerment/responsibility	6	3	9
M6	Good management	16	4	20
M7	Sense of belonging	14	1	15
M8	Work/life balance	7	2	9
M9	Working in successful company	2	1	3
M10	Employee participation	16	10	26
M11	Feedback	10	3	13
M12	Recognition	12	3	15
M13	Equity	3	2	5
M14	Trust/respect	4	6	10
M15	Technically challenging work	11	4	15
M16	Job security/stable environment	10	2	12
M17	Identify with the task	20	6	26
M18	Autonomy	9	7	16
M19	Appropriate working conditions	6	0	6
M20	Making a contribution/task significance	6	3	9
M21	Sufficient resources	2	0	2
M22	Team quality	-	4	4
M23	Creativity/Innovation	-	4	4
M24	Fun (playing)	-	1	1
M25	Professionalism	-	2	2
M26	Having an Ideology	-	1	1
M27	Non-financial benefits	-	1	1
M28	Penalty Policies	-	1	1
M29	Good relationship with users/customers	-	2	2

Table A.1.: Motivators for software engineers identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011

	Demotivator	1980-2006	2006-2010	Sum
D1	Risk	1	0	1
D2	Stress	5	2	7
D3	Inequity	4	0	4
D4	Interesting work going to other parties	1	0	1
D5	Unfair reward system	2	1	3
D6	Lack of promotion	5	0	5
D7	Poor communication	5	1	6
D8	Uncompetitive pay	6	1	7
D9	Unrealistic goals	4	0	4
D10	Bad relationship with users and colleagues	4	0	4
D11	Poor working environment	9	1	10
D12	Poor management	7	1	8
D13	Producing poor quality software	3	0	3
D14	Poor cultural fit/stereotyping/role ambiguity	3	0	3
D15	Lack of influence	2	0	2
D16	Task Complexity (too easy or too difficult)	-	2	2

Table A.2.: Demotivators for software engineers identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011

	Aspect	1980-2006	2006-2010	Sum
A1	Problem solving	3	1	4
A2	Team working	2	1	3
A3	Change	4	0	4
A4	Challenge	4	1	5
A5	Benefit	3	0	3
A6	Science	2	0	2
A7	Experiment	2	0	2
A8	Development practices	2	2	4
A9	Lifecycle	1	0	1
A10	Creativity	-	1	1
A11	Relationships with users/customers	-	1	1

Table A.3.: Aspects of software engineers identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011

	Characteristic	1980-2006	2006-2010	Sum
Ch1	Need for stability	5	4	9
Ch2	Technically competent	5	4	9
Ch3	Achievement orientated	4	3	7
Ch4	Growth orientated	9	6	15
Ch5	Need for competent supervising	4	1	5
Ch6	Introverted	7	1	8
Ch7	Need for involvement in personal goal setting	1	0	1
Ch8	Need for feedback	2	3	5
Ch9	Need for Geographic stability	1	2	3
Ch10	Need to make a contribution	3	2	5
Ch11	Autonomous	7	9	16
Ch12	Need for variety	4	2	6
Ch13	Marketable	2	0	2
Ch14	Need for challenge	4	5	9
Ch15	Creative	2	2	4
Ch16	Need to be sociable	5	1	6
Ch17	Competent in Management	-	2	2
Ch18	Flexible / Team worker	-	2	2
Ch19	Have fear of punishment	-	1	1

Table A.4.: Characteristics of software engineers identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011

	External sign	1980-2006	2006-2010	Sum
E1	Retention	12	6	18
E2	Project Delivery Time	2	0	2
E3	Productivity	5	1	6
E4	Budgets	1	0	1
E5	Absenteeism	1	0	1
E6	Project success	1	0	1
E7	Organizational Commitment	-	3	3
E8	Benevolence	-	1	1

Table A.5.: External signs of motivated software engineers identified by Sarah Beecham et al., 2008; A. Franca, Gouveia, et al., 2011

	Theory	Number of Articles
1	Job Characteristics Theory	35
2	Herzberg's Motivational Theory	21
3	Maslow's Hierarchy of Needs	11
4	McClelland's Theory	9
5	Goal-setting Theory	9
6	Expectancy Theory	6
7	Equity Theory	5
8	Stimulus Response Theory	3

Table A.6.: Used theories in SE documents identified by Tracy Hall et al., 2009

Appendix B.

Motivator Mapping

In the following table motivator mappings are shown, which have been applied in this thesis to avoid using different names for a single motivator. Scientific works, where the motivator mapping is applied to: O. Melo, Santana, and Kon, 2012 (M), Bosnić et al., 2011 (Bo), Sach, H. Sharp, and Petre, 2011 (S), França and Fabio Q. B. da Silva, 2010 (F), Ye and Kishida, 2003 (Y), Wang, He, and J. Chen, 2005 (W).

