



Jautz Julian, Bsc

# **Introducing Key Performance Indicators into a Free and Open-Source Project to evaluate Human Resource Management Practices and Project Performance**

## **Master's Thesis**

to achieve the university degree of  
Master of Science

Master's degree programme: Software Engineering and Management

submitted to

**Graz University of Technology**

Supervisor

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

Institute of Software Technology

Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Franz Wotawa

Graz, October 2024





Jautz Julian, Bsc

# **Integration von Leistungskennzahlen in ein Freies- und Open-Source-Software- Projekt zur Evaluierung von Personal- management und Projektleistung**

## **Masterarbeit**

zur Erlangung des akademischen Grades eines  
Diplom-Ingenieur  
Masterstudium: Software Engineering and Management

eingereicht an der

**Technische Universität Graz**

Betreuer

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

Institut für Softwaretechnologie

Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Franz Wotawa

Graz, Oktober 2024



# 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 indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

---

Date

---

Signature



# Eidesstattliche Erklärung

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

---

Datum

---

Unterschrift





---

## Acknowledgment

I would like to sincerely thank my parents for their consistent support throughout my education, making it possible for me to pursue and complete my studies. Their belief in me has been invaluable.

A special thank you to my partner, Caroline, who constantly motivated me when I struggled with writing and provided essential advice during the process. I am grateful to be able to learn so much from her.

I also want to thank my brother and sister for their continuous support since my school years, especially my brother, who was a role model and inspired my choice of study.

I am much obliged to the Catrobat coordinators who kindly agreed to be interviewed for this thesis. Their insights and perspectives were essential for my research, and I appreciate their time and cooperation.

Lastly, I am grateful to my supervisor, Prof. Wolfgang Slany, for enabling me to write this thesis, to my main advisor, Stefan Kutschera, for his guidance and recommendations, and to Patrick Ratschiller for his helpful advice.



# Abstract

Every day, people worldwide rely on the functionality and reliability of computer applications. Many are developed as Free Open-Source Software (FOSS) or Open-Source Software (OSS). In very simple terms, this means that profit realisation is often a minor concern, and the source code is publicly accessible, contributed by a large number of volunteers. Therefore, human resource management (HRM) is crucial for creating a suitable collaboration framework. This type of development therefore differs greatly from its proprietary counterparts. This raises the question of how project progress and success can be measured when the financial aspect only plays a minor role or is even insignificant. The non-profit FOSS project Catrobat was used as an example to explain which key performance indicators (KPIs) can be used to illustrate the success of such a project. Guided expert interviews with the development team coordinators provided insights into the project organisation. Based on this qualitative analysis, critical success factors (CSFs) tailored to Catrobat, and subsequently KPIs, could be defined. Furthermore, deficits in the current management processes were identified. By combining interview results with quantitative KPI measurements, recommendations for HRM practices were developed to likely enhance project success. This success can be tracked through continuous data collection and regular updates to the presented KPIs. This work illustrates that even non-profit FOSS projects can benefit from the introduction of KPIs and the visualisation of project performance they enable.

**Keywords:** Free Open-Source Software; Open-Source Software; FOSS; OSS; Project Management; Human Resource Management; Key Performance Indicators; KPI; Critical Success Factors; CSF; Dashboard; Agile Software Development



# Kurzfassung

Tagtäglich verlassen sich Menschen weltweit auf die Funktionalität und Zuverlässigkeit von Computeranwendungen. Darunter befinden sich viele, die als Free Open-Source Software (FOSS) oder Open-Source Software (OSS) entwickelt werden. Stark vereinfacht bedeutet das, dass der Quellcode öffentlich zugänglich ist, die Gewinnerzielung oft nicht im Vordergrund steht und die Entwicklung von einer Vielzahl von freiwilligen Mitwirkenden übernommen wird. Daher ist besonders das Personalmanagement gefragt, um einen geeigneten Rahmen für die Zusammenarbeit zu schaffen. Somit unterscheidet sich diese Art der Entwicklung stark von ihren proprietären Gegenstücken. Dabei stellt sich die Frage, auf welche Art und Weise Projektfortschritt und -erfolg gemessen werden können, wenn der finanzielle Aspekt nur eine Nebenrolle spielt oder gar unwesentlich ist. Am Beispiel des non-profit FOSS Projekts Catrobat wurde erläutert, welche Leistungskennzahlen (KPIs) eingesetzt werden können, um den Projekterfolg eines solchen Projekts darzustellen. Ein Einblick in die Projektorganisation wurde durch leitfadengestützte Experteninterviews mit den Koordinatoren der Entwicklerteams ermöglicht. Aufgrund der Analyse dieser qualitativen Forschungsmethode, konnten auf Catrobat zugeschnittene kritische Erfolgsfaktoren (CSFs) und im Anschluss KPIs definiert werden. Weiters konnten Defizite in den aktuellen Managementprozessen aufgezeigt werden. Mittels einer Kombination aus den Interviewergebnissen und den quantitativen Messungen der KPIs wurden Empfehlungen für Personalmanagement-Praktiken ausgesprochen, die unter aller Voraussicht den Projekterfolg steigern können. Dieser ist wiederum durch eine kontinuierliche Datenbereitstellung und Aktualisierung der präsentierten KPIs messbar. Diese Arbeit veranschaulicht, dass auch ein non-profit FOSS Projekt Nutzen aus der Einführung von KPIs und der damit ermöglichten Visualisierung der Projektleistung ziehen kann.

**Schlüsselwörter:** Free Open-Source Software; Open-Source Software; FOSS; OSS; Projektmanagement; Personalmanagement; Key Performance Indicators; KPI; Leistungskennzahlen; Critical Success Factors; CSF; Kritische Erfolgsfaktoren; Dashboard; Agile Softwareentwicklung



# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Problem Statement . . . . .	2
1.3. Methodology . . . . .	2
1.4. Research Questions . . . . .	3
1.5. Hypothesis . . . . .	3
<b>2. Background and Related Work</b>	<b>5</b>
2.1. Project Management in Software Development Projects . . . . .	5
2.1.1. Agile Development . . . . .	6
2.2. Human Resource Management . . . . .	7
2.2.1. Recruitment, Selection, Motivation, and Retention . . . . .	7
2.2.2. Performance Evaluation and Compensation . . . . .	8
2.2.3. Orientation, Training and Development . . . . .	9
2.2.4. HR Effectiveness and HR Measurement . . . . .	10
2.2.5. The Status Change of HRM . . . . .	10
2.3. Metrics . . . . .	11
2.4. Performance Measures . . . . .	12
2.4.1. The Benefits and Pitfalls of Measuring Performance . . . . .	12
2.4.2. Critical Success Factors . . . . .	13
2.4.3. The Balanced Scorecard Approach . . . . .	14
2.4.4. (Key) Result Indicators . . . . .	15
2.4.5. (Key) Performance Indicators . . . . .	15
2.4.6. Selecting and Defining Suitable Performance Measures . . . . .	16
2.5. Free and Open-Source Software . . . . .	16
2.5.1. Definition . . . . .	17
2.5.2. Management in FOSS Projects . . . . .	18
2.5.3. KPIs in FOSS Projects . . . . .	19
2.6. The Catrobat Project . . . . .	19
2.6.1. Organisational Structure of Catrobat . . . . .	20
2.6.2. Communication and Productivity Tools . . . . .	23
2.6.3. Motivation to Participate . . . . .	24
2.7. Interview Evaluation . . . . .	24
<b>3. Implementation</b>	<b>27</b>
3.1. Methodology . . . . .	27

3.2.	Limitations of the Implementation Area . . . . .	28
3.3.	Interviews . . . . .	28
3.3.1.	Interview Method . . . . .	28
3.3.2.	Expert Selection . . . . .	29
3.3.3.	Interview Guideline Structure . . . . .	30
3.3.4.	Interview Scope . . . . .	30
3.3.5.	Adaption of the Guideline After the First Interview . . . . .	32
3.3.6.	Recording and Transcription . . . . .	33
3.3.7.	Evaluation . . . . .	33
3.4.	KPI Implementation . . . . .	34
3.4.1.	The Mission and Vision of Catrobat . . . . .	35
3.4.2.	Noteworthy Specifics of the Catrobat Project . . . . .	36
3.4.3.	Definition of Critical Success Factors . . . . .	37
3.4.4.	The Balanced Score Card Perspectives . . . . .	43
3.4.5.	Developing the KPIs . . . . .	47
3.5.	Summary . . . . .	54
<b>4.</b>	<b>Results and Evaluation</b>	<b>55</b>
4.1.	Interview Results . . . . .	55
4.1.1.	Field of Activity . . . . .	57
4.1.2.	Team Organisation . . . . .	59
4.1.3.	Development Process . . . . .	62
4.1.4.	Recruitment, On- and Offboarding . . . . .	64
4.1.5.	Performance Measures . . . . .	66
4.1.6.	Team Member Motivation . . . . .	70
4.1.7.	Knowledge Management . . . . .	72
4.1.8.	Conclusion - Deficiencies in the Current Processes . . . . .	73
4.2.	Measurement and Visualisation of the KPIs . . . . .	75
4.2.1.	Tools for HRM and KPIs . . . . .	75
4.2.2.	Processing the Data . . . . .	76
4.2.3.	Visualising the Data . . . . .	78
4.2.4.	Dashboard Representation . . . . .	80
4.2.5.	Analysis of the KPI Values and Benefits of the KPIs . . . . .	88
4.3.	Recommendations for Management Practices . . . . .	90
4.3.1.	The HRM perspective . . . . .	91
4.3.2.	The Technical Perspective . . . . .	94
4.4.	Findings . . . . .	94
4.5.	Limitations . . . . .	95
<b>5.</b>	<b>Conclusion and Future Work</b>	<b>97</b>
5.1.	Conclusion . . . . .	97
5.2.	Future Work . . . . .	99
	<b>Bibliography</b>	<b>101</b>



<b>A. Interview Field Manual</b>	<b>111</b>
<b>B. Interview Transcript</b>	<b>115</b>
B.1. Interview with <i>Paintroid</i> Coordinator . . . . .	115
B.2. Interview with <i>IDE</i> Coordinator . . . . .	125
B.3. Interview with <i>Stage</i> Coordinator . . . . .	137
B.4. Interview with <i>Catty</i> Coordinator . . . . .	151
<b>C. Interview Consent Agreement</b>	<b>165</b>



# List of Figures

2.1.	Changing Roles of HR Management . . . . .	11
2.2.	Project's Strategy . . . . .	13
2.3.	Six-Perspective BSC . . . . .	14
2.4.	Types of Performance Measures . . . . .	15
2.5.	Pocket Code Android . . . . .	20
2.6.	Catrobat Team Structure . . . . .	22
2.7.	Qualitative Content Analysis . . . . .	25
3.1.	CSFs and KPIs Dependencies . . . . .	35
4.1.	Interview Code Tree . . . . .	56
4.2.	KPI - WMA . . . . .	81
4.3.	KPI - SPI . . . . .	82
4.4.	KPI - AHRT . . . . .	83
4.5.	KPI - RSR - Pocket Code . . . . .	84
4.6.	KPI - RSR - Pocket Paint . . . . .	85
4.7.	KPI - TTIP . . . . .	86
4.8.	KPI Dashboard . . . . .	87



# List of Tables

3.1. CSFs in relation to the BSC . . . . .	44
3.2. Overview of Catrobat's KPIs . . . . .	48



# 1. Introduction

## 1.1. Motivation

Since their first appearance in 1984, Free and Open-Source Software (FOSS) projects have developed into essential building blocks of the software landscape. Many doubted the success of the projects at the beginning, and many software companies turned down free software, concluding that there was no money to be made with it. With the spin-off of the open-source movement, a new era began and the advantages of open-source became known to a wider community. While the differences and further development will be discussed in more detail in Chapter 2.5, it should only be mentioned here that it was thus possible to use open-source code commercially, even if this was not the main driver of this movement (DiBona, 1999, pp. 2-4).

Linux, Firefox (Mozilla), LibreOffice and Apache HTTP Server are all based on open-source code, just to name a few FOSS projects that were essential for the further development of the digital world. At the beginning, it seemed unlikely that such large projects, which relied largely on voluntary labour, could produce a usable end product. Not only were the results usable, but the quality often surpassed that of closed source software (CSS) (Raymond, 2001, p. 21-22). Empirical tests could even prove, that the quality of open-source is in no way inferior and might even surpass CSS (Kuan, 2002, pp. 25-26).

Although the quality and the product are comparable to closed source or proprietary software, the organisation behind it differs significantly. On the one hand, there is a small, permanent core team that works together regularly, while on the other hand there are a large number of voluntary contributors. These contributors can be spread across the globe and most only contribute to the project a few times. Usually there is also a group of so-called core developers in a FOSS project who contribute the most to the source code, but this is still not equivalent to a development team in the CSS environment (Apache et al., 2002, pp. 316-317).

Regarding the mentioned differences, the question arises how project management, and especially human resource management (HRM) is applied at FOSS projects. The changed team structure and a high contributor turnover pose new challenges for the management. Another interesting question is how project success can be measured in FOSS projects. Since financial success plays a very subordinate role in many, especially free software projects, the success criteria of proprietary software cannot be adopted without any changes. Nevertheless,

it is still possible to implement key performance indicators (KPIs), that measure the project progress and success. However, the selection of appropriate ones poses a challenge.

To gain an insight into these issues, the International Catrobat Association was used as an example of a FOSS project. Because this non-profit organisation is based at TU Graz, it was the perfect representative for this research. This thesis therefore deals with the introduction of KPIs into the Catrobat project in order to make project performance visible and subsequently also with considerations as to how this success can be further increased.

## 1.2. Problem Statement

The implementation of a free and open-source software (FOSS) project significantly differs from the conventional, closed source software project. As a result, also the management of a FOSS project faces other requirements and has to be handled in a different way. Nevertheless, HRM is a crucial part of every successful project. In fact, due the volatile nature of FOSS projects and their high fluctuation in team members, HRM might be even more important than in a closed source project (Ruhe, 2014, p. 321).

However, Aberdour (2007, p. 59) argues that FOSS projects often lack in organisational structure. This includes a lack of definition of the development method, no structured software tests, hardly any documentation and only a few measurable project goals. Using Catrobat as an example, the current organisational status will be evaluated with a particular focus on HRM in order to identify any existing shortcomings. It can be assumed that a (software) project without a well-structured HRM has significant trouble producing high quality results in predictable quantities, as a lack of management on the project level makes it almost impossible to coordinate the project teams.

Additionally, without objective performance measures, it is more difficult to gain an objective overview on performance and to counteract productivity loss. Due to the current management situation in the Catrobat project, performance estimates are based on subjective assessments of every team's coordinator. Furthermore, the Catrobat project does not have a definition of project success and performance, not to mention any metrics that could provide information about the current state of the project. Despite the fact, that Catrobat is a very successful FOSS project, without objective evaluation methods and a lack of management practices, there is no guarantee for sustainable success.

## 1.3. Methodology

This master thesis takes several steps to introduce KPIs into the Catrobat project. Chapter 2 begins with an explanation of the literary background to project



management and HRM in particular. It also takes a look at metrics, performance measures and FOSS projects using the example of Catrobat.

In order to gain an understanding of the current processes, particularly the HRM practices, in the Catrobat organisation, guided experts interviews were conducted as a qualitative research method. The coordinators of the four main development teams were consulted as interview partners. Explanations on the structure and procedure of the interviews can be found in chapter 3.3. Afterwards, the detailed analysis of the interview results is discussed in chapter 4.1.

After evaluating and analysing the current state, the process of implementing KPIs for the Catrobat organisation is explained. Chapter 3.4 covers the entire process, starting with the examination of Catrobat's mission and vision, through the creation of critical success factors (CSFs) with regard to the balanced scorecard (BSC) perspectives, to the actual definition of the KPIs.

These KPIs are capable of depicting the progress and success of the project. Before the KPIs are presented using sample data from a Catrobat team, the visualisation options and possible tools for the presentation are briefly discussed in chapter 4.2.

Finally in chapter 4.3, (human resource) management practices, based on the results of the interview analysis, that can increase performance and productivity in the organisation are discussed. This also illustrates the link to the KPIs that are used to indicate a change in performance.

## 1.4. Research Questions

In order to successfully implement KPIs and to improve project performance in the Catrobat FOSS project, this thesis answers the following questions:

- **Question 1:** Which KPIs (Key Performance Indicators) are suitable for measuring productivity and success in a FOSS project?
- **Question 2:** Which tools and processes are needed to track the KPIs continuously?
- **Question 3:** Which actions can be taken to improve the project performance according to the defined KPIs?

## 1.5. Hypothesis

It is hypothesised that not only profit-orientated companies, but also non-profit FOSS organisations can successfully introduce KPIs, if the fundamental mission and vision of the organisation are taken into account. These KPIs will differ from the typical ones found in economic guidebooks, as the focus and long-term goal of a FOSS organisation are also different. However, those KPIs

## 1. Introduction

---

will also be suitable for the continuous display of project performance and an increase in productivity will be recognisable if the daily work is aligned accordingly. Furthermore, it is assumed that a lack of potential in relation to (human resource) management practices is only insufficiently recognised if there are no objective methods for determining performance. Therefore, it will be possible to recommend HRM practices, which relate to the defined KPIs, in order to increase project success in the long term.

## **2. Background and Related Work**

The following chapter discusses the terms Project Management and Human Resource Management (HRM) in software projects. Furthermore, it takes a look at the importance and the role that human resource management plays in software projects according to literature research. Moreover, the chapter contains a conceptual delimitation regarding several terms, that concern the measurement of performance, and deals with findings of enterprises, which introduced performance measures into their software projects. Lastly, the term free and open-source (FOSS) project is defined and an overview about the Catrobat project, which is the target FOSS project for introducing HRM and KPIs, is given.

### **2.1. Project Management in Software Development Projects**

Successfully implementing a project without defined project management processes supporting the execution is hardly possible. A project is seen as an organized method to reach a specific objective, which has a targeted schedule and usually a defined budget (Lientz, 2001, xv). Each activity and task during the project consumes resources, which need to be provided in the right place at the right time. Such a complex endeavour requires a process that controls the achievements of the project objectives (AK Munns and BF Bjeirmi, 1996, p. 81).

This project management process combines several managerial disciplines, namely Scope, Time, Cost, Quality, Human Resources, Communications, Risk and Procurement Management. These segments are orchestrated by the Integration Management to form the superstructure of Project Management (Webster and Knutson, 2010, p. 9).

Despite the fact, that the project-oriented mode of operation is not exclusively used in the IT sector, in fact construction is innately a project-oriented industry and pharmaceuticals and aerospace operate in a project manner too, this chapter focuses on project management in software development (Webster and Knutson, 2010, p. 1).

### 2.1.1. Agile Development

When applying the traditional project management views to software development projects problems arise, because the requirements of software development cannot be met by these management approaches. The main reason is that the traditional method views development as a linear sequence of well-defined activities and the precondition for that is an almost perfect information about the project's goal and expected solution. However, software development projects are often prone to deviations in scope, schedule, or resources (Dybå et al., 2014, pp. 277-279).

Given this circumstance, a new way of handling projects was constructed. The principles of Agile Development are most typically defined via the *Manifest for Agile Software Development*<sup>1</sup>. According to the manifesto the foundation of the agile movement can be depicted by four phrases:

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

There are several Agile Methods like SCRUM and Extreme Programming, which follow those basic agile principles. However, these collaboration frameworks value some agile principles more than others and cannot be seen as a perfect implementation of the Agile Manifesto (Laanti et al., 2013, p. 247).

Despite these differences and the disunity on how exactly to define Agile, Agile can be seen as a loosely structured solution development paradigm, that builds upon iterative development and incremental delivery. Integrated in this paradigm are several main elements, which are crucial for agile development. The *adaptive* aspect considers the ongoing change of requirements and circumstances, that demand adaptive planning and effective feedback loops.

As already noted in the maxim of the Agile Manifesto the *value driven* approach focuses on the incremental delivery of working software and the frequent bilateral interchange with the stakeholders about the project status. Furthermore, *collaboration* is key and self-organized, multi-disciplinary teams are trusted with effective solution finding. *Empowerment* highlights the differences in the traditional management approach and the agile approach. Agile teams work in an environment of trust, respect and courage and there is no need for the servant-leadership relationship of traditional management (Moran, 2015, pp. 1-4).

---

<sup>1</sup><https://agilemanifesto.org/> visited on 14. May 2024

## 2.2. Human Resource Management

Usually, four types of assets that affect organisational performance can be differentiated in every organisation. These assets are *physical* (buildings, land, furniture, equipment, vehicles, computers, etc.), *financial* (cash, financial resources, stocks, financial securities, etc.), *intangible* (specialized research capabilities, patents, information systems, designs, operating processes, etc.) and *human* (individuals with talents, capabilities, experience, professional expertise, relationships, etc.). The human assets represent a special case, because this human capital entails much more than solely the people in organisations. Human resource (HR) is about the capabilities and skills each person provides to the organisation. In fact, these human resources are crucial to leverage all the other asset types in a successful manner (Mathis and Jackson, 2008, pp. 4-5). Ivancevich and Konopaske (2013, vi) agree that human resource management (HRM) differs distinctively from other management disciplines as its emphasis lies on people in work settings and their well-being and comfort in an organisation.

There is strong evidence that a thoroughly thought-out system of HRM practices has a significant impact on individual and organisational performance. What must be considered, however, is that many individual HRM practices are active at the same time and affect employees simultaneously. To avoid counteracting of methods, a dynamic bundle of HRM practices, that mutually reinforce each other, has to be designed and implemented according to the needs of the individual organisation (Kaifeng Jiang et al., 2012, p. 73). The essential goal of HRM is to make working people more productive and satisfied. Each of the following activities as described by Ivancevich and Konopaske (2013, pp. 1-5) contributes its significant part to the overall goal of HRM:

1. Equal employment opportunity (EEO) compliance
2. Job analysis
3. Human resource planning
4. Recruitment, selection, motivation, and retention
5. Performance evaluation and compensation
6. Orientation, training and development
7. Labor relations
8. Safety, health, and wellness

Although none of those activities can be neglected, the following explanations will focus on the most crucial elements regarding the HRM of the Catrobat FOSS project.

### 2.2.1. Recruitment, Selection, Motivation, and Retention

The recruitment process is started after the job analysis has been done and the required characteristics of the potential applicants are listed as the job

## 2. Background and Related Work

---

specifications (Ivancevich and Konopaske, 2013, p. 219). On the one hand, recruitment refers to the activities needed to generate a pool of qualified applicants and on the other hand, recruitment is also accountable for applicants accepting the offered job (Breaugh, 1993, p. 4; Mathis and Jackson, 2008, p. 193).

The whole process is affected by several external influences, which impede a successful staffing. organisations have to follow rules established by the government and restrictions set by labour unions. Apart from the legal requirements, the fluctuating labour market conditions have a tremendous impact on the effort needed. For example, a stagnating economy, where no new jobs are created, leads to a surplus of qualified workers and therefore to a high probability finding new employees. Vice versa, skilled labour shortage poses major challenges for companies (Ivancevich and Konopaske, 2013, pp. 190-191).

Following the recruitment phase, the selection process aims to choose the right employee for the vacant job position. Apart from the obvious fit of technical skills, the social abilities of the applicants must not be neglected. Especially in independent agile teams is the capacity for teamwork essential (Moran, 2015, p. 113). Furthermore, reasoned selection is the first step for keeping a high employee retention, as high turnover rates are linked to poor selection screening efforts in various companies. (Mathis and Jackson, 2008, p. 77).

More and more companies recognise the importance of human resources and implement people-first practices to increase to focus on employee motivation and retention (Ivancevich and Konopaske, 2013, p. 11). An example is the food-service company *Sysco*. Their workforce analysis could clearly link high employee satisfaction to higher revenues, lower costs, greater employee retention, and superior customer loyalty (Davenport et al., 2010, p. 4).

Retention strategies are manifold and must complement each other. High retention rates go along with agreeable working conditions like time and work flexibility, training and mentoring, career planning and an attractive rewards program (Mathis and Jackson, 2008, pp. 77-81).

### 2.2.2. Performance Evaluation and Compensation

Especially in times of a shortage of skilled labour, it is essential for companies to evaluate the performance of their employees in order to retain high-performing employees. In principle, it is not just a question of filling vacancies, but of filling key positions with qualified employees on a permanent basis (Cappelli, 2005, p. 12). Aguinis (2013, p. 2) describes performance management as *"a continuous process of identifying, measuring, and developing the performance of individuals and teams and aligning performance with the strategic goals of the organisation"*.

One instrument of employee management is leveraging a reward and penalize system. If goals are met, staff members are rewarded with positive incentives like pay rises or increased responsibilities. However, if the performance targets are not achieved, negative consequences for the employee follow. It is the

responsibility of the HRM to help the individual to overcome the shortfall in performance. Possible options are training, changing the team or even the job. If the required performance can still not be delivered, termination remains the last possible solution (Ellig, 2005, pp. 129-130).

This is precisely the difference between performance *management* and a performance *appraisal* system. Aguinis (2013) clearly emphasises in his definition, that performance management is a continuous process. The main task is not a one-off evaluation but responding to the evaluation results in order to improve performance in the long term. This iterative process is crucial for performance management and enables to align employee's tasks with the company's long-term objectives (Aguinis, 2013, pp. 2-3).

However, any reward and penalize system relies on a stable and fair performance evaluation system. If the employees feel unfair treated by the system, the incentive will trigger exactly the opposite and the employee performance will decrease (Alam and Kamal, 2024, p. 3). A refined and reliable performance management system contributes to the success of a business in many forms, Aguinis (2013, pp. 6-7) lists 16 advantages:

1. Motivation to perform is increased.
2. Self-esteem is increased.
3. Managers gain insight about subordinates.
4. The definitions of job and criteria are clarified.
5. Self-insight and development are enhanced.
6. Administrative actions are fairer and more appropriate.
7. Organisational goals are made clear.
8. Employees become more competent.
9. Employee misconduct is minimized.
10. There is better protection from lawsuits.
11. There is better and more timely differentiation between good and poor performers.
12. Supervisors' views of performance are communicated more clearly.
13. Organisational change is facilitated.
14. Motivation, commitment, and intentions to stay in the organisation are enhanced.
15. Voice behaviour is encouraged.
16. Employee engagement is enhanced.

### 2.2.3. Orientation, Training and Development

A substantial part of HRM is the guidance of the staff through their career steps. New employees need to get familiar with the organisation's conventions and need support finding their way into the job. This orientation phase is crucial in order to sustain the positive attitude of the new staff member in the long term.

## 2. Background and Related Work

---

The Orientation phase is followed by the training and development phase. Whereas the focus at training lies in improving the skills needed at the current position, development focuses on the abilities for likely future functions assigned (Ivancevich and Konopaske, 2013, pp. 391-393). This continuous education in combination with resulting work-related challenges has a positive impact on employee retention. However, employers need to consider how to use the newly acquired expertise, otherwise employees might feel that their potential is not fully utilised and get dissatisfied (Mathis and Jackson, 2008, pp. 78-79).

It is worth noting that the method of training has been changing from individual job-related training towards understanding business practices, corporate culture, and team building, that aims to improve employees' ability to work more effectively in groups (D. Davis and Davis, 2008), which are considered as soft management practices (von Glinow et al., 2005, pp. 395-396).

### 2.2.4. HR Effectiveness and HR Measurement

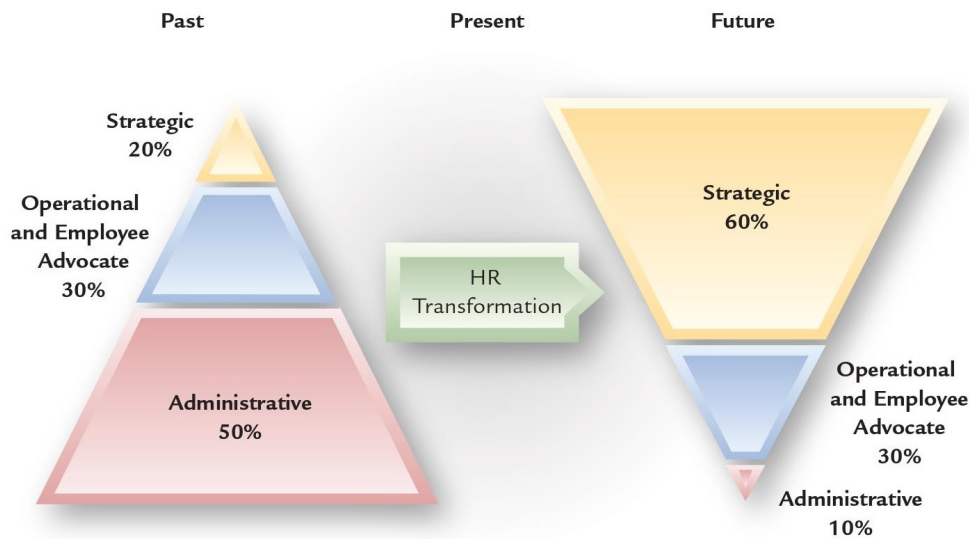
In order to successfully implement a HRM strategy in an organisation a reliable and continuous measurement system has to be implemented. Led by the misconception that the benefit of HR functions cannot be measured, the value of HRM is often not recognized and its impact devalued. In reality, the outcome of HRM activities can and must be evaluated in the same way as it done for other business units. There are several guiding principles when implementing HR measurement (Mathis and Jackson, 2008, p. 58).

It is not productive to generalise the HR metrics for all organisations as those have to be adjusted individually. However, when considering real life examples, key dimensions for HR metrics might include work climate or satisfaction level, the ratio between employees and productive output (e.g. amount of objects sold or produced, customer service case completed) and employee turnover per business unit (Cascio, 2005, pp. 106-107). These key result areas prove that human resource management measurements are not restricted to survey results based on employee polls but provide vital information on the financial aspects of human capital management (Flamholtz, 2005, p. 274).

### 2.2.5. The Status Change of HRM

HRM experienced a transformation in regard of its status as a management discipline. While it had once been treated as a compulsory *cost centre* responsible for administrative tasks it turned into a strategic *profit centre* in many organisations. Due to this shift, the functions of HRM came into focus of performance measurement. The contribution that HRM has to the goals and mission of the firm is an important parameter and therefore, measuring and evaluating





Note: Example percentages are based on various surveys.

Figure 2.1.: Changing Roles of HR Management from Mathis and Jackson, 2008, p. 11

the activities and the performance of the HRM operations is mandatory for companies' success (Ivancevich and Konopaske, 2013, pp. 7-9).

Understandably, also the shares of the three main roles of HR management have shifted. Whereas the *administrative* tasks like record-keeping comprised the largest part of HR duties in the traditional management style, *strategic* functions are in the focus of HRM nowadays. The third role, *Operational and employee advocate*, which task is to resolve employees' concern and problems, kept its percentage approximately on the same level (Mathis and Jackson, 2008, pp. 10-12).

## 2.3. Metrics

Metrics are the basic building blocks for all performance measures. Their intended purpose has been changing over the years from delivering data for the result report at the end of a project to a continuous information stream during the project lifetime (Webster and Knutson, 2010, pp. 6-7). A common misconception is to use the term metrics and key performance indicator interchangeably. While KPIs are a special type of metric, not all metrics are KPIs.

A metric itself is just the resulting number of a measurement related to the organisation or project monitored. The number does not hold any value on its own, the usefulness of the metric is defined solely by the value attributed to it by the stakeholders who interpret it in terms of project progress (Kerzner, 2013, pp. 87, 94). This emphasises the importance of carefully selecting the project metrics. Metrics that are not in line with the project objectives can have an

extremely negative impact on project performance (Englund, 2010, p. 380).

### 2.4. Performance Measures

Parmenter (2010, p. 24) defines the term performance measure as an *"indicator used by management to measure, report, and improve performance"*. The main purpose of implementing performance measures is to help a company aligning daily activities to strategic objectives. However, when organisations put their performance measures, and especially their KPIs, into action, they often cannot overcome the obstacles this critical endeavour entails (Parmenter, 2010, ix).

The more data the better is often the motto when establishing measurements for the first time. However, collecting large amounts of data can be overwhelming and distract from the really important metrics. It is therefore important to ensure that only the essential data is made available to the right people at the right time (Kerzner, 2013, vii). A common issue is that the persons responsible are not aware what a KPI really is and that there are in fact four types of performance measures, not solely KPIs. According to Parmenter (2010, p. 1) these types are:

1. Key Result Indicators (KRIs)
2. Result Indicators (RIs)
3. Performance Indicators (PIs)
4. Key Performance Indicators (KPIs)

The following chapters will define the four types of performance measures and provide an explanation on how to differentiate them and how to utilize each of them in the correct way.

#### 2.4.1. The Benefits and Pitfalls of Measuring Performance

The obvious reason for implementing a performance measure system is to increase organisational efficiency and effectiveness, believing in maxims like *"If it cannot be measured, then it cannot be managed"* or *"What gets measured gets done."* (Kerzner, 2013, p. 88). When ideally executed, implementing a performance measurement system will undoubtedly lead to a performance improvement, as it is impossible to find flaws and inefficiency in ongoing processes without being able to rely on significant data. Flamholtz (2005, p. 270) highlights the huge savings potential in the HR field when performing management actions backed with performance data. Despite this, if performance measurement is introduced carelessly, unexpected, severe backlash can occur, which can even have a negative impact on the organisation's output (Gray, 2015).

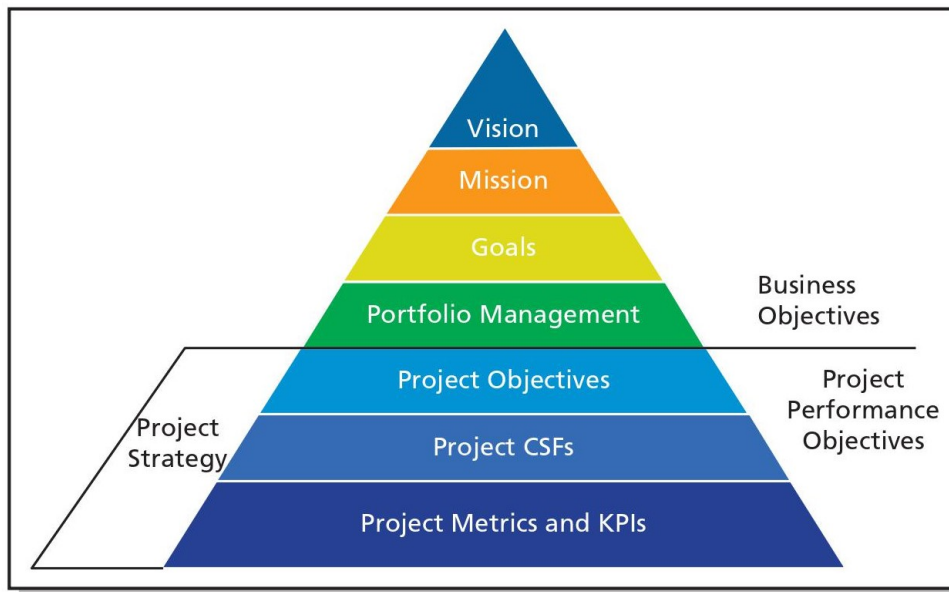


Figure 2.2.: Project's Strategy Structure by Kerzner, 2013, p. 103

### 2.4.2. Critical Success Factors

Critical success factors (CSFs) are used to quantify the organisation dependent concept of success (Kerzner, 2013, p. 104) by illustrating aspects of organisational performance (Parmenter, 2010, p. 25). Parmenter (2010, p. 199) defines the term CSFs at the organisational level, considers five to eight CSFs to be sufficient, and does not address possible differences in companies with several projects. As shown in figure 2.2, Kerzner (2013, pp. 102-103) defines the CSFs according to the project's objectives and the definition of project metrics and winning KPIs is highly dependent upon the collective understanding of the project's CSFs. Whereas CSFs can be quite different when comparing organisations, even in the very same business sector, it is essential that every employee is aware of the objectives of their organisation and aligns their daily work with the CSFs (Parmenter, 2010, p. 19). Moreover, a clear and comprehensible description of the CSFs is just as essential as a well-founded distinction between success factors and critical success factors (Parmenter, 2010, p. 199).

There can be around thirty success factors in an organisation, which are non-negligible elements for a successful operation of the company, but they are significantly less important than CSFs and without distinctions they would distract from the most crucial endeavours. One requirement that a CSF has to fulfil is to affect at least one balanced scorecard (BSC) perspective, probably a well-defined CSF will be relevant for several BSC perspectives (Parmenter, 2010, pp. 25, 37).

## 2. Background and Related Work

---



Figure 2.3.: Six-Perspective Balanced Scorecard by Parmenter, 2010, p. 37

### 2.4.3. The Balanced Scorecard Approach

R. S. Kaplan and Norton (1996) introduced the concept of the balanced scorecard approach in order to provide companies with a future-oriented approach, that should ensure a sustained competitive advantage. Therefore, the habit of only using traditional financial values for evaluating success was being questioned. Assessed as past-orientated and not sufficient for creating future value, the financial perspective was extended by three new perspectives: Customer, Internal Business Process and Learning and Growth (R. S. Kaplan and Norton, 1996, pp. 7-8).

Parmenter (2010) takes the creation of the BSC one step further by introducing two more perspectives. He states that *Employee Satisfaction* and *Environment and Community* are not neglectable influence factors, which have to be considered as vital for the entrepreneurial success. By upgrading the employee satisfaction to a BSC perspective it gains more attention from the management. The focus shifts to a continuous approach to increase satisfaction, which is passed on to customers and shareholders. Similarly, the increased focus on social issues enables the company to drive progressive values. Supporting and engaging with the community creates positive press and a favourable public perception of the company and finally, both customers and future employees are part of this very community. The updated version of the BSC is depicted in figure 2.3 (Parmenter, 2010, pp. 16-17).

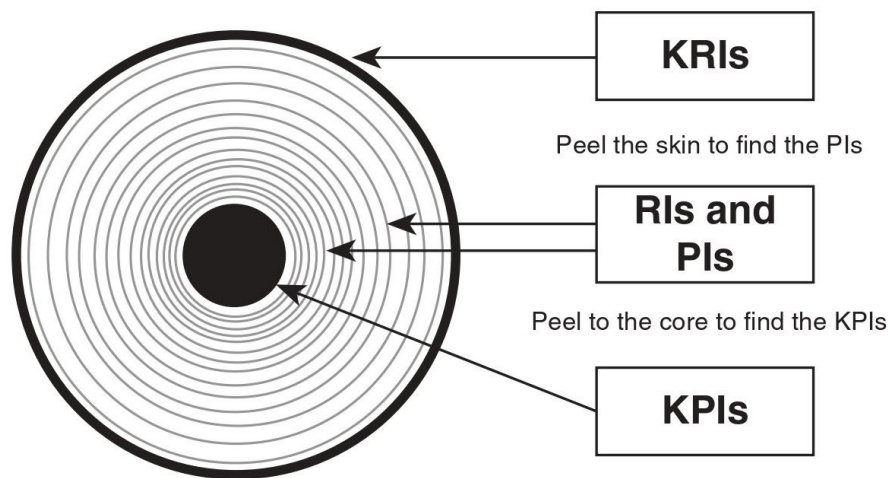


Figure 2.4.: The Four Types of Performance Measures by Parmenter, 2010, p. 2

#### 2.4.4. (Key) Result Indicators

Key result indicators (KRIs) and result indicators (RIs) can deliver a comprehensive status report on how the results of the ongoing process or project look at a certain point of time. Whilst they are perfect for checking if the success requirements of the customer and/or the organisation are being met, it is generally not possible to draw conclusions from them for improving performance in the future (Mendelssohn and Howell, 2010, p. 136). Kerzner (2013) mentions profitability as example, because it clearly depicts if the organisation is performing well, but it does not give an indication on how to improve the overall performance. Another example are financial metrics, because they are on the one hand *"linked to long-term strategic objectives"* and on the other hand they are comprised out of many components (Kerzner, 2013, p. 94).

As depicted in figure 2.4, Parmenter compares the performance measures to the layers of an onion. The outermost layer is formed by the KRIs, which directly reflects environment impacts during the growing phase as well as how it was handled on the way to the supermarket. The result indicators are visible after removing the KRIs layer. They are similar to KRIs, but do not report on CSFs or other key business processes (Parmenter, 2010, pp. 1-4).

#### 2.4.5. (Key) Performance Indicators

Continuing with the onion analogy (see figure 2.4), performance indicators (PIs) form the inner layers, before reaching the core, which is considered as the key performance indicators (KPIs). PIs do not report on key business factors, nonetheless they carry relevant information for individual teams, which use PIs to review their cadence with the project strategy. Just like KPIs they are

## 2. Background and Related Work

---

nonfinancial, but depending on the team's focus, different PIs will be used (Parmenter, 2010, pp. 3-4).

In general, a KPI provides information regarding the performance of the whole organisation or less often of a single project in a specific *tactical or strategic activity*. This activity is vital for the persistent success of the enterprise (Eckerson, 2006, p. 294).

Kerzner (2013) does not agree on the fact, that (K)PIs must be nonfinancial metrics, but also sees *actionable* as one of the most important characteristics of a KPI. If it is not possible to know how to react on the current KPI report in order to improve the performance in the future, the reporting metric is no KPI (Kerzner, 2013, p. 125; Eckerson, 2011, p. 199). In order to narrow down the indicators to the most crucial ones, a limit on the amount has to be set. R. S. Kaplan and Norton (1996, ix) previously recommended not exceeding a maximum of 25 KPIs. More recent recommendations even speak of a reduced maximum of 10 KPIs, as well as 10 KRIs (Parmenter, 2010, p. 12).

### 2.4.6. Selecting and Defining Suitable Performance Measures

Selecting the right performance measures is crucial for success, but it is also one of the biggest challenges in the KPI implementation process. In the rarest of cases, it is possible to select the correct measures straight away; usually, readjustments have to be made after a certain period of time. Of course, selection errors should be minimised right from the start, but it makes sense to review the measures and adjust them after a few months if necessary (Kerzner, 2013, pp. 133-134; Parmenter, 2010, p. 41).

In any case, it must be possible to track the progress of the project in relation to the CSFs with the selected KPIs, otherwise the endeavour will most likely not increase productivity in the organisation. Chapter 3 goes into more detail on the exact selection and implementation of the measures. However, this ongoing process starts with a planning phase, followed by an implementation phase. After an initial measurement phase, the data obtained is analysed and the findings are reported to the stakeholders. The last step is continuous improvement, which means that the steps before are analysed and adaptations according to the gained knowledge are made (Pennypacker, 2010, p. 331).

## 2.5. Free and Open-Source Software

The following subchapter deals with the term free and open-source software (FOSS) and respectively with the differences between free software and open-source software. Afterwards, chapter 2.6 gives an insight into the Catrobat project, which is also run as a FOSS project and serves as the central starting point for the practical part of this master's thesis.

### 2.5.1. Definition

The abbreviation FOSS combines two philosophies, namely free software and open-source software. Both have their origin in the GNU (Gnu's Not Unix) project, which was started in the year 1984 by Richard Stallman. In Stallman's opinion all source code of software projects should be free, envisioning source code as a kind of scientific knowledge, which should not be subject to restricted access. By simply sharing source code another problem would have arisen. Companies could leverage the freely available code, adapted and extended it and thereby gotten profit out of it. This was remedied by the GNU General Public License (GPL). This license demands resulting source code, based on source code under GPL license, must also be freely accessible and must not restrict its usage in any way (DiBona, 1999, pp. 2-3).

The free source code movement is directly linked to these principles. It represents the belief, that restricted access to software is unethical, antisocial and simply wrong. Acting as a political movement, free software is concerned with public issues and the impact proprietary software has on social equity (Sandred, 2001, pp. 37-45).

However, the open-source movement separated from the free software movement as in their opinion the strict anti-business policy prevented free software from realising its full potential and being recognised as a future-proof methodology for software development (DiBona, 1999, pp. 3-4).

Essentially, the biggest difference of these methodologies lies in the utilisation of licenses. Licenses that are conform with the open-source definition, allow to use open-source in proprietary programs. This would be against the principles of the free software movement, but in the sense of open-source it enables this software methodology to become profitable and interesting to a broader audience, while keeping the advantages of openly accessible source code (Sandred, 2001, pp. 37-38, pp. 46-50).

Based on the *Debian Free Software Guidelines (DFSG)*<sup>2</sup>, the principles of the open-source movement have been summarised in ten criteria as *The open-source Definition*<sup>3</sup>:

1. *Free Redistribution of software (even as a component of an aggregate software distribution)*
2. *Programs must include Source Code*
3. *Modifications and derived works are allowed*
4. *Maintain the integrity of the author's source code*
5. *No discrimination against persons or groups*
6. *No discrimination against fields of endeavour*
7. *Distribution of License without need for execution of an additional license*
8. *License must not be specific to a product*

---

<sup>2</sup><https://www.debian.org/social.contract#guidelines> visited on 02. June 2024

<sup>3</sup><https://opensource.org/osd/> visited on 02. June 2024

## 2. Background and Related Work

---

9. *License must not restrict other software*
10. *License must be technology-neutral*

Raymond (2001) emphasis the differences when he compares proprietary software (closed source software - CSS) to building a cathedral and open-source software to the disordered actions at a lively bazaar. At first, it seemed against basic understanding, that such a "*chaotic*" approach could lead to a superb software output. However, the success of Linux, which was developed in an open-source style, undermined the perception, that software could only be built by a small self-contained team of experts (Raymond, 2001, pp. 21-22).

Summarised, the term free software automatically implies, that this software is open-source too. Contrariwise, just because a software is open-source, it does not have to be free (Scacchi et al., 2006, p. 96).

### 2.5.2. Management in FOSS Projects

As the goals of a FOSS project significantly differs from those of a CSS project, also the management approaches must be adapted accordingly in order to meet the requirements of FOSS development. While it is obvious that revenue is not a priority for free source projects, this also applies to a limited extent to open-source projects. Compared to CSS projects, cause and effect are switched to an extent. Generating knowledge and solving problems is the focus in open-source projects, seizing business opportunities then follows naturally as a result of the solved problem. On the other hand, in profit-orientated organisations the implementation of new solutions often only serves as a means to increase their profits (Sandred, 2001, pp. 173-175).

