

**A Systematic Personalized Team Onboarding Design Process
In Software Development Companies**

By

Khounkham Phengthangsy

**A Thesis submitted in the requirement of the Degree of
Master of Computer & Information Sciences**

**School of Engineering, Computer and Mathematics Sciences
Auckland University of Technology, New Zealand**

**Primary Supervisor: Jim Buchan
Secondary Supervisor: Professor Stephen G. MacDonell**

2021

Abstract

A new team member may join an existing software development team due to a multitude of reasons, such as: replacement for a loss of existing members, new team formation, team member reshuffle, or creation of new roles in the team. It may take some time for the new team member to become integrated and productive and often an onboarding program is used to support the new team member during this time. The design of these onboarding programmes seems quite adhoc and often there is a one-size-fits-all approach. This may result in onboarding programs not well-suited to the particular characteristics of the new team member and the team. This research proposes a method of systematically designing a personalised onboarding program to take into account the diverse characteristics and experience of onboarders, as well as the specific contexts of the teams. The research also proposes some onboarding program implementation strategies as well as a supporting design tool.

The research first investigates how team onboarding programs of software development companies are conceptualized in the literature and then proposes a systematic personalized team onboarding design process. A set of main onboarding factors are investigated including: desired onboarding goals, activities to support the achievement of the goals, and possible challenges and risks to the implementation of the program of activities. Different perspectives of related parties of the software development teams as well as diverse personalities of onboarders are considered. The designed process is based on an empirically-based guide to customize the onboarding goals, select the tasks that the onboarder can do to achieve these goals, and consider possible risks that could be preventing the success of onboarding programs.

A Design Science Research approach was adopted as an iterative processes of problem identification, identifying design process requirements, and the design process development. Onboarding design process evaluation was based on simulated use-case scenarios only at this stage. The understanding of problem, the onboarding goals, related onboarding tasks, and the risks that form the basis of the proposed onboarding design process is gained from a Systematic Literature Review of relevant research. A thematic synthesis method was applied to categorize the set of onboarding goals, activities, and challenges/barriers/risks for analysis of extracted data from systematic literature review. Two indicative use-cases were used to evaluate the utility of the proposed onboarding program design process by construct personalized team onboarding programs for these use-cases, and reflecting on these. To support the proposed onboarding design process, a prototype tool was also developed, as a proof-of-concept .

It is expected that the results of this thesis will contribute to professional practice in a couple of ways. First, it is hoped that this thesis will raise the awareness of team onboarding among practitioners, and the need to design onboarding programs systematically and proactively. Second, the work in this thesis provides onboarding program designers with an empirically-

based onboarding design process that will allow the creation of personalized programs to suit the specific characteristics of the onboarder and teams. Researchers may benefit from this work also by building on the conceptualization of team onboarding presented.

Keywords: Onboarding, team onboarding, software development, personalized onboarding program

Table of Contents

ABSTRACT	II
LIST OF FIGURES.....	IV
LIST OF TABLES	V
ATTESTATION OF AUTHORSHIP	VI
ACKNOWLEDGEMENTS.....	VII
<u>CHAPTER 1: INTRODUCTION</u>	<u>1</u>
1.1 MOTIVATION AND BACKGROUND	1
1.2 MOTIVATION FOR BUILDING THE TOOL	2
1.3 RESEARCH PROBLEM IN SOFTWARE DEVELOPMENT INDUSTRIES.....	3
1.4 RESEARCH METHOD, OUTCOME AND CONTRIBUTION.....	3
1.5 THESIS OUTLINE	4
<u>CHAPTER 2: LITERATURE REVIEW AND RELATED WORK.....</u>	<u>5</u>
2.1 DEFINITION OF ONBOARDING AND ITS PURPOSES.	5
2.2 DIFFERENT ONBOARDING CONTEXTS	6
2.2.1 GENERAL ONBOARDING PROGRAMS OF AN ORGANIZATION.....	7
2.2.2 ONBOARDING PROGRAMS OF OPEN-SOURCE SOFTWARE DEVELOPMENT PROJECTS.....	8
2.2.3 ONBOARDING PROGRAMS OF VIRTUAL TEAMS/GLOBALLY DISTRIBUTED SOFTWARE PROJECTS	9
2.2.4 ONBOARDING PROGRAMS OF SMALL/MEDIUM-SIZED SOFTWARE DEVELOPMENT PROJECTS	10
2.3 ONBOARDING CHALLENGES, POTENTIAL SOLUTION.....	11
2.4 ONBOARDING GOALS IN SOFTWARE DEVELOPMENT INDUSTRIES.....	12
2.5 RECENT ONBOARDING MODELS AND CONCEPTS.....	16
2.5.1 BAUER AND ERDOGAN’S MODEL.....	17
2.6 ORGANIZATIONAL, DEPARTMENTAL, TEAM ONBOARDING.....	21
2.7 ONBOARDING DEFINITION SUMMARY OF LITERATURE REVIEW	23
<u>CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY.....</u>	<u>26</u>
3.1 RESEARCH PURPOSE AND QUESTIONS.....	26
3.2 RESEARCH METHOD	27
3.3 RESEARCH PROCESS	30
3.3.1 SYSTEMATIC COLLECTION OF RELEVANT ARTICLES	31
3.3.2 INCLUSION AND EXCLUSION CRITERIA.....	34
3.3.3 SYSTEMATIC LITERATURE REVIEW	34
3.3.4 DATA ANALYSIS	35
3.3.5 A DESIGN WORKFLOW FOR A PERSONALIZED TEAM ONBOARDING PROGRAM.....	39
3.3.6 A PROPOSED SUPPORTING TOOL FOR A DESIGN WORKFLOW.....	43

CHAPTER 4: FINDINGS AND DISCUSSION	44
4.1 SYSTEMATIC LITERATURE REVIEW	44
4.2 RESULTS OF DATA ANALYSIS.....	45
4.2.1 RESEARCH QUESTIONS.....	45
4.2.2 ONBOARDING GOALS, CHALLENGES/RISKS, AND TASKS/ACTIVITIES	46
CHAPTER 5: PROPOSED ONBOARDING DESIGN PROCESS.....	58
5.1 PROCESS DESIGN OVERVIEW	58
5.2 USE CASES	62
5.2.1 USE CASE 1.....	62
5.2.2 USE CASE 2.....	67
5.3 COMPARISON OF USE CASES 1 AND 2.....	71
5.4 A PROPOSED TOOL	71
5.4.1 TEAM SETTING	71
5.4.2 ONBOARDER’S CHARACTERISTICS	73
5.4.3 A SET OF ONBOARDING GOALS AND SUPPORTING TASKS	74
CHAPTER 6: CONCLUSION AND FUTURE WORK	77
6.1 CONCLUSION.	77
6.2 LIMITATIONS AND THREATS TO VALIDITY.....	78
6.3 FUTURE RESEARCH.....	79
REFERENCES.....	80
APPENDICES.....	90
APPENDIX A - EXTRACTED DATA FROM SYSTEMATIC LITERATURE REVIEW	90
APPENDIX B - A SUMMARY OF THREE DISCOVERED CATEGORIES OF ONBOARDING RESEARCH CONTEXTS ..	127
APPENDIX C - A SUMMARY OF DISCOVERY-DESIRED ONBOARDING GOALS OF ONBOARDING PROGRAM...	130
APPENDIX D - A SUMMARY OF DISCOVERY-SUGGESTED ONBOARDING TASKS TO SUPPORT DESIRED GOAL	132
APPENDIX E - A SUMMARY OF DISCOVERED ONBOARDING BARRIERS/CHALLENGES/DIFFICULTIES/RISKS FROM REVIEWED ARTICLE.....	137

List of Figures

Figure 1	17
Figure 2	22
Figure 3	24
Figure 4	29
Figure 5	30
Figure 6	33
Figure 7	39
Figure 8	41
Figure 9	45
Figure 10	55
Figure 11	56

Figure 12	57
Figure 13	60

List of Tables

Table 1	15
Table 2	15
Table 3	40
Table 4	63
Table 5	63
Table 6	66
Table 7	67
Table 8	68
Table 9	70

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signed

Khounkham Phengthlangsy

Date: 21 July 2021

Acknowledgements

Above all, it has been the most wonderful life in New Zealand as a fully funded scholar which is the largest dream in my life to gain the Master's degree. This would never come true without the cordially tremendous support from many of those that I am unable to list. Essentially, without constantly academic guidance and assistance from my supervisor **Jim Buchan** and **Prof Stephen G. MacDonell** throughout the entire process of this research, I could never have succeeded in this critical journey. Please accept my deepest thanks.