Appendix B. Motivator Mapping

	Naming used in Thesis	**Other names**	**Other names**
1	Good relationship with users/customers	Relationship with users	
2	Team quality	Team working (M)	Team atmosphere (B)
3	Employee participation	Employee participation/involvement/ working with others (M)	
4	Project success	Working on a successful product (M)	
5	Management/Coordination	Good management (M)	
6	Grading	Grade (B)	
7	Technical challenging work	Challenging project (B)	Challenging goals (F)
8	Learning	Learning new things (B) (Y)	Learn and develop new skills (W)
9	Meet new people	New people/cultures (B) (W)	
10	Empowerment/Responsibility	Responsibility (B)	Empowerment (F)
11	Identify with the Task	Important tasks (S)	
12	Career path	Career goals (S)	Career development (F)
13	Work life balance	Balance between personal and professional life (F)	
14	Successful organization	Working in successful company (F)	
15	Creativity/Innovation	Creativity (F)	

Table B.1.: Motivator mapping for similar names

Appendix C.

Retrospective Preparation

In this guide important steps for performing retrospectives are discussed. Then two examples for performing a retrospective are described. This guide is adapted from Derby and Larsen, 2006.

Before: For the preparation of retrospectives, the retrospective leader has to think about topics like *outcome of the iteration, environment and history of the team, goal of the retrospective, duration of the retrospective* and *activities using in the particular steps of the retrospective* (Derby and Larsen, 2006).

During: During performing the retrospective, the retrospective leader has to have an eye on *group dynamics, time management, managing activities* and *managing the overall process*. He is responsible for the effective and efficient performing of the retrospective (Derby and Larsen, 2006).

Afterwards: After the retrospective, the retrospective leader has to remind the team about their working agreements and improvements which they have elaborated. He is also responsible to capture the results of the retrospective. For example he can create a poster with results and place it in the working room of the team.

Retrospective Planning Example One

Set the Stage: Lay the groundwork for the session by reviewing the goal, attendance and agenda. (3 minutes)

Gather Data: Activity - MAD, SAD, GLAD. Participants write down words, when they felt mad, sad or glad during the last iteration. If all participants have finished writing, everyone presents his own identifications. (20 minutes)

Generate Insights: Activity - Brainstorming / Filtering. Brainstorm causes and effects of the gathered data. Cluster words which belong together. After clustering, ideas for improvement are worked out via brainstorming. (13 minutes)

Decide What to Do: Decide together democratically which improvements are relevant for the next iterations. Select up to two improvements. (10 minutes)

Close the Retrospective: Have a recap on worked out improvements and thank people for their participation. (6 minutes)

Shuffle Time: Buffer and time for switching between steps (8 minutes)

Total: Outcome: Concrete working agreements (60 minutes)

Retrospective Planning Example Two

Set the Stage: Lay the groundwork for the session by reviewing the goal, attendance and agenda. (3 minutes)

Gather Data: Activity - Topic Radar. The retrospective leader creates scales for different topics on a flip chart. These topics can be: Teamwork, quality, iteration outcome, learning and other topics. Then participants assess the last iteration after these topics. (20 minutes)

Generate Insights: Activity - Prioritize with Dots. Have a look at all scales. Then let participants make dots to mark topics which seem to be the most interesting ones. This topics should be discussed in depth. At the end of this step, a brainstorming is done to identify activities and possible improvements for the chosen topics. (13 minutes)

Decide What to Do: Activity - Start, Stop, Stay. Then assign these activities and improvements, identified in the previous step, to the categories Start, Stop and Stay. With this categorization it is possible to discuss about starting,

stopping and staying activities or improvements. New activities can be started to counteract identified issues. Staying with activities or slightly modifying them can be done to keep useful activities. If activities are not longer needed or they are causing issues, they can be stopped. (10 minutes)

Close the Retrospective: Have a recap on worked out working agreements and thank people for their participation. (6 minutes)

Shuffle Time: Buffer and time for switching between steps (8 minutes)

Total: Outcome: Concrete working agreements (60 minutes)

Further information about retrospectives and a lot more activities can be found in literature: Derby and Larsen, 2006.

Appendix D.

Original Survey in German

In this section, the survey is appended, which has been used for gathering data from Catrobat members.

Herzlich Willkommen!

Deine Teilnahme ist anonym. Zum Aussenden von Erinnerungsmails wird lediglich gespeichert, wer die Umfrage ausgefüllt hat und wer nicht. Antworten können nicht einzelnen Personen zugeordnet werden. Bei der Angabe wann du mit Catrobat angefangen hast reicht das Monat oder das Quartal (damit sich niemand Sorgen über die Anonymität machen muss).

Das Zwischenspeichern der Umfrage erfolgt durch Klick auf den 'Später fortfahren' Button in der linken unteren Ecke. Danach musst du einen Benutzernamen und ein Passwort angeben (Achtung Passwort wird in Plain Text gespeichert).