Furthermore, also the motivational reasons of software developers, joining OSS projects differs from developer that are employed in a regular employment relationship. Neither monetary compensation nor social security drive people towards open-source projects. As surveys show, they gain benefit out of the learning and knowledge sharing in the project, achieving renown and simply experience joy while contributing to projects (Scacchi et al., 2006, pp. 96-99).

These circumstances suggest that the areas of human resource, communication and time are particularly important. A listing of main tasks in open-source project management is given by Hahn and Zhang. One of the challenges is the *human resource staffing* as it is difficult to select the right candidates in the large pool of interested developers in order to create a participatory setting with the right amount of personnel, covering the required skills and know-how.

Another obstacle is the *communication and coordination* between the contributors. Usually, the developers are scattered around the whole world and collaboration takes place exclusively in a virtual setting. Furthermore, fluctuating amounts of time developer contribute to the project and people, who only infrequently participate in the project, make the coordination even more difficult (Apache et al., 2002, pp. 316-317). These uncertainties also make sustainable



release management more challenging (Hahn and Zhang, 2005, pp. 5-6). However, a frequent release cycle makes an open-source project more appealing to contributors and also improves the quality and applicability of the resulting program, as already proven by Linus Torvald's approach when developing Linux (Raymond, 2001, pp. 21-22).

### 2.5.3. KPIs in FOSS Projects

Understandably, the differences in the basic structure of FOSS and CSS project also have their impact on the selection of KPIs, that are valuable for the respective project. Especially a non-profit FOSS project will value financial performance measures less than other organisations. Nevertheless, establishing KPIs is always an individual project tailored to the organisation and FOSS projects for sure need measures that track customer complaints, participation of developers, overdue projects, fundraising campaigns, and so on, on a daily or at least weekly basis (Parmenter, 2010, pp. 246-250).

## 2.6. The Catrobat Project

The Catrobat Project<sup>4</sup> was founded in 2010 by Wolfgang Slany, professor at the Institute of Software Technology at Graz University of Technology (TU Graz). His aim was to provide a graphical programming application for mobile devices, which was inspired by Scratch<sup>5</sup>, a visual programming framework for desktop computers, created by MIT Media Lab. The requirements included a graphical user interface, which was easy to use, a low entry barrier for new users and thus resulting in a high fun factor. The implemented application provides an easy to use drag and drop interface of colourful bricks, which represent the programming commands available (figure 2.5). The primary user group has always been young people, which can dive into the world of programming without the need of a PC.

What started as a small student project with a few internal participants at TU Graz turned into an international FOSS project with contributors all over the world. However, the core developer team still consists out of students of TU Graz, as participating in the Catrobat project is an elective course in several degree programs. Students benefit immensely from participating in a large agile software project as it prepares them for subsequent jobs in the private sector.

The first application released was *Pocket Code*<sup>6</sup> on *Google Play*<sup>7</sup> in 2014 (figure

---

<sup>4</sup><https://catrobat.org/> visited on 05. June 2024

<sup>5</sup><https://scratch.mit.edu/> visited 04. June 2024

<sup>6</sup><https://catrob.at/pc> visited on 05. June 2024

<sup>7</sup><https://play.google.com/> visited on 05. June 2024

## 2. Background and Related Work

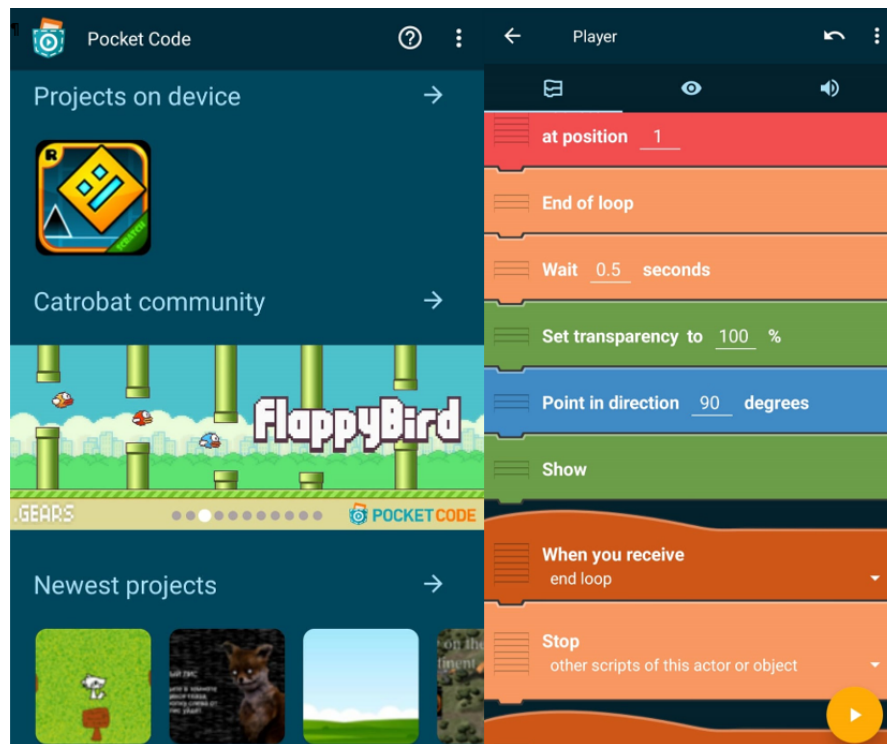


Figure 2.5.: Pocket Code Welcome Page and Programming Blocks on Android Phone

2.5). Another application, *Pocket Paint*<sup>8</sup>, completes *Pocket Code* by providing an image processing tool, which can be used separately as well as integrated in the visual programming process for creating own graphics. Each of the applications has more than one million downloads as of June 2024 and can be considered a huge success. Furthermore, games and applications developed with the *Pocket Code* application can be published and shared with other users, which creates a lively community (Müller et al., 2019).

### 2.6.1. Organisational Structure of Catrobat

The Catrobat project combines the development method of FOSS projects with the requirements of the university context. In itself, Catrobat is an independent FOSS project, which attracts many international contributors. However, the core team still consists out of university students. Overall, Müller et al. (2019, pp. 7721-7725) defined several roles, that interact with each other and play a part in the successful development of the Catrobat applications:

1. *Peripheral Contributor*: sporadic contributions, no active communication with the community
2. *Active Contributor*: active community member, frequent contributions

<sup>8</sup><https://catrob.at/pp> visited on 05. June 2024

3. *Senior Contributor*: seasoned contributor, with additional competences like member onboarding, code reviews and code merges
4. *Coordinator*: team member, who is responsible for coordination of the team activities and with authority over team decisions
5. *Product Owners (PO)*: standing, very active contributors, which control the overall vision and direction of the project

In recent years, another role emerged in the project teams. *Scrum Masters* oversee the operational methods of the teams. Although the Catrobat project does not follow to all principles of the agile methodology *Scrum*, the wording *Scrum Master* was kept, in order to emphasize that their tasks have a great similarity to those of *Scrum Masters* according to the *Scrum* definition. The ascribed responsibilities include, among others adherence to the agile approach, promoting teamwork, removing obstacles and distractions and making sure that the working conditions are suitable for every team member (Asproni, 2006, pp. 1-4).

It is entirely possible and desirable for members' roles to change during their time at Catrobat. On the one hand, members are promoted to Senior Contributors by their respective coordinator after a longer period of membership, which entails additional responsibility. On the other hand, it is also possible to have different roles. For example, one can work in one team as a developer and support other teams as a Scrum Master at the same time. If members decide to continue their work on the project for a longer period of time and if they have already acquired enough skills and experience, the possibility of becoming a Coordinator of a team emerges.

### **Agile Teams in Catrobat**

The Catrobat project is characterised, as typical for agile development (Moran, 2015, p. 1), by self-organising teams, where each team is responsible for a certain aspect of the project.

## 2. Background and Related Work

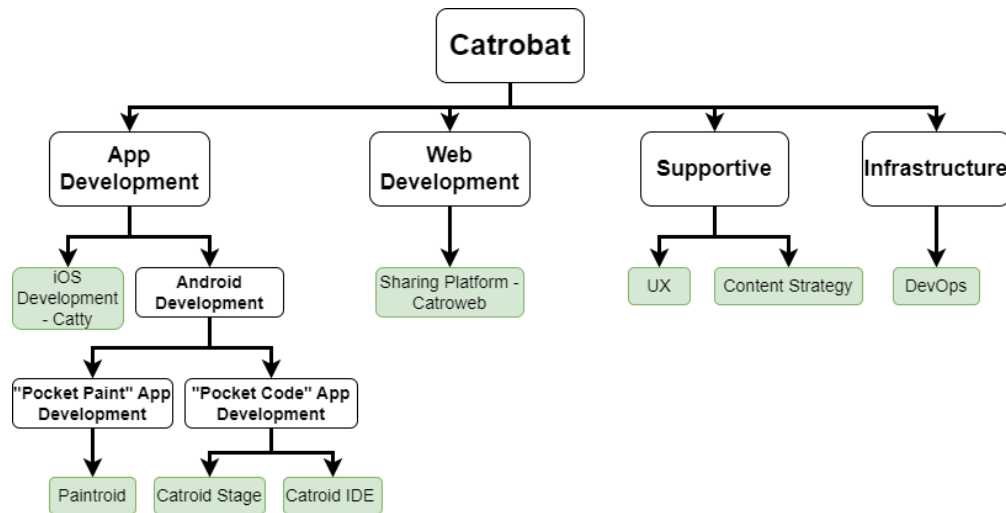


Figure 2.6.: Catrobat Team Structure adopted from (Müller et al., 2019, p. 7725)

Figure 2.6 displays the four main divisions of the Catrobat project. Each division consists out of one or more teams, labelled with their internal name, which functions can be roughly summarised as follows:

1. **Catroid IDE** and **Catroid Stage** are two separate teams, which both work on the *Pocket Paint* mobile programming application. The division is made based on the two distinct scopes of application. *Catroid IDE* is accountable for the visual coding integrated development environment (IDE), on the other hand *Catroid Stage* covers the functionality of the interpreter of the visual programming language<sup>9</sup>.
2. **Catty** or the official name *Pocket Code for iOS* is, according to the name, the iOS equivalent of the Android application *Pocket Code*<sup>10</sup>.
3. **Paintroid** is in charge of the stand-alone Android application *Pocket Paint*. Furthermore, it is also directly integrated in the *Pocket Code* application and provides a graphical paint editor for, but not limited to, *Pocket Code*<sup>11</sup>.
4. **Catroweb** works as a web-based sharing platform for *Pocket Code* projects. Uploaded projects can be accessed by the community, downloaded and even modified and extended. Furthermore, it works as a social media platform for the community<sup>12</sup>.
5. **Content Strategy**'s focus lies in defining new features for the applications.
6. **UX**'s responsibilities cover all usability and design issues for all Catrobat teams and therefore is directly linked to the work of the *Content Strategy* team. The team is also responsible for testing the applications.

<sup>9</sup><https://github.com/Catrobat/Catroid> visited on 17. June 2024

<sup>10</sup><https://github.com/Catrobat/Catty> visited on 17. June 2024

<sup>11</sup><https://github.com/Catrobat/Paintroid> visited on 17. June 2024

<sup>12</sup><https://github.com/Catrobat/Catroweb> visited on 17. June 2024

7. **DevOps** adheres to the *DevOps* definition so that they assure an integrated approach of development and operation activities, which leads to a high overall performance in the software development life cycle (Azad and Hyrynsalmi, 2023, p. 1). In principle, this team administrates the infrastructure of the whole Catrobat project. They take care about server and services and make sure that safety regulations are complied with.

### 2.6.2. Communication and Productivity Tools

Communication is a vital aspect of each FOSS project. Especially the circumstance that members of a project team are usually spread over different locations requires a reliable and quickly to hand communication tool. Particularly young FOSS projects that do not manage to set up communication and collaboration in a sustainable way often cannot avert the failure of the project (Ehls, 2017, p. 5333). In the case of Catrobat, members work with the *Slack*<sup>13</sup> productivity platform, which enables real time chat between individuals, as well as the creation of channels. Channels facilitate the collaboration of team members, because every team has their own virtual space where they can share information and meet in a *Huddle*, which is an audio chat and optionally offers video transmission too. However, face to face meetings at the University are still common and an important opportunity for the core team to exchange critical project information.

Great emphasis is also placed on another type of collaboration, which is the so-called *pair programming*. This method lets two developers work together on one computer, in person as well as connected via a video conference. While one person is actively coding, the other one has full capacity to think about how to solve the given problem. The two roles are switched from time to time. The idea is that with little personnel overhead a well thought-out and high-quality solution can be implemented (Anchit Shrivastava et al., 2021, p. 3).

Aside from communication, an open-source project relies heavily on sound documentation. This documentation should entail guidelines for new programmers on how to set up the project and how to get started with contributions. Furthermore, the current state of overall project progress, tasks in progress, responsibilities and meeting outcomes need to be tracked consistently and accessible at any time. Another main pillar, of not only FOSS development, but software development in general, is a version control system, that engenders the collaboration of software developers on a shared code base (Fogel, 2009, no pagination).

The Catrobat project builds on the *Atlassian*<sup>14</sup> tools *Jira* and *Confluence*. *Jira* is used as a project management tool, which depicts the current progress of each team on an adapted *Kanban* board (Fellhofer et al., 2015, pp. 18-20). On a

---

<sup>13</sup><https://slack.com/> visited on 18. June 2024

<sup>14</sup><https://www.atlassian.com/> visited on 18. June 2024

## 2. Background and Related Work

---

*Kanban* board tasks are visually displayed in at least three columns. The basic version has the three states *waiting*, *in progress* and *completed*. Those columns can be adapted according to specific project needs and can also be extended to include more states. By all means, the overall goal is to have a clear overview on the status of each project task and have a distinct separation of tasks in progress and finished ones (Dalton, 2019, p. 187). *Confluence* supports the organisational activities of the project. It offers team areas, where knowledge items important to the team can be shared. Moreover, meeting notes of the regular team meetings and other occasional meetings like retrospectives are kept at this central repository. For the important task of version control, Catrobat makes use of *GitHub*<sup>15</sup>, a developer platform where the project is organized in several well-defined repositories (Fellhofer et al., 2015, pp. 18-20).

### 2.6.3. Motivation to Participate

Müller conducted a survey among the participants of the Catrobat project in order to identify the biggest motivators for contributing to this FOSS project. An important factor for almost three quarters of the students is the fact that they receive credits for their ongoing university studies. Even more motivating is the vision and mission of the Catrobat project.

Apart from that, around two thirds value the new skills and experiences they gain through participating. This is largely in line with the findings of Hars and Ou (2001). In an online survey they found out that while intrinsic factors are not negligible, external factors have more influence on the motivation. The most important element is gaining human capital, which is equivalent to acquiring new skills and experiences. However, the two polls differ in the intrinsic areas of altruism and community identification, which, translated to the answer options of the Catrobat survey, is ranked much higher in the Catrobat environment (Hars and Ou, 2001, pp. 2-7).

## 2.7. Interview Evaluation

As interviews are being conducted to determine the current status of HRM processes within the Catrobat project, the content analysis procedure is briefly presented below.

There are several available methods for evaluating the interviews after transcribing them. Available approaches, that also can be combined, are, according to Bogner et al. (2014, p. 71), the qualitative content analysis, the code-based method and the sequential analytical process.

---

<sup>15</sup><https://github.com/> visited on 18. June 2024

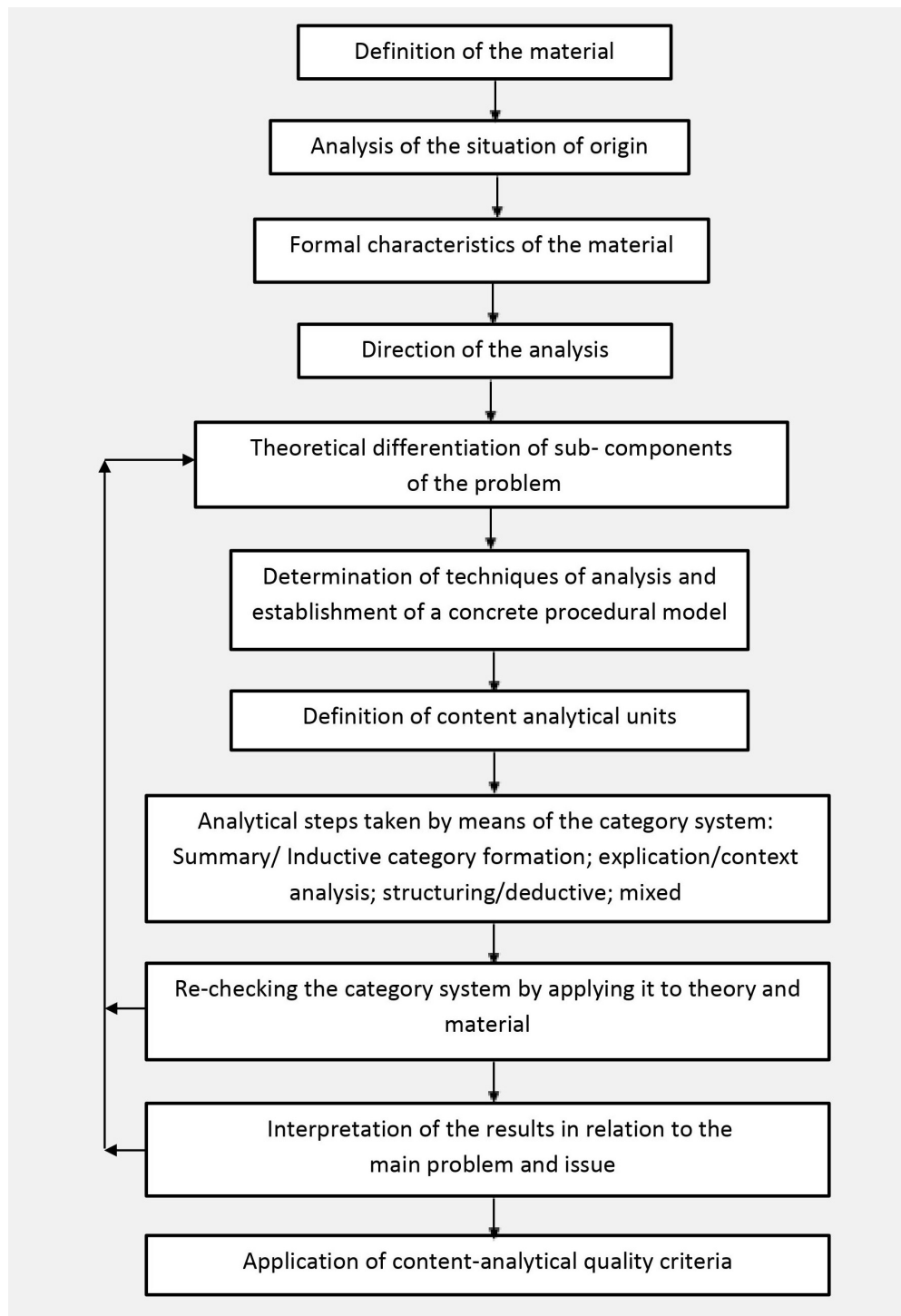


Figure 2.7.: General Step-by-step Model of Qualitative Content Analysis (Mayring, 2014, p. 54)

## 2. Background and Related Work

---

The evaluation in this master's thesis is based on the qualitative content analysis by Mayring and Fenzl (2019, pp. 633-641). Mayring introduced several steps (figure 2.7) into the interpretation process. The aim of this approach is to create a documented and verifiable end product, due to the systematic rules of the individual analysis steps. The individual steps of the General Step-by-step Model of Qualitative Content Analysis by (Mayring, 2014, pp. 56-62) (figure 2.7) can be summarised as follows:

1. **Definition of the material:** It is necessary that an exact definition of the material is done at the beginning, which must not be altered or extended during the analysis, unless it is absolutely necessary for the meaningfulness of the results.
2. **Analysis of the situation of origin:** The exact circumstances of origin need to be documented. This includes parties responsible for creation of content, motivational background of the author(s) and target group of the material.
3. **Formal characteristics of the material:** In this step the attributes of the material are defined. The entire material used has to comply with the rules defined.
4. **Direction of the analysis:** The distinction is made between concentrating on the content aspects or focusing on the intentions and emotional background of the material.
5. **Theory-oriented differentiation of the problem:** The content analysis is always based on rules and on a theoretical background, which ensures that the results provide new insights.
6. **Determination of techniques of analysis:** This step deals with analysis techniques such as summarising, explication and structuring. Depending on the material and the aim of the interview evaluation, a fitting approach is selected. This also lays the foundation for later coding of the material.
7. **Definition of content analytical units:** Analysis units are required for the evaluation, which are defined at this stage. There are three measurement variables: coding unit, context unit and analysis unit. The coding unit represents the smallest material part that is classified independently.
8. **Execution of the material analysis:** At this point the actual analysis is conducted. Afterwards a verification run of the category system is done. If there is a need for changes, the step needs to be repeated.
9. **Interpretation of the results in relation to the main problem and issue:** The aim of this step is to interpret the analysis with regard to the research questions so that the results from the content analysis are combined with the theoretical foundations.
10. **Application of content-analytical quality criteria:** In order to confirm the objectivity of the research results, quality criteria should be applied. These criteria include the reliability and validity of the results.



## 3. Implementation

In the following chapter, the practical operations that were carried out to answer the research questions are examined in detail. First of all, the fundamental approach is explained, and limitations of the research structure are discussed. Afterwards, the utilized approaches for insight finding are presented. All findings, their interpretation and analysis can be found in chapter 4.

### 3.1. Methodology

In order to gain insight into the inner workings of the Catrobat project, interviews with experts were conducted. In the case of this master's thesis the *coordinators* of the Catrobat teams are seen as experts. The coordinators take up a management position inside the team. However, other than *Product owners* they are still involved in the daily business. Which means that they participate in the weekly meetings, are direct contact persons for other team members and are very familiar with the code base. In fact, they act as an interface between the developers and the upper management, which is an ideal position to be knowledgeable about the long-term goals and the vision of the project defined by the upper management as well as the struggles in the daily development tasks conducted by the team members.

The aim of the interview was to review to status quo of HRM processes in the project, to uncover weaknesses in the organisation and to get an understanding on what the critical success factors of the project are. This qualitative research method is used as a base for defining the KPIs that are able to deliver information on the project performance. In view of the current project processes, recommendations for the introduction and improvement of HRM practices will then be given. A one-off data extract is intended to provide an outlook on the visualisation possibilities of the KPIs. In the long term, these measurement methods ensure that the effectiveness of improved and newly introduced HRM practices, among other things, can be quantified. Detailed information about the interviews can be found in chapter 3.3, respectively the implementation of the KPIs is described in chapter 3.4.

## 3.2. Limitations of the Implementation Area

Defining KPIs is always a unique endeavour tailored to a specific project or enterprise (Kerzner, 2013, p. 119). Hence, all implementations and findings are customised to the Catrobat project characteristics. In order not to go beyond the scope of this master's thesis, four coordinators were interviewed. This provides a precise insight into the internal processes of these four development teams, which bear the main responsibility for the development of Catrobat's mobile applications. However, it only allows limited conclusions to be drawn about the other teams within the project. These other teams are responsible for support activities such as interface design and application development management.

Even so, the organisational structure and approach of the individual teams are very similar. Apart from the different technical areas of responsibility, the teams differ primarily in the number of members. While coordinators in smaller teams may struggle less with the lack of HRM processes, identified problems can largely be transferred to the entire project. Although, the implementations are Catrobat specific, conclusions can also be drawn for other FOSS projects in general. FOSS projects have some specific traits, that set them apart from CSS projects, e.g. the approach in hiring new contributors compared to the hiring process at a private enterprise (Hahn and Zhang, 2005, p. 5). Therefore, findings that relate to the specifics of FOSS projects, both in HRM and in the implementation of KPIs, can be valuable for similar endeavours in other FOSS projects, even though they cannot be adopted without adjustments.

## 3.3. Interviews

This chapter deals with the theoretical basis of the expert interview and explains why this form of interview was chosen to interview the coordinators of the *Catrobat* teams. In addition, the creation of the field manual, the selection of the interviewees and the scope of the interview are highlighted.

### 3.3.1. Interview Method

The interviews were conducted as expert interviews. This method is a less structured survey instrument and is part of the qualitative, empirical social research (Meusner and Nagel, 2009, pp. 465-466). It is to be considered, that two types of expert knowledge can be differentiated. On the one hand there is operational knowledge and on the other hand contextual knowledge exists. Contextual knowledge describes the conditions in the expert's environment and the problem structure that the expert is dealing with. In contrast to this, the operational knowledge is oriented towards the actions of the expert. Based on these actions, conclusions regarding the inner workings of certain processes

can be made and a better understanding of them can be achieved. Usually, both types of knowledge are considered as important and should be queried in an interview. While the contextual knowledge is usually a type of explicit knowledge and can therefore be communicated directly by the interviewee, the operational knowledge presents the interviewer with greater challenges. For this reason, the guided expert interview is recommended to reconstruct both the explicit and implicit parts of the operational knowledge (Meusner and Nagel, 2009, pp. 470-472).

In order to conduct a productive interview, the interviewer must acquire thematic expertise. This is done, among other things, by creating the guideline for the interview (see sub-chapter 3.3.4). The questions should be defined in a way, that the answers can reveal the *How* of the decisions and actions. The expert's logic of action can then be put together from the answers (Meusner and Nagel, 2009, pp. 473-475).

A clear distinction must be drawn between the expert interview and the biographical interview, even if the expert interview can contain narrative passages, which is beneficial for analysing the action of the expert and gaining access to the implicit knowledge of them (Meusner and Nagel, 2009, pp. 473-475). It should be emphasised that the expert interview is never about the person themselves, but only about the knowledge that this person has acquired and can pass on (Liebold and Trinczek, 2009, p. 37).

### 3.3.2. Expert Selection

The possible group of candidates for the interview is limited anyway, as only Catrobat members at the management level are suitable. As already explained, the role of the coordinator functions as link between the operational level, which are the developers responsible for the implementation of the applications, and the executive level, which is constituted by the POs and the founder of the International Catrobat Association. While facilitating the collaboration within the team as well as between the teams, the coordinators gain detailed insights into the challenges of day-to-day implementation work and are committed to removing obstacles wherever possible.

Another decision was made regarding which team coordinators to interview. In order to make an informed decision, several factors were taken into account. The key factors were the size of the team, the length of time the coordinator has already held this position and that the team is a development team, responsible for a Catrobat application. The size of the team is seen as significant, because smaller teams can more easily mitigate the lack of HRM practices. This is manifested by the frequent lack of strategic HRM procedures in small companies (Marlow, 2006, pp. 474-475).

The specification of a certain amount of time as coordinator is necessary so that experience could be gained in the process of onboarding new members

### 3. Implementation

---

as well as in the process of offboarding. However, also coordinators with a shorter experience were chosen, as they can offer more insight on their own onboarding process and the challenges they face at the beginning of their new position. Furthermore, they are more likely to scrutinise established processes and may see potential for improvements (Cunha and Chia, 2007, pp. 559-561). The condition that the teams must be development teams of one of the mobile applications, which represent the main applications of the project, enables a better comparison between the teams, as they all have a very similar approach to day-to-day business. This led to the selection of the *Paintroid* team, in charge of implementing the *Pocketpaint* app for Android, the *Catty* team, responsible for the iOS application, the *Catroid IDE* team and the *Catroid Stage* team, which together are in charge of implementing the *PocketCode* Android app.

#### 3.3.3. Interview Guideline Structure

The interview guide serves as the basis for the interview to be conducted. According to Kaiser (2014), it is an instrument for data collection and fulfils three essential tasks. Firstly, it serves to structure the interview and helps the interviewer to conduct a coherent, easy-to-follow interview. It is important to define the number and order of the questions. The sequence of questions should be coherent for the interviewee and enable a fluent conversation. The number of questions can vary greatly depending on the topic of the interview but should primarily be based on the estimated duration of the interview and reduced where possible. In order to facilitate the structure of the interview, the individual questions are assigned to topic blocks that clearly show which area of interest the question is trying to discuss.

Furthermore, the guide must once again inform the interviewee about the aim and significance of the interview, as well as explain the upcoming interview situation, regardless of whether the interviewee has already been informed in advance. The guideline must also contain information on the handling of personal data and the possibility of rendering it anonymous. Finally, the guide underpins the researcher's acquisition of knowledge in the expert's area of specialisation, for example by including explanatory statements. Based on the principle of openness, deviations from the guidelines are possible if, for example, the interviewer asks interim questions (Kaiser, 2014, pp. 52-54).

#### 3.3.4. Interview Scope

The complete guideline can be found in the appendix A, but this chapter will briefly summarise its contents. As an introduction to the interview, the expert is given the opportunity to talk informally about their role and experiences in the Catrobat project. It is also pointed out once again that there is enough time

for the interviewee to answer all questions in peace:

*"First of all, I would like to ask you to tell me something about your field of activity and your role in the Catrobat project! You can take as much time as you like for this. I won't interrupt you, just make a few notes, which I will discuss later."*

The following three questions also deal with the field of activity of the expert. On the one hand, they should enable the expert to find his way into the dialogue, on the other hand, insights should also be gained regarding particularly important and very time-consuming tasks of the coordinator, as well as conclusions drawn about challenges at the beginning of their activity. This focus at the start is set, because as there is a possibility that the coordinator has come to terms with the inadequacies of the processes over time and now sees them as less serious (Prielipp et al., 2010, p. 1499).

Afterwards, questions regarding the topic *Team organisation* are asked. At first, some general information about the composition of the team is requested:

*"Please give me some information about the team structure you coordinate!"*

Some questions also contain sub-questions, which are only asked if the answer has not already been given in the first question. In this case, the current size of the team as well as the consistency of the team size is a crucial factor that is covered by a sub-question if required. Moreover, this topic block tries to find answers regarding the duration of contribution of the team members, the importance of regular team meetings and the cooperation within the team as well as between the teams

Another focus is set on the development process of the team. Apart from a general description of the life cycle of a development task, a special emphasis is set on the planning phase and the amount of information that is already available in that phase.

Furthermore, the onboarding and offboarding processes need to be analysed in more detail. The next topic block is therefore dedicated to these. The onboarding process at the Catrobat project was already examined in great detail in a master thesis conducted by Paul Schreiner and is not the focus of this interview guideline (Schreiner, 2022). However, it is worth to take a closer look at knowledge management (KM), which is particularly valuable in the transfer of knowledge between new members and long-standing members who leave the project (Mathis and Jackson, 2008, pp. 262-263). The takeover by a new coordinator is of particular interest here, as inadequately documented HRM processes, which are carried out improvised, can lead to major difficulties after the transfer (Omotayo, 2015, pp. 8-9):

*"I would like to speed forward to the moment when you are going to resign from*

### 3. Implementation

---

*your role as coordinator. Please tell me how you imagine the selection of your successor! How smooth will this transition be in your expectation?"*

The second to last topic area is called *Performance Measures*, which indicates its importance with regards to this master's thesis, which tries to find approaches to improve the performance and productivity of the Catrobat project. The core question of this part is as follows:

*"Please elaborate on the options and measures you have to evaluate the contribution and performance of your team members!"*

This question attempts to ascertain the current status of performance evaluation inside the team. Based on this, steps can then be defined that make the performance of a team more clearly visible. Another crucial question relates to the CSFs of the project and tries to find out, which underlying factors are relevant for the success of the whole project:

*"What do you believe are the key factors that have contributed most significantly to the achievements of the Catrobat project, and why?"*

Furthermore, the coordinators are also asked about the benefit of the 24-hour threshold. This threshold requires that every collegiate contributor spends at least 24 hours per month at the Catrobat project and attempts to achieve a minimum level of participation of each member.

Last but not least, a topic block is dedicated to employee motivation, a central topic of HRM, which goes hand in hand with a targeted increase in productivity, as without motivation, productivity also decreases (Shahzadi et al., 2014, p. 159).

#### **3.3.5. Adaption of the Guideline After the First Interview**

After the first interview had been done, some minor changes were applied to the guideline. Those changes were not regarding the content, apart from question thirteen, but aimed to improve the wording and comprehensibility of the questions. This affected questions eight, nine, twelve, fourteen and eighteen.

Other changes concerned the division of questions into further sub-questions in order to obtain answers to all parts of the questions nine, twelve and twenty. Although, the first expert, which was the coordinator of the *Paintroid* team, managed to answer all questions as intended, the alterations assured that also the other experts could understand the questions in the expected manner. Question thirteen was rephrased from:

*"In your opinion, how is success defined in the Catrobat project? In other words, which factors need to be met that you would evaluate the whole project and espe-*

*cially your team as performing well?"*

to:

*"What do you believe are the key factors that have contributed most significantly to the achievements of the Catrobat project, and why?"*

The first wording could be understood to mean that the expected answer was aimed at KPIs that had not yet been implemented to measure the success of the project continuously. However, the improved phrasing should point to success factors that are deep-rooted in the Catrobat project and responsible for the success so far, despite not being measured.

### 3.3.6. Recording and Transcription

Both interviews were held online, and an audio recording was made for each interview to facilitate the transcription. The verbatim transcription of the interviews can be found in the appendix B, questions are marked with *I* for *Interviewer* and answers are marked with *E\** for *Expert*. The *\** is replaced by the numbers 1 to 4 in order to differentiate the experts. To numbering is done in this order:

1. Paintroid Coordinator
2. IDE Coordinator
3. Stage Coordinator
4. Catty Coordinator

To increase comprehensibility, expletives and pauses were removed. Furthermore, unfinished sentences were converted into a legible form. However, care was taken not to change the content of the sentences.

### 3.3.7. Evaluation

Applying the qualitative content analysis by Mayring (see 2.7), a short explanation on how the individual steps of the general step-by-step model of qualitative content analysis were conducted is given:

1. **Definition of the material:** The interview transcripts are used as material for this study.
2. **Analysis of the situation of origin:** The aim of the interviews conducted is to depict the present state of HRM processes in the Catrobat project. Therefore, team coordinators, which had agreed voluntarily to being interviewed, shared their opinions in an online interview setting with the author of this master's thesis.

### 3. Implementation

---

3. **Formal characteristics of the material:** The characteristics of the transcripts created are defined at the beginning of this chapter.
4. **Direction of the analysis:** In this master's thesis the content aspects are paramount.
5. **Theory-oriented differentiation of the problem:** In this case, the analysis is based on the theoretical findings of the chapter 2.
6. **Determination of techniques of analysis:** The content-based structuring is used in this work, as particular emphasis is placed on the content-related aspects of the interviews.
7. **Definition of content analytical units:** In this master's thesis the coding unit is defined as a single sentence in an interview and each interview is a separate context unit. The chronological sequence of the interviews is used for the unit of analysis as a higher order.
8. **Execution of the material analysis:** Each conducted interview underwent all steps.
9. **Interpretation of the results in relation to the main problem and issue:** The interpretation of the results can be found in chapter 4.1.
10. **Application of content-analytical quality criteria:** In the course of the content analysis and the limited time frame and scope of the master's thesis, as well as the restricted, specific group of experts, it is not possible to apply these criteria sufficiently.

#### 3.4. KPI Implementation

The focus of this section is to present the concept on the basis of which the KPIs for the Catrobat project have been selected. The development is based on Parmenter's (2010, pp. 41–105) multi-stage process, although this had to be simplified and adapted within the scope of the master's thesis. A major difference is that suggested workshops, that would involve many participants and last several days were not feasible, primarily due to the limited availability of decision-makers, but also because of the specified content framework of this master's thesis. Figure 3.1 depicts the links between the individual parts of the organisation's strategic management.



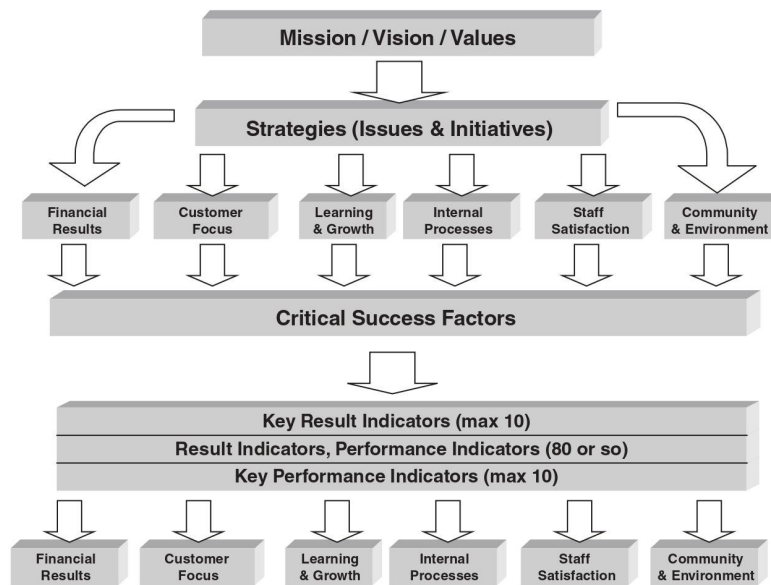


Figure 3.1.: How CSFs and KPIs Fit Together and Link to Strategy (Parmenter, 2010, p. 206)

### 3.4.1. The Mission and Vision of Catrobat

In order to be able to define CSFs and subsequently KPIs, the objectives of the organisation must first be analysed (Aguinis, 2013, p. 38). Catrobat has a mission and vision statement, that describes what the whole organisation is aiming for. Mission and vision differ from each other in their scope and life span. On the one hand, the vision is an entrepreneurial goal that is reachable in a defined time frame, if the organisation operates successful. On the other hand, the mission is formulated in such a way that it is considered the ultimate goal of the company, the accomplishment of which may never be fully achieved and can therefore drive the company for decades (Parmenter, 2010, p. 38).

Aguinis (2013, p. 69) claims that a well-crafted mission statement can answer the following questions:

1. Why does the organisation exist?
2. What is the scope of the organisation's activities?
3. Who are the customers served?
4. What are the products or services offered?

The Catrobat Mission reads as follows:

*The nonprofit Catrobat project aims at developing solutions which inspire teenagers and adults to learn coding, get creative and be prepared for the digital future of our world. Simple logical processes, creative solutions and easily understandable connections are essential in the world of tomorrow.<sup>1</sup>*

<sup>1</sup><https://catrobat.org/about/> visited on 20. July 2024

### 3. Implementation

---

The Catrobat Vision is defined similarly:

*"Our vision is to provide young people with the chance to include these principles in their everyday digital-life. We believe that free, age-based and motivating software, which can be used directly by teenagers or in education, is essential in order to prepare the next generation for their future."*<sup>2</sup>

These mission and vision statements are capable of answering the four previously asked questions. The reason why Catrobat exists is to prepare people for the challenges of the digital future. Which is also a mission statement, that will always be prevailing as technological progress will not stop in the future and people will continuously be challenged by new developments. The main customer group are teenagers and also adults who are eager to become familiar with the field of information and communications technology (ICT).

In order to define a concrete way on how to reach the mission, the vision statement describes the offered product as software and the scope is delimited by applications for creative coding that can be incorporated in everyday life.

#### 3.4.2. Noteworthy Specifics of the Catrobat Project

In order to craft meaningful metrics for the Catrobat project, several considerations about the unique circumstances of the organisation must be taken into account. There are two major factors in which Catrobat differs from other ventures that are used in the literature as the basis for the KPI definition process. First of all, there is the fact that Catrobat is managed as a FOSS project. Though, there are references in literature on how to implement KPIs in FOSS projects and what to pay particular attention to.

However, the spirit of a FOSS project is combined with the status of a university project. The core team, which consists out of students of the TU Graz is a rather unusual arrangement for a FOSS project. This places Catrobat somewhere in between a CSS and a FOSS project. Of course, it adheres to all FOSS principles, but the team structure has similarities with that of a profit-oriented company. But the fact that the majority of contributors are students entails further peculiarities, which were revealed by the interviews that were conducted with the coordinators of the project (see chapter 4.1). These include among other things a very high fluctuation rate, which is due to limited hours students have to complete based on their selected courses, great uncertainty regarding the availability of the contributors, because of other obligations that students must fulfil, and the aspect, that students do not receive payment but work for gaining ECTS-credits.

---

<sup>2</sup><https://catrobat.org/about/> visited on 20. July 2024

All these differences must be taken into account when defining the CSFs and subsequently the KPIs. This is because it changes both the organisation's business objectives and the individual participants' incentive to participate, and the metrics must reflect these changed conditions accordingly.

### 3.4.3. Definition of Critical Success Factors

Finding an assessment of success, that suites all stakeholders of a project is nearly an impossible endeavour, because the perception of success differs corresponding to the personal point of view and goals of the involved stakeholders as demonstrated by K. Davis (2014, pp. 196-199) by an extensive academic research. Lamprou and Vagiona (2022, p. 251) ranked CSFs based on their occurrences in the consulted references and came to the conclusion that most common CSFs were *"project mission/goals/objectives/scope/vision, top/senior management support and commitment, project communication and the project planning/monitoring/control"*.

Parmenter (2010, p. 207) suggests concentrating on the senior management team and the organisation's oracles, which are the employees with the most experience and the most insight into the organisation, in order to find success factors for further consideration. These must then be analysed and reduced to the really critical ones. However, in an ideal process, everyone involved in the project should have a say in how they would define success in the project (Kerzner, 2013, p. 102).

According to Parmenter (2010, pp. 205-206), there are some guidelines for characteristics of CSFs that can really drive project progress:

- *Are worded so a 14-year-old can understand them and run the company*
- *Will be no surprise to management and the board as they will have talked about them as success factors*
- *Apply to more than one balanced scorecard perspective*
- *Have a great influence on other success factors*
- *Are focused in a precise area rather than being the bland statements that strategic objectives often are*

Comprehensibility for 14-year-olds refers to the fact that the statements should be written as specifically as possible so that every employee can clearly understand what exactly the statements relate to. Examples like *"increased customer satisfaction"* or *"increased profitability"* are not specific enough to enable employees to work in line with them (Parmenter, 2010, pp. 204-206). The CSFs are reviewed by comparison to the BSC perspectives. This is discussed in more detail in the next chapter (3.4.4).

Understandably, the development of CSFs does not bring any added value if not all employees know what the company's objectives are. Only if the CSFs are clearly communicated to all employees can they align their daily work accordingly (Parmenter, 2010, p. 212).

#### **CSFs in the Catrobat Project**

Based on the analysis of the interviews conducted with the coordinators of the project teams (see chapter 4.1), taking into account the mission, the vision and the specifics of the Catrobat project and the adherence to guidelines compiled from academic sources, the following five CSFs were identified for the project:

1. Ensuring collaboration and knowledge transfer within the organisation (CSF 1)
2. Fostering an environment of open-mindedness and mutual appreciation to encourage contributors to meet their full potential (CSF 2)
3. Attracting quality members to the organisation and retaining experienced contributors (CSF 3)
4. Valuing and acting on customer feedback (CSF 4)
5. Keeping up with technological advancements and innovations (CSF 5)

For the sake of clarity, the CSFs are referenced in the following chapters with CSF 1 to CSF 5, as indicated in the enumeration in the brackets.

In addition, the creation process of the CSFs is shown below based on the findings generated from the coordinator interviews. The experience and expert knowledge of the interviewees made it possible to draw well-founded conclusions about the critical success factors in the Catrobat project. Instead of conducting a workshop, the final CFS statements were then presented to and approved by project founder *Wolfgang Slany* and two PhD students, who hold positions as decision-makers in the project.

#### **Ensuring collaboration and knowledge transfer within the organisation**

Catrobat's most valuable resources are the people who invest their time, knowledge and skills to drive the project forward and make it a success. While the highly motivated and best trained university students, who participate in the project as part of their university education, are undoubtedly a huge asset, they also present a challenge for the project management.

As most student's participation is limited to a few hundred hours, there is a lively turnover of participants. In the absence of cooperation and knowledge sharing, these circumstances harbour the risk of constantly losing the knowledge recently obtained. If knowledge is not actively passed on, employees disappear after completing their hours, along with their knowledge. This in turn requires new participants to start from scratch and acquire the necessary knowledge on their own. With the very limited participation time already mentioned, these participants hardly have any time to work productively on the project.

In order to counteract this, two successful practices have already been implemented. This refers to the mentoring system for newly hired members and in general the pair programming development method, which enables two contributors to work together on a problem and exchange ideas and knowledge

on an ongoing basis:

*"[...] when we have new team members, we ask more experienced team members to mentor them for the first couple of tickets. That's when a lot of knowledge transfer can happen."* (E3, L. 364-366).

The benefits and effects of the mentoring are discussed in more detail in the interview analysis in the chapters 4.1.4 and 4.1.7. As explained in the conclusion of the interview analysis (chapter 4.1.8), the current processes are not fully developed yet. Under unfortunate but recurring circumstances, knowledge is nevertheless lost. Furthermore, communication between the teams is particularly limited. There is still a lot of potential, especially in improving this:

*"So, yeah, even if sometimes it would be great, because we have three Android native teams, so they can easily work with each other, but, yeah, we don't have quite that community, I think"* (E1, L. 131-133).

Contributors miss the opportunity to directly communicate and collaborate with members of other teams, requiring the coordinators to be involved unnecessarily, as discussed in chapter 4.1.2.

This CSF reminds all contributors that keeping and sharing knowledge is key for the success of the Catrobat project. It emphasizes the adherence to the procedures, that are already in place and proven beneficial. At the same time, it raises the attention towards the impediments that still hinder collaboration across teams and requires further developments to reduce the knowledge loss.