From the bottom of my heart, I would like to express my sincere gratitude and grateful appreciation to the people and government of New Zealand for granting this tremendous opportunity to pursue this master's degree in the most amazing country in the world. The programme not only offers me a chance to master my academic field that could drive to an occupational success in near future, it allows me to enable a window of knowledge for my angel daughter joining me in New Zealand and she has a marvellous time to experience such a great primary education system here with me. My deepest gratitude to International Scholarship Office Team at AUT, they are not only provide professional supports but also offer warm advice to drive pass all critical concerns throughout the journey.

Finally, I might even not be able to apply to this New Zealand Aid scholarship without inspires, heroic sacrifice, and greatest support from my wife. She has to work harder than usual for filling up all family's expenditure and bills as well as looking after my little son in Laos during my absence from earning family income and staying with my little daughter for having an academic life experience in New Zealand. Staying away from the loved ones could experience difficult time in a life certainly a single mum to taking care everything. At the same time, it is the meaningful two years to prove how much the love we all have to each other in the family. A big thank to you Latsamy Phanchackhanh for all loves, and the most fabulous dedication for our kids and family, I will be forever in your debt.

CHAPTER 1: INTRODUCTION

In this chapter the background, motivation, and the purposes of the research are explained. An overview of the purposes of onboarding, significant aspects, scenarios, and expected outcomes are presented. In addition, the main aim of this research, research scopes, research questions, expected contribution, and structure of this thesis are stated in the final section of this chapter.

1.1 Motivation and Background

For most organizations involved in software development, the development work is done by small teams. It is not uncommon that the individual team members change for one reason or another. It may be that a team member leaves the team and perhaps the organization and needs replacing. Or it may be that a team needs additional resources to get their work done, or even that a new team is created from scratch. The new team members may be new hires, new to the organization, or recruited to the team internally. Whatever the reason and situation, sourcing, recruiting, and transitioning new members to the team is a task that is critical to continuously maintaining organizational plans, especially work quality. The process of integrating new hires to the team is commonly referred to as organizational socialization or onboarding in the literature (Batistič & Kaše, 2015; Bauer & Erdogan, 2011; Britto et al., 2018; Kowtha, 2018; Perrot et al., 2014). A recruitment and selection process of new hires would be effective or not depend on the how much understanding and facilitation by the organization is (Perrot et al., 2014). Employees who have working attitudes and remain in the organization for a longer time have generally had successful organizational socialization, whereas the early departure of employees from the organization or low productivity on the job can be the result of ineffective onboarding, which consequently leads to a cycle of recruitment and selection for the company, and this costs time and resources (Bauer & Erdogan, 2011). Judith (2001) pointed out that, in the findings of a survey, almost all organizations agree on the onboarding progress being a significant component of the effectiveness of newcomers' socialization.

A number of studies have investigated how an onboarding program is implemented, its effectiveness and barriers, and created a recommended workflow for practitioners. For instance, Britto et al., (2020) conducted research to evaluate the onboarding outcomes of large-scale, globally distributed software development projects, they found that having a remote mentor, using a formal training approach, and assigning over and distributed workload in early stages of team onboarding, negatively impact the outcome of onboarding results. Fagerholm et al., (2013) studied the onboarding process of virtual open-source software teams. The study reported that onboarders with assistance from mentors performed more effective than the ones without support in earlier stages of onboarding programs. This indicates that although some strategies are required for the onboarding like the mentors, different onboarding approaches like formal training still needs to be carefully applying for the program in the onboarding

programs in software development contexts. Buchan et al., (2019) synthesized essential onboarding goals/techniques for an effective team onboarding in Agile software development. The study reported that a variety of possible team onboarding processes/techniques could be applied to different situations of new team members as well as characteristics of teams, work requirements and organizational goals. Thus, all these factors need to be thoroughly analysing in the onboarding process, especially a personalized onboarding design process which the aim of this research study.

Apart from spending time studying his/her new roles, roles which relate to other team members' roles, working culture, and the key values of the company, the greatest challenge is acquiring knowledge of and understanding the huge amounts of complex legacy code of an application (Britto et al., 2019). Furthermore, inappropriate onboarding techniques/processes sometimes are applied to certain newcomers, and unexpected outcomes are produced. Therefore, the team designers or project managers need to design the appropriate onboarding processes and techniques for newcomers and generate the most productive results (Buchan et al., 2019b). With the many different ways of global software development, the decentralization of projects to participants in a wide range of geographical locations and from a wide range of cultural backgrounds (Fagerholm et al., 2013), personalized team onboarding processes and techniques could be designed and applied in different situations. In addition, monitoring and evaluating newcomers is essential to ensure that they can become a part of the team and able to contribute the expected amount of effort rapidly. To assist in meeting these requirements, a support tool can be designed/developed to assist the team designers/project managers with designing a team onboarding program in order to filter the appropriate onboarding techniques and apply appropriate resources to facilitate new team members during the onboarding process in ways that suit a particular onboarding situation, especially in Agile software development teams or small-medium software development team.

1.2 Motivation for Building the Tool

With the results of the previously mentioned studies of the factors influencing high performance team, and of onboarding programs using various techniques in different situations depending upon personality traits and organizational circumstances, an onboarding support tool can be designed and developed to facilitate the design workflow of a personalized onboarding program. It could benefit not only the selection of desired goals and supporting tasks of a personalized team onboarding plan in an early stage, but it could also be used to monitor newcomers' progress and performance as well as maximizing the efficiency of entire software development teams in the long term. The functional requirements of the tool would be based on the results from a systematic study of the literature and suggested onboarding program workflows in Buchan's (2019) report. In an example of building tools to facilitate an onboarding program by Steinmacher et al. (2016). The tools aimed to help newcomers to overcome a

set of onboarding barriers when joining open-source software development projects. For instance, the difficulty of finding a task to start with, the tool allow the newcomers to select a set of easy tasks. Also, the tool facilitates newcomers to overcome communicating issues with the community such as a suggesting message template for newcomers to first communicate with team members or experienced developers. Evaluation outcome of the tools suggested that although they were unable to lower the technical barriers, it addressed many problems which newcomers could handle by employing the tools. However, this tool is just for guiding onboarders while the purpose of tool development in this research study is about supporting designers.

1.3 Research Problem in Software Development Industries

Several studies in the software development context have been conducted from different perspectives to investigate onboarding problems. For instance, an investigation looked at mentors' and newcomers' barriers in onboarding in open-source software development (Balali et al., 2018; Steinmacher et al., 2014). The findings of the study showed a set of barriers and provided possible strategies to overcome the reported problems. For instance, personal and interpersonal challenges (e.g fear of judgement, inability to improve upon criticism, cultural differences) could be addressed by having a mentor and/or team supervisor support. Other facets of onboarding are still requiring further investigation, such as team onboarding design for different newcomers' perspectives or different software development contexts. This is similar to the aim of this study which is a personalized onboarding design process. The findings from those previous studies (a list of barriers and strategies) are significant for this research where they would be a set of factors for process designers to considers when a personalized onboarding process is being planned. Yang (2017) studied on-boarding processes in Agile software development teams and, although a set of desired goals is clearly reported with common suggested activities, this does not go as far as applying that to designing a team onboarding process. This design is required for diverse personal characteristics, the different backgrounds of onboarders, so that the process could be employed for any onboarder and team. With this requirement in mind, an explicitly personalized team onboarding process is an importance of area of study that extends current understanding and practice and could have significant impact on onboarding programs and software development teams' productivity. This shows that the research that is the topic of this thesis is potentially both significant (high impact) and contemporary (not solved).

1.4 Research Method, Outcome and Contribution

To achieve the goal of this study, a Design Science Research process (DSR) is adopted as the overarching approach to guide the research (Peffer et al., 2007). The literature review in Chapter 2 identifies the onboarding problem and its context, which are the first phases of the DSR process. A Systematic

Literature Review (SLR) is then used to extract data from existing literature to inform the design of the onboarding program design process. This personalized onboarding design process is the main artefact of the DSR approach. The evaluation phase of the DSR process is based on simulation of its application using two types of onboarder archetypes. Feedback was from my supervisor and was used to improve the model.

It is expected that the results of this thesis will contribute to professional practice in a couple of ways. First, it is hoped that this thesis will raise the awareness of team onboarding among practitioners, and the need to design onboarding programs systematically and proactively. Second, the work in this thesis provides onboarding program designers with an empirically based onboarding design process that will allow the creation of personalized programs to suit the specific characteristics of the onboarder and teams. These contributions should benefit the productivity of onboarders' transition into teams in the software development context.

Researchers may benefit from this work also by building on the conceptualization of team onboarding presented.