Bitte nimm dir auch Zeit, um Fragen mit Freitextfeldern zu beantworten - je mehr Informationen wir von dir bekommen desto besser.

Los gehts.

Teil A: Demographie

A1. Was ist deine Staatszugehörigkeit?

- [ ] Österreich
- [ ] Deutschland
- [ ] Slowenien
- [ ] Kroatien
- [ ] Mazedonien
- [ ] Bosnien und Herzegowina
- [ ] Türkei

Italien ☐

anderes Land ☐

anderes Land

A2. Wie alt bist du?

unter 20 ☐

20 bis 23 ☐

24 bis 26 ☐

27 bis 30 ☐

über 30 ☐

A3. Geschlecht

weiblich ☐

männlich ☐

A4. Was ist deine höchste abgeschlossene Ausbildung?

Matura ☐

Bachelor ☐

Master ☐

PhD ☐

andere Ausbildung ☐

andere Ausbildung

A5. An welcher Hochschule bist du eingeschrieben?

Technische Universität Graz ☐

FH Joanneum ☐

Karl-Franzens-Universität ☐

andere Hochschule ☐

andere Hochschule

A6. Was ist deine Studienrichtung?

Informatik ☐

Softwareentwicklung-Wirtschaft ☐

Telematik (Information and Computer Engineering) ☐

Informatik Lehramt ☐

andere Studienrichtung ☐

andere Studienrichtung

A7. Hast du bereits berufliche Erfahrung im IT Umfeld gesammelt?

keine berufliche Erfahrung ☐

unter 1 Jahr ☐

1 bis 3 Jahre ☐

über 3 Jahre ☐

A8. Wie gut kennst du diese agilen Softwareentwicklungsprozesse und die dazugehörigen Methoden?

z.B. Wenn Scrum der Prozess ist, ist das Daily Scrum bzw. Daily Stand Up Meeting eine dazugehörige Methode.

Geringe Kenntnisse = kenne die meisten der dazugehörigen Methoden, nicht/kaum angewendet

Durchschnittliche Kenntnisse = kenne die meisten der dazugehörigen Methoden und habe ein paar angewendet

Fortgeschrittene Kenntnisse = kenne (fast) alle der dazugehörigen Methoden und habe einige angewendet

Experte = kenne (fast) alle der dazugehörigen Methoden und habe (fast) alle schon verwendet

	Noch nie gehört	Nur davon gehört	Geringe Kenntnisse	Durchschnittliche Kenntnisse	Fortgeschrittene Kenntnisse	Experte
Agiles Testen	☐	☐	☐	☐	☐	☐
Behavior Driven Development (BDD)	☐	☐	☐	☐	☐	☐
Crystal	☐	☐	☐	☐	☐	☐
Extreme Programming (XP)	☐	☐	☐	☐	☐	☐
Feature Driven Development (FDD)	☐	☐	☐	☐	☐	☐
Kanban	☐	☐	☐	☐	☐	☐

	Noch nie gehört	Nur davon gehört	Geringe Kenntnisse	Durchschnittliche Kenntnisse	Fortgeschrittene Kenntnisse	Experte
Lean Software Development	☐	☐	☐	☐	☐	☐
Scrum	☐	☐	☐	☐	☐	☐
Scrumban	☐	☐	☐	☐	☐	☐

A9. In welchem Ausmaß hast du diese agilen Softwareentwicklungsprozesse bereits verwendet?

z.B. Wenn Scrum der Prozess ist, ist das Daily Scrum bzw. Daily Stand Up Meeting eine dazugehörige Methode.

	Noch nie	Nur kurz im Rahmen einer Lehrveranstaltung oder eines Praktikums ausprobiert	geringe Verwendung von nur wenigen Methoden	durchschnittliche Verwendung der meisten Methoden	intensive Verwendung von (fast) allen Methoden
Agiles Testen	☐	☐	☐	☐	☐
Behavior Driven Development (BDD)	☐	☐	☐	☐	☐
Crystal	☐	☐	☐	☐	☐
Extreme Programming (XP)	☐	☐	☐	☐	☐
Feature Driven Development (FDD)	☐	☐	☐	☐	☐
Kanban	☐	☐	☐	☐	☐
Lean Software Development	☐	☐	☐	☐	☐
Scrum	☐	☐	☐	☐	☐
Scrumban	☐	☐	☐	☐	☐

Teil B: TIPI

B1. Im Folgenden findest du eine Reihe von Persönlichkeitseigenschaften, die mehr oder weniger stark auf dich zutreffen. Bitte markiere für jede Aussage, inwieweit sie auf dich zutrifft oder nicht. Du sollst diese Einstufung jeweils für Paare von Eigenschaften vornehmen, auch wenn möglicherweise die eine Eigenschaft stärker zutrifft als die andere.