#### **Fostering an environment of open-mindedness and mutual appreciation to encourage contributors to meet their full potential**

Most of the participating students are inquisitive and curious contributors who want to finally put the theoretical knowledge they have acquired at university into practice. This is a great opportunity for the project to transform this existing motivation into project progress. The coordinators mentioned several times during the interviews how motivated participants affect the overall success of the organisation, which is analysed in chapter 4.1.6 in greater detail:

*"when people really felt that their ideas were being heard and considered, that's when I experienced in the upcoming weeks that participation was better. If you put people in the right team and you see that the things they are interested in, that they could put that to use, that's when they're really motivated. That's something, for instance, that I see from former team members that are now working in the Godot team, where they're working on a pipeline, on DevOps things, getting everything to work. That's what they're really interested in and that's also resulting in a high*

### 3. Implementation

---

*contribution.*" (E3, L. 550-557).

This quote provides several indications of how motivation is created or rather kept high among the participants. First of all, the contributors need to feel valued and to be seen as an important part of the project. If coordinators are open for ideas put forward by participants, even if they may not yet have much experience in the project, the participants feel recognised. If it is then even possible for contributors to realise their own ideas, their motivation will be far higher. This is the case, because that approach increases their intrinsic motivation, that means that pursuing the activity itself rewards the contributor. This yields better performance results than focusing on extrinsic motivation, which needs to link the completion of the activity to some kind of external reward (Bernhard Schmid and Jonathan Adams, 2008, p. 61).

Another positive effect comes with the increased identification with the project. The review process, which is often somewhat disregarded (see chapter 4.1.8), could receive increased attention. Contributors, that implement their own ideas, are keen on seeing them included in the final application, available to the customers. As well as they might show higher incentives to review code written by other contributors as they are intrinsically motivated, to keep the quality of *their* app high.

Moreover, there are special teams, like the *Godot* team mentioned by the interviewee, which use the newest technologies to make the applications future-proof. Godot<sup>3</sup> is a free and open-source engine for developing video games. Participants with a particular affinity for technology flourish here and can pursue their interests.

This CSF highlights the importance of the single contributor and their ideas. It stands for the fact that the Catrobat project is open for differing opinions and objective discussions. Everyone is heard and allowed to contribute their views. Mistakes are allowed to be made and serve the personal learning progress. That also describes the grading process explained by one contributor (see chapter 4.1.5):

*"I do this most important question for me is did they learn something and how much did they improve? I try to compare the person they were when they started, and I try to compare the person they are, when they are leaving"* (E4, L. 511-514).

All of these merits lead to increased intrinsic motivation and thus to increased personal performance and greater project success.

---

<sup>3</sup><https://godotengine.org/> visited on 15. August 2024

### Attracting quality members to the organisation and retaining experienced contributors

Repeating the fact, that the motivated and skilled contributors are essential for the success of the Catrobat project, the organisation must continue to ensure that sufficient qualified members are available. Students are attracted to become new members as they are offered the unique opportunity to earn ECTS while contributing to a FOSS project, which feels different in comparison to most theoretical university courses (see chapter 4.1.6).

However, keeping the students in the project after they received their grade proves more difficult. Chapter 4.1.2 illustrates the challenges, which come with the high fluctuation of the team members. It also influences the knowledge transfer as discussed in chapter 4.1.7. When experienced members leave and are replaced by several inexperienced contributors there need to be enough overlap time, in order to engender a thorough onboarding process for the newcomers. Adverse effects on the project performance can occur, when this overlap is not possible due to shortages of members (see chapter 4.1.4):

*"And if many of the team members leave at the same time and we can't motivate them to take additional courses at Catrobat, then the team can shrink pretty rapidly. [...] And then we saw that if you have less members, you also have less people for code review. So we saw that many of the tickets that were kind of already done were stuck in code review" (E3, L. 324-330).*

Furthermore, there are some roles in the project, which require previous experience in the project in order to hold it. These include the coordinator role and the scrum master role. However, the number of potential candidates is limited due to the experience required and finding successors is a challenge. This is why, at the time of the interview, there are two coordinators, who are simultaneously coordinator and scrum master of their team, as shown in the chapter 4.1.1. Furthermore, one coordinator also doubts that they will be able to find a replacement for their position in time:

*"What could be an issue is to find a coordinator, so right now I have a team with a lot like more senior people, so I also already talked to them that I need a new coordinator next year, because I'm really finished with my hours [...]" (E1, L. 284-287).*

This CSF is used for indicating the importance of a continuous influx of new members as well as for remembering that keeping experienced contributors as long as possible in the project is incredibly important. Although there is no shortage of interested students, there are from time to time vacancies in the teams that cannot be filled quickly enough. A proactive approach to recruitment can result in an improvement here. However, the major deficits

### 3. Implementation

---

exists in employee retention. It is therefore critical to be constantly aware that great project progress can only be achieved with the help of experienced contributors. Only then will new incentives be actively created to persuade members to remain loyal to the project for an extended period of time.

#### **Valuing and acting on customer feedback**

In its mission statement, the Catrobat project describes the goal of preparing people, and especially teenagers, for the technological challenges of the future while acting as a non-profit organisation. Since there is no incentive to achieve profit, the top priority is to develop applications that fulfil the needs of users in the best possible way. A certain self-benefit can be recognised in that young people who can be inspired by technology, and in particular computer science and programming, at an early age are more likely to consider a technical education. Thus, also increasing the number of potential future participants. In order to achieve the goal of getting as many people as possible excited about technology and making them fit for the technological challenges of the future, new customers must be acquired and existing ones retained.

The CSF urges to appreciate the feedback received by customers and to see it as valuable information that can and must be used to continuously improve the applications. Therefore, end user requests often serve as source of ideas for new features. This feedback is then incorporated into the organisation's medium- and long-term goals, which are defined by the decision-makers and communicated to the teams by the POs. Having these long-lasting objectives in mind has again a positive effect on the motivation of the individual contributors:

*"[...] especially if people higher up in the organisation, product owners and essentially the bosses, the people responsible, if they come to meetings, if they tell us about news, if they tell us like visions for the next middle to long-term plans for the product for the next couple of months, that also helps to motivate people to work on things because then you have a goal in mind, what you're working towards" (E3, L. 424-429).*

The social initiatives of the Catrobat organisation also go beyond the development of learning applications, as can be seen from the timeline of the organisation's achievements<sup>4</sup>. This includes events like the *Girls Coding Week*, that encourage girls to get involved in application development and to consider the Information Technology (IT) sector as a potential career path. Another project, called *Code'n'Stitch*<sup>5</sup>, was also dedicated to promoting gender equality in the IT sector and was even honoured with an award for research based on culture and gender equality, awarded by the TU Graz.

---

<sup>4</sup><https://catrobat.org/timeline/> visited on 15. August 2024

<sup>5</sup><https://catrobat.org/mind-the-gap-award-for-codenstitch/> visited on 15. August 2024



In combination with Catrobat's vision, which aims at removing entry barriers regarding technical education for young people, regardless of origin or gender, the CSF puts the purpose of creating value for the customers at the centre of Catrobat's ambitions.

#### **Keeping up with technological advancements and innovations**

Helping people to *"be prepared for the digital future of our world"*, as declared in Catrobat's mission statement, implies that the organisation itself has the requirement to be future oriented and to drive innovation in its internal processes as well. This attitude is also appreciated by the contributors who, as university students, are given the opportunity to work with new technologies and apply them in a real project, which again is a motivational factor for many contributors (see chapter 4.1.6):

*"That's something, for instance, that I see from former team members that are now working in the Godot team, where they're working on a pipeline, on DevOps things, getting everything to work. That's what they're really interested in and that's also resulting in a high contribution."* (E3, L. 554-557).

This as well ties in with the second CSF, which emphasises the independence and self-reliance of participants in promoting and implementing innovation.

The fifth CSF is intended to express the innovative idea behind the project. Whilst preparing the users of the Catrobat applications for new technologies is already an integral part of the organisation's mission, internal development is thus also encouraged to see technological progress as an opportunity and as essential for the continued existence of the project. Even if all members of the organisation are convinced anyway that openness towards technological advances has a positive impact on project performance, it is important to manifest that it is this attitude, among others, that causes the success of the project. Especially, to assure that this principle must not be forgotten, as well as to continue to organise the project work according to it in the years to come.

#### **3.4.4. The Balanced Score Card Perspectives**

As described in chapter 2.4.3, the BSC consists out of six perspectives. In order to validate the suitability of the chosen CSFs the potential success factors are matched with the six perspectives (Parmenter, 2010, p. 221). The five defined CSFs and their dependency on the individual BSC perspectives are clearly presented in the table 3.1. As the Catrobat project is a not-for-profit organisation and is not financed through its operating activities, the financial perspective is not considered. However, this in no way means that not-for-profit organisations can ignore the financial perspective in general. Although, increasing their profits

### 3. Implementation

---

is not an objective of these organisations, they still need to ensure financial stability, for example by maximising donations from outside sources (Martello et al., 2008, pp. 67-80). In the special case of Catrobat the funding is related to the university and holds a special position.

	BSC Perspectives					
CSF	CF	IP	LG	ES	EC	F
CSF 1	x	x	✓	✓	x	-
CSF 2	x	✓	✓	✓	x	-
CSF 3	x	x	✓	✓	✓	-
CSF 4	✓	x	x	x	✓	-
CSF 5	✓	✓	✓	✓	x	-

Table 3.1.: CSFs in relation to the BSC

In order to illustrate the correlation of the CSFs and the BSC perspectives, the following enumeration explains the six perspectives and their relevance regarding the CSF statements:

#### Customer Focus (CF)

The centre of this perspective are objectives that evaluate the interactions with the customers, as well as objectives that assess positioning of the organisation on the market. Overall, the objectives of this perspective help to gain new customers and equally important, to retain a loyal customer base. Depending on the customer target group, specific measures will vary, an example of generic objectives include: "*customer profitability, customer satisfaction and market share*" (R. S. Kaplan and Norton, 1996, p. 26).

The proposed CSF *Valuing and acting on customer feedback* is linked to the customer focus perspective. Caring for the opinion of loyal customers gives them the feeling that their opinion is important and that their concerns and ideas are heard. The more the customers are integrated in the further development of the application the more likely it is that they will stay loyal to the offered products and that they recommend the applications to others.

In addition, the CSF *Keeping up with technological advancements and innovations* ensures that the applications remain attractive to end users. Demand will only be maintained if the learning content offered is state of the art, which is why innovation is essential.

#### Internal Process (IP)

In contrast to the traditional approach, this perspective not only serves to optimise existing processes, but is also intended to create openness towards new

process implementations. This supports achieving the organisation's strategic goals in the best way possible, as established processes are often not sufficient and thinking outside the box generates the necessary progress. Overall, this perspective aims to outline the crucial processes, that are capable of attracting and retaining customers as well as those processes that are responsible for financial success. Furthermore, another aspect is innovation, as the future conditions might require completely new approaches and processes in order to still succeed as an enterprise (R. S. Kaplan and Norton, 1996, pp. 26-28).

Two of the five advised CSFs are related to this perspective. *Fostering an environment of open-mindedness and mutual appreciation to encourage contributors to meet their full potential* considers the actual development process of the applications, which is the core process of the Catrobat project. This process relies on the cooperation of the individual team members. Only by utilising collaborative practices like *pair programming* and open idea sharing can the development requirements be fulfilled effectively.

*Keeping up with technological advancements and innovations* reminds that especially in the IT sector innovation is omnipresent. All processes are evaluated against new developments and adapted where necessary in order to maximise the benefits of innovations.

#### **Learning and Growth (LG)**

The main aspect of this perspective is to make the organisation and its processes future-proof. Although the internal process perspective also aims to promote innovation, its main focus is on optimising current procedures. In order to compete long-term against other market actors, employees need training, technology and systems must be updated and procedures must be refined. The foundations for these improvements are laid with the objectives of this perspective (R. S. Kaplan and Norton, 1996, pp. 28-29).

Closely linked to this perspective is the increasingly important knowledge management system. A well-founded KM system enables continuous learning and further training of employees, whose level of knowledge is essential for the success of the entire organisation (Jelenic, 2011, p. 42).

Four out of five CSFs target this perspective in various means. A lively exchange of knowledge guarantees that all contributors are up to date regarding new advancements. An open-minded environment, accessible for new ideas and novel ways of thinking fosters future oriented ways of working. Attracting new members, which are still students and eager to work with visionary technologies, as well as a fundamentally positive attitude towards technological progress ensures that the Catrobat project stays competitive.

### 3. Implementation

---

#### **Employee Satisfaction (ES)**

This perspective values the employees as a very important, if not the most important, resource in the organisation. The objectives aim to deliver a regular reporting of the satisfaction rate, as well as providing incentives for increasing motivation and happiness (Parmenter, 2010, pp. 16-17).

As this is a crucial perspective, four out of the five CSFs deal with it. Putting emphasis on teamwork, knowledge transfer and idea exchange, encourages members to actively contribute to the project. Focusing on contributor retention implicitly demands to provide a pleasant work environment. In addition, the focus on innovation makes the work attractive for young people who want to realise the potential of today's fast-paced technological developments.

#### **Environment/Community (EC)**

The importance of engaging with the community and presenting the organisation as progressive and aware of its impact is ever increasing. This attitude appeals to young consumers as well as potential employees and generates positive media coverage (Parmenter, 2010, p. 17).

Hence, *Attracting quality members to the organisation and retaining experienced contributors* and *Valuing and acting on customer feedback* focus on this perspective. A lively exchange with the community makes it possible to adapt the products on offer to the needs of the consumers. Moreover, interest in participating in this project is thereby also increasing.

#### **Financial (F)**

This perspective presents a rundown of the economic performance of the organisation and it was therefore advised to place it at the top of the perspective hierarchy. Suitable objectives for this perspective include return-on-capital-employed, economic value-added or operating income (R. S. Kaplan and Norton, 1996, p. 25). For-profit organisations, that aim to generate financial returns to investors as their longstanding target, are advised to consider the objectives of the other perspectives as supportive for achieving the goals of the financial perspective (R. S. Kaplan and Norton, 1996, p. 61).

Later, Kaplan came to the conclusion that putting the financial perspective on the top of the hierarchical depiction of the perspectives is not appropriate for all organisations. After some not-for-profit organisations decided to move the customer perspective into focus, Kaplan recommended to define an "*overarching mission objective*" that is able to act as reference point for the objectives of the other perspectives (R. Kaplan, 2001, p. 360).

The Catrobat CSFs are intentionally not directly linked to the financial perspective as the financial success is not an inherent aspect of the mission and vision of Catrobat.

### 3.4.5. Developing the KPIs

After having the CSFs aligned with the BSC perspectives, the next step is to define the KPIs, which will be used to measure the accomplishment of the objectives the organisation strives for. In a similar manner as the CSFs, the KPIs were created based on the expert interviews and then presented to the project decision-makers and adapted, where required. Moreover, attention was paid to ensuring that the KPIs have the necessary characteristics defined by Parmenter (2010, p. 88):

- *Nonfinancial measures*
- *Measured frequently (e.g., 24/7, daily, or weekly)*
- *In accordance with the corporate philosophy exemplified by CEO and senior management team*
- *Understandable measures, that clearly indicate what action is required by staff*
- *Measures that tie responsibility down to a team*
- *Significant impact (in line with CSFs and covering as many BSC perspectives as possible)*
- *Encourage appropriate action (e.g., confirmed to have a positive impact on performance)*

A closer look at the KPIs reveals the connection to the CSFs and the benefits for the project. Moreover, implementation details like frequency of measurement and the data source for the KPIs are discussed. Table 3.2 provides a detailed breakdown of the KPIs. One must also consider that KPIs can cause adverse effects and behaviours instead of providing a performance boost, as already discussed in chapter 2.4.1. Hence, the issues that can arise when implementing the respective KPI are part of the analysis as well.

### 3. Implementation

KPI Acronym	KPI Name	Linkage to CSFs	Data Source	Frequency of measurement (daily, weekly, monthly)
WMA	Weekly meeting attendance (last 30 days)	CSF <sub>1</sub>	Confluence (Meeting notes)	weekly
SPI	Story points implemented/merged (last 30 days)	CSF <sub>2</sub>	Jira (Ticket Board)	daily
AHRT	Available human resources per team	CSF <sub>3</sub>	Time sheet	daily
RSR	Average rating since last release	CSF <sub>4</sub>	Android/iOS Statistics	daily
TTIP	Average time tickets stay in the implementation process phases	CSF <sub>1</sub> CSF <sub>5</sub>	Jira (Ticket Board)	daily

Table 3.2.: Overview of Catrobat's KPIs

#### Weekly meeting attendance (WMA)

The central hub of internal team communication is the - depending on the team - weekly or fortnightly team meeting. As shown in chapter 4.1.2, the regular meetings are crucial for efficient teamwork. Since most contributors work alone, or if pair programming is carried out, then mainly with recurring partners, the meeting is often the only opportunity to exchange ideas and get help:

*"And they will be also talking about some things that they maybe at the moment, block them from doing certain things or they are waiting on. And this is very important because otherwise we wouldn't like talk with each other for a very long time" (E4, L. 149-152).*

Particularly, newcomers strongly rely on the frequent meetings, as they often struggle to find their way in the huge codebase without the support of senior developers. Personal meetings at university had provided frequent opportunities to socialise in the past, but they have become very rare, especially after the SARS-CoV-2 pandemic:

*"I joined during COVID times, so I'm kind of used to this, but I know from people in the past that it was easier to find, for instance, pair programming partners*

*and work together when you were used to being at university on campus” (E3, L. 193-196).*

Another contact point is the team-internal group chat. However, enquiries for advice are often not answered or only answered very late due to various reasons like information overload (Eppler and Mengis, 2004, p. 1119). Hence, the weekly meeting is the first point of contact for questions and problems.

Therefore, a high attendance rate at the meetings generally promotes collaboration and the transfer of knowledge within the organisation. It is a simple approach to depict the actual interaction between the team members. Furthermore, high participation rates increase the likelihood of holding further technical meetings, as the necessity for them can be recognised in the team meetings. As a result, these technical discussions are again an event where knowledge sharing happens.

Attention must be paid to active participating of the collaborators. If members attend the meeting, but do not participate actively the KPI will display satisfactory results, but no improvement in collaboration is achieved. Even so, the interviews suggest that this problem is highly unlikely to occur. Most members draw their motivation from active participation and use the meetings for purposeful discourse (see chapter 4.1.6).

As meetings are held once a week in most teams, a weekly measurement interval is recommended. Each team can react independently to their own results, as the rate is measured and analysed on a team-by-team basis. The data is drawn from the *Catrobat Confluence* site, where notes are created for every meeting. These notes include the attendance information of every collaborator and can be exported for the statistics.

### **Story points implemented/merged (SPI)**

The main process in the development teams is the implementation of tasks for the mobile Catrobat applications. A more detailed description of the development procedure is provided in chapter 4.1.3. Briefly outlined, the process takes course in several phases. As soon as a ticket is set to *ready for development*, i.e. all the necessary preparations have been done to implement the requirements, the ticket can be selected by a developer. Only when the ticket is selected and pushed to *in development* does the active part of development begin. While the developer is engaged with developing the ticket, the ticket stays in the *in development* column. When the implementation is finished, the ticket is moved to the *code review* column. Another developer needs to review the ticket and verify the changes. When this step is completed, the ticket can be seen as implemented. However, the final affirmation is still missing. A final Quality Assurance (QA) review, executed by the coordinator, PO or in special cases by a senior developer, has to be approved before a merge is done.

### 3. Implementation

---

Measuring the number of tasks a team can complete in a given period of time is a common metric in software development projects. In order to realise the measurement, a time period for the measurement is required on the one hand and, on the other, a measurement parameter that makes individual tickets comparable, as the effort per ticket can vary considerably (Zheng et al., 2021, pp. 5-10).

In agile projects that use Scrum, or adapted versions of it, the measurement duration is aligned with the sprint duration. A sprint usually lasts between two and four weeks and the intermediate product at the end of the sprint should be usable software. For determining the effort of a single task, a planning poker event is held, where everyone involved in the implementation process can vote for the complexity of the ticket. Every ticket is awarded with story points and therefore the tickets are comparable to each other (Sachdeva, 2016, p. 16793, Schwab and Schwab-Matkovitz, 2013, pp. 210-211). Based on these definitions a team velocity can then be defined as the amount of story points implemented in a single sprint (Zheng et al., 2021, pp. 5-10).

As the involvement in Catrobat per coordinator is only a few hours per week, fixed sprints are not constructive. Instead, development is continuous, and planning sessions are only held when necessary:

*"But in a normal occasion, you have a so-called planning game where you discuss user stories that were given and decide with the developers" (E2, L. 232-233).*

Therefore, a fixed time period for measurement is not reasonable. Instead, using a continuous time interval that always covers the last thirty days is more meaningful. A shorter interval would not be robust due the relative high fluctuation in the available hours. Access to the data is provided via the development boards in *Jira*, which already include information on story points and when a ticket was implemented and merged.

This KPI compensates for two weaknesses in the implementation process. On the one hand, it gives the coordinators a certain degree of planning reliability, as they can use this KPI to better estimate how many story points can be realised in the coming weeks. As elaborated on in the interview analysis conclusion chapter 4.1.8, the lack of information regarding the availability of the members makes it difficult to plan ahead. Although this KPI does not manage to provide direct information on the member availability, which would not be realisable, it works with estimated amounts of story points able to be implemented. The collaborators appreciate the flexibility they have in the project and that they can always prioritize university exams and work obligations over the participation in Catrobat. If estimates were required, they would most likely be very conservative and imprecise in order to rule out the possibility of an undershoot.

Moreover, in the mind of the CSFs, the coordinators are given additional



incentive to implement features and contribute to the collective performance. In this respect, the KPI aims to utilise the full potential of the team members. Due to the QA review, which was recognised as a bottleneck in chapter 4.1.3, it is recommended to use two variations of the KPI. One measurement depicts the amount of story points without the QA review phase, which is called *Story points implemented*. Influencing this variant is entirely up to the developers and their performance. The other version, *Story points merged*, measures the story points of the tasks that have been merged into the project and is therefore also depended on the contribution of the coordinator and the PO of the team.

It should be noted that the assessment of the story points is highly subjective and is not comparable between the teams (Schwab and Schwab-Matkovitz, 2013, p. 210). Trying to rank the individual teams according to their performance based on story points would lead to misleading conclusions. Moreover, it could even happen that the story point estimations are increased over time, in order to seem more productive. Thus, it is recommended to provide the collaborators only access to the KPI of their team so that no meaningless performance comparisons can take place.

#### Available human resources per team (AHRT)

One major problem that affects all Catrobat teams is the high member fluctuation. Several coordinators state in the interviews, that they struggle to plan the availability of human resources in their team. As described in the analysis in the chapters 4.1.4 and 4.1.8, there are situations where the number of team members can be halved in a very short time period:

*"We are currently at the moment, that three, no I mean four members are leaving the team right now. So until now, we were seven members, eight members, I'm sorry. And after that, we will only have four anymore"* (E2, L. 96-99).

The importance of the human resources for the project manifests itself several times in the interviews and was taken as a reason to define CSF 3. After all, a contributor shortage has many negative aspects and must be avoided. In order to counteract these tough situations, this KPI can be used to predict when it will be necessary to recruit new members. To obtain a meaningful result, a simple formula is applied:

$$\text{AHRT} = A \times \frac{\text{Count}(\{i \in T \mid h_i > 80\})}{|T|} \quad (3.1)$$

$A$  represents the overall number of available hours in a team. To obtain this value, all available hours of the team members, as stated in the time sheet, are added. In addition, the percentage of members, that have more than 80 available hours, is derived from the time sheets.  $|T|$  represents the total number of team

### 3. Implementation

---

members and the number of hours left for team member  $i$  is denoted by  $h_i$ . By multiplying these two values, the KPI value is determined.

This multiplication further reduces the KPI value when many contributors are nearing the end of their time at Catrobat. Again, this factor counteracts two issues that would otherwise occur. One coordinator stated that members with less than 80 hours are asked not to begin with new tickets, but to focus on completing all the tasks they have started and also to pass on the knowledge they have acquired by mentoring newcomers. This reduces their ability to contribute to the project progress. Furthermore, there could be cases, especially in smaller teams, where a single contributor has a large number of open hours because they want to complete various university courses at Catrobat. Even if several other members are nearly finished with their hours, the KPI would still be within an acceptable range. However, the team would still be exposed to the risk of rapid shrinking.

Current values can be taken daily from the time sheets. Members are already required to record their hours worked on a daily basis. The time sheets also contain the total number of hours that still need to be completed. The correctness of the KPI is based on the proper documentation of the hours worked. If, regardless of the agreement, contributors record their hours significantly later, it could happen that an existing demand is not recognised despite the KPI. Nevertheless, this issue is manageable in all likelihood. At least, contributors need to track their hours in any case once a month in order to meet the required threshold of minimum participation.

#### **Average rating since last release (RSR)**

Unlike the other KPIs, this metric focuses on the customer perspective. In order to sustain a large user base, *valuing and acting on customer feedback* is defined as critical success factor. As the main product of Catrobat is its mobile applications it is apparent to take advantage of the many reviews available in the app stores. Bug reports have always been used to continuously improve the applications. While the number of bug reports also provides a certain amount of information about satisfaction, distortions can occur. Questions arise as to whether a bug that appears particularly often should be assessed differently from many different bugs that occur infrequently. Also, a bug that occurs may not always reduce user satisfaction. This depends very much on the type of bug.

The average rating, on the other hand, is a concise metric that provides immediate information about customer satisfaction. If it drops, the problem can be resolved based on the bug reports. Keeping a constant eye on the current rating motivates members to fix serious bugs as quickly as possible and to implement features requested by users in time in order to improve ratings with the next release. Because the application is improved with each release, only

the ratings of the latest version should be used for the current analysis. The latest ratings are drawn daily from the app store statistics.

One drawback of this KPI is that the effects on the ratings caused by application improvements only become visible after the next release with a severe delay. Certainly, that bears the risk of reduced motivation, when there is no immediate impact on the reviews visible. Despite that, it also opens up the opportunity of aiming for an increased release frequency, which is aspired already, in order to be able to showcase regular improvements.

#### **Average time tickets stay in the implementation process phases (TTIP)**

This KPI bears resemblance to the second KPI, as both measure the progress in the implementation process. Therefore, most of the elaborations made are also applicable to this metric. Both the reasons for the introduction and the method of measurement are essentially the same. However, there are significant differences that justify the coexistence of the two.

An initial indication is provided by the additional correlation with the fifth CSF. This CSF emphasises the importance of technical advancements in the Catrobat project. This also included innovations that must be implemented in the applications. That is why the amount of time a single ticket stays in a phase of the implementation process is highly interesting. If tickets stay for a long time in the *ready for development* column it means that new features are being implemented slowly.

Moreover, this KPI is used to mitigate bottlenecks in the process. In the interview it was mentioned several times that the *QA review* is often a bottleneck. Apart from this, it is always possible that temporary bottlenecks occur. To illustrate, it could happen that many intriguing tickets are *ready for development* at the same time. Naturally, the developers will start to code on these tickets as they are curious to implement exciting features. Thereupon, the time needed on average for developing will decrease and at the same time the value for the *code review* column will increase, because all developer resources are busy with implementing features. Now, the coordinator has the opportunity to counteract this trend and to assign a code review to every developer.

Issues could arise when dealing with outliers. Tickets, that have become stale and continue to stay in a certain column increase the KPI value significantly. Then again, when team members and coordinators actively make sure that no such tickets exist this KPI has the additional benefit of keeping the developer board up to date and tidy. If it turns out, that outliers are strongly influencing the metric, an adaption of the KPI, that excludes the outliers, should be introduced.

## 3.5. Summary

The restraint of only selecting coordinators as interviewee for the expert interview proved to be meaningful. In this way, the differences regarding team internal approaches and issues, that might be unique to a team, could be identified. Yet, the homogeneous selection allowed to determine the most crucial factors for the continuous success of the project. Both processes, that already work very well and others, that need some refinement were characterised. Although no large weaknesses were found, there is still a lot of potential for improving the project performance.

Therefore, the second part of the chapter took a closer look on the strategic and operational prerequisites needed for those refinements. The overall objective of defining KPIs needed some groundwork to be done in the first place. Based on the already existing vision and mission of Catrobat and keeping the six BSC perspectives in mind, the critical success factors of the project could be established. In doing so it became clear that the essential strength of the organisation are the many skilled and curious contributors. In the final step five KPIs were designed in order to track the vital factors of the organisation. Again, this was done in context with the CSFs and the BSC perspectives. Because all development teams act quite similar it was refrained from defining additional PIs.

Furthermore, no KRIs and RIs were defined in order to solely focus on the essential KPIs. Hence, it only remains to track the defined KPIs and to analyse their actual impact on the project performance, when the project work of the contributors is aligned with them.

## 4. Results and Evaluation

In this chapter the conducted interviews are analysed and based on those findings, key values of the Catrobat project, the actual processes of the project and shortcomings of these processes are discussed. These interviews were also used to define CSFs (see chapter 3.4.3) and subsequently KPIs (see chapter 3.4.5) for the project. A reporting of the KPIs as a dashboard application is presented, as well as suggestions for further actions in order to work in line with the KPIs. The limitations of the work are also highlighted, and an outlook is given on how the findings of this master's thesis can be used to continue working on an HRM and performance measurement system for the Catrobat project and subsequently for other FOSS projects.

### 4.1. Interview Results

The content of this chapter is the illustration and interpretation of the results of the qualitative content analysis from the coordinator interviews. A further subdivision into main and sub-categories is made to improve clarity (see figure 4.1 for an overview of all categories). Each of the four interviews is analysed according to all defined categories. There is no fixed order in which the individual interviews are analysed; instead, the answers were arranged in a meaningful and comprehensible order for the analysis. The concrete discussion about the current HRM practices and deficits in the project processes takes place in chapter 4.1.8, but the essential results, in order to deal with the research questions, are already highlighted at the end of each thematic category in an interim conclusion.

## 4. Results and Evaluation

Deficiencies in the current Processes	0
Tickets on hold	9
Duration of Development Process Unpredictable	5
High Contributor Turnover	3
Difficulties in Collaboration	5
Unfinished Tickets	4
Knowledge Loss	5
Inefficient Manual Checks	6
Lack of Information	11
Lack of Human Resources	9
Knowledge Management	0
Different Levels of Knowledge	1
Technical Documentation	4
Knowledge Transfer	12
Team Member Motivation	0
Socialising	7
Incentives for Motivation Increase	5
Influence of Motivation on Project	6
Reasons for Participating in Catrobat	5
Performance Measures	0
Individual Performance Evaluation	4
Monthly minimum Participation	6
Consequences when being below Threshold	4
Impact on Team Member Grading	5
Needs for Objective Performance Measures	4
Key Factors for Success	5
Recruitment, On- and Offboarding	0
Onboarding of new Members	4
Transition to new Coordinator	5
Offboarding Process	5
Recruitment Process	7
Experience with Shortage of Team Members	7
Influence of Coordinator	3
Development Process	0
Availability of Team Members	5
Implementation Process	5
Time Critical Tasks	4
Planning Phase	4
Bottlenecks	5
Duration of Ticket Implementation	6
Team Organization	0
Scrum Master Role	6
Collaboration	1
Collaboration within the Team	4
Collaboration across Teams	13
Team Meetings	5
Impact on Project Progress	4
Attendance	6
Team Structure	5
Amount of Team Members	3
Duration of Participation	5
Fluctuation of the Number of Members	5
Field of Activity	0
Challenges at the Beginning	4
Time consuming Coordinator tasks	4
Most important Coordinator tasks	4
Role in the Catrobat Project	4

Figure 4.1.: Code System depicting the Categories of the Interview Content Analysis

### 4.1.1. Field of Activity

All four interviewees hold the same position as a coordinator of a project team. Therefore, their explanations regarding their role are quite similar, however they differ in the level of detail. One coordinator emphasizes the manifoldness of their position as they *"[...] do everything the team needs"* (E1, L. 28).

Activities, that are mentioned by every coordinator include the team organisation, the support of team members and the communication and collaboration with other teams. The *Catty* coordinator has also further responsibilities as their team lacks a scrum master - in contrast to the other teams - and they have to take over this role:

*"I'm also some sort of, you could say, scrum master. So, it's not only I'm a coordinator, but I'm also holding our team meetings and our weekly meetings"* (E4, L. 32-34).

But also the *Paintroid* coordinator claims that they have to hold the meetings, as they *"[...] don't have any scrum master or something like that"* (E1, L. 37-38). However, this is a temporary issue due to a lack of scrum masters in the project, whereas there has never been a separate scrum master position in the *Catty* team.

#### Most important and time intensive tasks

Special attention was paid to the activities that are considered especially important, but also to the activities that the coordinators consider taking up a particularly large amount of time. The coordinators agree, that the organisational tasks, which are required in order to enable the developers to be productive are crucial. One coordinator provides a concise summary of their main tasks:

*"Most important tasks are having a rough overview of where tickets are. And giving support to team members, especially when I see that sometimes tickets end up taking longer. So that we kind of like, that we see impediments and then discuss them during team meetings and try to remove them."* (E3, L. 56-59).

Two interviewees also mention the recruiting and onboarding process as an important responsibility of theirs, *"[...] since most team members do not stay for a very long time, so that we get new members. So, we can continue with our work, [...]"* (E2, L. 39-40). Those two coordinators also consider the onboarding process and in addition, the documentation and knowledge transfer from leaving members to newcomers as very time consuming:

#### 4. Results and Evaluation

---

*"Yeah, the most time would probably be onboarding for new team members, since this is kind of a crucial part" (E2, L. 44-45).*

On the other hand, three coordinators mention that holding meetings and keeping track of all tickets and contributor progress is particularly time-consuming or at least requires the most mental capacity.

*"And if I'm setting up the team meeting and checking with when everyone's available and the time and dates, and sometimes we then have to reschedule or we have to sometimes maybe set it up again because nobody's got really that much of time, especially when there's lots to do at university. So I think it would be most of the time the organisational task of setting up, keeping in check, taking everyone into account" (E4, L. 58-64).*

Two of the coordinators are also highly engaged in technical discussion, often conducted as face-to-face meetings lasting several hours. It is worth mentioning, that those technical talks are *"[...] not often directly related to coordinating the team members itself, but also it affects their work in the future"* (E3, L. 80-81) and are therefore still handled by the experienced coordinators, reducing their capacities for their actual job.

#### **Challenges when becoming a Coordinator**

In order to get an understanding for the obstacles a new coordinator might face in their role, the coordinators were asked what they found particularly challenging, when they became a coordinator. Whereas one interviewee *"[...] got a quite good list of responsibilities and that made things a little bit easier"* (E3, L. 97-98), the others struggled with the lack of clarity regarding their exact role definition, as well as with the high complexity of the project and the partly meagre documentation available:

*"Since Catrobat is like a kind of a big project with many, many, many members, it's kind of hard to find your way on all the documentation that there exists, to understand how the default processes do work" (E2, L. 56-58).*

*"Probably just knowing what tasks I should do. Like, I think there wasn't a clear guideline on what are my tasks, what do I have to do, and also not having an active product owner at that time" (E1, L. 51-53).*

Considering a future transition to a new coordinator, all four coordinators agree that it is very difficult to get into this role without any guidance of the former coordinator. They also agree that *"[...] it would take some time since the project is huge, since our info, our letters, our information in the confluence pages are big"* (E2,



L. 366-367). In this context it is also mentioned that it *"[...] could be an issue is to find a coordinator [...]"* (E1, L. 284) and *"[...] if the old coordinator just leaves and the new coordinator has to work on everything, [...], questions usually arise. And that's when it's helpful that the old coordinator is still available"* (E3, L. 394-398).

### **Interim Conclusion**

The outline of the coordinator tasks indicates that this role is a not precisely delimited collection of activities that are essential for the functioning of the team and the progress of the project. Coordinators seem to mitigate weaknesses in the processes with their presence, be it collecting technical knowledge and passing it on personally, organising time consuming technical meetings, or manually checking the progress of the individual developers on their tickets. This makes it also quite difficult to facilitate a smooth transition to a successor without a long transition period in which both coordinators work together.

#### **4.1.2. Team Organisation**

All four teams have a similar structure, they consist out of a coordinator, a product owner (PO), several developers and optionally a scrum master. The number of developers is different for each team, but around three to six contributors. There is no hierarchy in the team, there is just a differentiation regarding the experience level of the developers:

*"And then we only have the team members, essentially, and we don't have a strong hierarchy there. I mean, the least little bit of hierarchy that we have is that we know that some team members are new because they've been onboarded recently"* (E3, L. 113-116).

### **Member Fluctuation**

However, the amount of team members is quite unstable in the Catrobat project. The high fluctuation is triggered by the approach on how to convert university course efforts and respectively ECTS in hours to be completed in Catrobat. It happened that contributors *"[...] did 200 hours, and then they were gone. So yeah, that can be quite challenging"* (E1, L. 68-69). This is unfortunate for the progress of the project, but hardly avoidable with the current approach:

*"Some people stay in touch with the project and really are interested to see everyone in the team succeed and the project succeed. And some people are, at the moment they are done with their hours, they are gone, which is also fair. That's*

#### 4. Results and Evaluation

---

*the agreement which is in place at the first time" (E4, L. 374-377).*

If several members complete their hours at the same time, it can lead to considerable changes in the team composition and also to bottlenecks among the developers, as happens in the IDE team:

*"We are currently at the moment, that three, no I mean four members are leaving the team right now. So until now, we were seven members, eight members, I'm sorry. And after that, we will only have four anymore" (E2, L. 96-99).*

This also has a direct impact on the productivity of the team affected:

*"Sometimes we have far too less members in our team, which then kind of holds back the productivity in our team" (E2, L. 88-89).*

#### **Team Meetings**

Every team has either weekly or biweekly meetings, which are *"like a daily in a scrum setting"* (E4, L. 147). This means that the members *"[...] discuss what everyone is doing right now, and what they plan to do in the future, like, in the next two weeks"* (E1, L. 81-83). It is also entailing the opportunity for members to raise awareness, if they face any obstacles:

*"And they will be also talking about some things that they maybe at the moment, block them from doing certain things or they are waiting on. And this is very important because otherwise we wouldn't like talk with each other for a very long time" (E4, L. 149-152).*

The importance of these meetings is confirmed by all interviewees, however, one coordinator points out that this is a subjective feeling, as these meetings have always been held and there is therefore no comparison to a team without regular meetings and the benefits are not measured objectively. Even so, a positive benefit of these meetings cannot really be denied with regard to the experts' statements:

*"I would say they are really important, since if you kinda are stuck as a, let's say, junior developer in our team, you are not really well known with the structural environment of the project. You have many questions, and the meetings are mainly also here to discuss those questions so that the developer can continue with the work. Without such meetings, it would be really hard to communicate a necessary knowledge or questions so that you could continue with your workflow" (E2, L. 132-138).*

In order to hold effective meetings, a high attendance rate of the contributors is mandatory. Especially coordinators that oversee a bigger team note that [...] *it's not that easy to get all of them together, since most of our members are students at the university. And therefore, they have other attendances as well for lectures, for example.* (E2, L. 121-124). Nevertheless, all coordinators are quite satisfied by an average attendance rate of 66% - 80% of the contributors. Then again, sometimes contributors still need a reminder by the coordinator to not forget about the meetings:

*"Sometimes we have some team members, some team members who are more often absent than others. But I think we have a quite good attendance rate. Sometimes we feel the need to mention to people that taking part in the meetings is mandatory"* (E3, L. 176-179).

### **Collaboration**

Considering the positive impact of team work on project performance (Lukusa et al., 2021, p. 890), reducing impediments in collaboration is key for success. In the interviews, it became clear that there are major differences between collaboration within a team and between teams. Cooperation within the teams works well overall, apart from individual cases where there is a lack of mutual sympathy, as reported by one coordinator. Language is also cited as an obstacle, as the majority of students are native German speakers, but there are also some contributors who do not speak German. In general, both the language of the course of studies and the working language at Catrobat are English, but German is often preferred among the students:

*"In the team, I think, yeah, it depends. Sometimes people, like, do not like each other, which, yeah, can be a major issue, which I already had. Also, I think the language is also sometimes an issue, like, not that big of an issue, but people who speak German tend to do more pair programming with people that are also speaking German."* (E1, L. 111-115).

The impression is created that the individual teams are silos, because members *"[...] don't know the members of the other teams, so they really don't like to work with those"* (E1, L. 130-131). The communication between the teams is handled almost exclusively by the coordinators, which leads to time-consuming and cumbersome processes:

*"So it was like, yeah, someone from my team spoke to me, I spoke to the coordinator there, the coordinator there spoke to the team member there, or maybe not, I don't*

## 4. Results and Evaluation

---

*know. And then maybe I reminded them. So it was like this back and forth with like some two coordinators in between, which maybe we didn't know and didn't need in that special example" (E4, L. 210-215).*

In general, collaboration between teams is reduced to the necessary minimum, although there is potential for collaboration and thus for increasing productivity:

*"So, yeah, even if sometimes it would be great, because we have three Android native teams, so they can easily work with each other, but, yeah, we don't have quite that community, I think" (E1, L. 131-133).*

### Interim Conclusion

The coordinators face several challenges regarding the team organisation. They have to deal with the high fluctuation of contributors, which in particularly unfavourable cases can lead to several valuable developers leaving the team at once, also losing expertise and knowledge. The meeting attendance rate is occasionally not as high as desired, because the contributors have other commitments with higher priority and therefore extra effort must sometimes be made to keep everyone up to date. Furthermore, collaboration across teams is limited and heavily relies on the coordinators, which is an additional burden for them.

### 4.1.3. Development Process

All four teams adhere to the same development process, which is explained by an interviewee:

*"So the theoretical process is that someone will code up the solution for it and then open up a pull request. This pull request will get reviewed by two independent senior developers. And these people will then merge it. And it will, as part of being merged, be part of the next release" (E4, L. 226-229).*

Contributors can choose which ticket they want to work on, *"they assign it to themselves, and they start working on it"* (E1, L. 139). After the ticket has been implemented and moved to the review column, it is ready for the reviews. However, the amount of time that the review needs is difficult to estimate, because *"[...] those reviews are sometimes a bottleneck"* (E3, L. 228). Another coordinator describes this bottleneck as highly problematic, as the implementation can be delayed to such an extent that the developer responsible leaves the project before the review is done:

*"But the actual bottleneck is that we sometimes wait way too long for a ticket to be merged. Even that long that people are coming, working on a big feature and are ready to leave, but the feature is not implemented yet until we release. And there is this big, I'd say... It's hard to find a word for it, but there is this big risk, let's say risk that the ticket will become stale so that the whole feature branch is getting stale because after the person who did multiple months of really good coding would just become obsolete because the person's leaving. No one's able to get into their shoes and just continue that work" (E4, L. 266-274).*

The reason for this delay is that the second review cannot be done by every contributor. It somewhat depends on the team and on the complexity of the ticket, but often only the PO or the coordinator itself are entitled to review and merge the ticket. As this is not the only responsibility they have, it can take quite some time. At least one team tries to mitigate this problem by allowing seniors to review and merge tickets, "[...] if it's just a bug or a task that is not that big, like, it has no real business value or stuff like that [...]" (E1, L. 187-188).

There is another unknown when estimating the duration of the development for an individual ticket. Contributors do not have to indicate in advance how much they will contribute to the project in the near future. Depending on their workload due to other university courses and work, the number of hours they spend on the Catrobat varies significantly. Coordinators "[...] have little information about that, about how many hours, for instance, they will contribute in the next week, in the next two weeks, in the next three weeks" (E3, L. 275-277). Hence, the experts can only roughly estimate on how much time a ticket will need in order to get implemented and merged:

*"For really big tickets, alone the development on itself, it would take several weeks. Checking on that code again would take more weeks, and then merging it would also take some time. Overall, for big tickets, I would say maybe around one to two months, and for small tickets, one to two weeks, depending on the code, depending on how experienced the developers are" (E2, L. 201-207).*

Especially when handling time critical tickets this uncertainty makes it difficult to plan ahead and to assure that tasks are implemented in time. If a ticket is really urgent "[...] and if no one does it, I just do it myself" (E1, L. 204-205), one coordinator claims. Nevertheless, it should be mentioned that the basic process is highly accepted, and a fundamental change would not find any supporters. Even if improvements in the processes are possible and desirable, everyone would most likely agree with the statement of one coordinator that they are *"very happy to use this system how it is"* (E2, L. 226).

### Interim Conclusion

The agile development approach in place is a reasonable process model under the highly volatile circumstances of a student FOSS project. However, it poses several challenges related to time management and process planning. Without hardly any information on the availability of the contributors and many dependencies between developers and experienced reviewers the implementation of features can be unnecessarily prolonged. This undoubtedly has a particularly negative impact on the progress of the project, especially in cases where the long waiting time requires a renewed adaptation and refactoring of the code that has already been completed.

#### 4.1.4. Recruitment, On- and Offboarding

Due to the fact that experienced employees are constantly leaving the project, two coordinators highlight the recruitment and onboarding process as one of their major tasks.

##### Recruitment Process

As a matter of fact, coordinators cannot decide on their own whom and when to hire:

*"We also have a document on Confluence where we can essentially request new members and then it's up to somebody else higher up to essentially distribute new members that have joined to the teams and then the onboarding starts" (E3, L. 304-307).*

However, coordinators are kept uninformed when exactly they will receive new team members. Coordinators might also need to ask repeatedly, if they do not receive the requested members. This is especially disadvantageous, because they *"[...] need the, like, the more senior ones to help the new member. And there should be enough overlap"* (E1, L. 210-211). The admission of a new member then involves a moment of surprise for the whole team, including the coordinator:

*"Like, most times, I just got, like, I just said I need a member. And then there is a, like, welcome text in my team's chat that tells us that there's a new member. And then I have to write them and have, like, an onboarding meeting" (E1, L. 221-224).*

This recruitment process is particular difficult for the iOS Team, due to the fact that applicants *"[...] need Apple hardware to be able to join our team"* (E4, L. 129).