1.5 Thesis Outline

This study consists of six chapters. First, the present introduction chapter has introduced the initial motivation for the research, provided background to onboarding problems, research outline, and main expected contribution. The second chapter presents related work on the onboarding process, models, the meaning of onboarding in different contexts from onboarding-related studies. The third chapter presents the research design, research questions, and the methodologies used in this study. The fourth chapter represents the findings and discussion of systematic literature review as well as common team onboarding process, techniques, and goals for different situations in software development teams, investigating and categorizing the onboarding challenges, gaps, and employed activities. The fifth chapter describes the proposed personalized onboarding processes designs based on the current onboarding gaps discovered in the systematic literature review, and used cases. Finally, the conclusion, limitations, and suggestions for future research are presented in Chapter six.

CHAPTER 2: LITERATURE REVIEW AND RELATED WORK

The aim of this chapter is to position my research in the current body of related research and provide a background to the main concepts related to designing a team onboarding program. In contrast, the systematic literature review described in Chapter 4 has the aim of specifically answering the research questions by synthesizing specific data from systematically identified literature.

This chapter describes what is known currently in general about onboarding from different perspectives and in different contexts. It introduces a working definition of onboarding, and the principles of onboarding from a wide range of onboarding research and relevant studies. While this thesis investigates the personalization of team onboarding into co-located Agile software teams, research into onboarding in different contexts (e.g., organizational onboarding, onboarding into virtual teams) may also give insights applicable to my research.

The first section of the chapter explains what onboarding is and provides a working definition, as well as confirming the importance of onboarding programs that are well designed. The next section identifies the different contexts of onboarding research and provides some principles for designing onboarding programs. This is followed by sections that provide an overview of the goals and challenges of an onboarding program as well as related onboarding activities, with the idea that an effective onboarding program will support the achievement of these goals and address the challenges by incorporating those activities. This provides a framework to research the design of a process to create a personalized team onboarding program. The systematic literature review investigates these areas in much more detail and applies the findings to describe a specific process for designing a personalized onboarding program.

2.1 Definition of Onboarding and its Purposes.

An onboarding program is a common practice which different industries apply to orient new hires into a team. A number of research studies refer to this process as “organizational socialization” where the goal is to support new employees in learning to adapt to the organization culture and team norms (Batistič & Kaše, 2015; Bauer et al., 2007; Wanberg, 2012). Similarly, Stein and Christiansen (2010) described it as a program that integrates a newcomer and their culture into the working culture of the team and likens it to putting a new costume on a newcomer. The onboarding program offers support for the new team member to understand organizational goals, share team knowledge, and develop the skills for their role. Ross et al., (2014) described an onboarding program as the process of gaining and fully understanding information about and the ideas of organizations. Similarly, Bauer & Erdogan (2011) defined an onboarding process as the process to allow organizational outsiders to become organizational insiders.

It can be seen that an onboarding program has many facets. For the purposes of this thesis I am defining an onboarding program as a package of practices, policies and procedures, formal or informal that supports the newcomers' early experience of behavior adaptation and learning in their new environment (Klein et al., 2015; Moe et al., 2020) as well as their socialization and performance in the new workplace (Britto et al., 2018). In the software development context, the team onboarding phase is a vital step for bringing new developers into the team and supporting them in becoming proficient and productive software developers (Buchan et al., 2019a; Pham et al., 2017)

Not all onboarding programs are equal, however. The design of an onboarding program can determine how effective or ineffective it is in achieving the requisite goals. An effective or poorly designed onboarding program can have a significant effect on the productivity of the team and the onboarder. There is a significant body of research which supports the idea that a new team member will integrate into an existing team and become a productive team member faster with a well-designed onboarding program (Bauer & Erdogan, 2011; Britto et al., 2018; Fagerholm et al., 2013; Steinmacher et al., 2014; Viviani & Murphy, 2019b). A successful onboarding program guides the primary rules and new practices which can improve the quality of onboarding in any industrial field (Stein & Christiansen, 2010). In contrast, the impact of poor onboarding of team members could be high, resulting, for example, in a lack of clarity in their responsibilities and role in the organization or team (Finn, 2006), leading to poor team productivity or even the onboarder leaving the organization. This illustrates the need to understand how to design an effective onboarding program for a team.

2.2 Different Onboarding Contexts

With the revolution that has taken place in software development industries, a wide range of onboarding research has been conducted in different software development contexts. Based on the explorations of the researchers involved, the following are the most common onboarding contexts currently found in either software development industries or general onboarding research:

- 1) A general onboarding of new hires into organizations (Bauer et al., 2007; Klein et al., 2015)
- 2) Onboarding programs in open-source software development projects (Balali et al., 2018; Davidson et al., 2014; Diniz et al., 2017; Dominic et al., 2020; Fagerholm et al., 2014);
- 3) Virtual onboarding programs for virtual teams/globally distributed software projects (Britto et al., 2020; Britto et al., 2018; Cunha et al., 2020; Moe et al., 2020; Smite et al., 2014) and
- 4) Onboarding programs of new hires into a team of small/medium software development companies (Hoffström, 2019; Viviani & Murphy, 2019a).

The findings of these above-mentioned onboarding contexts, particularly software development teams/projects, provide an essential set of onboarding aspects (onboarding goals, supporting strategies,

problems in different perspectives) that would be mainly used in the onboarding design process of onboarding in this thesis. The following sections describe these onboarding categories in general.

2.2.1 General Onboarding Programs of an Organization

Different industries may deploy various different tactics in influencing a new hire to learn the requisite social knowledge and understand key roles in order to be more productive and able to contribute to the efforts of the team (Kowtha, 2018) and increase retention for organizations (Stein & Christiansen, 2010). Organizational culture, roles, goals, and the personality traits and/or barriers of new hires are considered to be significant components for employers designing an appropriate mechanism to deploy into an onboarding process. For instance, office politics, pressure to achieve a quality of work assignments due to limited resources, and time constraints could negatively affect an onboarding program of new instructional designers in an organization (Rabel & Stefaniak, 2018). In medical fields, some desired goals are similar to other industries, but they are dependent upon the context, for example retaining a clinician, ensure the quality of care provided to patients, and promoting well-being, since if clinicians are happy and, healthy then they would deliver excellent care to patients (Morgan et al., 2020). In the software industry, although basic expected goals and techniques employed are similar, a set of certain goals and techniques may be different especially when diving in-depth into particular areas, for example, understanding how to code and test to a team's expectation where a mob programming technique might be the most appropriate approach to achieving these goals (Buchan et al., 2019a). In general, three different styles of onboarding could suitably apply for different organizational contexts (Babajide & yagoub, 2019), as follows:

- 1) instructor-led classroom program(Babajide & yagoub, 2019; Klein et al., 2015): this is a traditional type of training that is not only part of the onboarding session but also a regular internal session in organizations where instructors guide the session through a prepared agenda.
- 2) computer-based and e-learning training (Babajide & yagoub, 2019; Britto et al., 2020; Buchan et al., 2019a): this is a cost-related approach in which employers rely on applying technology as a tool to drive the session. Videos, online courses and assessment sections and the utilization of different tools may be included in this training.
- 3) rotational onboarding program: this approach is classified into two types:
 - a) an individualized strategy that could be considered for deployment based on new hires' elements or characteristics such as individual, informal, random, variables, disjunctive, and divestiture
 - b) an institutionalized strategy that is considered for use when new hires have comparable characteristics.

While the first two strategies may be useful for boot-strapping common skills or knowledge acquisition needed in the organisation and maybe even team, they do not consider the potential diversity of experience of onboarders and the different needs of different teams” It is not clear if there is a gradual support of skills and knowledge and the needs change of the onboarder or a feedback mechanism to check progress. These are all aspects of a personalised onboarding program design considered in the design approach of this thesis. The third strategy has more aspects of personalisation to the characteristics of the onboarder but the details of how that informs the onboarding program design are not clear. The system proposed in this thesis makes this link between onboarder and team characteristics and the onboarding goals.