	Trifft voll und ganz zu	Trifft eher zu	Weder zutreffend noch unzutreffend	Trifft eher nicht zu	Trifft überhaupt nicht zu
Extrovertiert, begeistert	☐	☐	☐	☐	☐
Kritisch, streitsüchtig	☐	☐	☐	☐	☐

	Trifft voll und ganz zu	Trifft eher zu	Weder zutreffend noch unzutreffend	Trifft eher nicht zu	Trifft überhaupt nicht zu
Zuverlässig, selbstdiszipliniert	☐	☐	☐	☐	☐
Ängstlich, leicht aus der Fassung zu bringen	☐	☐	☐	☐	☐
Offen für neue Erfahrungen, vielschichtig	☐	☐	☐	☐	☐
Zurückhaltend, still	☐	☐	☐	☐	☐
Verständnisvoll, warmherzig	☐	☐	☐	☐	☐
Unorganisiert, achtlos	☐	☐	☐	☐	☐
Gelassen, emotional stabil	☐	☐	☐	☐	☐
Konventionell, unkreativ	☐	☐	☐	☐	☐

Teil C: Motivation

C1. ***Bewerte die unten angeführten Motivatoren, wie sehr sie dich allgemein motivieren, also unabhängig vom Catrobat Projekt. Es gibt drei grobe Gruppierungen für die Motivatoren (Projekt- bzw. Arbeits-Ebene, Gruppen-Ebene, Persönliche Ebene).***

Projekt- bzw. Arbeits-Ebene alllgemein (unabhängig vom Catrobat Projekt)

	motiviert mich sehr	motiviert mich	motiviert mich etwas	kein Einfluss auf Motivation
In einer erfolgreichen Organisation zu arbeiten	☐	☐	☐	☐
Angemessene Arbeitsbedingungen	☐	☐	☐	☐
Vielfältigkeit der Arbeit	☐	☐	☐	☐
Technisch herausfordernde Arbeit	☐	☐	☐	☐
Finanzielle Belohnungen und Anreize	☐	☐	☐	☐
Nicht finanzielle Belohnungen und Anreize	☐	☐	☐	☐
Ausreichende Ressourcen (Hardware, Manpower)	☐	☐	☐	☐
Professionalismus	☐	☐	☐	☐
Beziehung zu Kunden/Benutzern	☐	☐	☐	☐

	motiviert mich sehr	motiviert mich	motiviert mich etwas	kein Einfluss auf Motivation
Projekterfolg	☐	☐	☐	☐

C2. Gruppen-Ebene alllgemein (unabhängig vom Catrobat Projekt)

	motiviert mich sehr	motiviert mich	motiviert mich etwas	kein Einfluss auf Motivation
Unterstützendes Management, unterstützende Koordinatoren	☐	☐	☐	☐
Möglichkeit aufzusteigen (Teamleiter, Management)	☐	☐	☐	☐
Zusammenarbeit mit anderen, Teamarbeit	☐	☐	☐	☐
Team Qualität	☐	☐	☐	☐
Gefühl der Zugehörigkeit	☐	☐	☐	☐
Neue Leute kennenlernen	☐	☐	☐	☐
Einen Beitrag leisten	☐	☐	☐	☐
Gleichbehandlung, Gerechtigkeit	☐	☐	☐	☐

C3. Persönliche Ebene alllgemein (unabhängig vom Catrobat Projekt)

	motiviert mich sehr	motiviert mich	motiviert mich etwas	kein Einfluss auf Motivation
Vertrauen und Respekt	☐	☐	☐	☐
Feedback erhalten	☐	☐	☐	☐
Selbstständigkeit	☐	☐	☐	☐
Anerkennung	☐	☐	☐	☐
Verantwortung, Zuständigkeit	☐	☐	☐	☐
Status, Reputation	☐	☐	☐	☐
Entwicklung für sinnvolles Ziel	☐	☐	☐	☐
Identifizierung mit der Aufgabe	☐	☐	☐	☐
Work Life Balance (Arbeitszeiten, Überstunden...)	☐	☐	☐	☐
Gute Benotung erhalten	☐	☐	☐	☐
Lernen, sich weiterbilden	☐	☐	☐	☐
Kreativität und Innovation (z.B. eigene Ideen umsetzen können)	☐	☐	☐	☐

Teil D: Demotivation

D1. Demotivatoren

Bewerte die unten angeführten Demotivatoren, wie sehr sie dich allgemein, also unabhängig vom Catrobat Projekt demotivieren.

	demotiviert mich sehr	demotiviert mich	demotiviert mich etwas	kein Einfluss auf Motivation
Ungeeignete Arbeitsumgebung	☐	☐	☐	☐
Mangelhafte Kommunikation	☐	☐	☐	☐
Fehlende Möglichkeit um Beziehungen aufzubauen	☐	☐	☐	☐
Unrealistische Ziele	☐	☐	☐	☐
Fehlende Möglichkeiten um sich weiterzuentwickeln	☐	☐	☐	☐
Schlechte Qualität der Software	☐	☐	☐	☐
Fehlender Einfluss, Möglichkeit mitzuwirken	☐	☐	☐	☐
Unfaires Belohnungssystem	☐	☐	☐	☐
Uninteressante Arbeit	☐	☐	☐	☐
Ungerechtigkeit	☐	☐	☐	☐
Stress, Druck	☐	☐	☐	☐
Task Komplexität ist zu hoch	☐	☐	☐	☐
Task Komplexität ist zu nieder	☐	☐	☐	☐
Organisatorischer Overhead	☐	☐	☐	☐

Teil E: Catrobat Projekt Allgemein

E1. Was ist deine aktuelle Rolle im Catrobat Projekt?