Because of this recruitment approach, three of the four interviewees have already been struggling with staff shortages. Two of them could even refer to critical low member counts, which had negative impact on the whole project:

*"And if many of the team members leave at the same time and we can't motivate them to take additional courses at Catrobat, then the team can shrink pretty rapidly. And we had that in last winter semester when we suddenly were only left with two or three team members, me including. And then we saw that if you have less members, you also have less people for code review. So we saw that many of the tickets that were kind of already done were stuck in code review. And we're still trying to get the time to get work on those code reviews and get those tickets merged" (E3, L. 324-331).*

### Onboarding Process

The onboarding process is well defined and structured in all teams as it is crucial for training new members and gaining independent and reliable contributors. The process contains several steps and support structures:

*"[...] when we have new team members, we ask more experienced team members to mentor them for the first couple of tickets. That's when a lot of knowledge transfer can happen." (E3, L. 364-366).*

*"We more or less have like a checklist for onboarding stuff so that every new member gets an information letter. And on that they have like check boxes that they can go through, make some small tasks, check them. And with this finished list, they more or less are finished with the onboarding" (E2, L. 278-381).*

### Offboarding Process

Contributors usually leave the project, when they have finished the number of hours they agreed on to spend at Catrobat. The coordinators need to be informed in advance by the member that they will soon stop contributing to the project. Contributors are then asked to *"focus on their tickets, not starting too many new tickets because that would be an issue if they are then just gone and they have 10 open tickets"* (E1, L: 249-250). However, it is often the case that some tickets are still in review, when they leave the project. In those cases, the coordinators hope, that *"if they develop the ticket and it gets to the code review and someone asks something about that, then they will, like, respond even if they don't need to do it anymore"* (E1, L: 251-253).

However, this does not always work out and semi-finished tickets are left behind:

## 4. Results and Evaluation

---

*"And the most critical part or important part is that they tell us what they were currently still doing. And if they could, in a meeting, explain to me or to, let's say, to a senior developer what they did for a specified ticket so that we are able to finish the work of the member that's leaving" (E2, L. 322-326).*

### Interim Conclusion

Acquiring new members in time poses a challenge for the coordinators of the teams. Until the new contributor actually starts participating, they will not know if and when they will receive the requested support. Therefore, it is almost impossible to plan ahead, and inevitable staff shortages reduce the project performance significantly. On the one hand, because the teams would need more developers to handle the tickets, on the other hand because without enough senior people the successful mentoring process cannot be realised.

New members receive support from senior contributors in order to implement their first tickets, which are tailored to developers inexperienced with the Catrobat project. In addition, they receive a checklist with tasks that allow an easy start, which makes the onboarding process well-thought-out.

Furthermore, coordinators must pay attention to the remaining hours of contributors. If someone leaves unexpectedly, the progress of semi-finished tickets, as well as expertise and knowledge might be gone with them.

### 4.1.5. Performance Measures

Performance measures are used to track the overall progress of the project as well as the achievements of the individual contributor. Before considering establishing new measures, a review of the actual situation is necessary.

### Monthly Contribution

In order to ensure a baseline contribution a threshold of a minimum number of hours per month was introduced several years ago. This threshold guarantees some planning certainty for the coordinators, and it helps the contributors *"[...] to focus, to remember that they have to do something and not just be gone for a month"* (E1, L. 341-342). However, it is not uncommon for a contributor to fall below the threshold in a certain month, due to various reasons like university exams or a high workload in another occupation. While this certainly has impact on the project like *"[...] slower performance, slower velocity than that could be"* (E2, L. 504), it is still accepted, because an occasional undershoot is not significant for determining *"[...] if someone is willing and participating in a meaningful way [...]"*



(E4, L. 407-408).

### Individual Performance and Grading

As most of the contributors participate in the Catrobat project as part of a university course, they are rewarded with a grade at the end. Grading is an intricate process, which not only involves the coordinator. However, they give their "[...] *recommendation for a grade [...]*" (E3, L. 523-524), which is based on several parameters:

*"I do this most important question for me is did they learn something and how much did they improve? I try to compare the person they were when they started, and I try to compare the person they are, when they are leaving. And I try to take everything into account. So, I try to take into account how was their skill level, could they even write a line of code like a proper line, or could they talk in a group, or could they set up meetings by themselves, or could this person be trusted with a task that is maybe not that straightforward? So it was like, and could this person at the end be trusted with a task that is not that straightforward?"* (E4, L. 511-519).

Apart from the personal development, also some further technical aspects, commitment and reliability are considered:

*"I look at what kind of tickets they've been working on, so whether it's a complicated ticket, whether it's writing tests, whether it's a simple or complicated refactoring ticket, and that's when I evaluate people's performance, and also that helps to get a good, how should I say, to develop a good baseline on where contribution should usually be, so I now know, for instance, how many tickets are appropriate for 200 hours of work in the team"* (E3, L. 447-453).

*"[...] a lot of it is how well they did with working together in the team, communicating, being like answering if you message them, being in team meetings, showing like motivation on changing stuff in the team, like there's a difference if someone is like really like working on the team, creating new tickets, asking for a technical meeting because they want to change stuff, they want to restructure something [...]"* (E1, L. 307-312).

Although the coordinators carry out a comprehensive evaluation of performance in order to obtain a well-founded assessment, they are nevertheless dependent on a large number of subjective judgements. As proven in a meta-analysis by John M Malouff and Einar B Thorsteinsson (2016, p. 5) it is hardly possible to provide a bias free subjective grading. One coordinator states that it

#### 4. Results and Evaluation

---

is “[...] pretty hard to objectively measure the contributions [...]” (E4, L. 443) based on the available information. However, when being asked if they feel that the availability of more objective measures would facilitate the grading process, all four experts disagree. This attitude suggests that the Catrobat project values individual growth over achieving objective goals. One also expresses concern that parameters such as lines of code could then be measured, which evidently provide no added value in the measurement of performance (Forsgren et al., 2018, p. 45).

Nevertheless, three of the interviewees would appreciate some kind of tool or dashboard, which would offer a condensed view of useful information and statistics:

*“[...] I already mentioned reporting, and the timesheet helps, but yeah, you only have individual entries and no option to aggregate over data, because sometimes what I wanted to do is I wanted to get an overview on how much time is invested in individual tickets, and there I had to essentially compile the information myself. I had to download this, export it as an Excel, and then try to wrangle the data a little bit, but having something like a dashboard where I can see, okay, those are the tickets that my team member is currently working on, and that’s how many time goes into each of those tickets. That’s also helpful for performance evaluation, also finding if there’s any impediment. For instance, if you see that it’s a simple ticket, and you see that they’ve already been working on for it for 40 hours, then you definitely know that something is wrong [...]” (E3, L. 463-474).*

The timesheet is used by every team member to track their time spent on working at Catrobat. Each line of the sheet contains information about the start and end time of the contribution, of which type the contribution was (e.g. meeting, development or code review), optionally a link to the corresponding ticket in the project management tool *Jira* and a description of the work done. Coordinators have to compile needed information of the timesheets manually, because there is no automatic aggregation and reporting of the timesheet content.

Another coordinator would like to use planning tools to be able to better estimate how much can be realised in the coming month, for example:

*“[...] I think it would be great to try to have like more, like do more planning in the team, like what should we do in the next month, and should we try to get it done, and also check how many like story points or hours we got to do in that month” (E1, L. 323-326).*

### Reasons for Project Success

When changing the perspective from the individual performance assessment to the project-level evaluation, also the focus shifts to different measures of success. As well as on the team member level, there are currently no real objective measures of success. The subjective perception of accomplishment is described by one expert:

*"[...] right now there's not anything we can measure our performance, but now it's just like the feeling like there's progress, the app still lives, the app is still in the app store, and yeah, that's kind of it" (E1, L. 299-301).*

Despite the lack of measurement, the Catrobat project and its applications are thriving as millions of downloads prove. According to the interviewees, the knowledge and motivation of the contributors is the key for the success of the project:

*"[...] we get really, really motivated people who are really interested and you can often see that in the first couple of meetings when people are not just sitting in the meetings and doing what's necessary and what's required from them, but then really contributing with ideas and asking often critical questions, why we're doing things the way we do them. And if we have really important, really interested team members, really important team members, we saw two things. We saw that they were dragging other people along. So they were motivating other people. And when you see that there's movement within the team that we're gaining momentum, usually people pick up on that and that helps the whole team a lot. And it also helps in terms of contribution because those people often get a lot of things done" (E3, L. 409-419).*

Particularly the proximity to the university allows the project to access many young and talented people, who are eager to prove their acquired skills in a practical environment and to contribute to an interesting project.

Furthermore, the upper management, which consists out of PhD students and the project leading university professor engage contributors with their *"[...] visions for the next middle to long-term plans for the product [...]"* (E3, L. 426-427). On the one hand, those product development ideas keep the contributors intrigued to stay at the project and on the other hand they keep the applications interesting for the end users.

### Interim Conclusion

Currently, there are hardly any objective performance measurements in place. While coordinators need to grade collegiate contributors after they are finished with their hours, they do not feel that additional measures would aid them in

the process of grading. Nevertheless, the coordinators would value tools for reporting and planning, in order to gain a better overview of the current team as well as project progress. Moreover, the time-consuming manual timesheet checks could also be replaced by automated solutions. Overall, the coordinators strongly cherish the high level of education the contributors bring with them, which is essential for the success of the whole project.

### 4.1.6. Team Member Motivation

Motivation is what drives the students to invest their time into the Catrobat project and also gets them started to contribute in the first place. It is therefore worth taking a closer look at the reasons for participation and also analysing how motivation can be increased within the project.

#### Reasons for Participating in Catrobat

The coordinators give several reasons why students decide to participate in the project. Two of the interviewees assume that the main incentive is *"[...] their grade, since they are doing it as part of the university or of their academic career, and [...] they want to get the best grade that they could possibly get"* (E2, L. 459-462). Another coordinator sees it also beneficial, that it is possible to receive ECTS for the work at Catrobat. However, they believe that it is more about the uniqueness of the course design in comparison to other lectures, that attracts people:

*"[...] three teams that do mobile stuff, they may be interested in mobile apps. That's why I joined, because we didn't have that, like, that many lectures on mobile development. So, I wanted to try it out. And, yeah, that's why I came here. But I also think some people just think, like, they're better on working, like, in that team and doing, like, working, like, at a job later on than doing multiple lectures. So, that's also something I really love at Catrobat, that I just, it's like a little bit of work experience, because for the last few years, I had multiple teams. I could learn a lot"* (E1, L. 383-390).

This quote also makes it clear that the project offers students a great opportunity to familiarise themselves with processes in their later professional life. This is because the project, the individual teams and the development process are very similar to the agile approach practised by many IT companies. Furthermore, the project provides a platform to experiment with new technologies and contributors are granted a lot of freedom to realise their own ideas, which highly motivates curious students to participate:

*"That's something, for instance, that I see from former team members that are now working in the Godot team, where they're working on a pipeline, on DevOps*

*things, getting everything to work. That's what they're really interested in and that's also resulting in a high contribution" (E3, L. 554-557).*

The socialisation aspect should also not be neglected, because "[...] most people just join because they hear that it's a great project from other people" (E1, L. 380-381). In addition, the project also offers "[...] the chance to get to meet new people, which could be like-minded" (E4, L. 555-556).

### **Incentives for Motivation Increase**

Most of the incentives also emphasise the social interaction between the contributors. Three of the four experts mentioned the annual Christmas Party, which brings together all the contributors of all teams, which does not happen usually. Another incentive are the team days, *"where people really meet and then work together on things, that's motivating"* (E3, L. 574-575). Furthermore, two coordinators mentioned workshops, that are offered together with IT companies, which cooperate with the TU Graz. These workshops enable students to get in touch with their possible future employers and they can see which professional fields are open to them after their studies:

*"[...] we are doing some events or some workshops to motivate those team members, which they can participate at, so they can get even more knowledge, not only code in the project, but get more knowledge that might be helpful for their later work life in the industry [...]" (E3, L. 464-467).*

### **Influence of Motivation on the Project**

When being asked, if high contributor motivation is important for the project, all coordinators agree. Apart from the fact, that with more motivation tickets are developed faster, individual motivation has also a great impact on the team as a whole:

*"Obviously, if you're working on your tickets faster, if you're doing code reviews, that helps the team as a whole, objectively, but also if you're motivated and if you're engaging, then you, as mentioned before, you tend to carry other people along and that helps also other people's motivation" (E3, L. 563-566).*

### **Interim Conclusion**

The Catrobat project offers a welcomed variety compared to the usual university lectures. Many students join the project, because they value the way the project works and its strong resemblance to the private sector. They have far more

decision-making freedom than in lectures and can contribute and implement their own ideas. Furthermore, the project functions as socialising hub, bringing together like-minded people and facilitating interpersonal exchange, be it at team days, the Christmas party or at workshops.

### 4.1.7. Knowledge Management

Since processes and tools are not the only decisive factors for a competitive advantage, but rather employees and their knowledge are the key to sustainable business success, knowledge management is a particularly important aspect of the management perspectives (Omotayo, 2015, p. 16).

For the purpose of documentation, Confluence, which is a team workplace for collaboration, is used as an archive for technical knowledge and how-to guides. If a coordinator sees it as necessary, they *"ask people to write documentation on [the team's] Confluence page to have a little bit of knowledge transfer, especially when it comes to a little bit more complicated things"* (E3, L. 354-356). Another coordinator questions the usefulness of documentation on Confluence, because they rate it as quite likely, *"that no one ever, like, checks again or also checks if it's still needed, if it's still correct"* (E1, L. 269-270). They prefer to use a learning by doing approach and rely on the mentoring system, which is used by all four teams interviewed:

*"And when a new member joins our team, we typically assign one of the seniors as a mentor for a junior developer for a new member. And they most likely then work at the simpler tickets together so that the new member gets kind of a taste how the project works, how to work on tickets and so on and so forth. So we basically have a mentor and that mentor works with the junior so that this knowledge gets spread very well"* (E2, L. 335-340).

Especially because it is common for contributors to only stay for a short amount of time, a constant knowledge transfer is indispensable for a successful continuation of the project. One coordinator describes the ideal contributor replacement process for them, which is not the norm in their team, however:

*"So best case scenario is someone is leaving and two months, three months before that, someone new is joining. Unfortunately, I've never had that opportunity yet. So most of the time it is knowledge gets picked up and put together by someone who is new. And my part in the knowledge transfer process is that I try to retain as much of that as easily accessible as possible"* (E4, L. 388-392).

In the possible event that a successor is not found quickly enough, there is an unfortunate loss of knowledge:

*"And then when there's someone new, I try to transfer it the best way I can, which*

*is honestly also a huge loss, but it's better than nothing"* (E4, L. 397-398).

### **Interim Conclusion**

Knowledge transfer in the Catrobat project is characterised on the one hand by the mentor system, which brings senior developers together with new contributors and relies on the verbal transfer of knowledge. On the other hand, written documentation is also produced and stored on Confluence. However, not all coordinators are convinced of the benefits, as the documentation is incomplete, not well organized and is not reliably updated. Recurring shortages in the process of filling vacancies lead to insufficient knowledge transfers and thus to a loss of knowledge. Overall, the current processes cannot guarantee a sustainable knowledge management.

#### **4.1.8. Conclusion - Deficiencies in the Current Processes**

By and large, the reports from the coordinators draw a picture of functioning processes that make it possible to work successfully on the applications of the Catrobat project. Nevertheless, some processes could be identified that would require optimisation. Even though these were already touched upon in the interview analysis, they are again summarised here.

#### **Impediments in the Development Process**

It was mentioned several times, that tickets could not be finished because of the missing final review done by the PO or the coordinator. In many teams only one person has the entitlements to finally approve and merge the feature implementation. However, those people are often involved in many crucial decisions and have other responsibilities outside of the Catrobat project too. In the worst case, it takes even that long, that the contributor developing the feature, already leaves before merging. Even worse, if the second review comes to the conclusion that some adjustments are still necessary, the whole development might be discarded.

Similar, if a contributor leaves because they are finished with their agreed hours and there are semi-finished implementations left behind, it is likely that the progress will be rejected. Although it is improbable, that the worst-case always happens, waiting months before getting the final approval is highly unfavourable:

*"And we are waiting for the actual code review. We are waiting way too much time. And I'm talking about months here"* (E4, L. 277-278).

#### 4. Results and Evaluation

---

This makes it impossible for the coordinators to estimate how long it is going to take until a ticket is merged. This leads to problems, in particular if there are tickets that depend on each other or if one ticket is blocked by another and cannot be started until the other one is finished.

##### **High Contributor Turnover**

Most of the contributors start working for Catrobat as their bachelor's thesis. According to the ECTS credits awarded for the thesis, this means 210 hours of work at Catrobat. As it takes quite some time to familiarize with the project, 210 hours are not enough to implement more than a few tickets. If they leave the project right after the thesis their full potential cannot be realised. There are more options to contribute during the master studies, but not all contributors decide to stay longer or even do a master's program.

Hence, two major problems arise: There are particular difficulties with knowledge transfer and the size of the teams can suddenly shrink tremendously. As a result, the development process can come to a standstill and, above all, the already subpar review process is no longer carried out as required. In addition, without any seniors left in a project, the knowledge transfer to and the mentoring for the new junior developers is considerably more difficult to almost impossible.

##### **Lack of Information and Knowledge Loss**

As already mentioned, the high fluctuation of the members can lead to a massive knowledge loss. Most of the knowledge is transferred orally during the onboarding and mentoring processes. This harbours the risk of knowledge being lost forever. Documentation is done sporadically on various Confluence pages, however due to a lack of structure, this is not a very promising endeavour.

Moreover, information in general is often lost or does not reach its intended recipient. This affects the coordinators as they do not know how much team members plan to contribute in the next one or two weeks and overall, they have very little information on how long the contributors are going to stay at the project. If they want to know anything about the plans of the contributors, coordinators have to ask them personally in meetings. Besides, the amount of work done in the past few weeks must also be checked manually via the time sheets of the contributors.

##### **Inefficient Manual Checks**

Several processes rely on manual actions, which make it tedious and laborious to gain the information needed. This applies to time sheet checks as well as to the pull request checks for grading. Coordinators must compile the information on which tickets a student worked and how much time they needed by themselves.



There is no tool or similar to automatically obtain information on a contributor's performance.

### **Difficulties in Collaboration**

The issue with missing information concerns also the collaboration in the project. Bigger teams struggle with finding time slots for meetings in order to gather all team members. Understandably, contributors assign a higher priority to university exams or their paid work, compared to the team meetings at Catrobat. In addition, work is mostly done online, and face-to-face meetings rarely take place. It is therefore often a challenge to find someone when help is needed.

Furthermore, there are often communication problems when teams work together. Despite the fact that teams work on similar topics or even rely on the output of another team, they tend to operate as silo units. If there is the need to talk to each other, the communication is done via the coordinators of the teams. This often involves four people for discussing a simple inquiry and delays receiving a response and also makes communication prone to errors. It seems obvious that direct communication between team members from different teams could improve the process.

## **4.2. Measurement and Visualisation of the KPIs**

In order to utilise the KPIs, it is essential that the frequently updated values are accessible in a comprehensive visual format. Only then it is possible to act on KPIs that are running in an unsuitable direction (Parmenter, 2010, p. 96). The following chapters consider tools and representations, that provide KPI representations and their usefulness in the Catrobat environment.

In addition, an initial outlook on the processing of the actual Catrobat data is given. The importance of the KPI values and how they can be integrated into the daily work at Catrobat is also highlighted.

### **4.2.1. Tools for HRM and KPIs**

When searching for tools and applications that facilitate the digital processes of HRM and KPI tracking a variety of proprietary solutions can be found. Some of them are even open-source, however, a purely free source software could not be found. The sales policy usually uses free trials and a free stripped-down starter version. In order to utilise the software's full potential, a monthly subscription has to be acquired. Due to a rather limited budget at Catrobat, a pay-as-you-go plan is not feasible. Especially, as all of the considered providers charge per-employee and Catrobat relies on a fairly large contributor base.

It is also noteworthy that these HRM solutions are designed to provide a complete system for everything HRM related. This in turn makes it difficult to use only the individual parts of the system that are required. As other systems such as Jira and Confluence are already in use, the introduction of another extensive software program would unnecessarily increase the organisational effort and is not very expedient. Parmenter (2010, p. 58) also recommends waiting at least six months before introducing a new system for KPI tracking and using simple visualisation methods until then. This means that less time is lost during the implementation of the KPIs and the effort of selecting a system can be postponed until later. In other words, for the initial implementation, a tool that is as simple as possible and offers a quick update of the KPIs and a clear presentation of the results is preferable.

Therefore, the only tool used for the realisation of the findings of this thesis is PowerMetrics<sup>1</sup> from Klipfolio<sup>2</sup>. PowerMetrics is as well a proprietary application, but it provides an unlimited free version, that offers everything needed for simple data processing and visualisation of the KPIs. Klipfolio describes its tool as a *self-serve metrics platform and complementary analytics solution*. It enables the creation of metrics via automated connections to various data sources and displaying the metrics in well-arranged dashboards. While this is not the complete scope of features provided, these are needed for fulfilling the requirements of Catrobat. Providing new data is easy and is done quickly, which is essential for daily KPI updates. To summarise, PowerMetrics stands out from other options in terms of cost, clarity and simplicity.

### 4.2.2. Processing the Data

The data needed for the representation of the KPIs is taken from several different sources. The exports from those systems are usually .csv files, which stands for comma-separated values. PowerMetrics is capable of processing .csv files, however small adaption may be needed before a file can be analysed by PowerMetrics. This data processing can easily be automated for the purpose of exports. This means that a new data value can be added to the KPIs daily. For example, via connecting PowerMetrics with a cloud storage instance and keeping the data on the storage up-to-date. The data source and the extent to which it must be prepared before visualising it in PowerMetrics is specified below. The examples depicted below are based on the 5<sup>th</sup> of September, if data was available. In some cases, older data had to be used. This is then noted in the corresponding explanatory text.

It should be noted that although this is real data from the Catrobat teams, it was taken for a one-off snapshot and serves primarily as an example to illustrate the possibilities for a continuous KPI representation.

---

<sup>1</sup><https://www.klipfolio.com/PowerMetrics> visited on 20. September 2024

<sup>2</sup><https://www.klipfolio.com/> visited on 20. September 2024

### **Weekly meeting attendance (WMA)**

The weekly meeting attendance is manually tracked by the scrum master or the coordinator while writing the meeting notes. For displaying this KPI only the date of the meeting and the percentage of attending team members is needed, which can be added to the data source directly by the scrum master.

### **Story points implemented (SPI) and Average time tickets stay in the implementation process phases (TTIP)**

The data regarding the development progress of the applications is tracked via the agile project management tool Jira. Jira already offers some reports directly available on its graphical user interface (GUI). Yet, it does not allow to adapt existing reports according to the needs of the defined KPI without acquiring a reporting add on. An export is necessary anyway in order to be able to display the data collectively in the dashboard.

It is possible to download data via the GUI, but also to establish a connection to the Jira application programming interface (API). Scheibelmasser (2024) used this API in his master's thesis for a reporting solution for Catrobat that collects data from different sources. This means that with minor adjustments to the tool, the data required for the KPIs can be made available.

Each data record represents a Jira ticket, including timestamps for each change made to this ticket and the amount of story points assigned. These changes include when a ticket was moved to another column on the board, which is all the data needed to display both KPIs. However, some tickets do not have story points assigned, in those cases, they are assigned one story point.

### **Available human resources per team (AHRT)**

The available human resources per team were tracked indirectly via the timesheet tool on Jira. For every team member the total hours agreed on and their team affiliation were stored. While the Jira and Confluence instances were migrated to the Atlassian Cloud some time ago, the timesheet was continued on the old instance as this plugin was no longer available. However, due to security concerns, access to the plugin had to be blocked at the beginning of July and the working times are recorded locally by each member now. This means that no current timesheet data can be obtained at the moment. Although, a replacement is being sought.

For this reason, data from June and July is used for the visualisation. The remaining available hours per employee are automatically adjusted daily. If the total number of team members changes, this must be modified for the next export. One disadvantage of the former implementation is the lack of differentiation if the master's thesis is done via Catrobat. Doing the master's thesis at Catrobat requires spending nine hundred hours, which are added

to the overall available hours, however these ours are spent on the individual thesis, not on team progress. This means the value without master's thesis hours must be calculated every time a new report is created. This is feasible as every time sheet entry is categorised, but a separate value in the new implementation is recommended. Another downside is that there is no distinction between members of the IDE and the Stage team. Internally all team members belong to the Catroid team. Therefore, the visualisation is done for both teams together.

### **Average rating since last release (RSR)**

The mobile applications are available via the Android and iOS app stores. Both app stores provide access to daily ratings, which can be exported. The stores also generate automated reports according to their statistics. Nevertheless, it is favourable to collect and depict all KPIs on a centralised dashboard.

### **4.2.3. Visualising the Data**

The functionalities of PowerMetrics are used to present the current KPI development graphically to all Catrobat members. Illustrative examples of how the individual KPIs can be visualised are shown here. KPIs are represented on the team level, therefore the examples are based on the data of a single team, namely the IDE team.

The time period for data to show can be customised. For example, it can be set to show the *last 30 days*, which is suitable for most of the KPIs. Another feature is to compare the data with the previous period. The time span of the previous period depends on the selection of the current period. If the current period is set to 30 days, then the previous period shows the 30 days before the current period.

Another crucial functionality is to set goals. These goals are used as KPI thresholds. According to literature, a lower and a higher threshold create a desired KPI range. The idea is that the lower target can be reached with medium effort, which creates already a sense of achievement, but there is still incentive to reach the higher target. Switching between both KPI goals is possible, however the lower goal is shown by default until reached.

### **Weekly meeting attendance (WMA)**

Figure 4.2 displays the weekly meeting attendance of the IDE team between beginning of August and beginning of September. The comparison to the dotted attendance rate of the previous period, shows that the rate in August was surprisingly low. Although the lowest value in August might be attributed to the holiday period, such developments at other times of the year should be subject to analysis. The lower KPI target is set to 70% attendance, which is

manageable to achieve, if the attendance rate of the previous period can be kept constantly.

### **Story points implemented/merged (SPI)**

A comparison between the story points implemented (were moved to the column QA review) and merged for the period between 6<sup>th</sup> of August and 5<sup>th</sup> of September is shown in figure 4.3. The graph separates the months but shows the total for the whole period in the last two bars. There is a considerable difference here, while 17 story points have been fully implemented, only six have been merged. Due to the comparison, no target can be displayed in this view. A lower target of 20 and an upper target of 24 were set for separate visualisation. These targets are based on the previous months, in which 20 or slightly fewer points were realised.

### **Available human resources per team (AHRT)**

According to figure 4.4 the IDE and Stage team together have around 1000 available hours. The last drop is a simulation what happens when the available hours of one of the 14 active team members fall below the 80 hours threshold. Although, the overall hours are reduced by five, the KPI loses more than 100 hours. The KPI threshold is set to 300, below this value it is critical to acquire a new member.

### **Average rating since last release (RSR)**

The rating presented in figure 4.5 is the rating of the Android Pocket Code app, which is the main Catrobat application. Basically, the last release date is used as the start date. However, the last release, which was carried out on the 2<sup>nd</sup> of July for Pocket Paint, was suspended for Pocket Code. Nevertheless, this date is selected here as the start date. Despite the fact that the rating improved compared to the previous period, the app still needs a new release with some improvements to reach the lower KPI goal of 4.0.

The Pocket Paint application receives better ratings (see figure 4.6), therefore the KPI goal was also set at a higher value of 4.4. This rating is almost reached and a second, higher goal of 4.6 is recommended to be used for additional motivation.

### **Average time tickets stay in the implementation process phases (TTIP)**

A second export was carried out for this KPI at the beginning of October, so figure 4.7 can be used to compare the status at the beginning of September (left) with the beginning of October (right). As little has changed this month, the two statistics are quite similar. There are many different display options for

this graph. The individual phases can also be displayed separately, for example to compare them over a longer period of time. It would then also be possible to set KPI targets that cannot be displayed in this view. Even if the values are updated daily, it makes more sense to use a value from the previous period for the comparison. The values change little within a day and due to the fact that four bars need to be displayed per day, the space on the graph is limited.

### 4.2.4. Dashboard Representation

Without an appropriate representation of the KPIs the benefits are almost zero. A dashboard is recommended to represent the KPIs on a single page, which offers a concise overview of all important project developments. Complying with Parmenter's (2010, pp. 155-169) design recommendations, a dashboard view in PowerMetrics was created.

As already noted for the individual KPIs, the data was exported on 6<sup>th</sup> September. The dashboard therefore also shows the preceding 30 days up to that date. Additional filters have been applied to the KPIs for which longer time periods are defined. A different time period is applicable to the Meeting Attendance KPI because, as mentioned, no more recent data is available in this instance. The example dashboard (see figure 4.8) shows the KPIs for the Catrobat IDE team. It should be emphasised that all important KPI targets can be seen at a glance and actions can be set according to the target development. In the actual dashboard, the size of the individual KPI graphs can also be varied to obtain the best view. Due to the limited space on a page of the thesis, the individual statistics had to be reduced in size.