2.2.2 Onboarding Programs of Open-Source Software Development Projects.

Many open-source software development projects are emerging where many developers across different locations contribute their efforts to share various functions and advantageous information in order to address a variety of challenges in software industries (Abdulkadir et al., 2021; Balali et al., 2018; Fagerholm et al., 2013). Onboarding tasks to bring developers with diverse characteristics into a project is a predominant task. A number of studies related to onboarding programs in open-source software development projects have been published, mainly analyses of its methods, barriers from the different perspectives of involved parties, and evaluation of deployed strategies (Balali et al., 2018; Steinmacher et al., 2014). One of the main desired goals was expecting newcomers to contribute to the project (Balali et al., 2018; Hannebauer & Gruhn, 2017; Silva et al., 2020; Steinmacher et al., 2014), although a set of risks were reported. Steinmacher et al. (2014) investigated a set of barriers for newcomers who were being onboarded onto open-source software projects, and a group of newcomers’ limitations were reported such as a lack of technical background, previous knowledge and communication skills, and their diverse cultures, while organizational efforts were also needed to improve such matters as reception issues (impolite behavior, delays, or answers that were too complex), unclear and/or out of date documents, and local development environments. An approach to minimize those reported risks was using a tool to support the process where newcomers were able to share their work as well as tracking daily activities inside the system and supporters could provide feedback at the same time (Steinmacher et al., 2016; Steinmacher, Wiese, et al., 2015). While social barriers were positively addressed by using the tool, technical hurdles were still unable to be solved and newcomers still encountered many programming difficulties such as understanding the architecture of the code, and locating the right place to find an issue in source code (Steinmacher et al., 2016; Steinmacher, Wiese, et al., 2015).

Even though a list of barriers and some suggested tasks were set out, as well as some supporting tools were developed and employed throughout the onboarding periods in previous research studies, the design process of a team onboarding program, especially one addressing the particular characteristics

of newcomers, has not been explained in this open-source software context. Therefore, a systematic design workflow of a personalized team onboarding plan for newcomers is required to permit further exploration.

2.2.3 Onboarding Programs of Virtual Teams/Globally Distributed Software Projects

With the revolution in technology, not to mention the effects of the Covid global pandemic, distributed global software development teams have become common in organizations (Cunha et al., 2020). However, virtual onboarding new software developers into the teams is also a critical task even if the team work is being done remotely. In this onboarding category, researchers have explored the particular practices, techniques, methods and technologies of onboarding which are employed by companies (Britto et al., 2020; Britto et al., 2018). In the three case studies by Britto et al. (2018), the onboarding workflow was similar in terms of the conduct of some of the main activities in both local and central workplaces, while some activities were done differently due to specific cultures. Overall, the onboarding model was reported as follows:

- a) recruitment was the first activity of the process of onboarding in which related senior developers from organizations participated in interviews to ask some technical questions in order to screen appropriate newcomers before providing an orientation session for a week.
- b) a formal training session was needed if many people were recruited, for roughly three months, with the focusing on transferring legacy, but it depended on the amount of work.
- c) coaching and support were provided remotely which took place for possibly two months, and was conducted by assigning senior developers or providing online bootcamp programs.

However, some weaknesses were found. Some main activities were deployed centrally while they were not implemented locally (Britto et al., 2020; Britto et al., 2018), and this could lead to some confusion. In addition, other critical issues for virtual onboarding included a lack of trust among sites due to the diverse cultures of newcomers (Moe et al., 2020), while trust is one of the main factors for success in virtual team performance (Judith, 2001). Another challenge was collaboration, as team members a spread across different geographic regions, but a lot of dependency tasks to be implemented and the transfer of knowledge between sites was also critical (Moe et al., 2020). By contrast, Judith (2001) pointed out that while there are a set of challenges to forming a team in different geographic environments, this could offer opportunities for team members to have intensive interaction through dialogues for building a shared future. Using online tools was recommended to support the process of onboarding in order to keep track of a participant's activities and for mentors to give feedback (Britto et al., 2020). A wiki was introduced as an alternative portal for supporting the learning process of newcomers in global software development, although some access and usability difficulties existed (Cunha et al., 2020).

Although some critical concerns were discovered in the virtual onboarding context of different case studies, the onboarding workflow is still very general, which is similar to the open-source software context where no onboarding plan designs reported. Hence, the design process of the program should be investigated, particularly team onboarding programs for different newcomer characteristics. Therefore, a systematic design workflow of a team personalized onboarding plan for newcomers is required for further exploration and is the main purpose of this thesis.

2.2.4 Onboarding Programs of Small/Medium-Sized Software Development Projects

While several technical (e.g. working with legacy code) and social skills are desired by large software development organizations, an effective onboarding program is significant in integrating new hires into the teams quickly (Britto et al., 2019; Minghui & Mockus, 2011). Similarly, in order to maximize effectiveness and efficiency in small/medium-sized software firms, an effective onboarding program is also required (Viviani & Murphy, 2019b) to ensure continuity of high performing teams. Despite the observation that a multitude of small software teams have emerged to provide diverse business solutions across the world (Escobar-Sarmiento & Linares-Vásquez, 2012), there is a lack of published research that specifically investigates onboarding programs in small/medium-sized software development companies. Some challenges reported in this category included a lack of resources to fulfil an integration process for new team members (Allison, 2010; Viviani & Murphy, 2019b), while software engineering techniques were critical tasks for small software firms (Mishra & Mishra, 2006). A buddy programming concept was reported as the most productive mechanism to onboard new members into teams, and two different ways were discussed by Viviani and Murphy (2019) in order to bring new developers to understand systems' source code. A few companies provided a proper session for coaching newcomers on the source code's structures, while the most firms expected new developers to explore, review, and attempt to write source code by themselves, and relied on the buddy concept to support them when they encountered problems (Buchan et al., 2019a; Pham, 2014) as well as ad-hoc assistance by senior developers.

Apart from the four onboarding contexts set out above, a program for onboarding new developers into an Agile software development team is considered to be an onboarding characteristic in which onboarders need to explore the Agile method and adjust themselves to the Agile way of development. The Agile method has quick development cycles, an iterative process, simple design, peer reviews and critical participation of onsite stakeholders for handling a complex, fast-moving and competitive marketplace (Ramesh et al., 2010). Mentoring is a crucial mechanism for supporting newcomers getting into the Agile workplace. Mentors are experienced persons who provide instruction, counseling, and interaction in a way that has not existed in normal guidelines of onboarding programs (Kumar et al.,

2016), as well as regular and specific support and face-to-face meetings with onboarders (Buchan et al., 2019a).

2.3 Onboarding Challenges, Potential Solution

A number of studies investigated onboarding obstacles, and factors which led the onboarding outcomes to be unproductive, or newcomers to prematurely leave the organization after a period of employment, which resulted in a need to repeat a recruitment process sourcing newcomers. This costs a lot of time and budget. Onboarding difficulties could be incurred by both experts and newcomers (Yates, 2014). For instance, research by Balali et al., (2018) generated four main categories of onboarding barriers (personal, interpersonal, process, and technical) which affected mentors and mentees in open source software projects. Within these four categories, there are 44 sub-onboarding barriers, of which 25 impacted newcomers only and 10 impacted mentors only, while nine barriers affect both mentors and newcomers. Similarly, Steinmacher et al (2014) identified seven categories of barriers, some of them are coding issues, documentation problems, newcomer behaviour, and technical knowledge which were encountered by newcomers. The results also pointed out that the onboarding processes of different software projects still required investigation on matters such as the implementation of a specific ERP system which requires some customization, or Agile software development teams working in the same project but being located in different geographical locations. For instance, (Britto et al., 2018) investigated globally distributed open-source software development projects, and one of the findings showed that some onboarding functions are executed centrally, while others are only executed locally, which makes it difficult to control onboarding results. The results also indicated that distributed projects with legacy code could be challenging due to the difficult connection between onboarded newcomers and the original developers. In addition, virtual mentoring, a formal induction program which is not appropriate to the socio-cultural characteristic of newcomers, over-distributed workload in the early stage of onboarding program, and unstable teams are associated with the productiveness of the onboarding programs (Britto et al., 2020), while the newcomers struggle to understand code written by others (Yates, 2014). Fagerholm et al (2013) also stated that virtual team environments and organizational management itself could be a challenge for onboarding programs. Some other challenges may be incurred in open-source software projects after the onboarding stages, such as the difficulty of keeping new distributors, and many developers not turning themselves into actual distributors (Pinto et al., 2018)

In order for newcomers to quickly thrive in the team and productively manage the onboarding challenges that have been mentioned, a high level of trust among team members seems to be one of essential aspects for a high-performance team (Cauwelier, 2016; Edmondson, 1999; Faraj & Sproull, 2000; Judith, 2001; Moe et al., 2020). Teamwork (Weimar et al., 2013) is another significant factor creating the productivity

of a software development team, while building teamwork requires interpersonal communication, coordination of expertise, cohesion, cooperation and trust. Apart from the formally oriented induction program (Britto et al., 2020; Kim et al., 2015; Milligan et al., 2013) mentoring and coaching seem to be the most practical approaches to guide new comers in the onboarding program of software development teams (Balali et al., 2018; Canfora et al., 2012; Fagerholm et al., 2014; Fagerholm et al., 2013; Kumar et al., 2016; Major et al., 2007; Pham et al., 2017; Wanberg, 2012). Buchan et al (2019) categorized the techniques to be implemented for supporting different onboarding goals, and found that mentoring, online communities, and support from project members are the most frequently used techniques. In the circumstances of the handover of complex legacy code to newcomers during onboarding programs in globally distributed software projects, the detailed plan must be well prepared in advance and must spare an extendable period of time for mentoring (Britto et al., 2019), while an extensive code review and buddy programming seem to be the most appropriate and efficient approaches to guiding on-boarders in mid-sized software development companies (Viviani & Murphy, 2019a).