Member ☐

Senior Member ☐

Koordinator ☐

Ehemaliger Member ☐

andere Rolle ☐

andere Rolle

E2. Wieviele Personen arbeiten in deinem Team im Catrobat Projekt?

1 bis 3 ☐

4 bis 6 ☐

7-10 ☐

über 10 ☐

E3. Wann hast du angefangen im Catrobat Projekt aktiv mitzuwirken?

E4. Wieviele Stunden hast du bisher am Catrobat Projekt gearbeitet?

E5. Du kannst deine Wahl im Kommentar noch gerne erläutern.

Wie geht es dir allgemein im Catrobat Projekt?

sehr gut ☐

gut ☐

mittel ☐

eher schlecht ☐

schlecht ☐

E6. Wie kommt dir die Stimmung im gesamten Catrobat Projekt allgemein vor?

- sehr gut ☐
- gut ☐
- mittel ☐
- eher schlecht ☐
- schlecht ☐

E7. Welches Image hat das Catrobat Projekt bei deinen Bekannten?

- sehr gut ☐
- gut ☐
- mittel ☐
- eher schlecht ☐
- schlecht ☐

E8. Was gefällt dir am Catrobat Projekt am besten bzw. was wird besonders gut gemacht?

E9. Was gefällt dir am Catrobat Projekt nicht so gut?

E10. Was kann deiner Meinung nach getan werden, um das Catrobat Projekt für andere Studierende noch interessanter zu machen?

E11. Bewerte die unten angeführten Statements.

Wie gut wird dieses Statement aktuell im Catrobat Projekt umgesetzt bzw. wie gut ist es vorhanden?

	sehr gut	gut	mittel	eher schlecht	schlecht
Faire Richtlinien und Rahmenbedingungen im Projekt	☐	☐	☐	☐	☐
Fähige und angmessene Administration im Projekt	☐	☐	☐	☐	☐
Ich kenne die Ziele, die im Projekt verfolgt werden	☐	☐	☐	☐	☐
Ich lerne interessante und agile Prozesse in der Softwareentwicklung kennen	☐	☐	☐	☐	☐
Ich kann meine Programmierkenntnisse verbessern und vertiefen	☐	☐	☐	☐	☐

Teil F: Catrobat Projekt Kommunikation

F1. **Du kannst bei allen Fragen deine Antwort im Kommentarfeld noch gerne erläutern (Gedanken, Probleme, Verbesserungsmöglichkeiten...).**

Wie würdest du die Kommunikation bzw. den Informationsaustausch allgemein im Catrobat Projekt beschreiben?

- sehr gut ☐
- gut ☐
- mittel ☐
- eher schlecht ☐
- schlecht ☐

F2. **Wie gut findest du IRC zur Kommunikation im Catrobat Projekt?**

- sehr gut ☐
- gut ☐
- mittel ☐
- eher schlecht ☐
- schlecht ☐

F3. Wie würdest du die Kommunikation in deinem Team beschreiben?

- [ ] sehr gut
- [ ] gut
- [ ] mittel
- [ ] eher schlecht
- [ ] schlecht

F4. Wie würdest du das regelmäßige Team-Meeting in deinem Team beschreiben?

- [ ] sehr gut
- [ ] gut
- [ ] mittel
- [ ] eher schlecht
- [ ] schlecht

F5. Was sollte an den regelmäßigen Team-Meetings verbessert werden?

Bitte beschreibe im Kommentarfeld detaillierter was wie verbessert werden sollte.

- [ ] Sehe keinen Verbesserungsbedarf
- [ ] Pünktlichkeit
- [ ] Protokollierung
- [ ] Regelmäßigkeit
- [ ] Anwesenheit der Teammitglieder
- [ ] Förderung des Zusammengehörigkeitsgefühles

- [ ] Moderation des Meetings
- [ ] Inhalt, Ziele, Sinn des Meetings
- [ ] Möglichkeit sich einzubringen
- [ ] Behandlung von Problemen
- [ ] Klare Verteilung der Aufgaben
- [ ] Sonstiges

Sonstiges

F6. Hast du schon einmal am BiWeCo-Meeting teilgenommen?

- [ ] Ja
- [ ] Nein

F7. Wie würdest du das BiWeCo-Meeting im Catrobat Projekt bewerten?