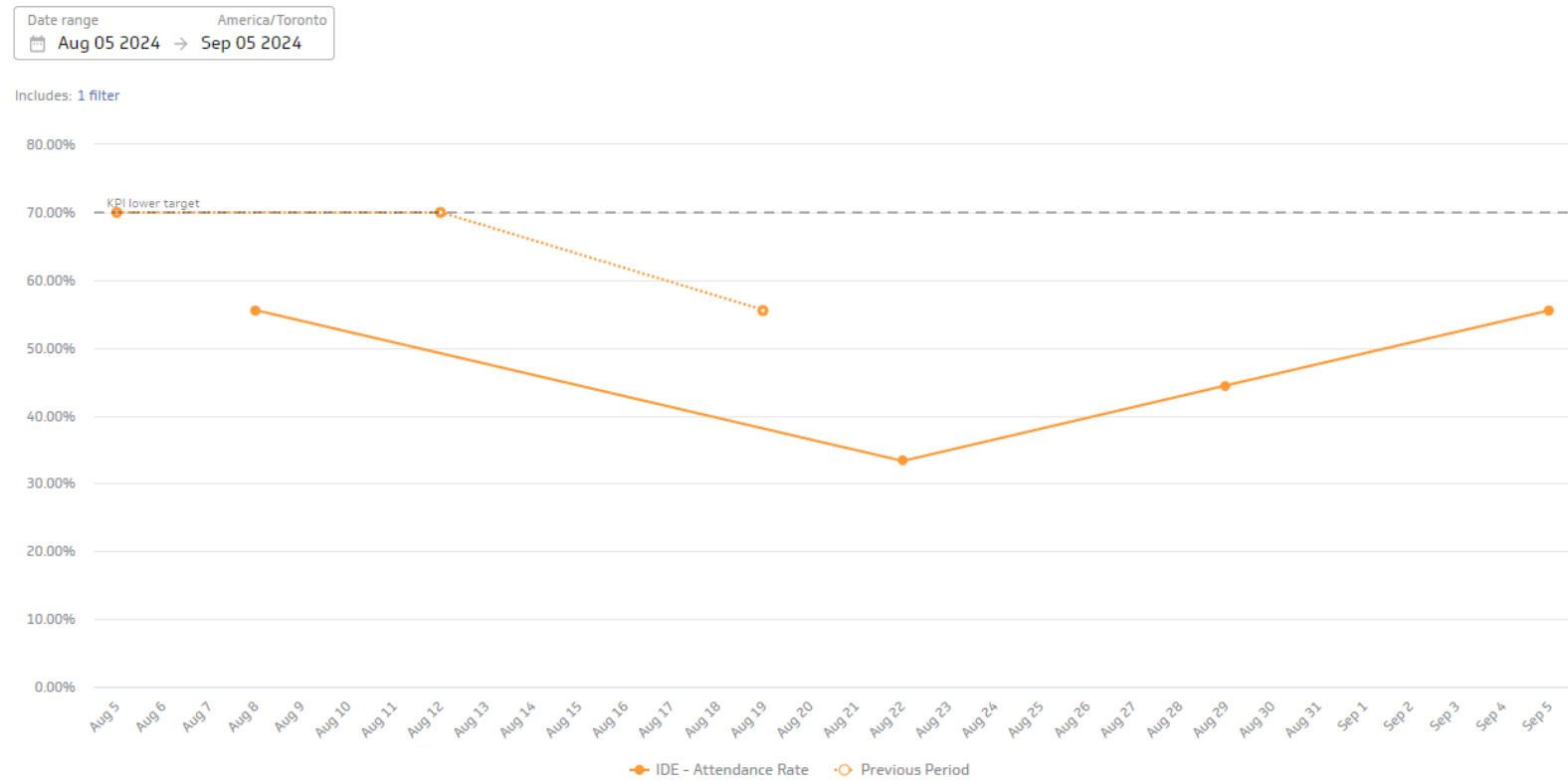


Figure 4.2.: Weekly meeting attendance rate - IDE team

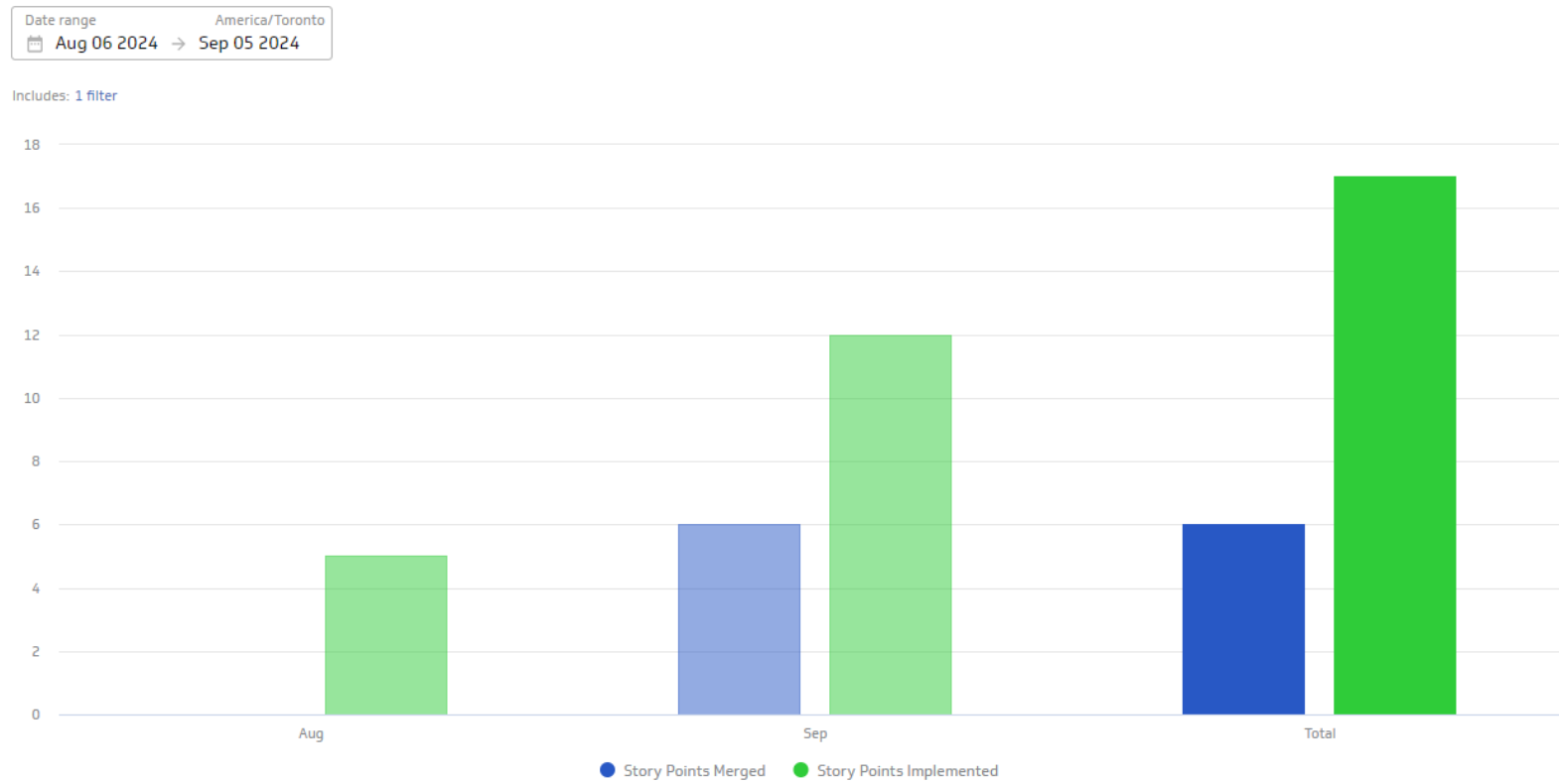


Figure 4.3.: Story Points Implemented vs. Merged - IDE team



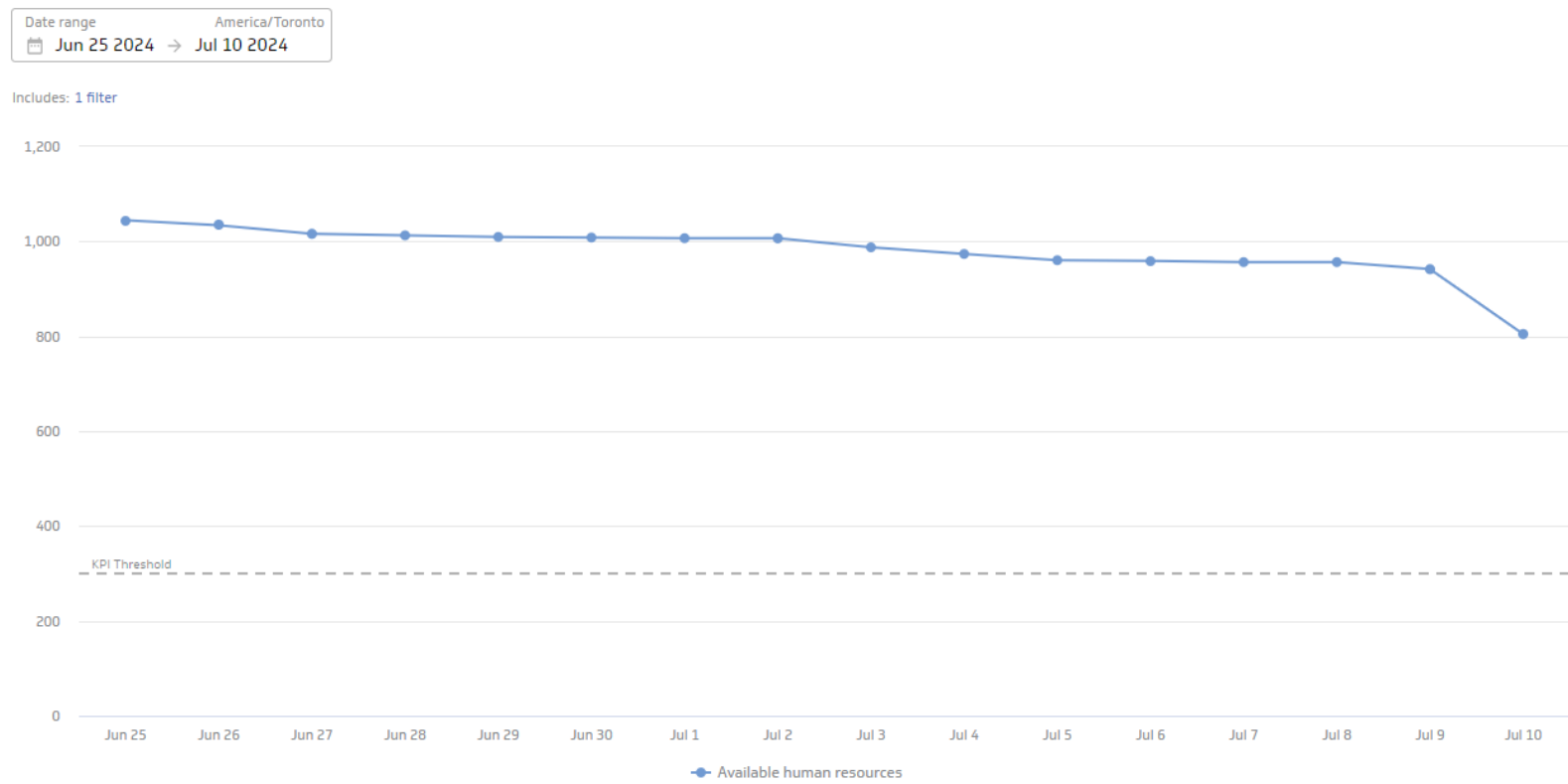


Figure 4.4.: Available human resources - IDE team

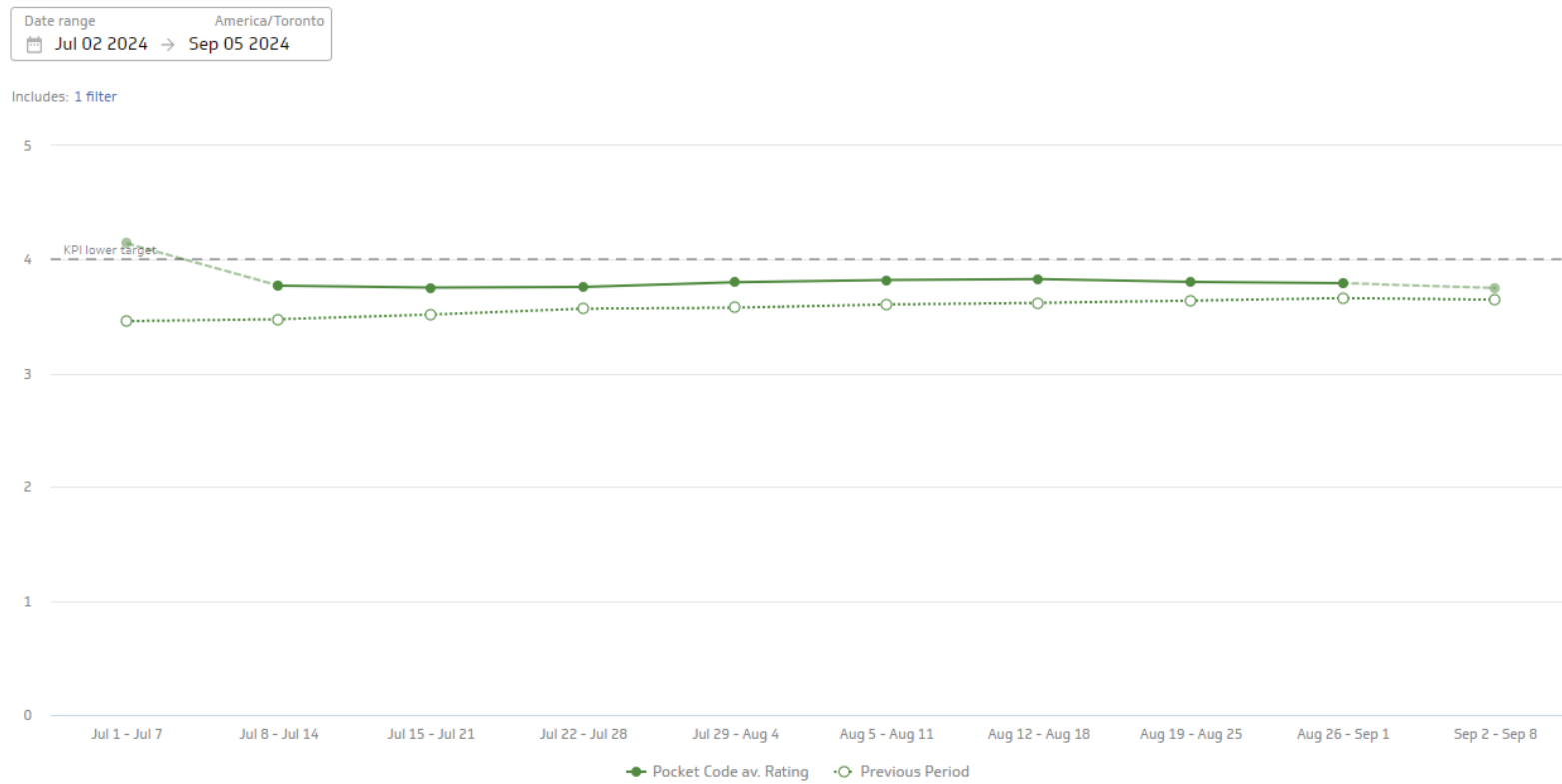


Figure 4.5.: Average rating since last release - Pocket Code

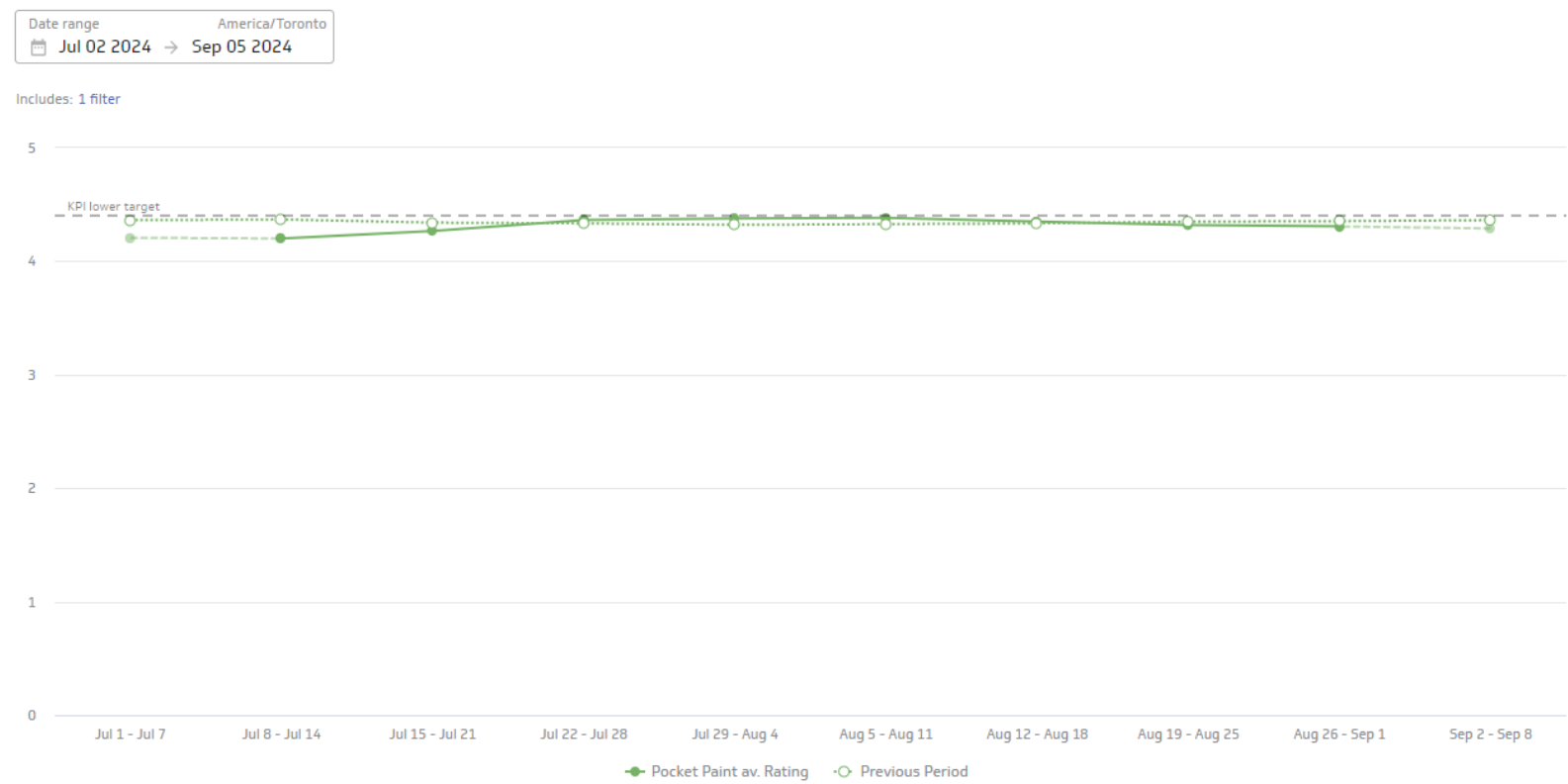


Figure 4.6.: Average rating since last release - Pocket Paint

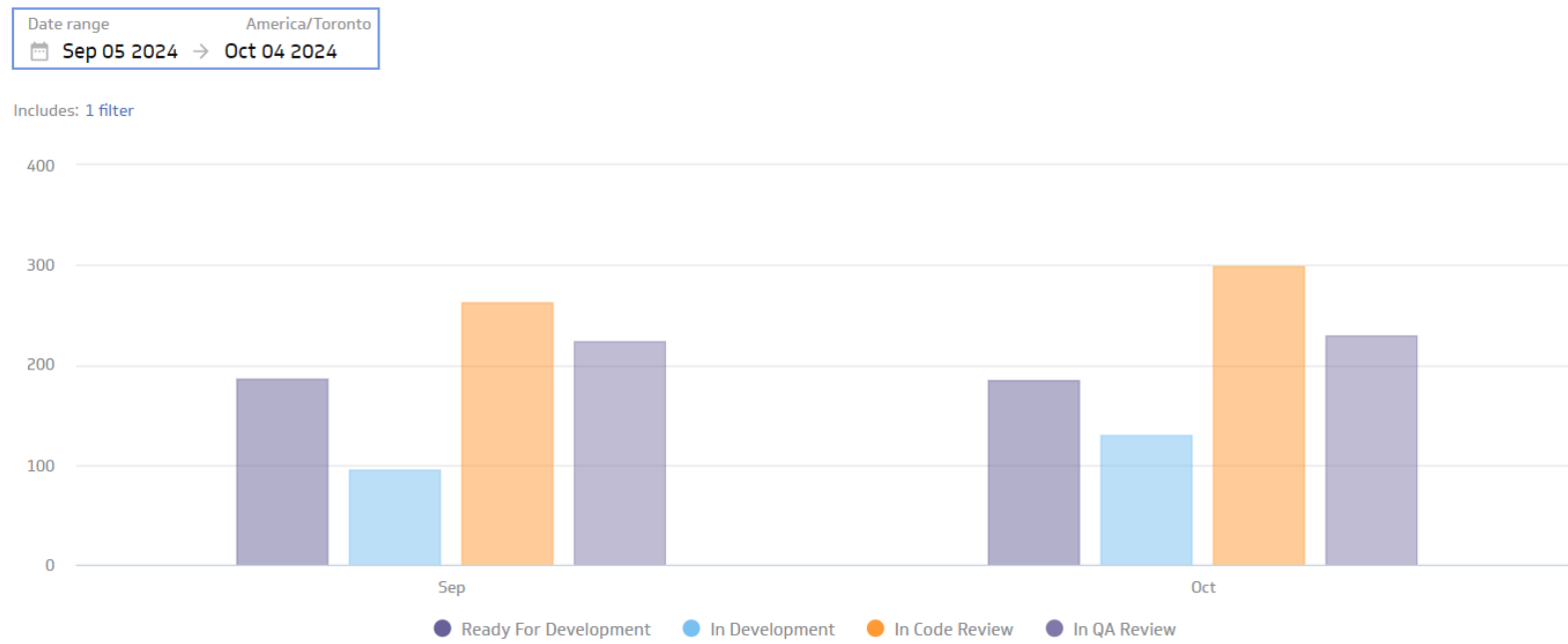


Figure 4.7.: Average time tickets stay in the implementation process phases

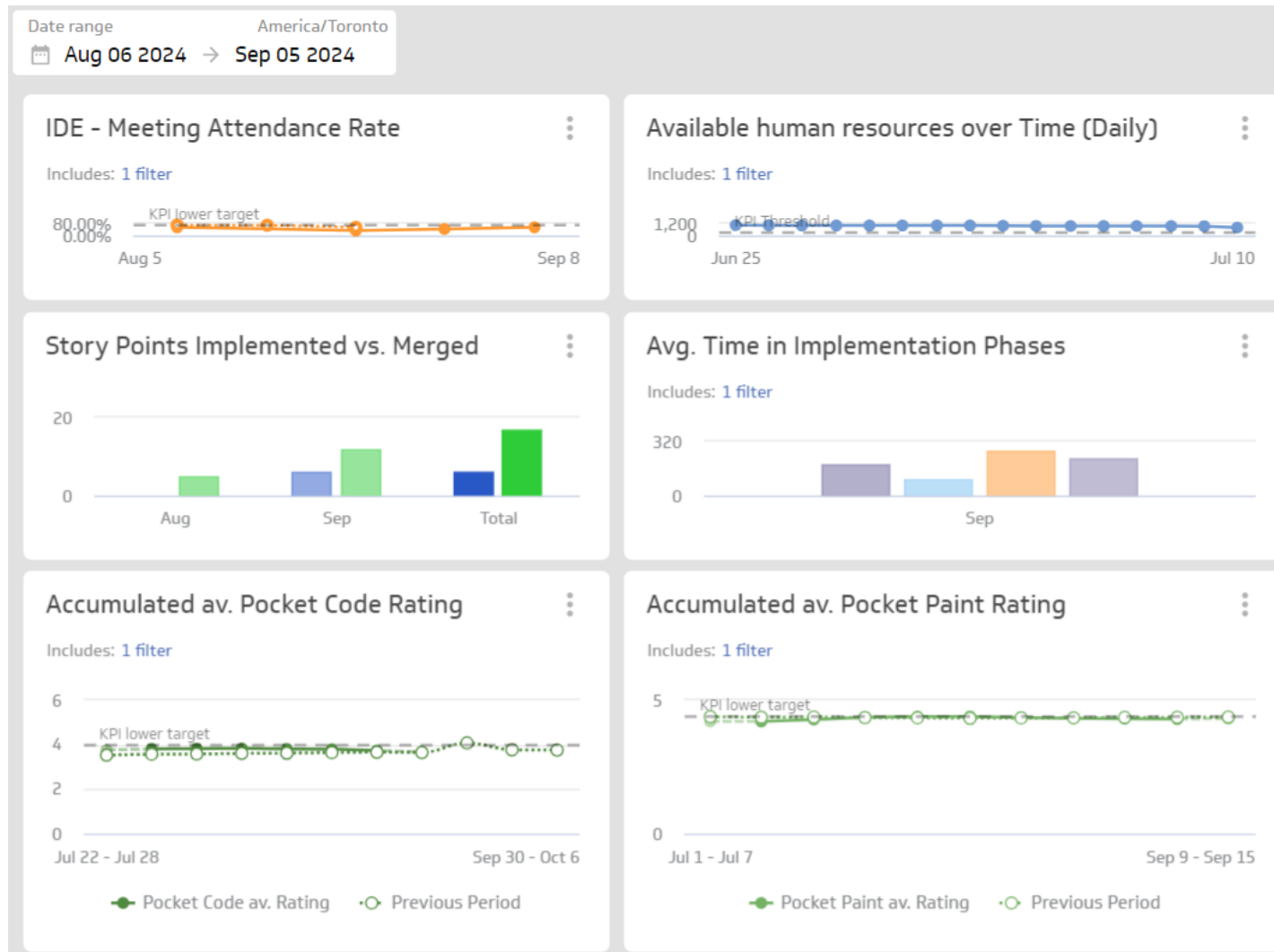


Figure 4.8.: KPI Dashboard - IDE Team

### 4.2.5. Analysis of the KPI Values and Benefits of the KPIs

Gathering the right insights from the KPIs is essential, but also involves some difficulties. Furthermore, it is particularly important to emphasise that this is a display of continuously changing data. The response to the data must be prompt and targeted. However, the overall picture of the data development must always be kept in mind so that conclusions are not drawn, and measures are not taken prematurely. Short-term changes or a punctual decline compared to values from previous periods are neither avoidable nor do they signal the need for action. In the worst-case scenario, overhasty intervention can even lead to a reduction in performance. On the contrary, it is necessary to recognise the patterns in the entirety of the data and to react on them in order to achieve corresponding improvements in performance (Kerzner, 2013, pp. 154-155).

Therefore, it is not reasonable to act immediately on the snapshot KPI data available now. Nevertheless, it is worth taking a look at the KPI values and considering what poorly performing KPIs mean for project progress. Especially in the event that the measured values turn out to be a long-term trend. The subsequent chapter (4.3.1) focuses on the optimisation of HRM processes, which have a direct and significant influence on the KPIs.

#### **Weekly meeting attendance (WMA)**

A high weekly meeting attendance rate is crucial in order to facilitate collaboration and knowledge sharing. Keeping the attendance rate constantly on a high level, is expected to accelerate the development process, as well as the acquisition of skills for the junior developers. Conversely, more experienced members increase the robustness of the project against high contributor turnover.

The interviews revealed that the coordinators are satisfied by the meeting attendance rate and estimate it between 66 and 80 per cent. Interestingly, the actual rate measured is significantly lower (see figure 4.2). The question arises as to whether this is related to the summer holidays. However, the coordinator of this team emphasised that participation rates are particularly high during the lecture-free periods. If the subjective perception of the coordinators actually deviates significantly from the actual values, this KPI can be used to counteract this discrepancy. The KPI asks the coordinators to actively contact their team members to encourage them to attend the meeting.

#### **Story points implemented/merged (SPI)**

This KPI acts as the decisive key figure to display the velocity of the project progress. It is also intended to give the coordinators a certain degree of planning security. Deviating values can also be an early indication that the contributors will not be able to fulfil their agreed monthly hours. In this case, the KPI enables the coordinators to intervene at an early stage so that external influences on

the participants can be taken into account. The KPI is able to indicate when the participants are doing constructive work, are involved in the project and are motivated.

The value of 17 implemented story points for August is slightly below the average of the months before. However, it is reasonable as the holiday season might decrease the amount of participation. The big difference to the amount of merged story points indicates the actual existence of the bottleneck in the QA process, which was mentioned in the interviews. In the long term, both values should converge so that the QA review is completed as soon as possible, and the implemented functionalities are available in the app in a timely manner.

### **Available human resources per team (AHRT)**

The lengthy recruitment process for new contributors and the associated lack of human resources was identified as a shortcoming on the basis of the interviews. This KPI can be used to check on a daily basis whether new team members are needed so that they can be recruited in time. The current example (see figure 4.4) shows the needs of the IDE and Stage team. One disadvantage is that there is no differentiation between the teams and the shortage cannot be localised precisely.

There is a high probability that readjustment is needed for this KPI, as it is difficult to estimate the number of hours at which the KPI limit should be set. The current limit is set at 300 hours, which should provide enough time to recruit new members, considering the duration of the recruiting process and the number of hours that are spent on average each month.

By observing this KPI, productivity losses, planning difficulties and loss of knowledge can be minimised.

### **Average rating since last release (RSR)**

Customer satisfaction is an important goal that the developers are working towards when implementing new features and immediate feedback is provided by the rating functionality of the app stores.

The KPIs depict a difference of around 0.4 rating points between the two Android apps. Contributors are given the opportunity to improve the rating by working on bug tickets and analysing crash reports. Furthermore, the rating difference ensures healthy competition between the implementation teams. In order to derive greater benefit from this KPI, the release frequency must be increased, which has been a long-standing goal of the project anyway.

### **Average time tickets stay in the implementation process phases (TTIP)**

Identifying and removing bottlenecks in the development process is a crucial task for performance improvement. It is easily detectable with this KPI where

a congestion of tickets happens and at which implementation step additional resources are needed. Coordinators need to react on the data presented and instruct team members to concentrate on the areas with accumulated tickets. Further actions might include the active assignment of long existing tickets to developers and frequent clearing out of the development board.

Comparing the provided data, it can be seen that the October values are slightly higher, as the tickets have been in the corresponding columns for one month longer. It is also possible that tickets that have only been in those phases for a short time have been moved on, so that tickets that have been in those phases for a longer time now have a stronger weighting. Only the value for *Ready for Development* has decreased minimally, which means that tickets have probably been moved to another phase or new tickets have been added from the backlog.

Generally, it is positive that the value for the development period is low. However, the discrepancy with the duration of the two review phases is recognisable here. Both phases take around twice as long as the development phase. It would therefore be better to put more emphasis on dividing the contributor resources between the phases. The fact that the code review column has a particularly high value is also due to many tickets that have not been updated since 2023. As an experiment, all those tickets for the October value were disregarded. In this case, the average duration would be reduced from around 263 to 98 days. Stale tickets were also found in the other phases, so a clean-up of the board is strongly recommended.

### 4.3. Recommendations for Management Practices

The interplay of the interviews conducted and the KPIs derived from them opens up the possibility of actively eliminating deficits that are currently present in the project processes. Those deficits could be tracked down by analysing the answers of the coordinators and the KPIs facilitate an objective measurement of the effectiveness of measures taken.

This chapter contains recommendations for management practices, which are capable of mitigating the process deficiencies, that are explained in detail in chapter 4.1.8, and a reference to the corresponding KPI, which will be used to evaluate the impact of the implementation. The following recommendations have been formulated in response to the identified shortcomings, employing scientifically proven strategies that address the underlying causes.

However, the success of them is highly dependent on the method and quality of execution and none of them is a guarantee for success. Only by repeatedly evaluating the implementation and effects can the practices be aligned with the specifics of the project and lead to improvements in the processes.



### 4.3.1. The HRM perspective

As Catrobat is heavily reliant on its skilled contributors, its human resources are the by far most important resource in the organisation. Therefore, HRM is the centre of gravity for organisational decisions and the recommendations for change are focussed on the area of HR.

An additional challenge is the rapidly changing team composition, which makes the implementation of comprehensive HRM practices considerably more difficult. For this reason, the recommendations are primarily aimed at smaller adjustments that can be implemented in a foreseeable period. Furthermore, not all recommendations should be worked on at the same time, but the focus should initially be placed on two to three suggestions. This is why they are also prioritised.

Apart from implementing concrete practices, there are other strategies, that facilitate productivity. By clearly defining the critical success factors of Catrobat, contributors are aware about the goals and values of the organisation. If enough emphasis is put on conveying the CSFs to every contributor, they will be able to orientate their work according to them. This will increase cooperation and motivation and in further consequence performance (Huzooree and Ramdoo, 2015, p. 11). Another factor for motivation increase is the appreciation for the implementations done by the contributors. Representing this valuation via customer feedback as a KPI aims at this principle (Tiwari and Saxena, 2012, p. 673).

The recommended HRM practices are:

1. **Facilitating cross team collaboration**
2. **Increasing contributor retention**
3. **Introducing an offboarding process**
4. **Actively distributing contributor resources**
5. **Automating contributor assessment**
6. **Focusing on meeting attendance and knowledge sharing**
7. **Recruiting contributors with foresight**

#### **Facilitating cross team collaboration**

The coordinators mentioned several times during the interviews, documented in chapter 4.1.2, that communication between teams is kept to a minimum and if collaboration is needed, all communication is done via the coordinators. There are various downsides of this approach. Coordinators waste their limited time to act as a dispensable intermediary and development is slowed down because of bureaucratic communication approaches and the lack of responsible persons.

Contributors are very capable of communicating directly with members from other teams, as this works smoothly within the team. The communication platforms have already been set up and Slack and Confluence are used regularly.

#### 4. Results and Evaluation

---

Mathis and Jackson (2008, pp. 169-170) recommend professional training of team members on effective collaboration. While this is certainly useful, as a start it is probably sufficient to extend the onboarding process of new members by a little. When contributors understand which team is responsible for the tickets they have to implement and how that team can be contacted, a direct outreach will come naturally. Assigning each newcomer the task of attending a single team meeting of another team and opening up the other Slack team channels are suggestions that are easy to implement. Furthermore, it must be emphasised to the coordinators that communication must happen directly between contributors.

The mere existence of KPIs increases the willingness to collaborate, as the set goals can only be achieved through co-operation. Furthermore, a direct impact should be recognisable in the two KPIs that measure progress on the Jira boards.

##### **Increasing contributor retention**

In order to decrease the high contributor turnover, incentives for contributor retention must be introduced. A reasonable approach is trying to keep the contributors who join Catrobat during their bachelor's degree. When they are almost done with their hours, they need to be actively informed about the options for continuing their participation in the master's programme. It is also recommended to focus more on socialising measures, such as workshops at companies, which make participation in Catrobat lucrative even independently of university courses. A lower contributor turnover means that there are more seniors in the project, the time required for new onboardings is reduced and positions such as coordinator and scrum master can be offered as an incentive for spending more time at the project.

This solves the shortage of contributors who are qualified to fulfil these roles and students who have experience in these roles have an advantage when looking for a job later on. The success of the measures can be easily determined by the available human resources KPI.

##### **Introducing an offboarding process**

Similar to the very well-functioning onboarding process, the offboarding process should also be better structured. Above all, to prevent people from leaving the project while leaving behind incomplete implementations (see chapter 4.1.4). Indicator for it is the available human resources KPI. If it drops, another team member is about to finish their hours. A clear guideline should regulate what team members have to do before leaving the project. Including also the limitation of starting new tickets as it is already in place in some teams. In addition, the practices of increasing contributor retention and increasing the

release frequency will also contribute to avoid leaving tickets unfinished.

#### **Actively distributing contributor resources**

Via the *Average time tickets stay in the implementation process phases* KPI bottlenecks can be located. That bottlenecks do exist was confirmed by the coordinators in the interviews, as described in chapter 4.1.3. The teams that also allow seniors to conduct QA reviews have already taken an initial measure to combat this problem. While it is part of Catrobat's principle that every contributor can decide they want to work on, in case of emerging bottlenecks it should be accepted that coordinators can decide where the resources are needed the most.

#### **Automating contributor assessment**

Assessing the performance of contributors is currently a highly labour-intensive task done by the coordinator (see chapter 4.1.5). The lack of a time sheet implementation that complies to all security regulations at the moment offers a chance to search for, and implement, a solution that allows greater functionality and automation. Creating reports that aggregate time sheet data and display the number of hours spent at each ticket and the hours spent per contributor per month are the minimum requirements. The time spent could also be recorded directly in Jira or one of several available add-ons for Jira could be included for automatic time tracking and reporting.

This would dramatically decrease the effort coordinators must invest every month. In addition, statistics about the actual implementation time of tickets would increase the planning accuracy. These improvements result in a productivity increase and therefore influence the two KPIs that measure progress on the Jira boards.

#### **Focusing on meeting attendance and knowledge sharing**

The weekly meeting was identified as the central hub for news exchange and knowledge sharing in chapter 4.1.2. Similar to the minimum hours per month, that must be completed by each university contributor, a minimum number of meetings that must be attended would guarantee a high attendance rate. The meetings are already defined as mandatory, although there is hardly any consequence if members do not attend regularly.

Even though, the concept of pair programming is a basic principle of Catrobat, it is rarely executed. Again, minimum hours that could be tracked via the time sheet would increase the occasions for knowledge sharing even further. As the meeting attendance is a vital metric there is a distinct KPI measuring it.

It should be noted that due to other commitments, there is often no regular date on which all team members are available. In this case, an exception would

have to be made, or the meeting could alternate between two dates, for example. However, this would also lead to an organisational overhead.

### **Recruiting contributors with foresight**

The oral transfer of knowledge reaches its limits when the recipients are missing. This happens when there is a lack of new contributors, which is sometimes the case at Catrobat, as shown in chapter 4.1.7. This occurs due to an untransparent recruitment process, whereby it is never clear how long the contributors have to wait for new members. This process must be streamlined, and the coordinators need to be integrated. Every team has different tasks and therefore requires different skills and personalities. As no one knows their team better than the coordinator, the decision to hire must not be made without their opinion. Among others, a limit must be set, which indicates the maximum time after that a team must receive a new member. Furthermore, the coordinator must be informed as early as possible when a new member has been found. Only then is it possible to coordinate the onboarding process in the best possible way.

This process is supported by the human resource KPI, that notifies the coordinator when recruiting a new member is necessary.

### **4.3.2. The Technical Perspective**

Apart from the seven HRM improvements listed, there is a single suggestion for an operational practice, that might be able to solve some issues of the current implementation process. Increasing the release frequency has already been aspired for some time. Nevertheless, the objective has only been pursued with insufficient commitment thus far. The long-term goal should be a continuous integration/continuous delivery (CI/CD) approach, which includes frequent and automated releases. This would decrease errors, manual work and stress (Humble and Farley, 2011, pp. 3-12).

As an intermediate step, however, an attempt should already be made to carry out releases at regular and short intervals, which will be able to mitigate occurring issues (Cesar Brandão Gomes da Silva, Antonio et al., 2017). It is important to see this as an obligation that cannot be circumvented. This would make it possible to provide an up-to-date version of the rating KPI and also to improve that rating, as errors are reduced and corrected faster.

## **4.4. Findings**

The extensive interviews conducted with the coordinators of the development teams led to a multitude of insights regarding the current organisational practices and the implementation process. Both strengths and weaknesses of the

approaches were found and addressed. There are only minor differences between the practices of each team, however some unique team methods were identified and recommended to other teams.

It was found out that the teams are already actively applying some HRM practices for agile organisations. This includes the use of collaboration tools such as Slack, Confluence and Jira. These tools enable simple, fast and location-independent collaboration. Both within teams and between teams. This is also a point for improvement, because while the infrastructure is in place, collaboration is currently only practised within the teams. Another practice that works well is the onboarding process. The fact that an experienced mentor is assigned to every newcomer eases the start in the Catrobat project tremendously.

For all other recommendations and metrics, it was crucial to recognise the importance of highly trained contributors and their collaboration. All KPIs are either defined on the team or on the organisational level, in order to facilitate teamwork and to prevent individual competition. A tool that is as simple and clear as possible was chosen for the visualisation, as there is little need for complex implementations, especially at the beginning. It is much more important that the KPIs have a strong presence, and that the data maintenance effort is minimised so that the metrics are actually integrated into the daily project work.

With regard to the weaknesses in the current processes, recommendations were made that have the potential to strengthen them. However, these are not intended as step-by-step instructions that can guarantee improvements. Rather, they are recommendations, whereby HRM practices for agile organisations from academic sources have been tailored to the specifics of this university FOSS project. Hence, the success depends on the actual implementation. Nevertheless, the implemented KPIs enable the continuous monitoring of organisational change and increase the success rate significantly.

## 4.5. Limitations

Selecting the right KPIs is a highly intricate process. Parmenter (2010, p. 41) describes this endeavour in a twelve-step process with the extensive involvement of a wide range of employees. For small-to-medium enterprises (SMEs) he reduces the twelve steps to eight, however it still requires many people involved and a joint effort to have a high chance of success in picking the matching KPIs for a specific enterprise (Parmenter, 2010, p. 243). Furthermore, even in an ideal context sixteen weeks are seen as the minimum time needed to reach the goal of winning KPIs (R. S. Kaplan and Norton, 1996, pp. 308-309). However, practical examples show that this undertaking can often take longer than a year due to underestimated complexity and subpar procedures (Parmenter, 2010, p. 253).

This master thesis was conducted in a time span of roughly six months.

#### 4. Results and Evaluation

---

Although it was possible to analyse the internal processes in Catrobat and define both critical success factors and key performance indicators, it was necessary to choose different approaches than those recommended in publications. Due to the limited time resources of Catrobat's decision-makers, it was not possible, as expected, to organise workshops lasting several days. With the interviews and a final validation of the decision-makers, however, an alternative was chosen that could produce convincing results.

As the effort of defining KPIs coincided with the amount of time stated in scientific sources, the actual introduction of new HRM practices and their control via the KPIs would have exceeded both the time frame and the scope of this thesis. This also includes a revision of the defined KPIs, which is recommended about six months after the implementation. During the one-time visualisation of the KPIs, it was found that some of the underlying data is incomplete or that there is simply no current data. Examples are the time tracking tool, which is currently offline and the meeting attendance rate, which has not been recorded for every meeting in every team. As a matter of fact, an accurate representation of the KPIs can only be guaranteed if the data quality is flawless.

Furthermore, it must be noted that the implemented KPIs and recommended HRM practices are specific to the Catrobat project. It is likely that similar FOSS projects could use some of the KPIs and/or practices, however it is not recommended to apply the presented measures at another organisation without adapting them according to the characteristics of that organisation.

## 5. Conclusion and Future Work

In the following chapter the results of this thesis are discussed. Prior to that, a reflection regarding the approach of the thesis is presented. This serves the purpose of indicating incalculable obstacles and the knowledge gained in the process of writing this thesis. To conclude, an outlook concerning further work, that builds on the findings of this thesis is shown.

### 5.1. Conclusion

One particularly important insight was the fact that comprehensive changes in an organisation can only be implemented effectively if the current situation has been analysed in detail beforehand. Without knowledge of current processes and problems, there is a risk that unrealistic proposals will be implemented that will be rejected due to their incompatibility with organisational reality.

This comprehensible conclusion raises the question as to why conducting an analysis of the current situation cannot always be taken for granted. Organisations are highly individual and complex bodies, figuring out the inner workings of it is an intricate and long-lasting process. Hence, it might seem intriguing to skip this examination process and to apply a standard procedure for increasing project productivity. In all likelihood, this will result in an unsustainable solution that cannot offer any added value.

For this reason, a great deal of time was spent during this thesis on understanding the special characteristics of the Catrobat organisation. While this was successful, the sometimes mediocre data quality posed a further problem. Therefore, there is still improvement of the data basis needed in order to draw meaningful conclusions from the implementations realised.

This thesis aimed to answer three research questions, in order to reveal which KPIs are useful for measuring performance and productivity in the context of a FOSS project. To gain insight into the project, the expert interview was chosen as a qualitative research method. Based on the findings from the interviews, it was finally possible to define KPIs that are able to visualise the success of the Catrobat project.

Usually, the definition of project success and project performance contains a financial aspect, which guarantees the economic viability of the corresponding business venture. As a result, CSFs and KPIs are also aligned in such a way that financial success is an essential part of the project success definition. In

## 5. Conclusion and Future Work

---

contrast, this unique case of a university driven FOSS project, where there is no need to achieve any financial goals nor to deliver application in a certain amount of time, opens up the possibility of an alternative view on success.

In order to answer the first question - *Which KPIs (Key Performance Indicators) are suitable for measuring productivity and success in a FOSS project?* - five KPIs, that are aligned with the CSFs and the BSC perspectives of the project, were defined:

- Weekly meeting attendance (WMA)
- Story points implemented/merged (SPI)
- Available human resources per team (AHRT)
- Average rating since last release (RSR)
- Average time tickets stay in the implementation process phases (TTIP)

A strong focus is set on the contributors of the project, which are the most valuable resource for Catrobat. Furthermore, the progress of the implementation process is visualised by another two KPIs. Last but not least, the customer perspective, which is represented by the ratings of the mobile applications, is also decisive.

While answering the second question - *Which tools and processes are needed to track the KPIs continuously?* - an interesting conclusion was drawn. After all, an overarching tool that also handles HR management and project management in addition to KPI visualisation is by no means the best solution. Introducing KPIs and a complex tool at the same time would create an excessive demand that would minimise the chances of a successful implementation. Thus, it is recommended to use a tool that is as simple as possible and allows daily automated data import and clear visualisation in a dashboard. The free basic version of the Klipfolio PowerMetrics tool was used as an example in this work. Comprehensive software solutions should only be considered once the use of KPIs has become part of the project routine.

After defining and visualising the KPIs, the third question remains to be answered: *Which actions can be taken to improve the project performance according to the defined KPIs?*

For this purpose, a comprehensive package of HRM practices was presented in chapter 4.3.1. The ranking of the seven proposed practices was done in order to prioritise importance and simplicity of implementation. First and foremost, this is again about promoting the individual contributors and their cooperation, especially across teams. There is also a lot of potential in trying to retain experienced members in the project, as they can contribute the most due to their expertise. As there will always be members who leave Catrobat, there should also be a stronger focus on a well-thought-out offboarding process, similar to the one already implemented for onboarding.

Apart from the HRM perspective, there is also a technical recommendation. The findings from this thesis reaffirm the plan to accelerate the release cycle, as



this could further increase the performance of the project.

## 5.2. Future Work

This work serves as a basic framework for measuring and managing performance in Catrobat. Further steps are necessary in order to gain a benefit for the organisation. First of all, CSFs and KPIs can only deliver value, when all contributors are aware of the main goals of Catrobat. Therefore, the next steps must include an information campaign, where everyone is informed about the CSFs and KPIs so that the daily work can be aligned according to the CSFs. In addition, all contributors need access to the KPIs, so that everyone is involved in delivering and measuring performance.

The required automatic data import must be implemented in order to be able to continuously display current KPI values. Furthermore, a re-evaluation of the defined KPIs is necessary after a few months of use. According to academic research, it is extremely rare that the KPIs selected at the beginning are actually perfectly suited for measuring performance in the specific organisation. The possible adverse effect of ill-defined KPIs was already discussed in chapter 2.4.1 and must be avoided at all costs.

Last but not least, the KPIs should also be used to measure the impact of organisational changes. A separate project is required to carefully implement the recommended management practices. The blueprints defined here provide an initial indication of which practices can offer added value, but the exact manner of implementation still needs to be defined.



# Bibliography

- Aberdour, M. (2007). Achieving quality in open-source software. *IEEE Software*, 24(1), 58–64. <https://doi.org/10.1109/MS.2007.2> (cit. on p. 2).
- Aguinis, H. (2013). *Performance management* (3rd ed.). Pearson. (Cit. on pp. 8, 9, 35).
- AK Munns & BF Bjeirmi. (1996). The role of project management in achieving project success. *International Journal of Project Management*, 14(2), 81–87. [https://doi.org/10.1016/0263-7863\(95\)00057-7](https://doi.org/10.1016/0263-7863(95)00057-7) (cit. on p. 5).
- Alam, M., & Kamal, I. (2024). Reviewing the effectiveness of reward and punishment in improving employee performance in jabodetabek. *Indonesian Journal of Multidisciplinary Science*, 3. <https://doi.org/10.55324/ijoms.v3i6.843> (cit. on p. 9).
- Anchit Shrivastava, Isha Jaggi, Nandita Katoch, Deepali Gupta, & Sheifali Gupta. (2021). A systematic review on extreme programming. *Journal of Physics: Conference Series*, 1969(1), 012046. <https://doi.org/10.1088/1742-6596/1969/1/012046> (cit. on p. 23).
- Apache, D., Mockus, A., Fielding, R., & Herbsleb, J. (2002). Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology*, 11. <https://doi.org/10.1145/567793.567795> (cit. on pp. 1, 18).
- Asproni, G. (2006). An introduction to scrum. *Software Developers Journal*, 6, 1–10 (cit. on p. 21).
- Azad, N., & Hyrynsalmi, S. (2023). Devops critical success factors — a systematic literature review. *Information and Software Technology*, 157, 107150. <https://doi.org/10.1016/j.infsof.2023.107150> (cit. on p. 23).
- Bernhard Schmid & Jonathan Adams. (2008). Motivation in project management: The project manager's perspective. *Project Management Journal*, 39(2), 60–71. <https://doi.org/10.1002/pmj.20042> (cit. on p. 40).
- Bogner, A., Littig, B., & Menz, W. (2014). *Interviews mit Experten: Eine praxisorientierte Einführung*. Springer VS. <https://doi.org/10.1007/978-3-531-19416-5> (cit. on p. 24).
- Breaugh, J. A. (1993). *Recruitment: Science and practice* (2. [print.]). Wadsworth. (Cit. on p. 8).
- Cappelli, P. (2005). Will there really be a labor shortage? In M. R. Losey (Ed.), *The future of human resource management* (pp. 5–14). Wiley. (Cit. on p. 8).

- Cascio, W. F. (2005). From business partner to driving business success: The next step in the evolution of hr management. In M. R. Losey (Ed.), *The future of human resource management*. Wiley. (Cit. on p. 10).
- Cesar Brandão Gomes da Silva, Antonio, de Figueiredo Carneiro, G., Brito e Abreu, F., & Pessoa Monteiro, M. (2017). Frequent releases in open source software: A systematic review. *Information*, 8(3). <https://doi.org/10.3390/info8030109> (cit. on p. 94).
- Cunha, M. P. e., & Chia, R. (2007). Using teams to avoid peripheral blindness. *Long Range Planning*, 40(6), 559–573. <https://doi.org/10.1016/j.lrp.2007.08.004> (cit. on p. 30).
- Dalton, J. (2019). *Great big agile: An os for agile leaders*. Apress L. P. <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5611944> (cit. on p. 24).
- Davenport, T., Harris, J., & Shapiro, J. (2010). Competing on talent analytics. *Harvard business review*, 88, 52–8, 150 (cit. on p. 8).
- Davis, D., & Davis, W. R. (2008). *Team performance inventory: Participant workbook*. (Cit. on p. 10).
- Davis, K. (2014). Different stakeholder groups and their perceptions of project success. *International Journal of Project Management*, 32(2), 189–201. <https://doi.org/10.1016/j.ijproman.2013.02.006> (cit. on p. 37).
- DiBona, C. (Ed.). (1999). *Open sources: Voices from the open source revolution* (1. ed.). O'Reilly. (Cit. on pp. 1, 17).
- Dybå, T., Dingsøyr, T., & Moe, N. B. (2014). Agile project management. In G. Ruhe (Ed.), *Software project management in a changing world* (pp. 277–300). Springer. (Cit. on p. 6).
- Eckerson, W. W. (2006). *Performance dashboards: Measuring, monitoring, and managing your business*. Wiley. <http://www.loc.gov/catdir/enhancements/fy0622/2005011800-b.html> (cit. on p. 16).
- Eckerson, W. W. (2011). *Performance dashboards: Measuring, monitoring, and managing your business* (2nd ed.). John Wiley & Sons Incorporated. <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=624516> (cit. on p. 16).
- Ehls, D. (2017). Open source project collapse – sources and patterns of failure. In *Hawaii international conference on system sciences (hicc-50)*, 2017. AIS Electronic Library (AISeL). (Cit. on p. 23).
- Ellig, B. (2005). What distinguishes the outstanding hr executives from the others. In M. R. Losey (Ed.), *The future of human resource management* (pp. 126–133). Wiley. (Cit. on p. 9).
- Englund, R. L. (2010). Dealing with power and politics in project management. In *The ama handbook of project management*. American Management Association. (Cit. on p. 12).

- Eppler, M. J., & Mengis, J. (2004). Side-effects of the e-society: The causes of information overload and possible countermeasures. *Proceedings of IADIS international conference e-society*, 2, 1119–1124 (cit. on p. 49).
- Fellhofer, S., Harzl, A., & Slany, W. (2015). Scaling and internationalizing an agile foss project: Lessons learned. In E. Damiani, F. Frati, D. Riehle, & A. I. Wasserman (Eds.), *Open source systems: Adoption and impact* (pp. 13–22, Vol. 451). Springer International Publishing. [https://doi.org/10.1007/978-3-319-17837-0\\_2](https://doi.org/10.1007/978-3-319-17837-0_2) (cit. on pp. 23, 24).
- Flamholtz, E. G. (2005). Human resource accounting, human capital management, and the bottom line. In M. R. Losey (Ed.), *The future of human resource management*. Wiley. (Cit. on pp. 10, 12).
- Fogel, K. (2009). *Producing open source software: How to run a successful free software project*. O'Reilly Media Inc. <http://gbv.ebib.com/patron/FullRecord.aspx?p=443307> (cit. on p. 23).
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science behind devops : Building and scaling high performing technology organizations* (First edition). IT Revolution. (Cit. on p. 68).
- Gray, D. (2015). *Measurement madness : Recognizing and avoiding the pitfalls of performance measurement* (1st edition). (Cit. on p. 12).
- Hahn, J., & Zhang, C. (2005). An exploratory study of open source projects from a project management perspective. <https://api.semanticscholar.org/CorpusID:47652335> (cit. on pp. 19, 28).
- Hars, A., & Ou, S. (2001). Working for free? motivations of participating in open source projects. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 9 p-. <https://doi.org/10.1109/HICSS.2001.927045> (cit. on p. 24).
- Humble, J., & Farley, D. G. (2011). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley. (Cit. on p. 94).
- Huzooree, G., & Ramdoo, V. (2015). Review of effective human resource management techniques in agile software project management. *International Journal of Computer Applications*, 114, 975–8887. <https://doi.org/10.5120/19972-1860> (cit. on p. 91).
- Ivancevich, J. M., & Konopaske, R. (2013). *Human resource management* (12th ed.). McGraw-Hill Irwin. (Cit. on pp. 7, 8, 10, 11).
- Jelenic, D. (2011). The importance of knowledge management in organizations—with emphasis on the balanced scorecard learning and growth perspective. *Management, Knowledge and Learning, International Conference*, 33–43 (cit. on p. 45).
- John M Malouff & Einar B Thorsteinsson. (2016). Bias in grading: A meta-analysis of experimental research findings. *Australian Journal of Education*, 60(3), 245–256. <https://doi.org/10.1177/0004944116664618> (cit. on p. 67).

- Kaifeng Jiang, David P. Lepak, Kyongji Han, Ying Hong, Andrea Kim, & Anne-Laure Winkler. (2012). Clarifying the construct of human resource systems: Relating human resource management to employee performance. *Human Resource Management Review*, 22(2), 73–85. <https://doi.org/10.1016/j.hrmr.2011.11.005> (cit. on p. 7).
- Kaiser, R. (2014). *Qualitative experteninterviews: Konzeptionelle Grundlagen und praktische Durchführung*. Springer VS. <https://doi.org/10.1007/978-3-658-02479-6> (cit. on p. 30).
- Kaplan, R. (2001). Strategic performance measurement and management in non-profit organization. *Nonprofit Management and Leadership*, 11, 353–370. <https://doi.org/10.1002/nml.11308> (cit. on p. 46).
- Kaplan, R. S., & Norton, D. P. (1996). *The balanced scorecard: Translating strategy into action*. Harvard Business School Press. (Cit. on pp. 14, 16, 44–46, 95).
- Kerzner, H. (2013). *Project management metrics, kpis, and dashboards: A guide to measuring and monitoring project performance* (2. ed.). John Wiley & Sons Inc. (Cit. on pp. 11–13, 15, 16, 28, 37, 88).
- Kuan, J. (2002). Open source software as lead user's make or buy decision: A study of open and closed source quality (cit. on p. 1).
- Laanti, M., Similä, J., & Abrahamsson, P. (2013). Definitions of agile software development and agility. In F. Mc Caffery, R. V. O'Connor, & R. Messnarz (Eds.), *Systems, software and services process improvement* (pp. 247–258). Springer. (Cit. on p. 6).
- Lamprou, A., & Vagiona, D. G. (2022). Identification and evaluation of success criteria and critical success factors in project success. *Global Journal of Flexible Systems Management*, 23(2), 237–253. <https://doi.org/10.1007/s40171-022-00302-3> (cit. on p. 37).
- Liebold, R., & Trinczek, R. (2009). Experteninterview. In S. Kühl, P. Strodtholz, & A. Taffertshofer (Eds.), *Handbuch Methoden der Organisationsforschung*. VS Verlag für Sozialwissenschaften. (Cit. on p. 29).
- Lientz, B. P. (2001). *Project management for the 21st century* (3rd ed.). Academic. (Cit. on p. 5).
- Lukusa, L., Geeling, S., Lusinga, S., & Rivett, U. (2021). Teamwork and project success in agile software development methods: A case study in higher education. *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 885–891. <https://doi.org/10.1145/3434780.3436648> (cit. on p. 61).
- Marlow, S. (2006). Human resource management in smaller firms: A contradiction in terms? *Human Resource Management Review*, 16(4), 467–477. <https://doi.org/10.1016/j.hrmr.2006.08.004> (cit. on p. 29).
- Martello, M., Watson, J., & Fischer, M. (2008). Implementing a balanced scorecard in a not-for-profit organization. *Journal of Business & Economics Research (JBER)*, 6. <https://doi.org/10.19030/jber.v6i9.2471> (cit. on p. 44).

- Mathis, R. L., & Jackson, J. H. (2008). *Human resource management* (12. ed., student ed.). Thomson South-Western. <http://www.loc.gov/catdir/enhancements/fy1215/2007933762-d.html> (cit. on pp. 7, 8, 10, 11, 31, 92).
- Mayring, P. (2014). *Qualitative content analysis: Theoretical foundation, basic procedures and software solution*. (Cit. on pp. 25, 26).
- Mayring, P., & Fenzl, T. (2019). Qualitative inhaltsanalyse. In N. Baur & J. Blasius (Eds.), *Handbuch methoden der empirischen sozialforschung*. Springer VS. (Cit. on p. 26).
- Mendelssohn, A., & Howell, M. (2010). Studies in project quality management: Achieving business excellence using baldrige, business process management, process improvement, and project management. In *The ama handbook of project management*. American Management Association. (Cit. on p. 15).
- Meusner, M., & Nagel, U. (2009). Das experteninterview — konzeptionelle grundlagen und methodische anlage. In *Methoden der vergleichenden politik- und sozialwissenschaft*. VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden. (Cit. on pp. 28, 29).
- Moran, A. (2015). *Managing agile: Strategy, implementation, organisation and people*. Springer. <http://gbv.ebib.com/patron/FullRecord.aspx?p=2094506> (cit. on pp. 6, 8, 21).
- Müller, M., Schindler, C., & Slany, W. (2019). Engaging students in open source: Establishing foss development at a university. In T. Bui (Ed.), *Proceedings of the 52nd hawaii international conference on system sciences*. Hawaii International Conference on System Sciences. <https://doi.org/10.24251/HICSS.2019.930> (cit. on pp. 20, 22).
- Omotayo, F. O. (2015). Knowledge management as an important tool in organisational management: A review of literature. *Library Philosophy and Practice*, 1(2015), 1–23 (cit. on pp. 31, 72).
- Parmenter, D. (2010). *Key performance indicators: Developing, implementing, and using winning kpis* (2. ed.). Wiley. (Cit. on pp. 12–16, 19, 34, 35, 37, 43, 46, 47, 75, 76, 80, 95).
- Pennypacker, J. S. (2010). Measuring the value of project management: A measurement system. In *The ama handbook of project management*. American Management Association. (Cit. on p. 16).
- Prielipp, R. C., Magro, M., Morell, R. C., & Brull, S. J. (2010). The normalization of deviance: Do we (un)knowingly accept doing the wrong thing? *Anesthesia and analgesia*, 110(5), 1499–1502. <https://doi.org/10.1213/ANE.obo13e3181d5adc5> (cit. on p. 31).
- Raymond, E. S. (2001). *The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary*. O'Reilly Media. <http://gbv.ebib.com/patron/FullRecord.aspx?p=443450> (cit. on pp. 1, 18, 19).
- Ruhe, G. (Ed.). (2014). *Software project management in a changing world*. Springer. (Cit. on p. 2).

- Sachdeva, S. (2016). Scrum methodology. *Int. J. Eng. Comput. Sci*, 5(16792), 16792–16800 (cit. on p. 50).
- Sandred, J. (2001). *Managing open source projects*. Wiley Computer Pub. (Cit. on pp. 17, 18).
- Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding free/open source software development processes. *Software Process: Improvement and Practice*, 11(2), 95–105. <https://doi.org/10.1002/spip.255> (cit. on p. 18).
- Scheibelmasser, P. (2024). *Development of an open source metadata-analysis/-reporting solution for it project management systems* [Master's Thesis]. Graz University of Technology. Institut of Software Technology. (Cit. on p. 77).
- Schreiner, P. (2022). *Analysis and improvement of onboarding at an open source project* [Master's Thesis]. Graz University of Technology. Institut of Software Technology. (Cit. on p. 31).
- Schwab, F., & Schwab-Matkovitz, I. (2013). *Systemplanung und projektentwicklung: Projektentmanagement und software-entwicklung*. Manz Verl. Schulbuch. <https://permalink.obvsg.at/AC11300160> (cit. on pp. 50, 51).
- Shahzadi, I., Javed, A., Pirzada, S. S., Nasreen, S., & Khanam, F. (2014). Impact of employee motivation on employee performance. *European Journal of Business and Management*, 6(23), 159–166 (cit. on p. 32).
- Tiwari, P., & Saxena, K. (2012). Human resource management practices: A comprehensive review. *Pakistan business review*, 9(2), 669–705 (cit. on p. 91).
- von Glinow, M. A., Drost, E. A., & Teagarden, M. B. (2005). Counterintuitive findings in international hrm research and practice: When is a best practice not best for practice? In M. R. Losey (Ed.), *The future of human resource management*. Wiley. (Cit. on p. 10).
- Webster, F. M., & Knutson, J. (2010). What is project management? project management concepts and methodologies. In *The ama handbook of project management* (pp. 1–10). American Management Association. (Cit. on pp. 5, 11).
- Zheng, M., Zada, I., Shahzad, S., Iqbal, J., Shafiq, M., Zeeshan, M., & Ali, A. (2021). Key performance indicators for the integration of the service-oriented architecture and scrum process model for iot. *Scientific Programming*, 2021, 1–11. <https://doi.org/10.1155/2021/6613579> (cit. on p. 50).