While these studies report specific onboarding challenges and suggest techniques to address these challenges, there is no overall onboarding process suggested that will address the challenges. The aim of our approach is to create a design process that will result in an onboarding process that addresses these challenges

2.4 Onboarding Goals in Software Development Industries

For onboarding programs in different industries, goal setting is a step to be defined at an early stage in order to use it as an indicator as to whether or not the execution of an onboarding program would be successful. A number of studies have been conducted on how to bring newcomers onto a team in effective ways especially in software development organizations. An effective onboarding program is a process to help onboarders obtain an understanding of team norms, company culture, job responsibilities, and the quality and expectations of the work, to assist in development and adaptation throughout a working process, and to emphasize the short-, medium- and long-term work structures, main purposes and work implication (Jensen et al., 2011; Singh et al., 2020; Stein & Christiansen, 2010; Steinmacher et al., 2014; Steinmacher et al., 2012) which were considered to be a set of onboarding goals. Buchan et al (2019) synthesized onboarding goals and activities from published onboarding research, and then mapped common onboarding activities to how strongly they supported the achievement of these goals based on a Repertory Grid study of recent onboarders to Agile software development teams. The research identified 11 main onboarding goals and 25 activities which the practitioners identified as supporting the achievement of each onboarding goal to different degrees. The

results mapped each onboarding activity to each onboarding goal with a high, medium or low level of support for the goal achievement.

From the perspective of high-performing teams, these onboarding goals are associated with the components of high-performing team factors which exist in effective working teams. Ideas for new onboarding goals may be discovered by considering such factors of high performing teams. For instance, according to one of the Google research projects, called Project Aristotle in the year 2012, 180 different active teams collected in both quantitative and qualitative data in the study. The findings revealed the top five significant factors of a high-performance team as follows:

- 1) “Psychological safety: Team members need to be open-minded and trust each other,
 - 2) Dependability: The completion of high-quality work on time by team members”,
 - 3) Structure and Clarity: The goals, roles and action plan must be clearly defined in a team”,
 - 4) Meaning of Work’: Tasks should be utterly important for team members,
 - 5) Impact of work: Team members agree that their work matters and creates change to the team”
- (Rozovsky, 2015, p.1).

One of these key factors, mutual trust within the team, was also reported by Muethel et al (2012) as an influential factor in the performance of information system projects. Mutual trust significantly improved team performance. Further, when team members trust, feel safe with, and are not scared of taking risks in front of others, difficult issues/problems were then freely raised for discussion in meetings and team members got a chance to brainstorm and share appropriate viable solutions. That was how the team members actively participated in an active learning session within a team and eventually affected the team performance (Cauwelier, 2019). Edmondson (1999) investigated 51 work teams in a manufacturing company in which a model of team learning was presented and tested in multimethod field study. The findings showed that team psychological safety (a shared belief or trust) among team members is associated with learning behavior (asking for feedback, sharing information, seeking help, discussing errors, and testing) when the team psychological safety is manageable. Further, learning behavior is considered to be an associated aspect sitting between the team psychological safety and team performance. Also, the results indicated that team outcomes are effectively shaped by an integration of team structures (contextualized support and team leader coaching), and shared beliefs. Similarly, effectively functioning within the team not only means adding people and simply assigning tasks to individuals, but also requires: 1) a deep understanding of the main objectives and the commitment of the team’s members to the target mission, 2) expecting more quality results than average results from the team, 3) transparency, accountability and member’s responsibilities being well-known, 4) a diversity of expertise to support the abilities of other team members, and 5) having interdependent trust between members (The Society For Human Resource Management, 2020).

Consideration of these factors of high performing teams suggest possible onboarding goals for a newcomer to a high performing team: the onboarding program should promote these 5 factors for the new on-boarder situated in the team.

To achieve different onboarding goals in software development teams, different team situations may require different approaches based on company onboarding strategies, newcomers' characteristics, and available team facilities. A number of techniques may be implemented in order to achieve the goals. Initially, organizations must plan prior onboarding strategies (Britto et al., 2019), and set the ultimate goals to be achieved, as well as indicators. Initial environments were significant for new developers to develop the broad technical and social skills, obtain effective training, and learn the company's strategies when joining the project (Minghui & Mockus, 2011). Specifically, team members could assist newcomers in getting more quickly into the source code of development environments by installation processes in order to address some onboarding concerns such as a long IDE installation/configuration process, learning unique tools, and inadequate documentation (Yates, 2014). Aside from that, Viviani and Murphy (2014) and Buchan (2019) pointed out that some of the effective onboarding practices in a software development team were mentoring, a buddy approach (pair programming), and the extensive use of code reviews to guide newcomers into becoming a rapid productive part of the team, while conducting a workshop to learn about company-related products is recommended to focus on technical and organizational knowledge (Hoffstrom, 2018). On top of that, Buchan (2019) systematically synthesized onboarding goals, and supporting techniques to be conducted by an engaged person in order to achieve those organizational onboarding goals. The mapping between goals and techniques is stated in his study report too. From the perspective of newcomers, Lui (2019) investigated how the personalities and previous work experiences of new employees influence their onboarding plan. The research focused mainly on the expectation of graduate employees. The onboarding activities expected by new graduates in the onboarding process were classified and arranged. The most common requirement from the graduates' point of view was help from team members, a preference of 57% of 58 interviewed graduates, while a training session/workshop was the second preference of 40% of interviewees. In addition, online learning was reported as the top need for onboarding resources while mentoring was the most significant aspect for effective onboarding process in open-source software projects (Fagerholm et al., 2013). Mentor selection to fit with personalized characteristics of newcomers is also important in order to guide newcomers to becoming a productive team member. For instance, Canfora et al (2012) proposed an approach called "Young and new comer developer assistant" or "Yoda" to identify the most appropriate mentors for on-boarders. Despite the results from Yoda showing that the top recommended mentors were not always the most appropriate for newcomers, the potential usefulness of Yoda was as a recommended tool to help software team leaders in the selection of appropriate mentors when new on-boarders are to be onboarded into a project.

Table 1*The List of Onboarding Goals*

No	Goals
1	Understand and fit in with company culture
2	Understand and fit in with the team norms
3	Understand and meet others' expectations of one's own role's responsibilities
4	Understand the responsibilities, expertise and authority of other team members
5	Understand what work to do and when
6	Understand how to code and test to the team's expectations
7	Understand and meet the team's standards of work quality
8	Understand and adopt the Agile mindset
9	Know how to use Agile artefacts and techniques used by the rest of the team
10	Understand the short-, medium- and long-term work structures, aims and implications
11	Understand the product/project domain knowledge and terminology

Note. Reprinted from “Effective Team Onboarding in Agile Software Development: Techniques and Goals”, Buchan, J., MacDonell, S. G., & Yang, J. (2019). Effective team onboarding in Agile software development: techniques and goals [Conference]. 1-11. <https://doi.org/10.1109/ESEM.2019.8870189>

Table 2*The List of Supporting Onboarding Activities for Desired Goals*

Onboarding Activities/Tasks	Description
Mentoring	Assign the right mentor who has the experience to regularly support, meeting and interact with newcomers.
Online communities	Searching online communities such as Stack Overflow to find answers to specific technical questions.
Peer support	Ad hoc opportunities to ask peers (in and outside the team) for information or guidance. Usually, face-to-face. This included observing others as they worked or met.
Team socializing	Interacting with other team members in a social setting (not related to work tasks).
Training course	Attend a formal course to achieve specific learning objectives or certification related to work. May involve availability of an “education stipend”. Also includes online courses.
Review code	Analyze and understand relevant existing source code. Attend code reviews. Access to code repository.
Internal documentation	Documentation capturing local knowledge about data structures, algorithms, and control flow of the project. May also include product information.
Product overview	A presentation, video or similar that shows the functionality and features of their product as well as the business value.
Pair program	Develop with another developer at one workstation, swapping between driving and navigating regularly.
Stand ups	Have regular team stand-up meetings as described in Scrum or adapted.
Simple tasks	Do tasks that are low risk and technically unchallenging, but provide experience with tools, process, technology, team norms.
Self-learning	Learn about libraries, tools and techniques with free access to books and online courses through sites such Lynda, Pluralsight, Udemy, Code Academy, MSDN.
Induction	Learn about the company's history, beliefs, values, long-term goals, and company structure as well as safety, security and health and job responsibilities, accountability and progression.