- [ ] sehr gut
- [ ] gut
- [ ] mittel
- [ ] eher schlecht
- [ ] schlecht

F8. Was sollte an den BiWeCo Meetings verbessert werden?

Bitte beschreibe im Kommentarfeld detaillierter was wie verbessert werden sollte.

- [ ] Sehe keinen Verbesserungsbedarf
- [ ] Pünktlichkeit
- [ ] Protokollierung
- [ ] Regelmäßigkeit (14 Tage)
- [ ] Anwesenheit der Teams
- [ ] Austausch mit anderen Teams

Förderung des Zusammengehörigkeitsgefühles mit allen Teams ☐

Moderation des Meetings ☐

Inhalt, Ziele, Sinn des Meetings ☐

Möglichkeit sich einzubringen ☐

Behandlung von Problemen ☐

Austausch mit Wolfgang ☐

Sonstiges ☐

Sonstiges

F9. Wie würdest du die Durchführung des Planning Games in deinem Team bewerten?

sehr gut ☐

gut ☐

mittel ☐

eher schlecht ☐

schlecht ☐

F10. Bewerte die unten angeführten Statements.

Wie gut wird dieses Statement aktuell im Catrobat Projekt umgesetzt bzw. wie gut ist es vorhanden?

	sehr gut	gut	mittel	eher schlecht	schlecht
Kommunikation mit anderen Teams	☐	☐	☐	☐	☐
Ich habe die Möglichkeit mich in wichtige Entscheidungen einzubringen	☐	☐	☐	☐	☐
Zur stetigen Weiterentwicklung und Verbesserung wird Feedback eingeholt	☐	☐	☐	☐	☐

Teil G: Catrobat Projekt Teamwork & Arbeit

G1. Wie würdest du das Teamwork in deinem Team beschreiben?

- ☐ sehr gut
- ☐ gut
- ☐ mittel
- ☐ eher schlecht
- ☐ schlecht

G2. Wie würdest du das Teamwork allgemein im gesamten Catrobat Projekt beschreiben?

- ☐ sehr gut
- ☐ gut
- ☐ mittel
- ☐ eher schlecht
- ☐ schlecht
- ☐ Kann ich nicht beurteilen

G3. Bewerte die unten angeführten Statements bezüglich Teamwork.

Wie gut wird dieses Statement aktuell im Catrobat Projekt umgesetzt bzw. wie gut ist es vorhanden?

	sehr gut	gut	mittel	eher schlecht	schlecht
Möglichkeit neue Freunde zu finden	☐	☐	☐	☐	☐
Teammitglieder in meinem Team kooperieren	☐	☐	☐	☐	☐

	sehr gut	gut	mittel	eher schlecht	schlecht
Meine Teamkollegen zeigen genug Engagement	☐	☐	☐	☐	☐
Teamerfolg wird als Teamleistung angesehen	☐	☐	☐	☐	☐
Ich werde in meinem Team wahrgenommen und respektiert	☐	☐	☐	☐	☐
Es macht Spaß in meinem Team zu arbeiten	☐	☐	☐	☐	☐

G4. Bewerte die unten angeführten Statements bezüglich Arbeit an sich.

Wie gut wird dieses Statement aktuell im Catrobat Projekt umgesetzt bzw. wie gut ist es vorhanden?

	sehr gut	gut	mittel	eher schlecht	schlecht
Ich habe klare Ziele und Aufgaben	☐	☐	☐	☐	☐
Ich bekomme die nötigen Informationen um meine Arbeit zu verrichten	☐	☐	☐	☐	☐
Ich habe die Freiheit Entscheidungen zu treffen, wie ich meine Arbeit verrichte	☐	☐	☐	☐	☐
Während der Arbeit schaue ich stets auf Qualität	☐	☐	☐	☐	☐
Der Task Komplexitätslevel ist angemessen	☐	☐	☐	☐	☐
Meine Arbeit im Projekt stresst mich nicht	☐	☐	☐	☐	☐

Teil H: Catrobat Projekt Koordination

H1. Wie würdest du die Koordination in deinem Team beschreiben?

Du kannst deine Antwort im Kommentar noch gerne erläutern (Gedanken, Probleme, Verbesserungsmöglichkeiten...).

- sehr gut ☐
- gut ☐
- mittel ☐
- eher schlecht ☐
- schlecht ☐

H2. Was sollte an der Koordination in deinem Team verbessert werden?

Bitte beschreibe im Kommentarfeld detaillierter was wie verbessert werden sollte.

- [ ] Sehe keinen Verbesserungsbedarf
- [ ] Erreichbarkeit
- [ ] Anwesenheit
- [ ] Verlässlichkeit
- [ ] Moderation der Meetings
- [ ] Kommunikation mit den Teammitgliedern
- [ ] Förderung des Zusammengehörigkeitsgefühles
- [ ] Behandlung von Problemen
- [ ] Mehr Feedback geben
- [ ] Informationen verteilen
- [ ] Sonstiges

Sonstiges

H3. Bewerte die unten angeführten Statements.