# Acronyms

<b>AHRT</b>	Available human resources per team
<b>API</b>	Application Programming Interface
<b>BSC</b>	Balanced Scorecard
<b>CD</b>	Continuous Delivery (or Deployment)
<b>CF</b>	Customer Focus (BSC Perspective)
<b>CI</b>	Continuous Integration
<b>CI/CD</b>	Continuous Integration/Continuous Delivery (or Deployment)
<b>CSF</b>	Critical Success Factor
<b>CSS</b>	Closed Source Software
<b>DevOps</b>	Development and Operations
<b>DFSG</b>	Debian Free Software Guidelines
<b>EC</b>	Environment/Community (BSC Perspective)
<b>ECTS</b>	European Credit Transfer and Accumulation System
<b>EEO</b>	Equal employment opportunity
<b>ES</b>	Employee Satisfaction (BSC Perspective)
<b>F</b>	Financial (BSC Perspective)
<b>FOSS</b>	Free and Open-source Software
<b>GNU</b>	GNU's Not Unix
<b>GPL</b>	General Public License
<b>GUI</b>	Graphical User Interface
<b>HR</b>	Human Resource
<b>HRM</b>	Human Resource Management
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ICT</b>	Information and Communications Technology
<b>IDE</b>	Integrated Development Environment
<b>IP</b>	Internal Process (BSC Perspective)
<b>ISO</b>	International Organization for Standardization
<b>IT</b>	Information Technology
<b>KM</b>	Knowledge Management
<b>KPI</b>	Key Performance Indicator
<b>KRI</b>	Key Result Indicator
<b>LG</b>	Learning and Growth (BSC Perspective)
<b>OSI</b>	Open Source Initiative
<b>OSS</b>	Open Source Software
<b>PC</b>	Personal Computer

## Acronyms

---

<b>PI</b>	Performance Indicator
<b>PO</b>	Product Owner
<b>PR</b>	Pull Request
<b>QA</b>	Quality Assurance
<b>RI</b>	Result Indicator
<b>RQ</b>	Research Question
<b>RSR</b>	Average rating since last release
<b>SPI</b>	Story points implemented/merged
<b>TTIP</b>	Average time tickets stay in the individual phases of the implementation process
<b>TU</b>	Technical University
<b>WMA</b>	Weekly meeting attendance

# Appendix



# Appendix A.

## Interview Field Manual

### **Start recording**

#### **Start the interview:**

Hello!

Thank you for taking the time to talk to me and for your willingness to conduct an interview.

As already mentioned, I am conducting a scientific survey as part of my master's thesis at the University of Technology Graz regarding human resource management and possibilities on how to measure performance and productivity in the Catrobat project.

The interview will take about 45 - 60 minutes and will be recorded. After the evaluation, the audio recording will be deleted for data protection reasons. You have already given your consent to the collection and processing of your data in the form of a signature, thank you for this. [Alternative: I still need your consent in the form of a signature for the collection and processing of your data. (sign now)]

I will ask you some questions about your role and responsibilities in the Catrobat project as well as the current state of organisational processes. If there is anything you don't want to answer, just let me know. Otherwise, I would like to ask you to talk as freely and informally as possible, everything you say is important and correct, the interview is about your experiences. The interview is divided into several thematic areas, I will always inform you when we proceed with the next topic.

Are there any uncertainties, do you have any questions about the interview or the process?

Then let's start with the first question. Please take as much time as you need to answer, you are welcome to talk in detail.

### **Topic area A: Field of activity**

1. First of all, I would like to ask you to tell me something about your field of activity and your role in the Catrobat project! You can take as much time as you like for this. I won't interrupt you, just make a few notes, which I will discuss later.

2. When you think about your work as team coordinator, what are your most

important tasks?

3. Please explain to me your tasks that take up the most time and explain why they are so time-consuming!

4. Please try to remember your first days as a coordinator. Which challenges did you face in your new position?

### **Topic area B: Team organisation**

5. Please give me some information about the structure of the team you coordinate!

*Additional question: How many team members does the team have and is this always the same?*

*Additional question: For how long have the team members already been contributing and how long will they still be part of the team?*

6. Please tell me about your team meetings in detail, for example when, how often, where, which types of meeting, ...!

*Additional questions: Please tell me about the attendance rate at the meetings, is it difficult to get all team members together?*

*Additional question: How important are regular meetings for the project progress and why?*

7. When you think about collaboration within the team as well as in between teams in Catrobat, please name and explain obstacles, that hinder this cooperation, if there are any!

### **Topic area C: Development process**

8. Please explain to me the process of task implementation, from moving a ticket from "ready for development/to do" to "merged/done"!

*Additional question: How long does it approximately take until a ticket has undergone this process from "ready for development/to do" to "merged/done"?*

*Additional question: Are there any bottlenecks in this process? If yes how are you trying to mitigate them?*

9. Please tell me about the planning of this development process, how do you decide which tickets should be moved to "ready for development"?

*Additional question: How much information do you have on the availability of your team members in advance, for example how many hours they plan to work the next week or if they are available at all?*

*Additional question: Are there some kind of time critical tickets in your project? If yes: how do you make sure that those tickets are done in time?*

### **Topic area D: Recruiting, On- and Offboarding**

10. When you think about recruiting new members, how does the process for filling open positions work in detail?

*Additional question: How much influence do you have on deciding who will join the team?*

---

*Additional question: Can you think of one or more situations, where the lack of new team members led to difficult conditions for the project progress? If yes, please tell me about it!*

11. Please tell me about your experiences regarding the offboarding process of team members!

*Additional question: Please elaborate on the knowledge transfer from experienced members, which leave the project to new members!*

12. I would like to speed forward to the moment when you are finished with your hours as coordinator. Please tell me how you imagine the onboarding process for your successor!

*Additional question: What difficulties could arise during this transition?*

### **Topic area E: Performance measures**

13. What do you believe are the key factors that have contributed most significantly to the achievements of the Catrobat project, and why?

14. Please elaborate on the options and measures you have to evaluate the contribution and performance of the individual team members!

*Additional question: Do you feel that you need to have a better overview and understanding of the performance of your team and if yes, what would you need for gaining this overview?*

15. Team members, which participate in Catrobat as part of a university course, agree to spend at least 24h per month working on the project. Why has this threshold been introduced?

*Additional question: How do you check if this 24h threshold is reached by every team member?*

16. What are the consequences for them personally as well as for the project performance if somebody does not meet the threshold?

17. What is your impact on the grade of the team member leaving? Additional question: Please elaborate on how you decide on the grade?

### **Topic area F: Team member motivation**

18. What do you think, what motivates students to contribute to the Catrobat project?

19. Which impact does the motivation level of each team member have on the team performance?

20. Please tell me about incentives to increase the motivation of participants in the Catrobat project!

*Additional question: If there are not any, what do you think, why haven't any incentives been established yet?*

**Is there anything else you would like to share?**

**Thank you very much for the interesting conversation!**





# Appendix B.

## Interview Transcript

This chapter contains the transcripts of the interviews conducted with the coordinators of four *Catrobat* developer teams.

### B.1. Interview with Paintroid Coordinator

1 I:[00:03] Okay, hello. Thank you for taking the time to talk to me and for your  
2 willingness to conduct an interview. As already mentioned, I am conducting a  
3 scientific survey as part of my master's thesis at the University of Technology  
4 Graz regarding human resource management and possibilities on how to mea-  
5 sure performance and productivity in the Catrobat project.  
6 The interview will take about 35 to 45 minutes and will be recorded. After the  
7 evaluation, the audio recording will be deleted for data protection reasons. You  
8 have already given your consent to the collection and processing of your data  
9 in the form of a signature.  
10 Thank you for this. I will ask you some questions about your role and respon-  
11 sibilities in the Catrobat project as well as the current state of organisational  
12 processes. If there is anything you don't want to answer, just let me know.  
13 Otherwise, I would like to ask you to talk as freely and informally as possible.  
14 Everything you say is important and correct. The interview is about your expe-  
15 riences.  
16 The interview is divided into several thematic areas. I will always inform you  
17 when we proceed with the next topic. Are there any uncertainties? Do you have  
18 any questions about the interview or the process? [01:27]  
19 E1:[01:29] No, I think everything is very clear. [01:31]  
20 I:[01:32] Okay, then let's start with the first question. Please take as much time  
21 as you need to answer. You are welcome to talk in detail. So, the first topic area  
22 is about your field of activity. And first of all, I would like to ask you to tell  
23 me something about your field of activity and your role in the Catrobat project.  
24 You can take as much time as you like for this. I won't interrupt you, just make  
25 a few notes, which I will discuss later. [01:57]  
26 E1:[01:58] Sure, I'm a coordinator at Catrobat for a specific product development  
27 team at the moment. Yeah, I am like the right hand of the PO, the product

28 owner. And so, I do everything the team needs. Sometimes, I'm a technical  
29 lead. Sometimes, I just fill in for the PO. I'm there for my team if there are any  
30 questions or anything or any organisational matters. So, yeah, that's what I'm  
31 doing currently. Also, I'm also having meetings with the other coordinators to  
32 get the news from other teams and to work together. Yeah, that's kind of what I  
33 do. [02:53]

34 I:[02:56] When you think about your work as team coordinator, what are your  
35 most important tasks? [03:01]

36 E1:[03:04] I think just having the team, checking what the team needs, having  
37 the progress in focus. Also, right now, team meetings because I don't have any  
38 scrum master or something like that. So, yeah, just having the team make good  
39 progress and do all the organisational stuff so they can work with ease. [03:34]

40 I:[03:37] Please explain to me your tasks that take up the most time and explain  
41 why they are so time consuming. [03:43]

42 E1:[03:46] I think the most time would be having technical meetings with the  
43 team. Also, because we mostly have in-person meetings for a few hours. So,  
44 that takes up a lot of time and also doing QA reviews. So, when the team has  
45 the ticket finished and also they did a code review, then the ticket is assigned  
46 to me and I check the ticket, I test the app, and I decide if it gets merged or  
47 not. So, this is how I also help the PO because that would be a PO task in other  
48 structures. But here, yeah, that's what I do the most. [04:42]

49 I:[04:48] Please try to remember your first days as a coordinator. Which chal-  
50 lenges did you face in your new position? [04:55]

51 E1:[04:59] Probably just knowing what tasks I should do. Like, I think there  
52 wasn't a clear guideline on what are my tasks, what do I have to do, and also  
53 not having an active product owner at that time. Like, in another team, not this  
54 team. [05:23]

55 I:[05:29] Okay. So, we'll switch to the next topic, which is team organisation.  
56 And so, please give me some information about the team structure you coordi-  
57 nate. [05:42]

58 E1:[05:45] Right now, we have a PO, a product owner, who is mostly there for  
59 meetings and also big decisions on stories we need to do and also for releases.  
60 Then there's me, the coordinator, well yeah, what I already explained. And I  
61 think currently six developers, mostly from the TU Graz, like internal members,  
62 but also one external member, which is also one thing I have to do to, like,  
63 communicate with external members. [06:29]

64 I:[06:35] Mm-hmm. Is the amount of team members always the same, or does  
65 this change a lot? [06:41]

66 E1:[06:43] Currently, the team members are the same as for the last few months.  
67 But I also already had, like, stages at my time at Catrobat where the team  
68 members changed so quickly. Like, they only did 200 hours, and then they were  
69 gone. So, yeah, that can be quite challenging. But right now, we have quite a  
70 strong team, because I have four members that are here for a long time, like,

71 over a year already, and they also plan to do more than a year. So, right now,  
72 it's great. [07:25]

73 **I:**[07:31] So, I think you answered it more or less. But do you know how long  
74 they will still be part of the team, your current team members? [07:43]

75 **E1:**[07:46] Yes, I think four of them will be there for another year. And I think  
76 the external member won't be there for that long, and other members also won't  
77 be there. Like, we also already have some that are almost finished, so I don't  
78 really count them right now. [08:07]

79 **I:**[08:14] So, please tell me about your team meetings in detail. For example,  
80 when, how often, where, and which types of meeting do you have? [08:22]

81 **E1:**[08:24] We have a general meeting every two weeks, where we discuss what  
82 everyone is doing right now, and what they plan to do in the future, like, in the  
83 next two weeks. Also, if there is anything to discuss, we can do it there. Also, if  
84 anyone needs help, they will ask for help in that meeting. And other than that,  
85 we have, like, retrospectives, maybe twice a year, where we discuss what we  
86 can what we're doing good, what we're not doing as good, and what we can  
87 do better. And we try to, yeah, just become a better team. And, like, clearing  
88 up impediments that make us not as great of a team. Other than that, yeah, we  
89 just have, like, technical meetings, which would be, I don't know, maybe every  
90 second month, every month, depends. Because we're, yeah, just doing a big  
91 restructure. And yeah, we want to have a good amount of in-person meetings  
92 to discuss that matter. [09:46]

93 **I:**[09:50] So, please tell me about the attendance rate at the meetings. Is it diffi-  
94 cult to get all team members together? [09:56]

95 **E1:** [09:57] Yes, I think all, having all together, I'm not sure when was, when  
96 it was the last time we had all members at one meeting. But I think, like, 80%  
97 should be there every time. So, I think it's not that big of an issue, because we  
98 can just, like, they also text me, like, message me if they need anything. So, it's  
99 not a big problem for me. [10:27]

100 **I:**[10:29] Mm-hmm. And how important are regular meetings for the project  
101 progress and why? [10:34]

102 **E1:** [10:36] I think even if we do not have, like, a lot to talk about, it's good  
103 to keep the motivation up. Because I think, like, having a team and know  
104 the others are working and also speaking to them helps everyone with their  
105 motivation and not forgetting that there is Catrobat, while they're doing, like,  
106 all their university or maybe also social work stuff. So, I think it's good, great  
107 to check in every second week and, yeah, focus again. [11:17]

108 **I:** [10:21] When you think about collaboration within the team as well as in  
109 between teams in Catrobat, please name and explain obstacles that hinder this  
110 cooperation, if there are any. [11:33]

111 **E1:** [11:37] In the team, I think, yeah, it depends. Sometimes people, like, do not  
112 like each other, which, yeah, can be a major issue, which I already had. Also,  
113 I think the language is also sometimes an issue, like, not that big of an issue,

114 but people who speak German tend to do more pair programming with people  
115 that are also speaking German. So, if someone only speaks English, I think they  
116 will not as, like, the other people won't work with them as much as they would  
117 with the German ones, like, German speaking ones. Other than that, I think in  
118 my team it currently works really good.

119 With other teams, I don't know, I think we don't know each other that good. We  
120 have, like, biweekly meetings with the coordinators, and everyone talks about  
121 what they are doing in the team, what's happening there, but it's not really, like,  
122 discussing team matters with each other. So, it's just not just everyone, yeah,  
123 tells the other what they're doing, and there's, like, no real talk about it. So, I  
124 think it can be hard for some coordinators to, like, go to the other team and ask  
125 for help or also work together, especially if they're not at Catrobat for a long  
126 time. Like, I've been here for years, so I know the people, like, I'm not afraid to  
127 write anyone if I don't know them, but I think new coordinators have, like, a  
128 little bit issues with that. And also, like, other team members, other than the  
129 coordinator or POs, they don't really work together.

130 Like, they don't know the members of the other teams, so they really don't like  
131 to work with those. So, yeah, even if sometimes it would be great, because we  
132 have three Android native teams, so they can easily work with each other, but,  
133 yeah, we don't have quite that community, I think. [14:09]

134 I:[14:14] So, we'll move to the next topic area, which is the development process.  
135 And I'd like to ask you to explain the process of task implementation, from  
136 moving a ticket from ready for development to merge. [14:30]

137 E1:[14:33] So, yeah, the tickets are in the ready for development, like, lane on  
138 the Jira board, and everyone who's working on the team can pick a ticket. And,  
139 yeah, they assign it to themselves, and they start working on it. Sometimes  
140 they are doing pair programming, sometimes they're just working on it alone.  
141 They write tests, they implement the feature, and, yeah, when they are finished,  
142 they put the ticket in code review, and sometimes, sometimes not, ask another  
143 member to code review it. Yeah. Like, you mean to merged, or just this process?  
144 [15:24]

145 I: [15:25] Yes, to merged. [15:27]

146 E1: [15:29] Yeah, then maybe the other code reviewer, like they talk to, will code  
147 review it. If they didn't find someone, it will just stay in the code review lane  
148 on the Jira board until someone picks the ticket again and code reviews it. If  
149 there are any issues, any questions, the ticket goes back to in development, so  
150 the other, like the original developer knows they have to do something with the  
151 ticket. Maybe answer, maybe change some stuff, maybe add more tests, things  
152 like that. And when the code reviewer says, okay, now it looks good, they put  
153 it in the QA lane, which is there for especially the PO, the coordinator, and also  
154 for senior devs, because they are allowed to merge tasks and bugs. So, yeah,  
155 that's mostly my job, but so I pick the ticket, I check out the code again, because  
156 I think it's good for the code quality to have the coordinators check the code

again, also the POs, because I think everyone in our team is technical, so they can check the code even if they are PO. And if I find anything in there that I don't like, I will, like, text them and write on the pull request and ask them to change it. And if everything is good, I also download the APK to my phone, I test the app, I test the new feature, I just go through everything I can imagine that would break with this, like, that could break with this ticket. And, yeah, if everything is great, I will merge it and put the ticket to merged status. [17:40]

**I:**[17:42] How long does it approximately take until a ticket is merged? [17:47]

**E1:**[17:49] Depends. Could be days, could be months. So, it's, like, do you mean from QA lane to merge? [17:59]

**I:**[18:00] No, this whole process. [18:02]

**E1:**[18:02] Yeah, it can be months. I also already had tickets that were over a year, but most likely they will just land in in development again and get restarted, because it makes more sense. But, yeah, if the developer takes a lot of time, the code reviewer, like, no one wants to code review it, and also the QA reviewer takes a lot of time, if there are a lot of issues, it could also be, like, very big tickets with new features that are, like, really complex, it can take a lot of time. [18:36]

**I:**[18:39] Are there any bottlenecks in this process? And if yes, how are you trying to mitigate them? [18:45]

**E1:**[18:47] Yeah, so, code review is one of the bottlenecks, because I think it's just more fun to code than to review the code. But, yeah, I think right now it does not work as bad as before, because people are, like, talking to each other and just ask their teammates to review it with them, also, like, do a pair review. So, they just do a call and show the others the code, explain it, so it's quicker and does not take, like, it's synchronous, so it does not take that much time. Also, if they do pair programming, they are allowed to skip the code review, because, like, one is the official developer, one is the official code reviewer. And the other bottleneck is the QA review, which, yeah, we already worked on that. So, the seniors, as I already mentioned, are allowed to do some QA reviews. So, if it's just a bug or a task that is not that big, like, it has no real business value or stuff like that, then they are allowed to merge it themselves. So, that makes it a little bit quicker. [20:18]

**I:**[20:21] Please tell me about the planning of this development process. How much information do you have on the availability of your team members in advance? For example, how many hours they plan to work or if they are available at all? [20:35]

**E1:**[20:38] We currently don't plan anything in that case. Like, the team members have, like, an amount of hours they are working for the team or the project. But we don't plan, like, how much, like, how many hours would they do in a month, because it's really depends on their life. And, like, sometimes they're just doing that, like, 24 hours. And sometimes they're doing, like, 60 hours. So, there's no process on planning that right now. [21:16]

200 I:[21:17] Okay. But are there some kind of time critical tickets in your project?  
201 And if yes, how do you make sure that those tickets are done in time? [21:26]  
202 E1:[21:28] Yes, they are. Mostly, like, API updates or, like, tickets that are really  
203 critical. And what I do is I just ask my developers if anyone can, like, focus on  
204 that and do it quickly, because it's, like, critical. And if no one does it, I just do  
205 it myself. So, yeah. [21:54]  
206 I:[21:57] So, we'll go on to the next topic area, which is on- and offboarding.  
207 And the next question is, when you think about recruiting new members, how  
208 does the process for filling vacancies work in detail? [22:10]  
209 E1:[22:13] Yeah. Like, I check my team's hours, they have still left at the project.  
210 And if I see, like, I need a new member, because I need the, like, the more senior  
211 ones to help the new member. And there should be enough overlap. I just put  
212 the, like, the information in a sheet that is provided from Catrobat. Like, there's  
213 a sheet with, like, all the teams. And, like, how many members we need, how  
214 critical it is, and what they should know. So, I put my number in there. And if  
215 it does not get, like, if they don't check it and I don't hear from anyone, I would  
216 just get to one of the other, like, organisational people in our project and ask  
217 them if I can have a new member and that I need one quickly, quicker. Yeah.  
218 [23:22]  
219 I:[23:25] How much influence do you have in deciding who will join the team?  
220 [23:30]  
221 E1:[23:33] I think almost none. Like, most times, I just got, like, I just said I need  
222 a member. And then there is a, like, welcome text in my team's chat that tells  
223 us that there's a new member. And then I have to write them and have, like, an  
224 onboarding meeting. But there was, there were, like, I think a few times in the  
225 past years, I got asked if I need that member. Like, someone said they are very  
226 good at Flutter. And I did not need any members. Then they asked me, hey,  
227 could you take this member? Because it would fit really good for your team.  
228 [24:21]  
229 I:[24:24] Can you think of one or more situations where the lack of a new team  
230 member led to difficult conditions for the project progress? And if yes, please  
231 tell me about it. [24:35]  
232 E1:[24:36] Yes, sure. Like, I think not, like, in the last two years. But earlier  
233 than that, we had some times that I did not get a new member. But my other  
234 members were already, like, finished with the hours. And they, like, they said  
235 they will stay a little bit without, like, having to do the hours. But yeah, I mean,  
236 you're a lot more motivated if you need to do them. So, yeah, it was just harder  
237 to, like, really onboard the new member because I, like, did not have someone  
238 to mentor them and have, like, more senior people to show them the code and  
239 also help them. So, yeah, that was an issue already. Not right now, though.  
240 [25:37]  
241 I:[25:41] So, please tell me about your experiences regarding the offboarding  
242 process of team members. [25:47]

**E1:**[25:49] Yeah, for the offboarding, it's usually that the team member just, I think, in the team sheet, there's, like, a message that if they're under 80 hours, they should contact their coordinator. So, sometimes someone will write me and say, hey, I should, it tells me I should write you because I'm almost done with my hours. So, we, yeah, just I just tell them, yeah, that's fine. Do you want to continue or not? And if they don't want to continue, I ask them to please focus on their tickets, not starting too many new tickets because that would be an issue if they are then just gone and they have 10 open tickets. But most times they also agree to, like, check their tickets, like, if they develop the ticket and it gets to the code review and someone asks something about that, then they will, like, respond even if they don't need to do it anymore. And I also ask them to write, like, information they think we would need in the future into the wiki. And yeah, that's kind of it. Like, we don't have, like, a big process for offboarding. Then I would just tell the team that, like, right now there's someone finished with their hours. So, I tell the team they won't be joining the meeting anymore, but they are still responding if they need anything. And yeah, after their tickets are done, they will be gone. [27:35]

**I:**[27:38] Okay, you already, sorry [interrupted the expert]. [27:40]

**E1:**[27:40] Yeah, I remember also, like, they will write me when they really finish with their hours to get a grade that I need to give them. [27:51]

**I:**[27:53] Okay. So, you already mentioned the wiki, but are there any other forms of knowledge transfer between the experienced members when they leave? [28:03]

**E1:**[28:05] Not quite when they leave. Like, we are doing pair programming, so that should be, like, the most knowledge transfer, like, also the most successful knowledge transfer, because I think it helps a lot more than just to write stuff in the wiki that no one ever, like, checks again or also checks if it's still needed, if it's still correct. So, but no, we don't really have, like, an offboarding system where we say, okay, now you go talk to that one and explain them something, because it's just, you can't just make up things that you have to transfer the knowledge. It should happen, like, all the time before that already. [28:55]

**I:**[28:59] Okay. So, now I would like to speed forward to the moment when you are going to resign from your role as coordinator. Please tell me how you imagine the selection of your successor and how smooth will this transition be in your expectation? [29:14]

**E1:**[29:17] So, when I am finished and I get a new coordinator? [29:21]

**I:**[29:22] Yes, exactly. [29:23]

**E1:**[29:25] Yeah. I already had that once, because I already, like, changed my team once. But, yeah, I'm here for a lot of years, so I will probably start to look for a new coordinator when I have only like 100 hours left, because I'm probably gonna be here even if I don't have to be here anymore for questions and for helping the new coordinator. What could be an issue is to find a coordinator, so right now I have a team with a lot like more senior people, so I also already

286 talked to them that I need a new coordinator next year, because I'm really  
287 finished with my hours, and then I will just have them take over the meetings,  
288 have them take over other stuff like more QA reviews and stuff like that, and  
289 yeah, then I think it would just be a smooth process, because I would start early  
290 and also I would be there after my hours are finished, so every time you need  
291 help, you can just invite me. [30:56]

292 I:[31:00] Okay, fine, so we're going to the next topic, which is performance  
293 measures, and the first question, in your opinion, how is success defined in  
294 the Catrobat project? So in other words, which factors need to be met that you  
295 would evaluate the whole project and especially your team as performing well?  
296 [31:22]

297 E1:[31:24] Currently, there is no measure, like we don't have any KPIs, there is,  
298 no, there's not, like we have story points, but we don't check how many we do  
299 in a given like time interval, so right now there's not anything we can measure  
300 our performance, but now it's just like the feeling like there's progress, the app  
301 still lives, the app is still in the app store, and yeah, that's kind of it. [32:01]

302 I:[32:04] Okay, please elaborate on the options and measures you have to evalu-  
303 ate the contribution and performance of your team members. [32:13]

304 E1:[32:14] Okay, so individual team members for the grade? [32:16]

305 I: [32:16] Yes! [32:17]

306 E1:[32:17] Okay, for that, I mean, a lot of that part is just like how we interact  
307 with them in the team, like a lot of it is how well they did with working together  
308 in the team, communicating, being like answering if you message them, being  
309 in team meetings, showing like motivation on changing stuff in the team, like  
310 there's a difference if someone is like really like working on the team, creating  
311 new tickets, asking for a technical meeting because they want to change stuff,  
312 they want to restructure something, so stuff like that. Also, I would check the  
313 codes they checked in, like the progress that got merged, tickets they did, just  
314 to also check if they really did something, if they, I mean, most times I would  
315 see that while they are in the team, but I just check it afterwards again to be  
316 sure that, yeah, they have worked in the team, that they left the trace in the  
317 code, and yeah. Also, I would check the timesheet so that they like not just  
318 input any fantasy hours they did and that it looks correct. [34:04]

319 I:[34:07] Okay. Do you feel that you need to have a better overview and under-  
320 standing of the performance of your team, and if yes, what would you need for  
321 it? [34:17]

322 E1:[34:19] Like for me personal, I'm not sure if I need that, like I don't need  
323 it for the grade, I think, but I think it would be great to try to have like more,  
324 like do more planning in the team, like what should we do in the next month,  
325 and should we try to get it done, and also check how many like story points or  
326 hours we got to do in that month. So, that would be great to like see that, and  
327 also to plan other months, but it's just really difficult in a team full of students.  
328 [35:13]



I:[35:16] So, as you already mentioned, team members which participate in Catrobat as part of a university course agree to spend at least 24 hours per month working on the project. In your opinion, why has this threshold been introduced? [35:32]

E1:[35:34] I think, I'm not sure if it's still 24 hours, because I think it's now changed and it's like specific to each person. I'm not sure, but something about that, yeah. Yeah, I think it's just to not have people slack and do nothing, because I already saw it a few times, although before the 24 hours were introduced, the people just were gone for half a year, so you like really didn't know how many members you had in a team. If they will come back, they, some members just didn't communicate at all. So, we had to do the 24 hours limit to, yeah, have students like a little bit to tick their ass, to have them like really check that they are doing work, to focus, to remember that they have to do something and not just be gone for a month. [36:44]

I:[36:47] So, what are the consequences for them personally, as well as for the project performance, if somebody does not meet this threshold? [36:58]

E1:[36:59] I think for the project performance, it's, I mean, sure it will change something, but we are not like that progress driven right now. It's not like, oh, we have to do something in a strict amount of time. So, it hasn't been an issue for me or in any teams that I've been already. But yeah, for the team members, if they don't do the 24 hours, they get a warning and if they then don't respond or maybe, yeah, just do it again for a lot of times, because they have some, I think, three times they can go under that limit. I think, yeah, it depends on how many hours they are doing in the project. But I already saw someone get kicked out of the project, but it was like really in the beginning of the new limit and they were gone for, I think, almost a year before that. So, I understand that. But right now, I didn't really have that problem because I had a lot of motivated people in my team and that was not an issue. Like, even if they would have like 20 hours, I would not want to kick them out, because if they like do 60 hours the next month and I see they want to work and they have a lot of motivation, then it's not an issue for me. [38:45]

I:[38:48] So, you already talked about it a bit, but what is your impact on the grade of the team member leaving? [38:56]

E1:[38:58] So, the team member has to fill out like a form to get a grade and they will, like, there it's written that they have to ask me for like a kind of token or something, like, I'm not sure what it's called. But I have a page where, like, after I check all that stuff I already said before, like the timesheet and think about how they acted in the team, check the pull requests, I go on that page and I have, like, for grade one, there are multiple tokens that, so they don't know which grade I give them. So, they don't see what this token means, because they don't have access to the page. So, I can just give them that and they will fill it in in a form. And I think it has, like, a lot of weight, but I'm not sure, like, I think it's not just me giving the grade, I think there should also be other ones

372 that, like, the one that then really gives the grade, also sometimes checks the  
373 pull request, checks the timesheet, but I'm not sure, like, what happens after  
374 that. [40:20]

375 I:[40:24] Okay, so now we're approaching the last topic area, which is team  
376 member motivation. What do you think, what motivates students to take part  
377 in the Catrobat project? [40:38]

378 E1:[40:40] Like, why they start? [40:43]

379 I:[40:43] Mm-hmm. Why choose they to participate in this project? [40:47]

380 E1:[40:47] Okay. It was long ago. So, I think, like, most people just join because  
381 they hear that it's a great project from other people. Like, most come in because  
382 they have friends that are already on the team. And also because they, like,  
383 like, for them, like, three teams that do mobile stuff, they may be interested in  
384 mobile apps. That's why I also joined, because we didn't have that, like, that  
385 many lectures on mobile development. So, I wanted to try it out. And, yeah,  
386 that's why I came here. But I also think some people just think, like, they're  
387 better on working, like, in that team and doing, like, working, like, at a job  
388 later on than doing multiple lectures. So, that's also something I really love  
389 at Catrobat, that I just, it's like a little bit of work experience, because for the  
390 last few years, I had multiple teams. I could learn a lot. I could also, yeah, try  
391 out, like, roles like the coordinator role. And it's just, yeah, you get another  
392 level of experience than you do in, like, four months of lecture. So, that's also, I  
393 think, yeah, something that people really like, and also why they also stay here.  
394 [42:33]

395 I:[42:36] Okay. Which impact does the motivation level of each team member  
396 has on the team performance? [42:43]

397 E1:[42:45] I think a lot, because right now, the team is really motivated, and they  
398 really like working together and also do pair programming a lot of times. And  
399 I see that the motivation is really high. And I see that even though, like, I don't  
400 have that much time to, like, get the people to work, they just do it themselves.  
401 And I think if I had another team right now, because I also have work and a  
402 lot of stuff to do right now, if I had a team with other members, like, I already  
403 had teams like that before, then there would not be any progress right now. But  
404 because they are, like, they have intrinsic motivation, they're still getting stuff  
405 done, even though I don't, like, remember them all the time to do something.  
406 So, yeah, I think it has a lot of weight. [43:50]

407 I:[43:53] Okay. So, the next one will be the last question. Are there any incentives  
408 to increase the motivation of participants? And if yes, please elaborate on them.  
409 And if no, why haven't any incentives been established yet, in your opinion?  
410 [44:11]

411 E1:[44:15] I'm not sure. Let me think quickly. I mean, I think not really. Like,  
412 there is nothing, like, you get when you're more motivated. I mean, other than  
413 that, you will probably finish the course earlier and get a grade quicker. And,  
414 like, not an incentive, but, like, if you don't work good, you could also get

415 hours deducted. Like, I also already had that, because there was one member  
 416 who had been working on one ticket for, I think, a year. And when we checked  
 417 what ticket it was, we saw that, yeah, it's not possible. They, like, booked a lot  
 418 of hours on that without doing anything. So, they got hours deducted. But,  
 419 yeah, I don't think there are any incentives. But I'm not sure if you need them,  
 420 because, I mean, yeah, if I think about how it is at work, I mean, yeah, you may  
 421 get more money, but also, yeah, I'm not sure if there's anything equal that we  
 422 could do at Catrobat. [45:42]  
 423 I:[45:46] Okay. So, we are through with the questions. Is there anything else you  
 424 would like to share? [45:53]  
 425 E1:[45:57] No, I don't think so. [45:59]  
 426 I:[46:00] Okay, then thank you very much for the interesting conversation and  
 427 I'll stop the recording now. [46:05]

## B.2. Interview with IDE Coordinator

1 I:[00:03] Hello, thank you for taking the time to talk to me and for your will-  
 2 ingness to conduct an interview. As already mentioned, I am conducting a  
 3 scientific survey as part of my master's thesis at the University of Technology  
 4 Graz regarding human resource management and possibilities on how to mea-  
 5 sure performance and productivity in the Catrobat project. The interview will  
 6 take about 35 to 45 minutes and will be recorded. After the evaluation, the  
 7 audio recording will be deleted for data protection reasons. You have already  
 8 given your consent to the collection and processing of your data in the form of  
 9 a signature. Thank you for this.  
 10 I will ask you some questions about your role and responsibilities in the Catro-  
 11 bat project as well as the current state of organisational processes. If there is  
 12 anything you don't want to answer, just let me know. Otherwise, I'd like to ask  
 13 you to talk as freely and informally as possible.  
 14 Everything you say is important and correct. The interview is about your experi-  
 15 ences. The interview is divided into several thematic areas.  
 16 I will always inform you when we proceed with the next topic. Are there any  
 17 uncertainties? Do you have any questions about the interview or the process?  
 18 [01:23]  
 19 E2:[01:25] No, everything seems clear to me. [01:27]  
 20 I:[01:27] Okay, then let's start with the first question. Please take as much time  
 21 as you need to answer. You are welcome to talk in detail. So, we'll start with  
 22 the topic area field of activity. And first of all, I'd like to ask you to tell me  
 23 something about your field of activity and your role in the Catrobat project.  
 24 You can take as much time as you like for this. I won't interrupt you, just make  
 25 a few notes which I will discuss later. [01:56]  
 26 E2:[01:59] Yeah, sure. I am a part of Catrobat, especially in the Catroid IDE

27 team as a so-called coordinator. For my field of activities, I am more or less the  
28 interface between our team and other teams in the Catrobat project, let's say.  
29 And more or less, I do coordinate everything that is internally in my team and  
30 discuss if there are any questions or let's say any unconsciousness about some  
31 topics in our team with other teams or with the leads of the Catrobat project.  
32 Yes, that would be more or less the overall fields of activities, let's say. [02:50]  
33 I:[02:50] Mm-hmm. And when you think about your work as team coordinator,  
34 what are your most important tasks? [02:56]  
35 E2:[02:58] Most importantly, I'd say always having some meetings. So, internally  
36 with our Catroid team, and especially talking with the scrum master of the  
37 Catroid team, and talking with other scrum masters and coordinators of other  
38 teams, trying to get like new, let's say, members if we are kinda becoming into  
39 a short team, since most team members do not stay for a very long time, so  
40 that we get new members. So, we can continue with our work, and those things  
41 would be probably the main activities, let's say. [03:42]  
42 I:[03:44] Mm-hmm. Please explain to me your tasks that take up the most time  
43 and explain why they are so time consuming. [03:51]  
44 E2:[03:53] Yeah, the most time would probably be onboarding for new team  
45 members, since this is kind of a crucial part. And it's very important to help  
46 out the new team members, so that they can, or they will be able to start as  
47 early as possible with their project, to find their way and their place in the team.  
48 On one hand, for being productive at the project, but also on the other hand,  
49 for social networking parts, getting a good feeling. They always should be open  
50 to others, so that it's not only a working environment, but also where you can  
51 find new students, new friends, and so on. So, most likely onboarding for new  
52 members, or finding new members for the team, would be, I think, the answer  
53 for this question. [05:00]  
54 I:[05:01] Okay. Please try to remember your first days as coordinator. Which  
55 challenges did you face in your new position? [05:09]  
56 E2:[05:12] Yeah. Since Catrobat is like a kind of a big project with many, many,  
57 many members, it's kind of hard to find your way on all the documentation that  
58 there exists, to understand how the default processes do work. Like let's, for  
59 example, take an easy task, like holding meetings plus writing notes for them.  
60 It's always kind of a challenge, since you are not sure at the beginning what is  
61 necessary, what is unnecessary, and which information do I have to consider,  
62 and also how to work with questions from your team, especially when you  
63 were at the beginning. You most likely are not sure how to answer a specific  
64 question, since you do not really know how already the process do work in the  
65 team or in the environment. [06:23]  
66 I:[06:27] Thank you. So, we'll continue with the next topic, which is team organ-  
67 isation. The first question, please give me some information about the structure  
68 of the team you coordinate. [06:42]  
69 E2: [06:47] For my team, it's not that big or not that complicated. We are basi-

cally a small team currently of, if I am not mistaken, eight members, including senior and junior developers, as well as a scrum master. And more or less, my focus is only on this team and we do not really have a hierarchical structure or something like that. It's more like a very low-ground structure. We are all on the same level, let's say. The only difference is that our scrum master is always holding the meetings and trying to answer questions about, let's say, for example, tickets on a specific topic. My part would also be holding meetings if he is absent, or then coordinating anything that my team wants to do, wants to have, like, kind of knowledge they would need for some tasks, and discuss them with other teams then. But internally in the team, we do not really have a big structure, just like everything is on the same level, senior and junior developers, a scrum master, and myself as the coordinator. [08:15]

**I:**[08:17] Okay. Is the amount of team members always the same, or can that change? [08:22]

**E2:**[08:23] It's not always the same, unfortunately. We sometimes have a big overload of team members. And even at that point, if some students, for example, would like to participate in our team, that we have to neglect them and say, we are really sorry, but our team is at the moment overloaded with members. Sometimes we have far too less members in our team, which then kind of holds back the productivity in our team. So it's not really always the same. At sometimes we have way too many, at sometimes we have like way too less members. It jumps around, like, between, if we exclude the scrum master and the coordinator, around four to ten members, depending on which time in the year it is. [09:19]

**I:**[09:22] For how long have the current team members already been contributing, and how long will they still be part of the team, approximately? [09:31]

**E2:**[09:32] We are currently at the moment, that three, no I mean four members are leaving the team right now. So until now, we were seven members, eight members, I'm sorry. And after that, we will only have four anymore. But those four will stay approximately now, let's say for a year. And yeah, as I said, we currently are not losing, but they are leaving since they have done their work until now, four members. So we are exactly at this point where we were like a full team, and now we have too less members. [10:12]

**I:**[10:16] Okay. Please tell me about your team meetings in detail. For example, when, how often, where, and which types of meeting do you have? [10:24]

**E2:**[10:25] I normally have like two big weekly meetings. One is the Catroid IDE, which is the team, I am the coordinator of weekly meeting. It's basically a scrum meeting, and mostly held by the scrum master on its own. But it is always a nice thing if the coordinator is there as well to get information, if there is some need to give the team information that I got from other teams. And that is always currently on Tuesday around 4 p.m. And then we also have the coordinator meeting, which is always Friday on 4 p.m., where we discuss if we, for example, need new team members. Like in my case, that would be the next

113 topic for this weekly meeting, where I would ask if there are people that would  
114 like to join our team since we lost some team members. So yeah, Thursday for  
115 the scrum meeting and Friday for the coordinator meeting, and that's always  
116 every week, unless there's a holiday or something like that. But those two are  
117 the big meetings always every week. [10:42]

118 **I:**[10:46] So please tell me about the attendance rate at the meetings. Is it difficult  
119 to get all team members together? [10:53]

120 **E2:**[10:56] I wouldn't say it's difficult. It depends on the team size. When we are  
121 like a full team, let's say with ten members or nine members, let's say, it's not  
122 that easy to get all of them together, since most of our members are students at  
123 the university. And therefore, they have other attendances as well for lectures,  
124 for example. So it's not really that easy if we have a full team. On the other  
125 hand, if our team is smaller, like currently, it's not that hard. We all currently  
126 always have a full attended meeting, especially in the holidays, where they  
127 do not have like other work to do, let's say in the university. Yeah, so at the  
128 current state, it's not that hard since we are a smaller team now, but with more  
129 members, it normally is hard to get everyone at the meeting. [13:02]

130 **I:**[13:05] How important are these regular meetings for the project progress and  
131 why? [13:10]

132 **E2:**[13:11] I would say they are really important, since if you kinda are stuck as  
133 a, let's say, junior developer in our team, you are not really well known with  
134 the structural environment of the project. You have many questions, and the  
135 meetings are mainly also here to discuss those questions so that the developer  
136 can continue with the work. Without such meetings, it would be really hard to  
137 communicate a necessary knowledge or questions so that you could continue  
138 with your workflow. So, especially for the junior members, those weekly meet-  
139 ings are really important. Otherwise, without them, we would have like a high  
140 decrease in productivity for the project overall, I would say.

141 **I:** When you think about collaboration within the team, as well as in between  
142 teams in Catrobat, please name and explain obstacles that hinder this coopera-  
143 tion, if there are any.