Knowledge database	Access and contribution to a local knowledge database such as a wiki which may store complex structured and unstructured information. This may relate to product information, design decisions, testing architecture, coding standards.
Team Leader support	Ad hoc assistance from the Team Leader (may be Scrum Master, or Project Manager) answering questions or explaining decisions.
Course on Agile	Learn about the Agile way of working by attending a course (usually third party but may be run by internal coaches).
Team retrospective	Review challenges and learning with the team and learn from their challenges and learning. Some teams do this as part of their sprint retrospective meetings.
Review plan	Review the longer-term plan for the project to understand what has been done and is coming up.
Attend conference	Learn from others by attending a relevant technical national, regional or international conference.
Set expectations	Expectations about onboarding activities and goals are explicitly discussed and set before onboarding and reviewed during onboarding.
Electronic communication	Get assistance from others in the organization through electronic communications such as email, chat, social media.
Meet with other teams	Face-to-face meetings with other teams in the organization. May be at different branches geographically separated.
Location map	A diagram showing the distribution of every staff member on the floor. Information about staff such as authority, expertise and department is also attached to the floor map.
Checklists	Given checklists to assist with remembering aspects of work. For example, a checklist of points to look for when reviewing others' code.

Note. Reprinted from “Effective Team Onboarding in Agile Software Development: Techniques and Goals”, Buchan, J., MacDonell, S. G., & Yang, J. (2019). Effective team onboarding in Agile software development: techniques and goals [Conference]. 1-11. <https://doi.org/10.1109/ESEM.2019.8870189>

2.5 Recent Onboarding Models and Concepts

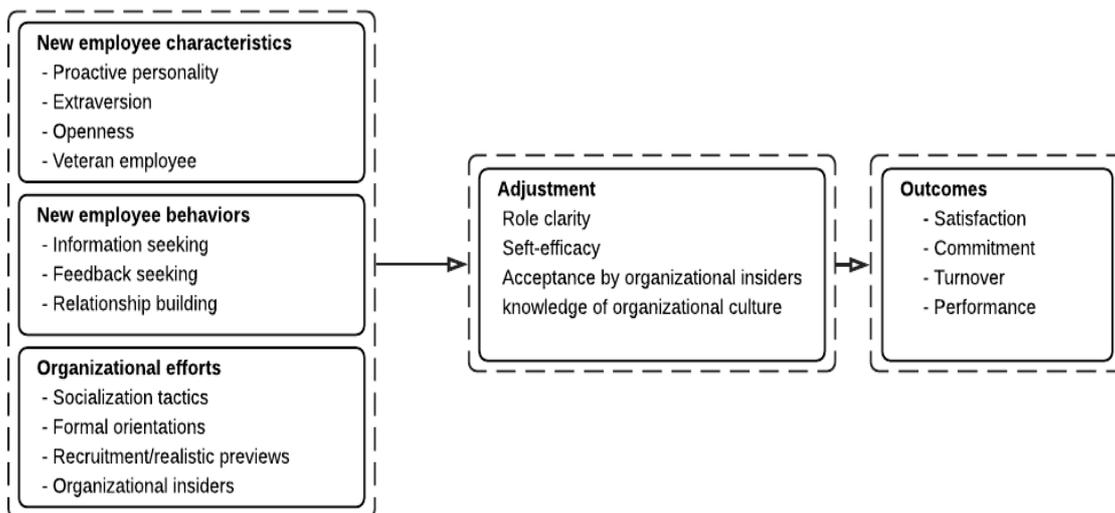
An onboarding program could use various concepts based on the socio-cultural background and aims of an organization. As an onboarding program usually refers to a period of time used to lift up new hires to become productive team members and able to contribute their efforts to the value creation of organizations (Bauer & Erdogan, 2011), the program also gradually adds a little more value than the traditional orientation programs. New hires are guided in a company introduction session within a half-day to learn about company history, staff policies, benefits, and code of conducts. Basic job directions are given to new hires, who are introduced to peers, supervisors, and leaders, and gain and understanding of and access to the available resources of the organization (Stein & Christiansen, 2010). The suggestion from researchers is that employees could take roughly three months to tailored themselves into a new job (Bauer & Erdogan, 2011). A number of empirical studies have conceptualized some practical onboarding models with detailed instructions on how to apply them into actual onboarding practices such as the onboarding models of Bauer, Jone, and Van Maanen and Schein. Bauer’s model will be discussed in the following section, particularly in regard to the characteristics and behaviors of newcomers and organizational roles

2.5.1 Bauer and Erdogan's Model

In the study of the effective onboarding of new employees by Bauer & Erdogan (2011), a number of factors influences the success of onboarding. The authors grouped those factors into three main categories which are: new employee characteristics/individual differences, new employee behaviors, and organizational efforts. These aspects are of paramount importance for newcomers' adjustment in order for them to thrive and satisfy the expectations of the onboarding programme. Figure 1 shows the three main parts of the associated components in Bauer's onboarding model of organizational socialization, with the influencing factors, adjustment, and outcomes of the onboarding.

Figure 1

Bauer's Onboarding Concepts



Note. A summary onboarding process for new hires to join an organization. From “Organizational Socialization: The Effective Onboarding of New Employees”, by Bauer, T., & Erdogan, B. (2011). In S. Zedeck (Ed.), *APA handbook of industrial and organizational psychology, Vol. 3. Maintaining, expanding, and contracting the organization* (pp. 51-64). <https://doi.org/10.1037/12171-002>.

The new employee characteristics refer to the personal characteristics of newcomers and cover educational background, previous experiences, socio-cultural knowledge, and personality traits. As the onboarding is a learning process, these characteristics significantly influence how the onboarding program could be effectively implemented to achieve its ultimate outcomes. The newcomers' behavior also accelerates the speed of becoming a productive team member such as prior searching of information, seeking feedbacks, but team mutual trust should exist too. Edmondson (1999) pointed out that one important aspect of team learning features was sharing different perspectives among the team members. Lastly, support from the organization is absolutely required in order to efficiently guide the entire onboarding process, such as a formal or informal induction session, sticking to all steps and

updating the onboarding plan from time to time as appropriate. For instance, according to Hillebregt (2019), a critical factor in proper onboarding is the level of comfort of new employees within the organization, specifically in regard to the culture supported by the organization.

Adjustment is the second part of the model and measures how fast newcomers are able to adapt into the organization. In this part, the adjustment consists of the following aspects: role clarity (clear responsibility and understanding on assigned tasks), self-efficacy (how confident employees could perform their duties), acceptance by organizational insiders (satisfaction of peers or supervisors who are currently working in the organization), and knowledge of organizational culture (understanding culture of organization). According to Fisher (1986), Feldman (1981), and Bauer, Morrison and Callister (1998), as cited in the study by Yang (2017), the performance assessment of newcomers' adjustment frequently used role clarity, self-efficacy, social acceptance and knowledge of the organizational culture.

In a high performance team, a better understanding of the organization's vision is required to identify an accurate solutions to the outstanding gaps and compliant with future needs of the organizations (Kirkpatrick, 2017, p. 2). Finally, desired outcomes are the third part of Bauer and Erdogan's (2011) model. Outcomes are indicated by the level of satisfaction, commitment, turnover and onboarders' performance at the completion point of the onboarding program. The three main inputs are discussed below as they are prominent in the consideration of a systematic personalized team onboarding design program in this thesis.

2.5.1.1 New employee characteristics

Bauer categorized employees' characteristics as follows:

- a) Proactive personality. The study stated that a new employee with proactive personality traits could be adjusting into new organizations faster due to having a more desirable ability to control surrounding environments by asking a lot of questions related to organizations/teams (Bauer & Erdogan, 2011). Proactive individuals could be more at ease with socializing and developing relationships among team members by joining social activities, which would provide valuation information for their political position to be more effective (Thompson, 2005). Core personal characteristics, such as broad interests, attraction to complexity, intuition, aesthetic sensitivity, ambiguity, and self-confidence, positively and consistently associate with the assessment of creative performance across different domains (Oldham & Cummings, 1996).
- b) Big five factors models traits. The study shown some connections between individuals and organizational socialization, for example, those who were open-minded to experience things are themselves more able to adjust to the job (Bauer & Erdogan, 2011). This is because feedback from colleagues, habits of curiosity of employees were more beneficial to employees rather than

drawbacks. Empirical research has shown that efforts of newcomers are predominant in a socialization process for building relationships with peers and supervisors (Griffin et al., 2000).