Wie gut wird dieses Statement aktuell im Catrobat Projekt umgesetzt bzw. wie gut ist es vorhanden?

	sehr gut	gut	mittel	eher schlecht	schlecht
Mein Koordinator macht einen guten und effizienten Job	☐	☐	☐	☐	☐
Ich bekomme regelmäßig Feedback über meine Leistung	☐	☐	☐	☐	☐
Mein Koordinator behandelt alle Teammitglieder gleich	☐	☐	☐	☐	☐
Ich kann meine Koordinator immer um Unterstützung bzw. Rat fragen	☐	☐	☐	☐	☐

Teil I: Catrobat Projekt Methoden

I1. Wie würdest du deinen Start ins Catrobat Projekt beschreiben (Arbeit, Team, Codebase, Tutorials...)?

Bitte beschreibe im Kommentarfeld detaillierter was wie verbessert werden sollte.

- sehr leicht und reibungslos ☐
- eher leicht und reibungslos ☐
- mittel ☐
- eher schwierig ☐
- sehr schwierig ☐

I2. Bewerte die unten angeführten Statements.

In welchem Ausmaß wird das Statement aktuell in deinem Team umgesetzt?

	viel zu viel	zu viel	passt	zu wenig	viel zu wenig
Unterstützung von Neulingen	☐	☐	☐	☐	☐
Dokumentation	☐	☐	☐	☐	☐
Pair Programming	☐	☐	☐	☐	☐
Refactoring	☐	☐	☐	☐	☐
Testing	☐	☐	☐	☐	☐

I3. Hier kannst du Probleme, Gedanken, Kommentare zu den oben genannten Statements einbringen.

I4. **Wusstest du dass das Catrobat Projekt ein Open Source Software Projekt ist?**

Ja ☐

Nein ☐

I5. **Hast du schon an einem anderen Open Source Software Projekt aktiv mitgearbeitet?**

Ja ☐

Nein ☐

I6. **An wievielen anderen Open Source Software Projekten hast du bereits aktiv mitgearbeitet?**

I7. **Was kannst du aus der Teilnahme am Catrobat Projekt mitnehmen und was hast du dazugelernt?**

Teil J: Catrobat Projekt Verbesserung

J1. **Was sollte deiner Meinung nach am Catrobat Projekt verbessert werden?**

J2. **Was sollte deiner Meinung nach am Handling mit neuen Mitgliedern verbessert werden?**

J3. Könntest du dir vorstellen, nach dem Abschluss deiner aktuellen Arbeit am Catrobat Projekt, noch weiter mitzuarbeiten?

Ja ☐

Nein ☐

J4. Was müsste verbessert werden damit du dich für "Ja" entscheidest?

J5. Bewerte die unten angeführten Ideen.

Wie gut würdest du diese Idee zur Verbesserung vom Catrobat Projekt finden?

	Super Idee	Gute Idee	Könnte helfen	Egal	Keine sinnvolle Idee
Scrum Master oder Coach zur Unterstützung bei Kommunikation, Meetings...	☐	☐	☐	☐	☐
Regelmäßige Meetings mit Wolfgang bzw. zuständiger Person für das Projekt	☐	☐	☐	☐	☐
Einstiegstest für das Catrobat Projekt (Clean Code...)	☐	☐	☐	☐	☐
Timesheet Meilensteine mit individuellem Feedback-Gespräch mit Koordinator	☐	☐	☐	☐	☐
Retrospektiven machen: was gut gemacht wird, was verbessert werden soll...	☐	☐	☐	☐	☐

J6. Fällt dir eine weitere Idee bzw. Maßnahme ein, die zur Verbesserung vom Catrobat Projekt beitragen könnte?

Teil K: Catrobat Projekt Motivation

K1. ***Bewerte die unten angeführten Motivatoren, wie zutreffend sie aktuell im Catrobat Projekt vorhanden sind.***

Projekt- bzw. Arbeits-Ebene im Catrobat Projekt

	voll und ganz zutreffend/vorhanden im Catrobat Projekt	eher zutreffend/vorhanden im Catrobat Projekt	eher nicht zutreffend/vorhanden im Catrobat Projekt	überhaupt nicht zutreffend/vorhanden im Catrobat Projekt
In einem erfolgreichen Projekt zu arbeiten	☐	☐	☐	☐
Angemessene Arbeitsbedingungen	☐	☐	☐	☐
Vielfältigkeit der Arbeit	☐	☐	☐	☐
Technisch herausfordernde Arbeit	☐	☐	☐	☐
Finanzielle Belohnungen und Anreize	☐	☐	☐	☐
Nicht finanzielle Belohnungen und Anreize	☐	☐	☐	☐
Ausreichende Ressourcen (Hardware, Manpower)	☐	☐	☐	☐
Professionalismus	☐	☐	☐	☐
Beziehung zu Kunden/Benutzern	☐	☐	☐	☐
Projekterfolg	☐	☐	☐	☐