144 **E2:**[14:31] Yeah, in the team, there are not really any obstacles, I would say. It's  
145 more like just always a time management thing. It's hard to always coordinate  
146 two people or more for a meeting, let's say, so that they can work on a problem  
147 together. Other than that, in a team, there wouldn't be really an obstacle. Ob-  
148 stacles across teams are also a time management thing. When to meet, do we  
149 really have time for that? But also, in different teams, you have different kinds  
150 of knowledge. So, setting up a meeting with someone from another team is also  
151 kind of hard, since you are not really on a baseline with each other. Everyone  
152 has different knowledge. Everyone has different opinions, especially in other  
153 teams. So, it's kind of hard to get your point through a discussion or to get on a  
154 baseline, so that you can continue with your work. And therefore, you probably  
155 need more meetings, and even with more members of other teams, so that you

get a big round of, let's say, a big meeting with multiple people, with multiple, let's say, coordinators, or masters, or developers, doesn't really matter in this case, and come together and discuss a topic. And with more members in the meeting, the discussion becomes even harder, since with more people, you get more opinions on stuff like that. And after that, you will probably decide to take even more meetings, and therefore, again, you get another problem with the time management overall. So, in the team, it's not really that hard. There aren't really any obstacles, but across other teams, it can be kind of challenging sometimes. [16:40]

**I:**[16:43] Okay. So, we'll approach the next topic, which is the development process. So, please explain to me the process of task implementation, from moving a ticket from ready for development until to merge. [17:00]

**E2:**[17:03] That's now specified for a developer, I guess, if I understood the question. Okay. Yeah, more or less, right at the beginning, kind of, before developing a ticket, a ticket has to be created, most likely by a product owner, which this position is actually for creating tickets for a project, or for a sub-project, let's say. And this ticket, when completely ready, can then be implemented by a developer. For that, we have got a board, a so-called scrum board, and in this scrum board, there is more or less, there are several tickets made from a PO. A developer can take one of these tickets and try to solve this problem on his own, in pair programming, doesn't really matter. The ticket itself just should be implemented until sometime. After this implementation, if the developer thinks, okay, I think I got the necessary requirements done, I would say, from my point of view, it's done. The developer can move this ticket from develop on the development part, or let's say column, scrum boards are always separated in columns, into the next column, which is most likely a PR or a CR review, which basically means like a pull request, or a code request, or a code review column, where a more experienced developer, like a senior developer, or in some cases also, maybe a scrum master, or a coordinator, depends how much they are also involved in coding themselves, in coding. So the developer moves the tickets from that column to the next column, and then let's say, senior takes, sees this, and now his task is to overall look at the code that the junior or another senior did, have a look on if the all requirements were implemented or not, if there are any major flaws, and after that, if the senior thinks that everything seems to be all good, he most likely can accept this state of the code, and after that, a PO or a coordinator is then able to merge this part of the code implemented by the developer into the overall project. That would be more like, more or less be the process of a ticket. [20:03]

**I:**[20:05] So how long does it approximately take until a ticket has undergone this process from ready for development to merged? [20:14]

**E2:**[20:16] It always depends also on the size of the ticket. We for that have some kind of code priority, prioritizing a ticket with points, and a small ticket has always like around one to two or a half point, let's say, and bigger tickets

199 have more points on it, which indicates that it's a harder, a bigger ticket, and  
200 it takes more time to do it. So it depends always if it's more workload or not  
201 for developing itself, and for like smaller tickets, I would say approximately  
202 one week to two weeks until it's fully merged. For really big tickets, alone  
203 the development on itself, it would take several weeks. Checking on that code  
204 again would take more weeks, and then merging it would also take some time.  
205 Overall, for big tickets, I would say maybe around one to two months, and for  
206 small tickets, one to two weeks, depending on the code, depending on how  
207 experienced the developers are. Yeah. [21:24]

208 I:[21:27] Are there any bottlenecks in this process? If yes, how are you trying to  
209 mitigate them? [21:32]

210 E2:[21:34] Hmm - Bottlenecks. Well, now I have to think a bit. I am not really  
211 sure if there are any. The only problem that might occur is that it's always in a  
212 stalling position, let's say, for the ticket itself. With the scrum board, you have  
213 multiple tickets, so if a developer is done with a ticket or has multiple tickets  
214 and has to wait for some feedback, the developer can always take another ticket  
215 and work continue on that. And sometimes, depending on how many seniors  
216 and coordinators and POs you've got, tickets might take a while until they get  
217 from this review aspect really to the merged, merged-finished part, let's say.  
218 And with that, it's not like the process itself has bottlenecks, I would say. In my  
219 opinion, it's great. I like to use it. But depending on if you have experienced  
220 members, experienced seniors, experienced POs, experienced coordinators, and  
221 if they have time right now, or if you have even enough of them, sometimes  
222 we lag also on the experienced people in our team. It might stall a bit so that  
223 it takes a while until a task, and therefore the code for it, the implementation,  
224 the functionality of that gets actually merged into the overall project. But the  
225 system itself, I think there aren't really any kind of bottlenecks. At least that's  
226 my opinion, and I am very happy to use this system how it is. [23:47]

227 I:[23:49] Okay. So please tell me about the planning of this development process.  
228 How do you decide which tickets should be moved to ready for development?  
229 [23:57]

230 E2:[23:59] Yeah. Normally, it's not always possible to do that. Sometimes I'll  
231 do it on a small group with myself and with an old coordinator and a scrum  
232 master. But in a normal occasion, you have a so-called planning game where  
233 you discuss user stories that were given and decide with the developers. So  
234 in a bigger meeting than absence from the weekly scrum meeting, you always  
235 create a new meeting. And in this group, you discuss and decide which of these  
236 user stories would be good to implement now for the next, let's say, part or for  
237 the next sprint. You evaluate them, you decide how big the ticket actually is,  
238 since a PO most likely does not code on its own and does therefore not have  
239 that much experience in it. So the opinion of the developers is very important  
240 at that. You discuss it in a planning game where you give those priority points,  
241 those points for each ticket. And in a bigger round that can take multiple hours



if needed, you rate those tickets. And after some while, you got hopefully multiple tickets that can be moved into ready for development. So basically, there's another meeting, you discuss it in a big round with the developers. And after this discussion, you hopefully have new tickets that can be implemented in the next sprint. [25:45]

**I:**[25:48] How much information do you have on the availability of your team members in advance? For example, how many hours to plan to work the next week, or if they are available at all? [25:59]

**E2:**[26:02] Directly, I do not have any information about that. But that's, for example, one point that I always ask in the weekly meetings. And therefore, like, for example, if one member doesn't really have much time in the next period, let's say, for example, in one month, since they got many lectures and exams to participate and to learn from it, they always beforehand say, I'm sorry, but I won't have much time this month. And if I already know that it would be great to get some functionalities done for the project, I always ask in those meetings or write them a direct message to them and ask if they are available for the next time. But I do not automatically have information about it. I ask about it. And the developers on their own also tell that information in the weekly meetings. [27:02]

**I:**[27:05] Are there some kind of time critical tickets in your project? And if yes, how do you make sure that those tickets are done in time? [27:13]

**E2:**[27:15] Currently not. We had some that were time critical. And those tickets normally get prioritized by the experienced developers. So in the case that we had a while ago, I asked if some of the senior developers could take this ticket and implement it as far as possible. One of them said, no, they do not have time this month. But the other senior said, there's no problem on his side. So he would take the ticket. And with the experienced tickets, also critical ones do get finished quite fast. And if there are no seniors, or if the seniors do not have time, then hopefully a junior developer is fast enough to do it. But yeah, if you do not have the member, then it is how it is. And we have to work with that with the current state. But yeah, until now, we were quite able to even do or finish critical tasks with critical deadlines, let's say, in a short period of time with experienced developers that were until now not really a problem. [28:35]

**I:**[28:39] So we'll continue with the next topic, which is on and off boarding. When you think about recruiting new members, how does the process for filling vacancies work in detail? [28:50]

**E2:**[28:53] We more or less have like a checklist for onboarding stuff so that every new member gets an information letter. And on that they have like check boxes that they can go through, make some small tasks, check them. And with this finished list, they more or less are finished with the onboarding. At the beginning, they most likely have an interview with one of our PhD in the Institute. And after that, this PhD does decide, depending on which teams are now full, which teams do need new members, decide with the new member

285 and which team he or she wants to participate in. And after that, usually I as a  
286 coordinator would also have a new meeting with the new member, explain a  
287 bit on what we are doing, how the organisation works indirectly, how the team  
288 itself is structured and works, and what the usual tasks would be, that we have  
289 meetings. I would introduce this member to the team and we would like have a  
290 small discussion about so everyone gets each other to know a little bit better.  
291 And yeah, and at the end, as long as then this checklist sheet with these small  
292 tasks, which have very detailed information on what to do now, I don't know  
293 them right at the out of my mind. But it's just like a huge list with small tasks.  
294 And after that, with this finished checklist, the onboarding itself would then be  
295 done at the end. [30:53]

296 I:[30:56] Okay, how much influence do you have on deciding who will join the  
297 team? [31:00]

298 E2:[31:05] It depends if we have the capabilities of re-deciding that. Let's say for  
299 example, we have multiple teams and let's say for example, one team doesn't  
300 have a lot of members, like wanna say Catroid IDE now, lost 4 team members,  
301 and we would need more team members so that we could have some help on  
302 our side. And all the other teams would be full, then my opinion wouldn't  
303 really be that important since we just have those empty spaces with the new  
304 members. And but if the case is that we have multiple teams with enough slots,  
305 I usually can bring up some points to convince that we would like to have this  
306 new member.

307 But that's only if the size of every team is more or less equal. Otherwise, the  
308 member will get into one of the teams that would need new members, which  
309 currently would be our Catroid IDE team. [32:27]

310 I:[32:30] Can you think of one or more situations where the lack of new team  
311 members led to difficult conditions for the project progress? And if yes, please  
312 tell me about it. [32:40]

313 E2:[32:42] Since I am a new Catroid IDE coordinator, the aspect of the coordina-  
314 tor, no. I, some time ago, worked as a developer on its own for the team. And  
315 even there, I didn't have the, let's say, the view, I guess, that we would be on a  
316 critical part since we didn't have any. So like, not really. I couldn't say yes on  
317 that question. [33:22]

318 I:[33:24] Then please tell me about your experiences regarding the offboarding  
319 process of team members. [33:28]

320 E2:[33:30] Yeah. Offboarding is kind of another, let's say, issue depending on the  
321 member. Most members normally have a quite easy offboarding where more  
322 likely they just inform us what they did, which tickets they did. And the most  
323 critical part or important part is that they tell us what they were currently still  
324 doing. And if they could, in a meeting, explain to me or to, let's say, to a senior  
325 developer what they did for a specified ticket so that we are able to finish the  
326 work of the member that's leaving. Other than that, there are no really big  
327 offboarding criterias on our side, just like what they did. We have also like a

time sheet that tracks the working hours of the team member. We will have a look on that always as well at the offboarding, but there are no really big steps for offboarding. [34:45]

I:[34:48] Please elaborate on the knowledge transfer from experienced members, which leave the project to new members. [34:54]

E2:[34:56] Yeah, they most likely... I'm not sure how other teams are working on that, but at our team, we more or less have always like a mentor positioning, which senior developers have. And when a new member joins our team, we typically assign one of the seniors as a mentor for a junior developer for a new member. And they most likely then work at the simpler tickets together so that the new member gets kind of a taste how the project works, how to work on tickets and so on and so forth. So we basically have a mentor and that mentor works with the junior so that this knowledge gets spread very well. If there are some information that the senior doesn't know, then information would also be spread throughout the weekly meetings. [35:56]

I:[35:59] I'd like to speed forward to the moment when you have finished a few hours as a coordinator. Please tell me how you imagined a transition to your successor. [36:09]

E2:[36:13] I'm not... Could you specify the question a bit more? [31:16]

I:[36:17] Yes. So when you are finished with being a coordinator, you will get another one who will do your job then. [36:25]

E2:[36:25] Exactly. [36:25]

I:[36:26] And what do you need to do that the next one is capable of doing your position? So what do you need to teach him or her? [36:37]

E2:[36:38] Okay. Yeah, there are some parts. As I said, you can't... At the beginning, you can't really teach the new coordinator everything. At the moment, it's always also a part of learning by doing. Like, for example, how to discuss information with other team members, how to get information that you need. Overall, I would go through with the new coordinator through our confluence pages, which pages are relevant, how and when to write meeting notes, when the meetings are, how to discuss topics with other coordinators. And there are also some notes in our confluence, which member is for which... Or which member of the Catrobat project is responsible for this team, which is responsible for that team, how the more or less hierarchy works in the project, so that if information is needed, where the new coordinator could ask for this information, how to hold the meetings, I would probably invite the coordinator in my last few meetings, so that he or she gets a taste of how to do a meeting. I would introduce them into our, let's say, into our internal team. And yeah, it would take some time since the project is huge, since our info, our letters, our information in the confluence pages are big. But yeah, that would more or less, I guess, be our end and also the how onboarding would work, since that's also a crucial part. But that would be more likely the introduction for the successor of the team. [38:49]

371 I:[38:50] Okay, so on to the next topic, which is performance measures. What  
372 do you believe are the key factors that have contributed most significantly to  
373 the achievements of the Catrobat project and why? [39:05]

374 E2:[39:06] I'm sorry, I had internet issues. Could you repeat the question, please?  
375 [39:10]

376 I:[39:10] Yeah, sure. What do you believe are the key factors that have con-  
377 tributed most significantly to the achievements of the Catrobat project and why?  
378 [39:20]

379 E2:[39:21] Hmm, that's a good question. That's a good question. At first, I think,  
380 since Catrobat is an NGO, we have many members, many developers around  
381 the world. But a big part of that is our students from the Technical University  
382 of Graz. And since the education that you get from this institute is pretty high,  
383 I would say, in my opinion, we always get members that are highly, that are  
384 well educated, that know about their skills that they've learned in university.  
385 And therefore, we always get good team members that can start from a very  
386 high level on coding, on understanding what problems we have, et cetera, et  
387 cetera. So that is one big part. Another big part is the leading, or let's say the  
388 PhDs, the professor that also work for Catrobat, which are leading this project  
389 overall, especially the PhD students, are very engaged into every problem that  
390 we get. They make up a lot of work on their own. They are highly, I don't want  
391 to say workaholics on that, but if there exists a problem, they at first are the  
392 ones that try to solve it. And with such educated and hardworking people, you  
393 will always get a good result at the end. So it's more or less a good combination  
394 of good people that are well educated, that work really hard, and that work  
395 together really, really well in the teams themselves or across the teams, I would  
396 say. And this combination leads to a high success at the end. [42:07]

397 I:[42:09] Okay. Thank you. So please elaborate on the options and measures  
398 you have to evaluate the contribution and performance of the individual team  
399 members. [42:22]

400 E2:[42:28] More or less, we do not have a tool per se that we can use on that. We  
401 track it by our already explained timesheet. We track it by the pull requests that  
402 the member has, since the pull requests are linked to their name. And we also  
403 ask then other team members, especially those who had the pair programming  
404 sessions with them, or let's say if a senior is going to leave, I would ask the  
405 juniors who were mentored by them, how was your mentoring with him? Did  
406 you learn a lot of stuff? And we would always base it on the information we  
407 would get from other team members. Yes. And that would be more like those  
408 three points would give us a pretty good view on how the member worked at  
409 his or her time for our team. [43:42]

410 I:[43:45] Do you feel that you need to have a better overview and understanding  
411 of the performance of your team? And if yes, what would you need for gaining  
412 this overview? [43:54]

413 E2:[43:58] Overall, I would say no. The thing is in every team and in every

project, you always have one or two members that are not really contributing that much as they should. That's always in the life of working with others that might happen one, two, three times in a period span. So overall, I would say no. For those special occasions, somehow it would be great, but I couldn't give you really an answer on how you could check that. Maybe with more pair programming, maybe one-on-one code reviews or stuff like that, or multiple reminders that they have to do more, or what they're doing is not the correct way, but I couldn't give you a straight answer on that. [44:53]

I:[44:55] Okay. So team members which participate in Catrobat as part of a university course agree to spend at least 24 hours per month working on the project. Why has this threshold been introduced? [45:09]

E2:[45:13] I think the base idea was just like, you are right, there are 210 hours, and I think that is based on the overall amount of ECTS that you're getting. Since ECTS are based on, I don't know how the EU came up with that, but one ECTS is approximately 25 to 30 hours, I guess. I'm not completely sure about that. The multiplication between the amount of ECTS, which for an equivalent of a Bachelor thesis is around, I think, seven ECTS, and that times the 25 or 30 hours would come up to this, more likely a very simple calculation for that. And I think it's, from my point of view, it's not that bad of an amount, since far more less would not really be productive for our side, since you always have to give these members some time to integrate themselves in the project. That always takes some time, and we wouldn't really give the students more hours for that thing that they would get, so we can't give these students, let's say, three ECTS, which would be around 100 hours worth, and at the same time, one from them, 500 hours for the project, that wouldn't be fair for them, and those 210 hours are more than enough to get a team member that is willing to work and works at a productive way, so that also we have something out of it. [47:20]

I:[47:25] So what is your impact on the grade of the team member leaving? [47:29]

E2:[47:33] More or less, I am the one that grades the members directly, so in our system, it works kind of like that, that I would get the information I want to get, as I already said, from the timesheet, from the pull requests, from the merged tickets that the member did, and from asking the other team members if they contributed well and what their experience with the member was, and based on all this information that I get, I would decide on which grade the student would get. I would then forward this information to a PO or to a lead from the project, so in this case, a PhD student, and this PhD student would then overlook it again, not really sure how this process works, and if he's fine with my result, he would then grade the member, and otherwise, we would have a meeting and discuss it on our own then, but my impact is very, very high on the grade of the student. [48:47]

I:[48:51] Okay, so we're going to the last topic, which is team member motiva-

457 tion, and the first question is, what do you think, what motivates students to  
458 contribute to the Catrobat project? [49:07]

459 E2:[49:10] The first point and the most obvious point, I would say, is their grade,  
460 since they are doing it as part of the university or of their academic career,  
461 and just like for every other lecture, they want to get the best grade that they  
462 could possibly get with the amount of work they do, and with that, they also  
463 contribute quite well for the Catrobat project, but what we are also doing are,  
464 let's say, we are doing some events or some workshops to motivate those team  
465 members, which they can participate at, so they can get even more knowledge,  
466 not only code in the project, but get more knowledge that might be helpful  
467 for their later work life in the industry, and workshops, some events that are  
468 like team building, every year we have like a Christmas event or a kind of new  
469 year event, let's say, where we can meet up together, where we sponsor food,  
470 music, where we can meet up, where we can network with each other, so that  
471 the overall motivation in the group and the dynamic in the group gets pushed  
472 a little bit, that we get to know each other a bit more, and therefore, at least in  
473 our experience, therefore the team members contribute a bit more on that side,  
474 yeah, but the baseline or the base motivation is for themselves to just get a good  
475 grade, and we are helping that out with some cool events, some networking  
476 events and stuff like that. [50:57]

477 I:[51:00] Okay, so which impact does the motivation level of each team member  
478 have on the team performance? [51:06]

479 E2:[51:10] I think it has a high impact, since I think we all at some point had  
480 team members that were not motivated at all at some lecture in our university  
481 life, and everyone was able to see that even if one team member is like highly,  
482 not say upset, but not motivated at all, it always decreases the motivation of  
483 the whole other team, therefore also decreases performance and productivity  
484 of the team, so getting everyone motivated as much as possible is a crucial part  
485 for the whole dynamic and the motivation of the whole team. [51:58]

486 I:[51:59] Nice, so we are at the last question, but you more or less gave already  
487 an answer to it, I will still read it to you, if you want to add something, feel free  
488 to do it. Please tell me about incentives to increase the motivation of participants  
489 in the Catrobat project. [52:18]

490 E2:[52:20] Incentives, yeah, but as I already said, we do not have to do much  
491 about it, they already are very motivated to do their stuff, since they get a  
492 grade, and therefore they want to do it on their selves, but as I already told, we  
493 are trying our best to motivate them even more with events, with some, and  
494 let's not say goodies, but yeah, some benefits that we can give them in form  
495 of events, in form of, I don't know, yeah, it's more or less always those events  
496 that we are trying to do, and those workshops also, they even get something  
497 for their later on career, and don't just think this part is just a small step or a  
498 small milestone in life, and after that I will forget about it, so these events and  
499 workshops are most likely the parts that we can do on our side. [53:30]

500 I:[53:33] Okay, great, so we are through with the questions, is there anything  
501 else you would like to share? [53:39]  
502 E2:[53:41] No, not really, I hope I was able to answer the questions for you.  
503 [53:46]  
504 I:[53:47] Yeah, for sure, so thank you very much for the interesting conversation  
505 and for your time, I'd stop the recording now. [53:54]  
506 E2:[53:55] Okay. [53:56]

### B.3. Interview with Stage Coordinator

1 I:[0:00] Hello, thank you for taking the time to talk to me and for your will-  
2 ingness to conduct an interview. As already mentioned, I am conducting a  
3 scientific survey as part of my master's thesis at the University of Technology  
4 Graz regarding human resource management and possibilities on how to mea-  
5 sure performance and productivity in the Catrobat project. The interview will  
6 take about 35 to 45 minutes and will be recorded. After the evaluation, the  
7 audio recording will be deleted for data protection reasons. You have already  
8 given your consent to the collection and processing of your data in the form of  
9 a signature. Thank you for this.  
10 I will ask you some questions about your role and responsibilities in the Catro-  
11 bat project as well as the current state of organisational processes. If there is  
12 anything you don't want to answer, just let me know. Otherwise, I'd like to ask  
13 you to talk as freely and informally as possible.  
14 Everything you say is important and correct. The interview is about your experi-  
15 ences. The interview is divided into several thematic areas.  
16 I will always inform you when we proceed with the next topic. Are there any  
17 uncertainties? Do you have any questions about the interview or the process?  
18 [01:24]  
19 E3:[01:26] No, I don't. [01:26]  
20 I:[01:27] Okay, then let's start with the first question. So, it's the first topic area,  
21 which is field of activity. And first of all, I'd like to ask you to tell me something  
22 about your field of activity and your role in the Catrobat project. You can take  
23 as much time as you like for this. I won't interrupt you. [01:50]  
24 E3:[01:52] Okay, so I'm the coordinator of the Stage team at Catrobat. And  
25 my main responsibility is coordinating the team. We have split up the work  
26 because our team also has a scrum master. And we kind of made the agreement  
27 that the scrum master is more responsible for interpersonal things. So, she's  
28 mostly working on the people side of things, to say that. And I'm working  
29 on the process side of things. So, I'm looking through the board. If there's  
30 any technical issues that team members have, then I'm usually handling those.  
31 So, what happens sometimes is that people, especially in the beginning when  
32 they're working on tickets, they run into a dead end. Or dealing with things

33 that they are kind of struggling with, but they might need help just to get to the  
34 right direction. Or to get some, how should I say, some data, some info about  
35 the field, about technical issues. And there I'm working on those. Yes, I'm also  
36 doing a little bit of coding on the side, but mostly my work is coordination.  
37 Since I've also been working on the Catroid, especially on the Stage code base  
38 for a longer period of time. I'm also often asked by members of other teams for  
39 assessments or for some expertise. For instance, when it comes to new tickets or  
40 new features. Then I'm asked how I would estimate certain aspects, the effort,  
41 or if it's doable or not. And then also at the end when team members leave, then  
42 I'm involved in grading. So, then I usually have a look at tickets they've done,  
43 how their participation was, and then kind of give my suggestion for a grade.  
44 What else have I not mentioned yet? Tickets. Yeah, when we're looking at the  
45 board for tickets, sometimes I'm creating new ones. Just to have training tickets  
46 or others. I'm also code reviewing. Most of the time, that's product owner code  
47 reviews. Sort of ones that are a little bit more intricate, a little bit more, how  
48 should I say, a little bit more careful. And when it comes to creating tickets,  
49 also looking a little bit at what things need to be done. Because if you work at  
50 the code base, if you have a look at other people's tickets, you kind of get a  
51 feeling on where work needs to be done. And that often results in some ideas,  
52 some suggestions for the future that could be done. I think that should be most.  
53 If not, maybe some things will come up during further questions. [05:18]  
54 I:[05:19] Mm-hmm. For sure. Okay. Second question. When you think about  
55 your work as team coordinator, what are your most important tasks? [05:28]  
56 E3:[05:32] Most important tasks are having a rough overview of where tickets  
57 are. And giving support to team members, especially when I see that sometimes  
58 tickets end up taking longer. So that we kind of like, that we see impediments  
59 and then discuss them during team meetings and try to remove them. [06:06]  
60 I:[06:10] Okay. So please explain to me your tasks that take up the most time  
61 and explain why they are so time consuming. [06:18]  
62 E3:[06:20] Hmm. Let me think. What takes the most time? Let's see. Past month,  
63 a lot of the time, clearing issues on tickets. Sometimes because I also find some  
64 other problems on the way that aren't related to that particular ticket. I mean,  
65 not directly related. Sometimes when we're working on things, we find other  
66 problems that should be handled. Often that leads to a lot of follow-up effort.  
67 Also, when it comes to then discussing those issues and evaluating whether  
68 or not how important they are. How much effort it would probably be. And  
69 because right now we're working on, for instance, on some bugs that we found  
70 at the release candidate. And for instance, that we saw there was a problem  
71 with just a little brick that is for the Arduino drones. And there we saw during  
72 code review, for instance, that there appears to be a general problem with  
73 Bluetooth for the new Android versions. That maybe also affects a little bit the  
74 question for what my responsibilities are. Very often I'm also then looking at  
75 what, if tickets run, if features run on different emulators and what the newest



requirements are for different Android versions. For instance, there is one ticket we saw that there's problems with, there might be problems with Bluetooth in general. That we might need to investigate. And those are the things that I'm also looking at as I kind of consider myself also one of the more technical responsible people for that area. That's not often directly related to coordinating the team members itself, but also it affects their work in the future. That was one thing that comes to my mind. And another thing, sometimes coordinating and then looking at tickets that people have, kind of evaluating where they are. Very often I can talk to people. And that also can sometimes take more time if the team is bigger. Recently the team is pretty small. We're three people that are coding that are working. So this part of work takes less time now. But before that we had a bigger team. And sometimes it was a little bit more time consuming to get an overview of the tickets. [09:57]

I:[10:01] Okay. Then, please try to remember your first days as coordinator. Which challenges did you face in your new position? [10:10]

E3:[10:14] The first days as coordinator, okay. I remember that I had quite good help because at that time both the Stage and the IDE team got new coordinators. So it was me for the Stage team and Maria for IDE team. And we both got a quite good handover done by Julia. She kind of gave us a meeting where she told about her past experiences of that. So I didn't have to find all the different responsibilities and tasks that I have in the future. I didn't have to find it out myself. I got a quite good list of responsibilities and that made things a little bit easier. I remember one of the first things that I did was to get this overview of tickets and team members that I wanted. Because we saw that, for instance, that a lot of the team members that we thought we had on the team page were no longer active. So one of the first things that we did was to find out, to get an overview of who was still active in a team. And then to redistribute tickets accordingly. So that led to a lot of tickets that we moved around to other people that we then moved back to ready for development so that they're available for new members. That helped a lot. And that took some time. It didn't take too long, but it still was a good chunk of work in the beginning to start with a clean slate, so to say. To have a good start where we have a good overview of who's in the team and who's working on what things. Yeah. [12:13]

I:[12:16] Okay. So then we'll move to the next topic area, which is team organisation. So please give me some information about the structure of the team you coordinate. [12:28]

E3:[12:31] The structure is pretty shallow, pretty flat. So we have me as the coordinator, we have a Scrum Master, Julia. And then we only have the team members, essentially, and we don't have a strong hierarchy there. I mean, the least little bit of hierarchy that we have is that we know that some team members are new because they've been onboarded recently. So those are usually the ones that will be put together with a more experienced member. But other than that, we have just those two, let's say, more people more involved with organizing

119 and coordinating the team. And the rest of the people are developing. And we  
120 also have a senior who's more involved in also giving feedback and help and  
121 maybe mentoring people. But other than that, that's the structure of the team.  
122 Two levels, I would say. [13:31]

123 I:[13:33] Okay. How many team members does the team have? [13:36]

124 E3:[13:37] Currently, we are four people, also counting our Scrum Master, Julia.  
125 So we have me, Julia, then we have Jakob, one of our more experienced team  
126 members and also a senior, and then a newly onboarded member. [13:55]

127 I:[14:00] Okay. For how long have the team members already been contributing?  
128 And how long will they still be part of the team? [14:07]

129 E3:[14:13] I can give rough estimates if that's fine. I mean, for me, I know I  
130 started in 2020. And I'll probably be contributing for some more time because I  
131 still have a couple of things that I want to get done. If I don't have any ECTS  
132 credits anymore, I'll be thinking about maybe contributing just on the side.  
133 Because I feel like this project is important and I still need to get a few things  
134 done, a few things. And Julia has joined, I think, a year or two years later. Jakob  
135 has, I think he's also been around for a long time. But before that, I think I  
136 even onboarded and mentored him a year after I joined. He will stay longer in  
137 the team because he's also working on a master's thesis. So he'll be available.  
138 He'll maybe shift focus more towards his master's thesis. But he said recently  
139 that he's still available in the team for code reviews and other expertise. And  
140 our newest team member joined two months ago, two, three months ago. And  
141 judging by the hours that he's done, he's doing his bachelor's. So he'll probably  
142 contribute until end of summer, end of summer, maybe in the fall. But he also  
143 gave us like a hint that he might be joining, might be contributing more after  
144 he's done with his bachelor's thesis. So we're kind of hoping, because he's very  
145 motivated, that he'll stay longer than that. That he'll stay for a year, year and a  
146 half. [15:58]

147 I:[16:02] Please tell me about your team meetings in detail. For example, when,  
148 how often, where, which types of meeting and so on. [16:09]

149 E3:[16:10] We have a team meeting once a week. That is just a weekly status  
150 update meeting where we kind of have a really, it's not a long meeting. It's  
151 usually 15 to 30 minutes at most, depending on how many news there are. And  
152 yeah, do you need also what we talk about in these meetings or just the kinds  
153 of meetings that we have? [16:37]

154 I:[16:39] You can tell me some information about the content, but you don't  
155 have as you want. [16:43]

156 E3:[16:46] Okay, I mean, maybe it's important for the master's thesis. So I'll just  
157 give you that information, I guess. We start off usually pretty easy because we  
158 think that a little bit of small talk, that's good for team building. Because we  
159 don't feel, we feel like actually a group of humans that are working together.  
160 That are related to each other in that kind of work. So we usually start off pretty  
161 easy just by talking about a little bit of things. Then usually we just go around

each one telling what he's been working on. And during that time, we often also mention issues. If somebody has problems with some of the tickets, we don't discuss them in detail, but we just acknowledge that they are there. And then usually have follow-up discussions and meetings individually. It's also a good place to start distributing information. Especially if Julia and I, we get information from product owners or from other meetings that are relevant for our team members. Then that's the place where we give that information to our team members. And it's also good to get a, there's a protocol, but it's also good if you're at a meeting to get a good, to have a good short impression on where the team is at right now. If somebody's working on the same tickets for weeks and even longer, then you hear that pretty easily if you're at the meetings regularly. think that's most, yeah. [18:26]

I:[18:28] Okay. Please tell me about the attendance rate at the meetings. Is it difficult to get all team members together? [18:35]

E3:[18:43] Sometimes we have some team members, some team members who are more often absent than others. But I think we have a quite good attendance rate. Sometimes we feel the need to mention to people that taking part in the meetings is mandatory. Sometimes we feel like that, how should I say? It's not the seriousness of the meeting, but the importance of the meeting is not always acknowledged by everyone the same. So there we sometimes mention, hey, people, we need to have you at the meetings. It's good for information exchange. So we had a couple of people who were missing every second time or something, but it's often also related to work. So that's when we try to find another slot for a meeting. But I would say that in the past, judging, just eyeballing it in the past, that two thirds to three fourths of all team members were there regularly. By regularly, I mean almost all the time. [20:00]

I:[20:04] Then I'll continue with the next question. When you think about collaboration within the team, as well as in between teams in Catrobat, please name and explain obstacles that hinder this cooperation, if there are any. [20:19]

E3:[20:27] We had this discussion a couple of times that I personally think that in the past it was a little bit easier with collaboration where people were still at university a lot of the time. I joined during COVID times, so I'm kind of used to this, but I know from people in the past that it was easier to find, for instance, pair programming partners and work together when you were used to being at university on campus. Because there if you meet people face to face and not just in meetings, then the kind of relationship that you have to those people is different. And also, I guess people feel a little bit more motivated. So we have been trying to get people back on site. We're also trying to get people more into meetings. And that is within the team. And between teams, one thing that I think is unless people know people from other teams personally, so for context, in this I'm mostly thinking about collaboration between IDE and Stage team. Unless people from my team know people from the IDE team, collaboration is not that good. Usually, the conversation goes over me. If we need something

205 from the IDE team, we try to do team days, joint team days between both teams  
206 so people get to know each other and so we can work more together. But work  
207 is usually pretty much separated between those teams. So we don't have those  
208 many points during work where we touch. Whether or not that's intended,  
209 because then people can specialize on the corresponding code-based parts. But  
210 yes. [23:00]

211 I:[23:04] Well, then we can move on to the next topic area, which is the develop-  
212 ment process. Please explain to me the process of task implementation from  
213 moving a ticket from ready for development to merged. [23:21]

214 E3:[23:24] Okay. I'm assuming that if tickets are kind of ready to be taken by a  
215 team member, that things like estimation, if necessary, have already been done.  
216 And that if the ticket's unclear, that this has also been dealt with. So often in  
217 meetings, people ask me which tickets they can take. Everybody has access to  
218 the board, but sometimes they still ask what they're working on or what they  
219 should be working on. Which kind of makes sense because then I can, if it's  
220 necessary for prioritizing things, then I can ask them to take up certain tickets.  
221 Usually they work on the tickets. I would say most of the time they work on  
222 it alone. So there should be a little bit more pair programming. We often had  
223 friends who came to the team together. They did more pair programming, but  
224 often work's done individually. I would say, depending on the difficulty of  
225 the ticket, if it's a really easy ticket and they have a lot of time, then they are  
226 done in a week. But usually it takes longer. And then the ticket comes into the  
227 first review cycle. And then after that's done, the second review cycle. And  
228 we often found that those reviews are sometimes a bottleneck. So people end  
229 up working on, well, when you put one ticket in the review cycle, they are  
230 working on another. So we often tried to get more reviews done to get those  
231 tickets finally merged. But usually, yeah, that's the rough development cycle.  
232 We have development, then first and second review cycle. Usually only minor  
233 adaptations have to be done unless we find bigger problems. But usually that's  
234 not the case. And then tickets get merged. [25:26]

235 I:[25:30] And how long does it approximately take until a ticket has undergone  
236 this process from ready for development to merged? [25:39]

237 E3:[25:50] If it's a simple beginner ticket, it's often done in one, two weeks, three  
238 at most, I would say. Because there we can do code reviews pretty easily. And  
239 other tickets, we often have more complex tickets. And those, especially if we  
240 find problems during review. And usually it also takes longer to review those  
241 tickets to make sure that things really work. And that can take, yeah, we're  
242 talking about months sometimes. But trying to give you a number, two, maybe  
243 three, I'd have to look that up, to be honest. But sometimes just from my feeling,  
244 it's in the range of months. [26:59]

245 I:[27:02] That's fine for me. Please tell me about the planning of this develop-  
246 ment process. How do you decide which tickets should be moved to ready for  
247 development? [27:14]

E3:[27:22] When I'm looking at new tickets that we have on the board, I do that on a pretty much regular basis every couple of weeks. Especially if people need new tickets, I usually look at the description of those tickets. Maybe as part of that work, I often try to reproduce the error and see what the effects of the problem are. And there I can implicitly make a little prioritisation. So those tickets that are important, they end up in ready for development earlier. But there we essentially have this big pool of tickets of various degrees of difficulty. We haven't had a planning game for a long time because most of the tickets that we're getting are our maintenance work and bugs. So most little to no features. And if there are any, they are blocked by other tickets that people have been working on. But yeah, so in this pool for ready for development, we have the tickets. And there people can just take new tickets. Unless, of course, as mentioned before, we have discussions in meetings and people ask what things are important. Which things should they be working on next. And that's where I, again, have influence over kinda steering things in the right direction. For instance, if we have now bug tickets for the release, I can distribute them amongst people faster. But yeah, that's the planning that we do. And the rest is mostly letting people work on their tickets. I would say that's more the agile way. A little bit less planning, a little bit. But yeah, we still try to get, still have to have a little, try to have a little control over the process by knowing where tickets are. And who's working on how many tickets and how many tickets are currently in code review. To know where the bottlenecks are and try to work against those. [29:30]

I:[29:33] How much information do you have on the availability of your team members in advance? For example, how many hours do you plan to work the next week or if they are available at all? [29:45]

E3:[29:49] Usually very little. I know how much they should be contributing each month. But usually I have little information about that, about how many hours, for instance, they will contribute in the next week, in the next two weeks, in the next three weeks. If I see that people haven't been contributing, I usually bring them up in meetings and try to ask them if they've been caught up in work or something else. And then every now and then in meetings we ask the question when people will be available to contribute so that I can get a rough estimate. But I have to gather information myself by talking to people. We don't have a process for that. [30:36]

I: [30:40] Are there some kind of time critical tickets in your project? And if yes, how do you make sure that those tickets are done in time? [30:47]

E3: [30:51] Yes, there are time critical, as I said. Usually they are important for releases. And I monitor them. I have them in my notes. I write down those tickets so that I can bring them up during meetings more often. And by bringing up, I mean asking for status. I'm asking people to code review those tickets. Yes. [31:20]

I:[31:23] Okay. So on to the next topic, which is on and off boarding. When you

291 think about recruiting new members, how does the process for filling those  
292 vacancies work in detail? [31:37]

293 E3: [31:41] Do you mean really the recruitment? I mean, when we recruit new  
294 members, how we get them or the onboarding process, how we get them into  
295 the group. [31:51]

296 I:[31:51] So if you know that you need a new member, how do you make sure  
297 that you get a new member? [31:56]

298 E3:[31:58] If we need new members, we bring them up during the bi-weekly  
299 coordinator and product owner meetings. Yeah, we have that. I mean, most  
300 of the time the team was sufficiently big. And in my opinion, it would have  
301 been more difficult to have a bigger team in terms of coordination effort. But  
302 yeah, in the last couple of months, we saw the team was getting smaller and  
303 smaller. So yeah, we brought it up in a COPO meeting and requested new  
304 members, essentially. We also have a document on Confluence where we can  
305 essentially request new members and then it's up to somebody else higher up  
306 to essentially distribute new members that have joined to the teams and then  
307 the onboarding starts. [32:57]

308 I:[33:02] And how much influence do you have on deciding who will join the  
309 team? [33:06]

310 E3:[33:11] A little bit. I can define requirements in that mentioned documents. I  
311 can define skills that are necessary to join our team and also nice to have. So  
312 for us, that's essentially experience in Java, Kotlin, a couple of programming  
313 languages. And it's nice to have if you have experience with gaming engines  
314 like libGDX and other things, that's helpful. But that's the influence I have over  
315 deciding who joins our team. Other than that, I cannot think of anything else.  
316 [33:50]

317 I:[33:52] Okay. Can you think of one or more situations where the lack of new  
318 team members led to difficult conditions for the project progress? If yes, please  
319 tell me about it. [34:03]

320 E3:[34:05] Yes. So that's pretty recent. Those were the last couple of months  
321 where we saw that many team members left, especially now with the smaller  
322 bachelor's thesis. People usually stay for 200 hours and that's enough to get  
323 a proper onboarding and then to work on a couple of tickets. So we have a  
324 quite high turnover. And if many of the team members leave at the same time  
325 and we can't motivate them to take additional courses at Catrobat, then the  
326 team can shrink pretty rapidly. And we had that in last winter semester when  
327 we suddenly were only left with two or three team members, me including.  
328 And then we saw that if you have less members, you also have less people for  
329 code review. So we saw that many of the tickets that were kind of already done  
330 were stuck in code review. And we're still trying to get the time to get work on  
331 those code reviews and get those tickets merged. Because with three people,  
332 especially with... Our third member, Dominik, doesn't do a lot of reviews yet,  
333 but right now it's just me and my senior. So we're working on code reviews

and two people, that's too little for code reviews to get a lot of tickets merged.

[35:39]

I:[35:43] I see. So please tell me about your experiences regarding the off-boarding process of team members. [35:49]

E3:[35:53] I cannot say anything particular about. . . I mean, because I'm usually not involved in off-boarding. That's something that our scrum master Julia does. She's then talking to people, getting feedback from them. I'm more involved when it comes to grading them. That's usually when I know that there's an off-boarding process. That's when I look at the tickets and see what has been finished, what hasn't been finished to redistribute the tickets or to maybe ask them to finish some things. That's the most involved I'm in off-boarding. And I felt that this process was not very formal in the past. Like I mentioned a couple of questions earlier, when I started as a coordinator, we saw that many people that were in the team on paper have been gone for many weeks and even months. And then we had to reach out to them and ask them if they were still contributing. So that off-boarding seemed sometimes pretty informal. People just left, got a grade, and then nobody knew what they were up to. [37:13]

I:[37:18] Please elaborate on the knowledge transfer from experienced members, which leave the project to new members. [37:24]

E3:[37:29] We do that in the form of two things. If people work on more complicated things, then I ask people to write documentation on our Confluence page to have a little bit of knowledge transfer, especially when it comes to a little bit more complicated things. For instance, we had one team member who worked on the internals on performance. So he gained quite some experience with the game engine that we're using with libGDX. And there he wrote some documentation on which features can be used on how things work like on a high abstraction. So we have some documentation. Most of the documentation is the code anyway, which is kind of our way of thinking about clean code and things. But on a higher level of abstraction to get a rough overview of things, we try to maintain a little bit of documentation. And that's one way of knowledge transfer. And the other is that when we have new team members, we ask more experienced team members to mentor them for the first couple of tickets. That's when a lot of knowledge transfer can happen. Because when they're working on a ticket together, usually... I mean, I'm also mentoring. And when I do that, I usually tell people not just about the tickets we're working on right now, but we also then talk about other things and how we do things, the way we do things, how we do code reviews and other things like that. And that's when a lot of knowledge transfer happens. Also on the technical side of things, when it comes to programming, best practices, do's and don'ts, we try to pass them on during mentoring for the first couple of tickets. [39:25]

I:[39:28] Thank you. So now I would like to speed forward to the moment when you are finished with your hours as coordinator. Please tell me how you imagine the onboarding process for your successor. [39:42]

377 E3:[39:50] If I have a new candidate for a successor, I think the onboarding that I  
378 got from Julia, I still have those notes lying around on my laptop. I would start  
379 off with that as a rough skeleton of responsibilities. And would that then add  
380 some of my personal experiences and notes to that. I had a couple of meetings  
381 where I talked to the new coordinator about things that are supposed to be  
382 done, on what responsibilities he or she will have. And then I think it would  
383 be a good idea to maybe have them contributing as a coordinator a little bit in  
384 parallel, a little bit like a mentoring process as a coordinator. Because I think  
385 that if we have this really strong cut in the middle where I just leave and then  
386 on the next day, he or she is fully responsible for everything. I think that's a  
387 little bit too much. I think there should be an overlap where both of us are still  
388 in the team and we slowly give responsibilities to the new coordinator. [41:18]  
389 I:[41:22] And what difficulties could arise during this transition? Can you think  
390 of anything? [41:29]  
391 E3:[41:30] Yeah, pretty much the thing I mentioned, if you have this very strong  
392 cut. Because when I took over from Adna, she was available for some time, but  
393 then she wasn't available anymore. I luckily got many of my questions asked  
394 by another coordinator. But yes, if the old coordinator just leaves and the new  
395 coordinator has to work on everything, usually if you really start working as  
396 a coordinator, on paper things are clear. But if you're then working on things,  
397 questions usually arise. And that's when it's helpful that the old coordinator is  
398 still available. So you can reach out to him or her and get information about,  
399 it's about getting guidance essentially. [42:24]  
400 I:[42:29] I see. Okay. Then we are coming to the next topic, which are per-  
401 formance measures. So what do you believe are the key factors that have  
402 contributed most significantly to the achievements of the Catrobat project and  
403 why? [42:47]  
404 E3:[42:50] Can you repeat that question one more time, please? [42:52]  
405 I:[42:53] Yes. What do you believe are the key factors that have contributed most  
406 significantly to the achievements of the Catrobat project and why? [43:02]  
407 E3:[43:03] The key factors that have contributed most to achieving, okay. Well, I  
408 think one thing is that I felt that if we had some really, sometimes every now  
409 and then we get really, really motivated people who are really interested and  
410 you can often see that in the first couple of meetings when people are not  
411 just sitting in the meetings and doing what's necessary and what's required  
412 from them, but then really contributing with ideas and asking often critical  
413 questions, why we're doing things the way we do them. And if we have really  
414 important, really interested team members, really important team members,  
415 we saw two things. We saw that they were dragging other people along. So  
416 they were motivating other people. And when you see that there's movement  
417 within the team that we're gaining momentum, usually people pick up on that  
418 and that helps the whole team a lot. And it also helps in terms of contribution  
419 because those people often get a lot of things done. So getting the right and



really motivated people, that helped a lot. That's one thing that comes to my mind. Just give me a couple of seconds. Maybe I can think of something else. [44:48]

I:[44:49] Just take your time. And if that's everything, it's also fine. [44:58]

E3:[45:33] One thing also is that, especially if people higher up in the organisation, product owners and essentially the bosses, the people responsible, if they come to meetings, if they tell us about news, if they tell us like visions for the next middle to long-term plans for the product for the next couple of months, that also helps to motivate people to work on things because then you have a goal in mind, what you're working towards. If it's just me and my team working amongst each other, then I see often that we need more conscious efforts to really keep a certain velocity in the team. Wait a couple of seconds. I have to plug in my laptop. I just got the notification that battery's low, but you can ask the next question right away. [46:48]

I:[46:49] Okay. [46:49]

E3:[46:50] That's my answer for this one question that we had. [46:52]

I:[46:53] Thank you. So, please elaborate on the options and measures you have to evaluate the contribution and performance of the individual team members. [47:06]

E3:[47:10] To my knowledge, we do not have. Okay. I think I'm safe now. Yep, battery life. All right. I haven't been using any like automatically generated performance measures, so I don't have something like a measure like ticket that somebody has been working on in the last months, so I have to compile those information on myself. I don't know if any other coordinators have done something like that, but I do that every once in a while when I'm investing some time to get an overview on tickets, but most of the time, really evaluating somebody's performance is done when they request a grade, because that's when I have a more deep dive into their tickets, so to say, then I look at what kind of tickets they've been working on, so whether it's a complicated ticket, whether it's writing tests, whether it's a simple or complicated refactoring ticket, and that's when I evaluate people's performance, and also that helps to get a good, how should I say, to develop a good baseline on where contribution should usually be, so I now know, for instance, how many tickets are appropriate for 200 hours of work in the team. [48:52]

I:[49:00] Do you feel that you need to have a better overview and understanding of the performance of your team, and if so, would you need for gaining this overview? [49:09]

E3:[49:11] Yes. Some reporting would certainly be helpful. I mean, right now, I'm only aware of the boards that we have, and I made a couple of queries just to know which tickets are available, which tickets are currently being worked on by members of my team, but having like a reporting where I can see with one click per team member what tickets they're working on, other things, that would be helpful, and you can do some things with Jira, but not in the way that

463 I think is helpful. Yeah, maybe, so yeah, I already mentioned reporting, and  
464 the timesheet helps, but yeah, you only have individual entries and no option  
465 to aggregate over data, because sometimes what I wanted to do is I wanted to  
466 get an overview on how much time is invested in individual tickets, and there  
467 I had to essentially compile the information myself. I had to download this,  
468 export it as an Excel, and then try to wrangle the data a little bit, but having  
469 something like a dashboard where I can see, okay, those are the tickets that  
470 my team member is currently working on, and that's how many time goes  
471 into each of those tickets. That's also helpful for performance evaluation, also  
472 finding if there's any impediment. For instance, if you see that it's a simple  
473 ticket, and you see that they've already been working on for it for 40 hours,  
474 then you definitely know that something is wrong, if they don't bring it up in  
475 the meeting themselves, but you have, I mean, I in the past had to compile that  
476 information myself, and that often, yeah, if there was an easier way to do that,  
477 that would certainly be helpful. [51:32]

478 I:[51:35] Okay, so team members which participate in Catrobat as part of a  
479 university course agree to spend at least 24 hours per month working on a  
480 project. Why has this threshold been introduced, in your opinion? [51:50]

481 E3:[51:54] I heard that in the past, it has been the case that people weren't  
482 contributing for a long time, and then they got back for a certain amount of  
483 time, and then they asked for a grade, and then they vanished again, things  
484 like that. It's also there to have a baseline of contribution, because if you know  
485 that if people join a team, and you know that they will be spending roughly  
486 that amount of time per month, it's easier for planning, because then you know  
487 how much, how should I say, how much capacity you have in a team. If you  
488 have five, six team members, then you know roughly how much work you can  
489 get done in a given amount of months. If, for instance, I mean, sometimes it  
490 happens that people contribute less, and then I usually talk to them, okay, that's,  
491 that's like, you can, you can remedy by contributing more in the following  
492 month, but if people, if that, if that limit weren't there, that people really ignore  
493 that, then the problem is that sometimes you, you can, you can come to a halt.  
494 [53:14]