- c) Experiences of new employees. How experienced newcomers adopt themselves through the process of joining new organizations is different compare with newly graduated employees. Bauer et al.'s (2007) research presented meta-analysis results that showed new graduates had a stronger self-efficacy than experienced newcomers had. In this situation, onboarding designers could better plan for putting effort into these personal characteristics; while unnecessary tasks could be avoided, more resources could be put into more urgently required areas of onboarders' characteristics. This could be used for identifying the onboarders' characteristics in order to personalize an onboarding program design.

2.5.1.2 New employee behaviors

Employee behaviors is a set of factors that can be used to speed up the success of an onboarding program for newcomers despite organizational support being well-provided for. Active participants could quickly adjust themselves into the designated roles and responsibilities. They could possibly understand company culture and team norms quickly, and be accepted by team members (Batistič & Kenda, 2018). Three elements of employee behaviors were classified in Bauer's model, as follows:

- a) Information seeking. This is a proactive behavior of newcomers in which they conduct an information search by different means to find information relevant to their work. For example in order to access the company's profile at earlier stages, such as reading employee's guidelines, reports, accessing the company website. The frequency of active information seeking of newcomers is associated with their adjustment ability (Bauer & Erdogan, 2011).
- b) Feedback seeking. Newcomers may seek feedback from senior employees and supervisors in order to assess whether or not their work results meet the team's standard or if any improvement is required. Feedback-seeking behaviors are required for getting useful information about an employee's onboarding progress, not only to minimize uncertainty about how to perform a job but also to self-evaluate and correct any mistakes that may previously have been made, or avoid them altogether (Morrison, 2002).
- c) Relationship building. This is also another important behavior to expand networking, in which newcomers could create trust in order to be accepted among team members. This is actually done through the normal events of organization like small groups talking, having lunch, or even a coffee break.

When onboarding designers plan an onboarding program, these general onboarders' goals should be supported for the achievement of the onboarding goals.

2.5.1.3 Organizational efforts

Significant support for fulfilling a personalized team onboarding program comes from the organization. If there is no organizational support, an onboarder may meet a lot of problems in the improvement of their performance (Cauwelier, 2019). Bauer et al. (2007) explained that different organizations could possibly employ various tactics to support their employees in an onboarding program. The organizational support is one of elements of her onboarding models. She classified strategies this effort as consisting of the following aspects:

- a) Socialization tactics. Organizational socialization tactics may be used by organizations as onboarding schemes to structure experiences for newcomers (Gruman et al., 2006; Korte & Lin, 2013; Saks & Gruman, 2011; Van Maanen, 1978). Organizational socialization tactics offer multiple opportunities for newcomers to learn about their organization, their fellow organizational members, domain skills and other related knowledge (Hatmaker, 2015).
- b) Formal orientations. This is where an organization provides a formal session to help employees understand company culture and structures, and make their newcomers feel welcome. A transition process to enable new employees to become a fully functioning member is not separate steps, but it is a continuous process that starts with orientation and training, and is achieved through a continuous support system providing periodical assistance, feedback, and a performance improvement (Sharma & Stol, 2020)
- c) Recruitment/realistic previews. While the recruitment process is a step that involves initially assessing a qualified newcomer joining a team, it is also a chance to provide information for newcomers about what life within the organization is really like. In addition, a realistic preview of jobs and company culture is an organizational strategy to provide accurate work information to newcomers before starting in their position in the organization (Bauer & Erdogan, 2011).
- d) Organizational insider. This covers significant support from members of the organization in various ways, with mentoring programs being an example. The mentor could help newcomers by giving advice, answering questions, explaining a suitable practice from implementing job instructions. Therefore, building relationships in a team is essential as good social relations and having a tight network could significantly influence the ability of newcomers to successfully join the organization (Korte & Lin, 2013).

With these categorized organizational efforts, a personalized onboarding design approach of this research will consider organizational resources by providing some guidelines and information about the organisational resources and commitment needed for different onboarding activities suggested by the

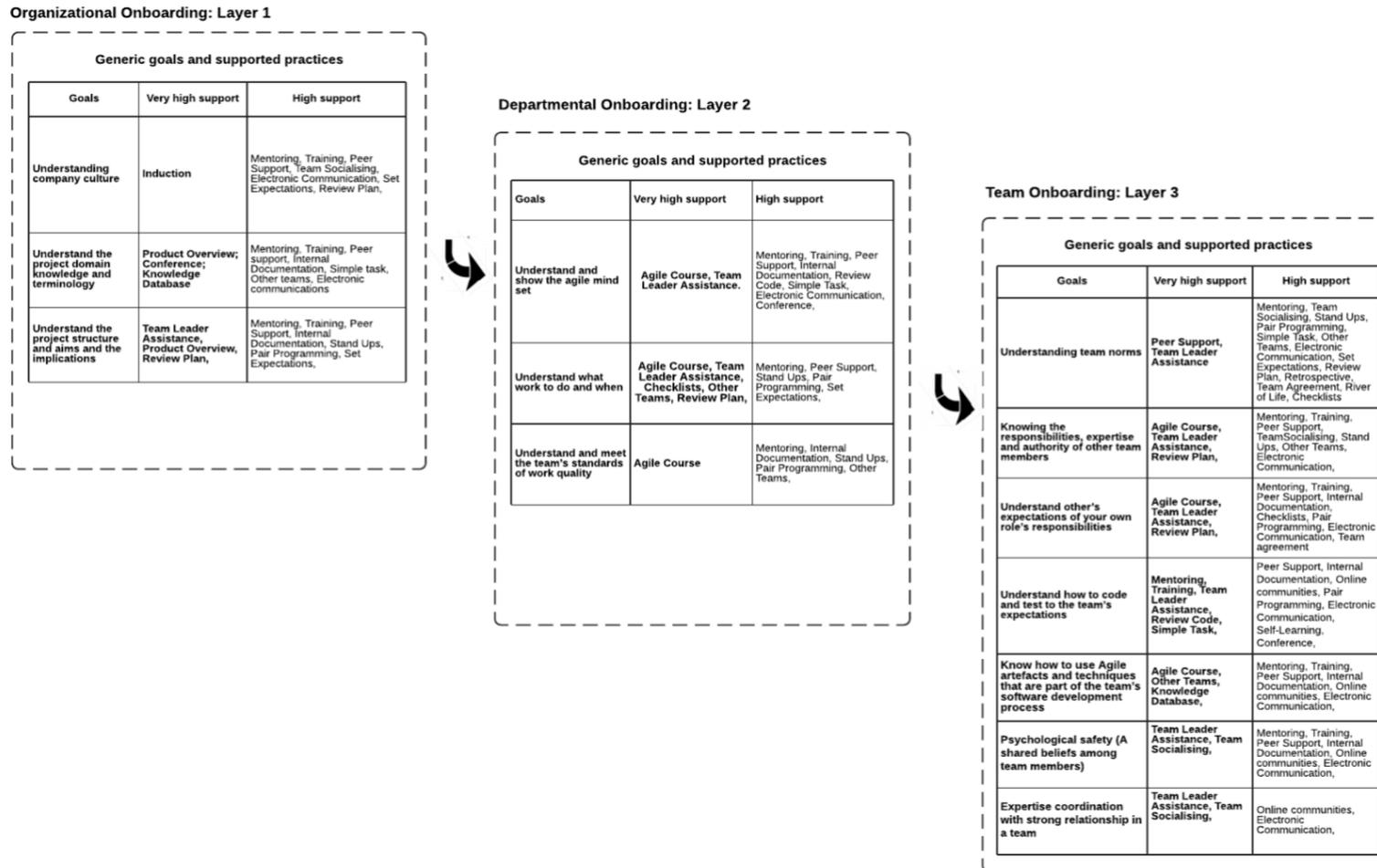
proposed design process. This will enable the designer to exclude any suggested activities that no resources are available for or that need some authorisation before including.

2.6 Organizational, Departmental, Team Onboarding

A number of researchers have studied onboarding barriers and challenges, and have proposed a potential onboarding model in organization levels in order to advise practitioners in different situations. Those studies reveal a set of significant components in the onboarding process; different onboarding models were proposed (Bauer & Erdogan, 2011; Jones, 1986; Van Maanen & Schein, 1977) while the effectiveness of globally distributed legacy projects was revealed through a multiple case study (Britto et al., 2019; Britto et al., 2020; Britto et al., 2018). Buchan et al. (2019) synthesized a list of onboarding goals in Agile team software development as well as support techniques for each desired goal. Despite the fact that those studies presented organizational and team onboarding, the results were not yet categorized explicitly in each layer of the onboarding process. Based on the literature review, it is useful to categorize the onboarding in different levels, namely organizational, departmental and team levels. Onboarding program designers could sort out a set of desired goals and techniques for each layer in order to minimize some overlapping goals or activities in the case of departmental and team onboarding. Figure 2 show an example of onboarding goals and techniques which could be classified in each onboarding layer in an organization.