K2. **Gruppen-Ebene im Catrobat Projekt**

	voll und ganz zutreffend/vorhanden im Projekt Catrobat	eher zutreffend/vorhanden im Projekt Catrobat	eher nicht zutreffend/vorhanden im Projekt Catrobat	überhaupt nicht zutreffend/vorhanden im Projekt Catrobat
Unterstützendes Management, unterstützende Koordinatoren	☐	☐	☐	☐
Möglichkeit Koordinator, Senior Member zu werden	☐	☐	☐	☐
Zusammenarbeit mit anderen, Teamarbeit	☐	☐	☐	☐
Team Qualität	☐	☐	☐	☐
Gefühl der Zugehörigkeit	☐	☐	☐	☐
Neue Leute kennenlernen	☐	☐	☐	☐
Einen Beitrag leisten	☐	☐	☐	☐
Gleichbehandlung, Gerechtigkeit	☐	☐	☐	☐

K3. Persönliche Ebene im Catrobat Projekt

	voll und ganz zu treffend/vorhanden im Catrobat Projekt	eher zutreffend/ vorhanden im Catrobat Projekt	eher nicht zutreffend/vorhanden im Catrobat Projekt	überhaupt nicht zutreffend/vorhanden im Catrobat Projekt
Vertrauen und Respekt	☐	☐	☐	☐
Feedback erhalten	☐	☐	☐	☐
Selbstständigkeit	☐	☐	☐	☐
Anerkennung	☐	☐	☐	☐
Verantwortung, Zuständigkeit	☐	☐	☐	☐
Status, Reputation	☐	☐	☐	☐
Entwicklung für sinnvolles Ziel	☐	☐	☐	☐
Identifizierung mit der Aufgabe	☐	☐	☐	☐
Work Life Balance	☐	☐	☐	☐
Gute Benotung erhalten	☐	☐	☐	☐
Lernen, sich weiterbilden	☐	☐	☐	☐
Kreativität und Innovation (z.B. eigene Ideen umsetzen können)	☐	☐	☐	☐

K4. Was motiviert dich, damit du zusätzliche "Extra" Energie für das Catrobat Projekt aufwendest?

Teil L: Catrobat Projekt Demotivation

L1. ***Bewerte die unten angeführten Demotivatoren, wie sehr sie dich im Catrobat Projekt demotivieren.***

	voll und ganz zu treffend/vorhanden im Catrobat Projekt	eher zutreffend/ vorhanden im Catrobat Projekt	eher nicht zutreffend/vorhanden im Catrobat Projekt	überhaupt nicht zutreffend/vorhanden im Catrobat Projekt
Ungeeignete Arbeitsumgebung	☐	☐	☐	☐
Mangelhafte Kommunikation	☐	☐	☐	☐
Fehlende Möglichkeit um Beziehungen aufzubauen	☐	☐	☐	☐

	voll und ganz zu treffend/vorhanden im Catrobat Projekt	eher zutreffend/ vorhanden im Catrobat Projekt	eher nicht zutreffend/vorhanden im Catrobat Projekt	überhaupt nicht zutreffend/vorhanden im Catrobat Projekt
Unrealistische Ziele	☐	☐	☐	☐
Fehlende Möglichkeiten um sich weiterzuentwickeln	☐	☐	☐	☐
Schlechte Qualität der Software	☐	☐	☐	☐
Fehlender Einfluss, Möglichkeit mitzuwirken	☐	☐	☐	☐
Unfaires Belohnungssystem	☐	☐	☐	☐
Uninteressante Arbeit	☐	☐	☐	☐
Ungerechtigkeit	☐	☐	☐	☐
Stress, Druck	☐	☐	☐	☐
Task Komplexität ist zu hoch	☐	☐	☐	☐
Task Komplexität ist zu nieder	☐	☐	☐	☐
Organisatorischer Overhead	☐	☐	☐	☐

L2. Was nimmt dir viel Energie oder zieht dich runter, bei deiner Mitarbeit am Catrobat Projekt?

Teil M: Feeback

M1. Willst du noch irgendwas zum Catrobat Projekt sagen? Fällt dir noch irgendwas ein?

M2. **Was fehlt in der Umfrage bzw. was kann verbessert werden?**

M3. **Glaubst du dass diese Umfrage interessante Erkenntnisse und Potentiale zur Verbesserung vom Catrobat Projekt identifizieren kann?**

Ja ☐

Nein ☐

Vielen Dank, dass du dir Zeit genommen hast!

Du wirst über die Ergebnisse dieser Umfrage informiert.

Falls du noch weiteres Interesse hast, kannst du mich gerne kontaktieren!