495 I:[53:20] And how do you check if this threshold is reached by every team  
496 member? [53:24]

497 E3:[53:25] I do, I do timesheet checks, and then if, and then if, if I see that there's  
498 no contribution, I often talk to people. [53:37]

499 I:[53:40] Okay, so what are the consequences for them personally, as well as for  
500 the project performance, if somebody does not meet the threshold? [53:49]

501 E3:[53:53] If somebody does not reach that threshold, then I, I have to talk  
502 to them, just to, just to make sure that, how should I say, the prioritisation  
503 is clear. The consequences for the, the consequences for the project, as I said  
504 before, could be like slower performance, slower velocity than that could be. If  
505 somebody, and personally for them, if somebody really doesn't contribute for a

506 long time, then we talk to them. And yeah, often this then leads to a discussion.  
507 And we had, we, I think we, often that was an indication that people were,  
508 were about to leave anyway, because they were mostly done with the hours.  
509 I haven't had a case where I really had to off-board somebody, because they  
510 weren't contributing. But when we off-boarded somebody, it was because they  
511 weren't contributing anymore, because they already got their grade, and they  
512 thought they were gonna contribute more, but then they wouldn't. And then  
513 we decided that it would be better to just off-board them, and to have to join  
514 later if they want to again. [55:12]

515 I:[55:18] You already gave kind of an answer to the next question. I would ask  
516 it anyway, if you want to add something still. What is your impact on the grade  
517 of the team member leaving? [55:28]

518 E3:[55:31] What, can you, can you elaborate on that? So what, do you think,  
519 what is it, what I have, what my say is in grading a team member, or how I  
520 grade a team member? [55:45]

521 I:[55:45] So how you do it, and what's, what your influence is. So is it only you  
522 who decides, or are there more people involved in the process? [55:56]

523 E3:[55:56] There's more people involved. I give, I give my recommendation  
524 for a grade, and I, from that moment on, I don't know which grade the event,  
525 they eventually get. So it could be that they're getting a worse grade, because  
526 somebody else thinks that my, my, that my suggestion is not, is too nice, or  
527 too bad. I don't know. But I, I look at a couple of things. I look at the tickets  
528 they have been doing and I also make notes when people contribute otherwise,  
529 so when people contribute with ideas, when people come to team days, when  
530 people participate in discussions, on how much they engage and those are the,  
531 the tickets are the more objective measure that I have. The other things that I  
532 mentioned are a little bit more subjective and all those together, they form my  
533 suggestion for a grade. [57:04]

534 I:[57:10] Okay, so we are approaching the last topic area, which is team member  
535 motivation. The first question is, what do you think, what motivates students  
536 to contribute to the Catrobat project? [57:26]

537 E3:[57:30] I think if you really can identify with the project itself, so the goals of  
538 the project, that certainly helps. If it's just for you personally, it's just another  
539 open source project like any other, then probably participation motivation is  
540 slightly lower. We saw that if people know each other, if they meet on site,  
541 then it's more motivating for them than if you are just sitting at home or in  
542 a room separate from other people and you only hear them every week in  
543 a meeting. That feels a little too distant, I think. If you're working together  
544 with people more closely, that's motivating. We also saw that if we have really  
545 short meetings where we essentially just have a five-minute meeting where  
546 everybody says, yeah, I've been working on the same ticket like the week before  
547 and I'm planning to do the same thing the next week, those meetings feel kind  
548 of redundant. But if we had meetings where we had discussions, where we

549 engaged a little bit more, where we discussed news, where we discussed tickets  
550 a little bit and other ideas, when people really felt that their ideas were being  
551 heard and considered, that's when I experienced in the upcoming weeks that  
552 participation was better. If you put people in the right team and you see that the  
553 things they are interested in, that they could put that to use, that's when they're  
554 really motivated. That's something, for instance, that I see from former team  
555 members that are now working in the Godot team, where they're working on  
556 a pipeline, on DevOps things, getting everything to work. That's what they're  
557 really interested in and that's also resulting in a high contribution. [59:54]

558 I:[59:57] Fine. So which impact does the motivation level of each team member  
559 have on the team performance? [01:00:04]

560 E3:[01:00:17] On the team's performance, so that's the performance of the team  
561 as a whole, you mean? [01:00:23]

562 I:[01:00:23] As a whole, yes. [01:00:24]

563 E3:[01:00:28] Obviously, if you're working on your tickets faster, if you're doing  
564 code reviews, that helps the team as a whole, objectively, but also if you're moti-  
565 vated and if you're engaging, then you, as mentioned before, you tend to carry  
566 other people along and that helps also other people's motivation. [01:00:49]

567 I:[01:00:53] Okay, so we are at the last question. Please tell me about incentives  
568 to increase the motivation of participants in the Catrobat project. [01:01:05]

569 E3:[01:01:09] I think a couple of ideas that we've been discussing, but some of  
570 them we get around to doing, some of them not. One thing that I think helps is  
571 more regular team days. It's not always easy, especially if the team is bigger,  
572 to get everybody on site on the same day. And sometimes when we had team  
573 days, people often joined for just an hour or two, which wasn't optimal, I think,  
574 but having team days where people really meet and then work together on  
575 things, that's motivating. And also then you, I don't know, you have a team day  
576 from 10 to 5 or something, and then you go for a drink out afterwards together.  
577 That's more the team building approach. Also Christmas party, that was also  
578 good for the team, I think. And one thing that we also thought about was to  
579 have little dedicated meetings and sessions where we have little "knowledge  
580 snacks", quote unquote, where people, if they've been working on something  
581 in their ticket that they think is valuable for other team members, that we have  
582 a meeting where they share those experiences. For instance, one of my team  
583 members, he's been working on the way how projects are stored. And so he  
584 worked a lot on the code files that are generated if you export a project. And  
585 if you have that kind of information available, then it's easier to understand  
586 some of the things and how to essentially artificially alter projects. And then we  
587 thought about that he could have a little tutoring session for the whole team on  
588 this. And that's also one of the things that I think could contribute, because then  
589 people feel that they contribute more than just with completing tickets, that  
590 their ideas are considered, that their experiences are considered and valuable  
591 for the whole team. And let me take, let me open my notes. Maybe I'll find

some other things. Team steps, regular knowledge transfer, yes. Yes, I think those are two things, the two things that I can think of. [01:04:09]

I: [01:04:12] Okay, perfect. I just made one note. You mentioned that the contribution time for the bachelor thesis decreased recently. Did I understand that right? And if yes, what did you mean about that? [01:04:30]

E3:[01:04:32] Oh, it's not been recently. I think it's already been a couple of years. When I started in 2020, many people, I mean, the newer ones, they have a bachelor thesis of roughly 200 hours. And before that, in the old curriculum, it was 400 hours. And of course, in 400 hours, you can get a lot, you have the team members longer in your team. So if they acquire some knowledge, you have that knowledge available for longer, and you have less turnover in your team. So right now, we often had it that people joined the team, then they worked on their checklist, they worked on the first ticket until they had good knowledge of the whole process. So they could like really start contributing and working on bigger tickets. They already had, I would estimate, 40 to 70 hours of their project done. So they were already done 30% of their work. And then if they work on bigger tickets, that can take a longer time. So often, people worked on a couple of tickets, and then they were gone from the team. But if you have double the amount of hours, then people can contribute more, they can contribute more to onboarding people, to onboarding new members, they eventually also stay longer for other courses. And if you have less turnover in your team, that's usually better for, I think that's usually better for keeping for knowledge transfer for, because then if people leave earlier, you can lose knowledge rather quickly if you don't have a good knowledge transfer process. [01:06:21]

I:[01:06:25] I see. Okay. So I'm through with my questions. Is there anything else you would like to share? [01:06:32]

E3:[01:06:36] Let me think. Nothing that comes to my mind now. [01:06:54]

I:[01:06:54] Okay. Then, thank you very much for the interesting conversation, and that you took your time. I'll stop the recording now. [01:07:05]

## B.4. Interview with *Catty* Coordinator

I:[00:01] Hello, thank you for taking the time to talk to me and for your willingness to conduct an interview. As already mentioned, I am conducting a scientific survey as part of my master's thesis at the University of Technology Graz regarding human resource management and possibilities on how to measure performance and productivity in the Catrobat project. The interview will take about 45 to 60 minutes and will be recorded. After the evaluation, the audio recording will be deleted for data protection reasons. You have already given your consent to the collection and processing of your data in the form of a signature. Thank you for this.

10 I'll ask you some questions about your role and responsibilities in the Catrobat  
11 project as well as the current state of organisational processes. If there is any-  
12 thing you don't want to answer, just let me know. Otherwise, I'd like to ask you  
13 to talk as freely and informally as possible.  
14 Everything you say is important and correct. The interview is about your expe-  
15 riences. The interview is divided into several thematic areas.  
16 I will always inform you when we proceed with the next topic. Are there any  
17 uncertainties or do you have any questions about the interview or the process?  
18 [01:20]  
19 E4:[01:22] No. We are good to go. [01:24]  
20 I:[01:24] Okay. Then let's start with the first question, which is from the topic  
21 area field of activity. And first of all, I'd like to ask you to tell me something  
22 about your field of activity and your role in the Catrobat project. You can take  
23 as much time as you like for this. I won't interrupt you. [01:44]  
24 E4:[01:47] Okay. So, I'd start with my official description of my role. That is  
25 that I am the coordinator for the iOS app of Pocket Code. So, I certainly care a  
26 lot about it, but it comes with a diverse set of responsibilities. Because, at the  
27 beginning, I am responsible for communications inside our team and looking  
28 after the flow of information. So, I'm here to remind people and to also com-  
29 municate with several people in and outside of our group and I'm also in a lot  
30 of contact with our product owner, and he is also the one who is responsible  
31 for more the general direction. I try to comply with that as much as possible  
32 and communicate that inside of our team. I'm also some sort of, you could  
33 say, scrum master. So, it's not only I'm a coordinator, but I'm also holding our  
34 team meetings and our weekly meetings and also being part of the bi-weekly  
35 coordinator meeting. We do have like this sort of big overarching theme of just  
36 being there if there is a question regarding the iOS team, which handles the  
37 Catty app, which it is called internally. [03:44]  
38 I:[03:49] Okay. When you think about your work as team coordinator, what are  
39 your most important tasks? [03:56]  
40 E4:[03:59] I think one of the most important tasks is to try to keep an overview  
41 of the different areas and the different people which are working on certain  
42 things. So it boils down to maybe just trying to keep things going. It's a lot of  
43 organisational stuff, at least for my role. It's definitely the most important area.  
44 Also, it's important that when we are, for example, getting new team members,  
45 doing the onboarding or doing the offboarding, it's kind of a lot of... I try to act  
46 as more of a hive mind for my group and try to keep most of the knowledge  
47 which we do have. I try to keep it in a place. So it's also a lot of documentation,  
48 which I try to do. And so I think it's like split across interacting with people  
49 and trying to know what everyone's doing and helping them there. And on the  
50 other side, if they're coming new people in or if they are leaving, people try to  
51 keep as much knowledge as possible. [05:30]  
52 I:[05:33] Please explain to me your tasks that take up the most time and explain

53 why they are so time consuming. [05:40]

54 **E4:**[05:43] That is a good question. I don't really know what's taking up the  
55 most time, to be honest, because I'm not that good at tracking it. I know what's  
56 taking up my most head space. So that's for sure the organisation part. So  
57 for example, setting up a meeting, which we sometimes have in person when  
58 we have a team day. It's like a bigger team meeting. And if I'm setting up the  
59 team meeting and checking with when everyone's available and the time and  
60 dates, and sometimes we then have to reschedule or we have to sometimes  
61 maybe set it up again because nobody's got really that much of time, especially  
62 when there's lots to do at university. So I think it would be most of the time  
63 the organisational task of setting up, keeping in check, taking everyone into  
64 account. That's taking up at least my most headspace. If I'm talking time only, I  
65 think it would be documentation, like doing, setting up new Confluence pages  
66 for various tasks, which new people should learn and they aren't documented  
67 yet. So that's something that takes a lot of time because you have to get in touch  
68 with the person that's doing the task right now. And when there is not like a  
69 follow-up candidate, let's say this way, you have to try to understand what are  
70 the most important steps, extract them out into a document of its own, get it  
71 up on Confluence. And then when the next person arrives, try to put the pieces  
72 back together and get that person started as good as possible. So I think that's  
73 taking up especially a lot of time to work as an intermediary of sorts. [08:00]

74 **I:**[08:04] Okay. [08:05]

75 **E4:**[08:07] I hope everything I'm trying to describe is good enough or if I should  
76 describe it in more detail, it's okay also. [08:16]

77 **I:**[08:16] No, sure. It's perfect. Thank you. I'd continue with the next question,  
78 which is, please try to remember your first days as coordinator. Which chal-  
79 lenges did you face in your new position? [08:31]

80 **E4:**[08:37] That's interesting. I do remember getting the chance to become co-  
81 ordinator of the Catty team, but it was quite a big transition for me because  
82 beforehand I was working at the Android part of Catrobat. So I was working at  
83 the IDE team and I got the opportunity to become the Catty coordinator because  
84 there wasn't one. So I came into a position which hasn't been previously been  
85 filled. So it was quite a challenge to know what you should do and what you  
86 shouldn't do. So I tried to a little bit enlist my former coordinator from the IDE  
87 team, which she helped me get things started. And she told me some things  
88 about the coordinator role, but she was also partly already leaving. So it was  
89 hard for me to understand where are my boundaries and what I am, I wouldn't  
90 say allowed to do because I'm not pushing boundaries or something, but what  
91 should I do? Like what is part of my job and what is part of also my essence of  
92 being a coordinator? Am I just like coordinating weeklies and that's it? Or do I  
93 need to like provide value and input for specific topics, which I'm maybe not  
94 even aware of because there hasn't been someone which I could learn from. So  
95 it was kind of like just trial and error. And that was very exhausting, actually. It

96 took a lot of time. It took a lot of effort. I tried to document it on this site for  
97 my successor, which hopefully will come at some point. But it was really hard  
98 time getting started as a coordinator. Yeah. [11:03]  
99 I:[11:06] Thank you. So we'll come to the next topic, which is team organisation.  
100 Please give me some information about the structure of the team you coordinate.  
101 [11:20]  
102 E4:[11:22] Okay. Should I be like, am I allowed to like name persons or should I  
103 just refer to the roles? [11:30]  
104 I:[11:34] Just refer to the roles.[11:36]  
105 E4:[11:36] Okay. Okay. Because I'm like, like, because of data protection, every-  
106 thing, I try not to like name anyone. And I stick to the roles then. [11:46]  
107 I:[11:46] Yeah. So just talk about developers, for example. [11:49]  
108 E4:[11:49] Okay. Yeah. So we do have, like, at the moment, we do have three  
109 developers, which I'm officially not part of, but I do still participate in devel-  
110 oping, but I wouldn't count myself into that because I'm already like my, my  
111 resources are already like distributed elsewhere. And we do have one of them  
112 is really junior. One of them is quite senior. And you could say that the third  
113 is I'd say it's also senior. Yeah. They both quite they, they know how to work  
114 to work the system quite well. So we also have our PO, which I would also  
115 count into our team, which is a little bit yeah, complicated because our PO is  
116 also at, I would say at some point, our technical lead, he does some of that.  
117 Um, he is responsible for the releases. So he, he's the one releasing, but he's  
118 also someone who is, for example, still doing like quality checks. And, and if  
119 there's like a pull request, he's like reviewing, he's like doing the reviews for  
120 the code. And he does still do a lot, but he is officially our, PO. And apart from  
121 that, I think that's, that are all the people which I am directly associated to in  
122 my team. Yeah. [13:38]  
123 I:[13:42] Okay. So is the amount of team members always the same or does it  
124 change? [13:47]  
125 E4:[13:49] The amount changes, there, there will be some, somebody who's  
126 leaving. He is finished with, with his degree at the TU and we will also get a  
127 new developer. I hope by fall he will join us because we do have, maybe that's, it  
128 has to be taken into account. We do have some special requirements, in contrast  
129 to other teams, we, you do need Apple hardware to be able to join our team.  
130 And you do need some programming experience at least to be able to join the  
131 team. And that weeds out a lot of people, unfortunately. And we, that's why  
132 we, we require sometimes are struggling with, with head count. Sometimes we  
133 could use like one or two people more, but it's hard to find people which, tick  
134 the boxes, which are unfortunately really required. So, yeah. [14:55]  
135 I:[14:59] Okay. So approximately for how long have the team members already  
136 been contributing and how long will the, stay, still be part of the team? [15:09]  
137 E4:[15:12] Um, okay. So one of the members has been contributing for, I think  
138 one plus year and the other one, the junior one also one plus year. And the



third one is he, he has been contributing for three or four months now. And our PO, which I also like counting in to a contributor. I think he has been there like way longer than anybody. So I think he's like three plus years at least. So, but he is only doing like requests anymore. [16:03]

I:[16:10] Okay. So please tell me about your team meetings in detail. For example, when, how often, where and which types of meeting do you have? [16:19]

E4:[16:20] Okay. So we do have team meetings every second week. They are like absolutely needed also and mandatory. And there is the structure of a team meeting. You can think of it as yeah, maybe like a daily in a scrum setting. So we, everyone talks about everything they have done and they are working on right now. And they will be also talking about some things that they maybe at the moment, block them from doing certain things or they are waiting on. And this is very important because otherwise we wouldn't like talk with each other for a very long time. And the meetings usually take about, I'd say half an hour to an hour on average. And then there's the other thing, which I would call, it's more like a team day where we have an extended team meeting. And we also have like certain topics we will be working on. So we have, for example, like this special project, we just need to get done and we need to do it before we release. And sometimes you're just working faster, more easily, especially with the PO when he's there. So it's also like an incentive for our PO to like attend these. So a lot of things can be done in a short amount of time for him because that's quite important because it's sometimes very hard to reach him and getting contact or something approved. So yeah, we have also like part of it is our classic meeting and the other is like the more special teams we are tackling at the moment. [18:40]

I:[18:45] Please tell me about the attendance rate at the meetings. Is it difficult to get all team members together? [18:51]

E4:[18:54] Actually not, no. It's like, it's, sometimes it takes more than one round to find a common date where everyone has the time to meet and also like the capacity mentally to be there. And I think the most, the hardest thing was to like get the PO to these meetings. Like he does attend some of them, but if I don't count him in and I just say like people who are not PO I would say we have a very good attendance rate. I'd even guess like it's three out of four people are always there, because we just like, yeah, we do place or, or at least I place a lot of value into that, into that meeting. So I try to find a good date. I try to encourage everyone. And I try to also like handle the meeting in a way that it provides value to every member. And not only that two people are talking about a special thing that they have, maybe they may be working on that can be done separately or after the meetings. I try to also like moderate it to the best of my abilities for every attending member. So I think that's apart from the correct date, I think is something that encourages to people. And if there is not that much to worry about, the meetings can also be like short. So like only half an hour and like everything's done. That's also, I think, very good. And yeah,

182 that's about it. [20:49]

183 I:[20:52] Mm-hmm. How important are those regular meetings for the project  
184 progress and why? [20:59]

185 E4:[21:06] I actually don't know. I think I can speculate because we've never  
186 been, or at least when I was part of the Catty team, we've never been without  
187 these meetings. So I'm actually not sure what the control group would be doing.  
188 But I think it helps some people in my team communicate and it provides a  
189 little bit of a platform for people to speak with each other when otherwise they  
190 maybe wouldn't do it that easily, like write someone or set up a meeting one-to-  
191 one with someone. And most of it, most of it happens in our team meetings.  
192 Most of the setup or follow up meetings or most of the other meetings outside,  
193 which where people collaborate with each other happen or get created in our  
194 team meetings. I think it's, I think we, we wouldn't have that much progress  
195 especially working together when there was no team meeting. [22:39]

196 I:[22:42] Okay. So when you think about collaboration within the team, as well  
197 as in between teams in Catrobat please name and explain obstacles that hinder  
198 this cooperation, if there are any. [22:57]

199 E4:[23:06] I'm actually not sure if there are any. I do think that I'm just thinking  
200 about if I've experienced any I do know that we had, for example, we had once,  
201 it was just a single time thing. So I'm not sure if that's like relevant even, but  
202 we do had some, like some waiting time between creating a ticket and receiving  
203 a mock-up for the specific thing we wanted to implement. So we wanted like to  
204 implement a specific yeah, maybe you could say like a new block or something.  
205 And we wanted to like, wanted to know how it should look like in the program,  
206 like visually. And we did need to wait quite some time because the people who  
207 were, who were responsible were not exactly pinpointed. Like it wasn't known  
208 who was responsible on our team. It wasn't known who was responsible on the  
209 I don't know what, no, I think it's the design team. So I think it wasn't known  
210 who was responsible there to create a mock-up. So it was like, yeah, someone  
211 from my team spoke to me, I spoke to the coordinator there, the coordinator  
212 there spoke to the team member there, or maybe not, I don't know. And then  
213 maybe I reminded them. So it was like this back and forth with like some two  
214 coordinators in between, which maybe we didn't know and didn't need in that  
215 special example. Otherwise, I've never experienced any, anything blocking or  
216 inefficient. [25:10]

217 I:[25:13] Okay, great. So the next topic area is the development process. And first  
218 of all, please explain to me the process of task implementation from moving a  
219 ticket from ready for development to merge or done. [25:34]

220 E4:[25:38] I think that's something I have to experience first before I can tell  
221 you the full story. I do know that we have, like, I can tell you the process,  
222 the theoretical process, because do you want the theoretical practical process.  
223 [25:59]

224 I:[26:01] As you experience it as a coordinator. So your view on this process. So

if it's theoretical, it's fine. [26:08]

**E4:**[26:10] Okay. So the theoretical process is that someone will code up the solution for it and then open up a pull request. This pull request will get reviewed by two independent senior developers. And these people will then merge it. And it will, as part of being merged, be part of the next release. So that's like the theoretical, the practical side is maybe a little bit more I'd say extended, let's say this way. And because it does, there's a lot of back and forth before something even gets to the point of being reviewed. Like the review officially happens at the end, but at least in my, for my part, we have never actually committed something which is open for review when it wasn't like absolutely fine. So we do like, there's a pretty high standard, I'd say, which leaves a lot, which there's a lot to do before it being reviewed. And then when there's the review it sometimes just takes some time to find two senior programmers who are able and willing to review it. So it's like the more practical description of the process. [27:41]

**I:**[27:44] Fine. So how long does it approximately take until the ticket has undergone this process from ready for development to merged? [27:55]

**E4:**[28:02] Like for all the tickets or only for the tickets that are being worked on in ready for development? [28:12]

**I:**[28:13] So yes, when somebody chooses to, to work on a ticket until it is then merged. [28:19]

**E4:**[28:20] Okay. Because we do have a lot of tickets, like not a lot, but we have a lot more tickets in ready for development than there are people like working on some of them. So when there is a ticket most of the time it's at least as long as I am here, it has been a bigger feature. So there has been this very long development cycles actually. And it took, and that's why maybe it's like hard to give an average because the sample size is so small. But if there were like these multi-month tickets and there were like several of them and I don't know if like, that's the norm. Maybe, unfortunately, maybe it has become the norm. I'm not like advocating for it, but there are like this very long feature tickets and it absolutely does take, I'd say at least three months for something to be, to get from ready to development until being like actually finished. [29:37]

**I:**[29:42] And are there any bottlenecks in this process? And if yes, how are you trying to mitigate them? [29:49]

**E4:**[29:57] Can you, can you give me an example of what a bottleneck would be? Because I don't, like... [30:03]

**I:**[30:04] Yes, for example, you need to wait for a specific person to do an action with this ticket until it can get merged and it is stuck at this stage for weeks, for example. [30:20]

**E4:**[30:21] Yeah, absolutely. So I would say it's the final stage because the first bottleneck I thought of was like, yeah, we don't have enough developers, so we could do more if we have more. But the actual bottleneck is that we sometimes wait way too long for a ticket to be merged. Even that long that people are

268 coming, working on a big feature and are ready to leave, but the feature is not  
269 implemented yet until we release. And there is this big, I'd say... It's hard to  
270 find a word for it, but there is this big risk, let's say risk that the ticket will  
271 become stale so that the whole feature branch is getting stale because after  
272 the person who did multiple months of really good coding would just become  
273 obsolete because the person's leaving. No one's able to get into their shoes and  
274 just continue that work. So it's like, if it's finished, we've got to implement it  
275 because otherwise people will leave and they will leave with a sour aftertaste.  
276 So it will be not as rewarding to leave the team as if you could finish it with a  
277 feature implemented. And we are waiting for the actual code review. We are  
278 waiting way too much time. And I'm talking about months here. And that's  
279 partly due because we just have our PO and partly due because he is our PO. So  
280 it's like, there's this challenge or I think I would say even clash of roles because  
281 on one hand he's responsible. He should be our PO and he should give a clear  
282 direction of what's to be done and when it's to be done. And on the other side,  
283 he's also responsible for implementing at least the final part. He's responsible  
284 for greenlighting the final parts. And so there's these two personalities which  
285 are fighting with each other because at one point he's like, okay, I need more  
286 time, but there's no one who can say, okay, you got more time except for himself.  
287 So it's like, it's this conflict of interest, which it's shining clear through because  
288 apart from that, the first half of the process is working quite well, I'd say. [33:25]  
289 I:[33:28] Okay. So please tell me about the planning of this development process.  
290 How do you decide which ticket should be moved to ready for development or  
291 which tickets should be worked on? [33:42]  
292 E4:[33:45] So that happens, that happens mostly, the decision is made mostly  
293 by external factors, I would say. So it's like, we do have some other team or  
294 we do have some other PO asking us for a slightly special topic or something  
295 we should work on, and then we will start working on it. And on the other  
296 hand, there are people which are quite intrinsically motivated, and they are  
297 able to find a topic, like they are working on, there is this big pool of ready  
298 to development tickets, which was already there when I got here. So they are  
299 working on one of them. And they find out, okay, there's a lot more stuff we  
300 could work on. And then they find work themselves. So it's like, these are, like  
301 I said, the main two things that decide what people are working on, except for  
302 when they have to do a beginner ticket. That's like, I do look that they have  
303 to do a beginner ticket, which is also like suited for them and their program  
304 knowledge level, like a skill level. So it's like that's the important part. And  
305 after that, it's a little bit, I'd say, open, a little bit more like open to, for the  
306 individual person to decide what to do. Or like, yeah, for example, they either  
307 decide that they want to continue working on something they already like, just  
308 work. And on the other side, they maybe get an outside notification that they  
309 will say, okay, like our PO or I get and I say, okay, we need to work on this  
310 specific topic and who wants to, and like, when we find a solution. [36:09]

I:[36:12] How much information do you have on the availability of your team members in advance? For example, how many hours to plan to work the next week, or if they are available at all? [36:23]

E4:[36:26] I don't have, except from team meeting, I don't have like any sort of knowing when they are available unless I ask them directly. [36:43]

I:[36:47] Are there some kind of time critical tickets in your project? And if yes, how do you make sure that those tickets are done in time? [36:54]

E4:[36:56] Yeah, we do have, at the moment, we do have one time critical ticket. And we started working on it the moment, like we got a new team member, which also was part of, like, you get a new team member, but he also wants to work in that field because that is important. And so this ticket that this team member is working on is time critical because they wanted to use it for summer break, because there's like this special, we wanted to implement a special brick, which is called like a draw brick, which enables the user to draw certain shapes, like you could stitch certain shapes. You could, in the future, also draw certain shapes and export it as an SVG, and then a plotter could be like cut out, could cut out your shape or the letters you designed or something like that. So that's one, that one's time critical. And actually, it was time critical until yesterday, because it was the deadline, and the feature was implemented, but it wasn't greenlit. So it wasn't, it isn't, we haven't done a release yet. So there's again, like this big part, where I try to make sure that we are on time by meeting up with the person several times. And also like supporting them asking what, what do you need? How can I help you? And I also did a little bit of pair programming with him. So I'm not the experienced one, but I still can provide some, I still can provide some value and some insights. And so we, we did that more or less together. And the second half, I did reminders, I did confirmations, like, yeah, is that correct? Okay, yeah, we, we got it working like this and that. And so I tried to keep in touch and then try to keep everything in check for like the code review and everything after that. But unfortunately, what it looks like to me at the moment, it's like, it hasn't gone through the release. So, we, I don't actually have a good, a good idea or a good strategy to combat the, this problem. Yeah. [39:54]

I:[39:58] Okay. So, onto the next topic area, which is recruiting and off-boarding. When you think about recruiting new members, how does the process for filling open positions work in detail? [40:13]

E4:[40:15] You want to repeat that real quick? [40:19]

I:[40:19] Yeah. When you think about recruiting new members, how does the process for filling open positions work in detail? [40:26]

E4:[40:28] Okay. So when we, when we do have the need for team members, we unfortunately still have like this, this stop of, of taking on new team members in a, in place, at least that's my knowledge. And when we do need someone new, we try to look at our personal circle. So we try to find someone which we sometimes already know, and we check if they have the available hardware

354 and if they also have some skills and if they want to work at Catrobat. And  
355 that's like the most important step of trying to get a new team members, like  
356 talking with them, what's part of the, like what's part of the project and what  
357 our team does and also like why they need the hardware and why they need  
358 some programming skills. And it's more than the hard thing is at one point  
359 you are exhausted, your social circle is exhausted. So you don't have enough  
360 people to ask. So maybe you think to yourself, who can I ask next, but there  
361 won't be somebody next so you just hope that new team members will maybe  
362 recruit new team members. But that's also not always the case, as you can think.

363 [42:12]

364 I:[42:16] Can you think of one or more situations where the lack of new team  
365 members led to difficult conditions for the project progress? And if yes, please  
366 tell me about it. [42:26]

367 E4:[42:29] Actually, not really. We didn't have, we were always low on team  
368 members, like in some we are four, I'd say five people. So we've always been  
369 low, but we've never been critically low that we, for example, can't do our pull  
370 requests or we can't, for example, onboard someone new. [43:04]

371 I:[43:07] Please tell me about your experiences regarding the off-boarding pro-  
372 cess of team members. [43:13]

373 E4:[43:16] Some people just vanish and some don't. I think that's the easiest  
374 way to describe it. Some people stay in touch with the project and really are  
375 interested to see everyone in the team succeed and the project succeed. And  
376 some people are, at the moment they are done with their hours, they are gone,  
377 which is also fair. That's the agreement which is in place at the first time. That's  
378 why we have it. And when I know someone is leaving, I try to talk with them a  
379 little bit more than also setting up a meeting one-on-one to get some insights of  
380 what has been good, what has been bad. Would you do it again? And if not,  
381 why? So I'll try to get a little bit of insight what's maybe working and what's  
382 not. And also before they're leaving, also trying to understand what they've  
383 been doing exactly in their role. So I know how it has to be maybe filled. So  
384 to actually know it, it has to be filled or to try to maybe extract some more  
385 knowledge out of them. [44:53]

386 I:[44:58] So please elaborate on the knowledge transfer from experienced mem-  
387 bers, which leave the project to new members. [45:05]

388 E4:[45:08] Okay. So best case scenario is someone is leaving and two months,  
389 three months before that, someone new is joining. Unfortunately, I've never had  
390 that opportunity yet. So most of the time it is knowledge gets picked up and  
391 put together by someone who is new. And my part in the knowledge transfer  
392 process is that I try to retain as much of that as easily accessible as possible. So  
393 that means most of the time creating a Confluence page and or doing just a  
394 document, like a Word document or something like that, where I try to write  
395 down what's important, what are the steps someone has to do to accomplish a  
396 specific task and so on. So it's like, I try to retain it like some kind of swimming

pool. And then when there's someone new, I try to transfer it the best way I can, which is honestly also a huge loss, but it's better than nothing. [46:30]

**I:**[46:35] I would like to speed forward to the moment when you are finished with your hours as coordinator. Please tell me how you imagine the onboarding process for your successor. [46:47]

**E4:**[46:49] Hmm. I imagine it as being well-structured and also, like if I could dream it up right now, I'd say it will be well-structured. And it has to be like there has to be some kind of grace period where someone who is still new at the position has the possibility to say like, okay, you told me, you taught me. It was interesting, but after like three weeks, I just realized that's not my type. I don't want to do anything in that direction. I want to work as a developer, which is fair. And I think that would be for me personally the best kind of onboarding there is. And also there would be like the transition period would be long enough, long enough for the new person to get settled or to like think, okay, it's not for me, but someone else still has time to take over. So that would be the ideal one. And also it would be like more or less seamless transition because I would like spend my remaining hours just training the new guy or girl. So I think that would be the best option, but I don't know if it's going to work. [48:30]

**I:**[48:34] Okay. So on to the next topic, which is performance measures. What do you believe are the key factors that have contributed most significantly to the achievements of the Catrobat project and why? [48:50]

**E4:**[48:59] I think one of the more important key factors is that there is an exchange of sorts, like you exchange some of your time for some of your ECTS. I think that's like one very important motivator for a lot of people in my team to get on board. And as a long-term factor, I'd say, which maybe isn't always used at the best of its abilities, maybe the identification with the project. So I think being like seeing the value and seeing what it does in the world, like what it does outside in the world, in the field, what's happening with that, how is it used, how are people enjoying it and where are we making a difference right now. Maybe that would be like another big motivator, but I don't think it's like that. I don't think it's that relevant right now. I don't think a lot of people spend one or many thoughts about it, unless you're like 100% invested in like Catrobat and the mission and everything it does. [50:34]

**I:**[50:38] Okay. Please elaborate on the options and measures you have to evaluate the contribution and performance of the individual team members. [50:49]

**E4:**[50:54] So can you clarify maybe how or what would be an example for that? [51:02]

**I:**[51:03] So, do you know or how do you know which of your developers contributes how much to the project? Or, so how experienced someone is, how do you know that? Can you measure this? [51:18]

**E4:**[51:22] Actually, most of it is self-reported. Like that's the key thing. And inside of my team, most of it is self-reported. There's also some kind of feel, I'd

440 say. I got to how much someone's doing or not. There's also like the timesheet,  
441 but this doesn't look right. It just forms more of a subjective way, my interpretation  
442 of the team and of the individual team members. I'd say it's pretty hard.  
443 It's pretty hard to objectively measure the contributions because I'm not going  
444 to start and counting how many lines of code someone has done, which I also  
445 don't think is like the way to go here. And I don't think apart from trusting my  
446 instinct and in some part trusting my team to report the truth and the accurate  
447 amount of what they've done. If someone's saying he's had a hard time doing  
448 something and he has spent a lot of time on it, I'm like, okay, that's true. We're  
449 here to learn. I'm a teacher. I do study that. Like it's part of my studies and I  
450 want to become a teacher. I'm not lenient, but I do think I'd spot someone's  
451 lacking, but also I'm very understanding if someone's here to learn and able to  
452 learn. It's hard for me to measure performance in an objective way. [53:35]

453 I:[53:38] Do you feel that you need to have a better overview and understanding  
454 of the performance of your team? And if yes, what would you need for gaining  
455 this overview? [53:46]

456 E4:[53:51] Well, I think it would be like, I'm not the tracking guy, honestly, so I  
457 don't think actually there would be a better, I need a better way of checking  
458 after what everyone has done and how they've done it. I think it would be  
459 nice to actually be able to review some time spent on something and review  
460 it in comparison to maybe some other sample group. I would be interested if  
461 someone's maybe spending this and that much amount of time on this and  
462 that task and if someone else is doing it more or less, but I wouldn't like rate  
463 performance on it. We would have to go through story points and doing that  
464 and this and that. The whole thing just becomes some game theory. I'd say it's  
465 just become some game theory example where someone who's trying to cheat  
466 the game will always be able to cheat it. I don't actually think I need more, no.  
467 [55:28]

468 I:[55:30] Okay. Team members which participate in Catrobat as part of the  
469 university course agree to spend at least 24 hours per month working on a  
470 project. Why has this threshold been introduced in your opinion? [55:44]

471 E4:[55:46] I think it has been introduced because people didn't take it seriously,  
472 I'd say, like serious in the sense of being able to reach someone or just going  
473 offline or just leaving without saying anything. I think it got introduced because  
474 there has to be some minimum bar and it was the easiest way to set a bar which  
475 is uniform across all teams and all tasks. I think it was an easy and it's a good  
476 way to talk with people that are not participating in their best interest. So I'd  
477 say it was introduced because of that, because you also have a little bit of, I'd  
478 say, I don't actually know how to word it in a good way, so let's leave it at that.  
479 [57:15]

480 I:[57:17] How do you check if this 24-hour threshold is reached by every team  
481 member? [57:21]

482 E4:[57:24] I do check it with the timesheet if it's possible to check and if it hasn't



been reached, I place much more emphasis on being given an explanation and being able to know is this person participating, does this person even want to participate or is it just here for free credit. It's important to distinguish and I do not have a problem with someone bringing in less than the 24 hours per month if there is, for example, they just started a new job as a teacher and they are at the end of the school year, they're having a hard time, but we do come to an agreement that in the summer holidays there will be more time invested. So I think if they are reachable and all the stuff I already said, if that's there, it's all good. [58:38]

**I:**[58:40] Yeah. You kind of already answered the next question, I will ask it anyway if you want to add still something to it, if not it's also okay. What are the consequences for them personally as well as for the project performance if somebody does not meet the threshold? [58:55]

**E4:**[58:56] Yeah, as I already said, I don't think it's not the gold standard having to do the 24 hours. I think it's more or less a good way to measure if someone is willing and participating in a meaningful way, but I do not have a problem if someone's working less because on the other hand, you have to think about if you're a small team and someone's working less, they do stay longer in general if they have if they're doing less hours per month, otherwise some members will be gone or would be already gone and we would be like critically low so it's like in some part I'm not the guy but the other part it's also like helping the team in general right now. [01:00:01]

**I:**[01:00:04] What is your impact on the grade of team member leaving? [01:00:07]

**E4:** [01:00:10] What sorry? [01:00:11]

**I:** [01:00:12] What is your impact on the grade of the team member leaving? [01:00:15]

**E4:** [01:00:18] Like how much I decide on? [01:00:18]

**I:**[01:00:21] And how... [01:00:21]

**E4:** [01:00:22] I just a lot, I do this most important question for me is did they learn something and how much did they improve? I try to compare the person they were when they started, and I try to compare the person they are, when they are leaving. And I try to take everything into account. So, I try to take into account how was their skill level, could they even write a line of code like a proper line, or could they talk in a group, or could they set up meetings by themselves, or could this person be trusted with a task that is maybe not that straightforward? So it was like, and could this person at the end be trusted with a task that is not that straightforward? So like where did this person improve and how much? And in all honesty like most of the people I've got the chance to grade did receive a very good grades. Just because of that because I could see at some point in everyone, which has been part of my team as of yet, I could see in everyone a big improvement from joining the team and also then leaving the team. [01:02:02]

**I:** [01:02:05] Great so we're approaching the last topic, which is team member

526 motivation and as well to this first question you already answered it more  
527 or less. Still, what do you think what motivates students to contribute to the  
528 Catrobat project? [01:02:23]

529 **E4:** [01:02:24] Yeah, credits and maybe in the long term identification with the  
530 project and making a difference. Also it I think for some, at least for my part,  
531 it's also interesting to take part in a university project and getting insights  
532 how things are run inside the university and not just being there to consume  
533 knowledge, but also like contributing in some way. [01:02:59]

534 **I:** [01:03:02] Which impact does the motivation level of each team member have  
535 on the team performance? [01:03:08]

536 **E4:** [01:03:13] The motivation level of each team member on the team per-  
537 formance? I don't really know how to answer it actually. Because I think, in  
538 general it does have, I would think that it does have a big impact, but I actually  
539 don't know, because I've never had... Like we do have differences and we  
540 do have discussions and everything, but we never had like someone, who's  
541 constantly like just being a donor. Like it hasn't happened, and I think at this  
542 moment I can't really like say something more than that, because I don't have  
543 the experience. [01:04:18]

544 **I:** [01:04:24] Okay. So we are approaching the last question. Please tell me about  
545 incentives to increase the motivation of participants in the Catrobat project, if  
546 there are any! [01:04:36]

547 **E4:** [01:04:39] Like team specific incentives or like in general? [01:04:43]

548 **I:** [01:04:44] Both. [01:04:44]

549 **E4:** [01:04:47] Like I think I'm repeating myself, but I the biggest one is the  
550 credits and there is also the... We've had a Christmas party, which was also  
551 very nice. We've also had a team day, which I unfortunately couldn't attend, but  
552 that was also very welcoming addition. Also there are several workshops we  
553 can attend and I've also been to I think most of them. And they were of different  
554 quality. Some of them were interesting, some of them maybe not targeted at  
555 my specific knowledge, but also interesting to see. And at last you do have the  
556 chance to get to meet new people, which could be like-minded. So it's easy if  
557 you are, if you love to work on projects like I do it's quite easy to find people,  
558 who also love to work on projects. Which maybe don't are part of university,  
559 maybe like they are at some part university some part maybe outside of it. So  
560 it's like interesting to see and to get to know these people and there's a higher  
561 concentration in Catrobat than there is in the general student group. [01:06:30]

562 **I:** [01:06:34] Okay, thank you. So we are through with the questions. Is there  
563 anything else you would like to share? [01:06:41]

564 **E4:** [01:06:42] No. [01:06:42]

565 **I:** [01:06:44] Great. Thank you very much for the interesting conversation and  
566 for your time. I'll stop the recording now. [01:06:52]

## **Appendix C.**

### **Interview Consent Agreement**

A scan of each signed declaration of consent is shown on the following pages.

## Interview Consent Agreement

**Interviewer:** Julian Jautz

**Purpose:** Master's thesis

**Topic:** Introducing KPIs and HR Management into a FOSS project and their impact on project performance

**Interviewee:** JULIA HEROLD

It is hereby assured that the interview and the data collected will only be used for the purpose of the master's thesis. The interview is designed to gather information about the current status of HRM processes in the project and to find out if there are any issues in the project processes, that make the work more difficult. Personal data will be anonymized, unless the interviewee agrees to be named.

The interviewee consents to the recording and transcription of the interview for the purpose of scientific processing regarding the master's thesis.

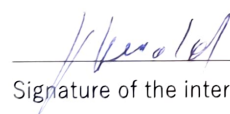
Please tick as appropriate:

☒ The interviewee agrees to be mentioned by name in the master's thesis.

☐ The interviewee does NOT agree to be mentioned by name in the master's thesis.

GRAZ, 09.07.2024

Place, date

  
Signature of the interviewee

  
Signature of the interviewer

For further information, please contact:

Name: Julian Jautz, BSc

Email: julian.jautz@student.tugraz.at

Phone: +436803280343

## Interview Consent Agreement

**Interviewer:** Julian Jautz

**Purpose:** Master's thesis

**Topic:** Introducing KPIs and HR Management into a FOSS project and their impact on project performance

**Interviewee:** Thomas REICHEL

It is hereby assured that the interview and the data collected will only be used for the purpose of the master's thesis. The interview is designed to gather information about the current status of HRM processes in the project and to find out if there are any issues in the project processes, that make the work more difficult. Personal data will be anonymized, unless the interviewee agrees to be named.

The interviewee consents to the recording and transcription of the interview for the purpose of scientific processing regarding the master's thesis.

Please tick as appropriate:

☒ The interviewee agrees to be mentioned by name in the master's thesis.

☐ The interviewee does NOT agree to be mentioned by name in the master's thesis.

GRAZ, July 14<sup>th</sup> 2024

Place, date

TRM

Signature of the interviewee

Julian Jautz

Signature of the interviewer

For further information, please contact:

Name: Julian Jautz, BSc

Email: julian.jautz@student.tugraz.at

Phone: +436803280343

### Interview Consent Agreement

**Interviewer:** Julian Jautz

**Purpose:** Master's thesis

**Topic:** Introducing KPIs and HR Management into a FOSS project and their impact on project performance

**Interviewee:** Edon Mucaj

It is hereby assured that the interview and the data collected will only be used for the purpose of the master's thesis. The interview is designed to gather information about the current status of HRM processes in the project and to find out if there are any issues in the project processes, that make the work more difficult. Personal data will be anonymized, unless the interviewee agrees to be named.

The interviewee consents to the recording and transcription of the interview for the purpose of scientific processing regarding the master's thesis.

Please tick as appropriate:

- ☒ The interviewee agrees to be mentioned by name in the master's thesis.
- ☐ The interviewee does NOT agree to be mentioned by name in the master's thesis.

11. Juli 2024

Place, date



Signature of the interviewee

Signature of the interviewer

For further information, please contact:

Name: Julian Jautz, BSc

Email: [julian.jautz@student.tugraz.at](mailto:julian.jautz@student.tugraz.at)

Phone: +436803280343

---

## Interview Consent Agreement

**Interviewer:** Julian Jautz

**Purpose:** Master's thesis

**Topic:** Introducing KPIs and HR Management into a FOSS project and their impact on project performance

**Interviewee:** Samuel Grausgruber

It is hereby assured that the interview and the data collected will only be used for the purpose of the master's thesis. The interview is designed to gather information about the current status of HRM processes in the project and to find out if there are any issues in the project processes, that make the work more difficult. Personal data will be anonymized, unless the interviewee agrees to be named.

The interviewee consents to the recording and transcription of the interview for the purpose of scientific processing regarding the master's thesis.

Please tick as appropriate:

- ☒ The interviewee agrees to be mentioned by name in the master's thesis.
- ☐ The interviewee does NOT agree to be mentioned by name in the master's thesis.

Graz 22. July 2024

Place, date



Signature of the interviewee



Signature of the interviewer

For further information, please contact:

Name: Julian Jautz, BSc

Email: julian.jautz@student.tugraz.at

Phone: +436803280343