Figure 2

Sample of Possible Desired Onboarding Goals in Layers



Note. Desired onboarding goals and support activities categorized in layers. Adopted from “Effective Team Onboarding in Agile Software Development: Techniques and Goals”, Buchan, J., MacDonell, S. G., & Yang, J. (2019). Effective team onboarding in Agile software development: techniques and goals [Conference]. 1-11.

<https://doi.org/10.1109/ESEM.2019.8870189>

For instance, if an employee moves from one department to another, some organizational onboarding goals may be not required as he/she has already experienced them since first joining the organization, and they just need to pay attention on the desired departmental and team goals instead. Another example is the movement of employees from one team to another team within the same department; in that case, the desired team onboarding goals only are focused by the onboarders. Nonetheless, if newcomers are completely new to the organization, they should probably participate in all layers of the onboarding program in order to achieve all expected onboarding goals.

In order to successfully complete a team onboarding program, newcomers need to achieve specific goals in the different layers of onboarding programs, like how the team works, and how Agile practices and technologies are used by the team. However, some of these team goals come from the department and company goals since the teams are part of the department and company. Therefore, to understand how the structure of team standards, newcomers should understand departmental, or may be even organizational standards. Therefore, this relationship could virtually affect the design of outcomes in the detail of team onboarding goals. Despite this valid conceptualization of team onboarding providing benefit to the program, the conceptualization for a particular newcomers could be more profitable to the team because the goals are narrow and specifically for newcomers.

Thus, the aim is to design a personalized team onboarding program. The process could be based on company, department, and team requirements, as well as onboarder characteristics, in order to create a set of goals with the supporting activities that have been retrieved from the reviewed articles.

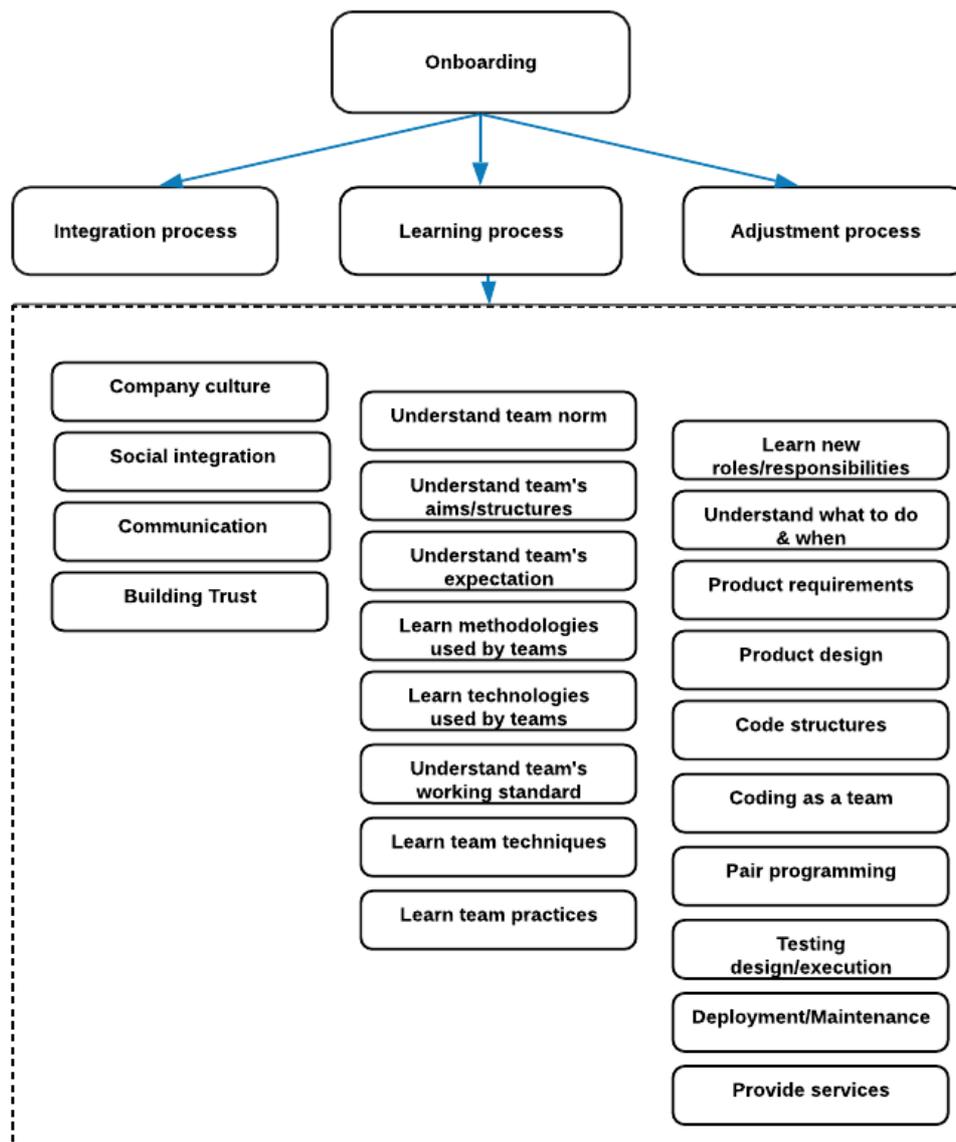
2.7 Onboarding Definition Summary of Literature Review

In conclusion, from reviewing various articles, it can be seen that a general team onboarding program has similar desired outcomes as team onboarding in software development contexts. The team onboarding program could be described as a process whereby newcomers learn about a wide range of aspects of the organization and team, as well as their new roles and responsibilities. It could be seen that there are some key capabilities to be focused in order to achieve the team onboarding programs: a) understanding company culture, b) team norms, c) learning products, d) understanding roles and responsibilities, e) trust development. Some significantly different aspects exist in software development team onboarding, as onboarders need to learn a set of team techniques of software development such as writing code as a team, standards of code structure, learning how to share the progress of tasks by using a version control tools (e.g., github), understanding how to structure the code as a team standard, learning from doing code reviews, learn how to write code with a buddy, learning how to use unit testing packages or other

dependency libraries to validate a group of written codes, and understanding the expectation of team work. In addition, the onboarding process is also considered to be an adjustment process in which the newcomers need to adjust to and reach the new team's working standard, learn all main tools and related technologies (full stack) that are being used by the team, learn any methodologies employed for team management (e.g., Agile, scrum) as well as the team's working style, in order that newcomers are able to contribute their efforts to team product development quickly once the onboarding program is completed, particularly in open-source and virtual software development teams. To summarize the definition of onboarding, particularly in the software development contexts (virtual team onboarding, open-source team onboarding, medium, small, and Agile software development teams), Figure 3 offers a brief summary with some of the key desired goals and sub-goals, as well as expected tasks, skills and practices which newcomers must learn from participating in the team onboarding program.

Figure 3

Definition of Team Onboarding From Literature Review Summary



Although a number of articles have been reviewed and some solid onboarding goals demonstrated in different contexts, outcomes are still not clearly shown in terms of how the onboarding programs could be designed, particularly personalized team onboarding programs, and how this specific set of goals and activities could be planned, especially when onboarding planners need to handle the diverse characteristics of onboarders. It is essential to conduct further investigation into how a systematic design process could be made in order to correctly plan a set of activities and goals for a specific newcomer. This is the main purpose of the thesis – to examine how a systematic personalized team onboarding program could be designed.

From the review of related work, the personalised onboarding program design problem is conceptualised as follows.

1. Identify a large set of onboarding goals as candidate goals to be customized for a particular onboarder and team
2. Identify the relevant details about the onboarder and team that will be the criteria for customizing the goals
3. Filter the onboarding goals to a set personalized to the characteristics of the particular onboarder and team
4. Identify what onboarding activities and strategies will most highly contribute to the achievement of the personalized onboarding goals
5. Identify the challenges in implementing these customized set of onboarding activities and strategies
6. Schedule a program of work that includes these activities and addresses the challenges

The next section describes the research approach and design to address each of these aspects of the personalised onboarding design problem.

CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY.

This chapter consists of the main aims, questions and the design of the research, and the methodologies employed. It introduces an overall research process, using a systematic literature review method for data collection. It starts the description of the data collection by outlining each stage of searching for and filtering relevant articles, and final selection of articles. A Design Science Research approach is presented as the main umbrella for an iterative process of data extraction, analysis, and the design workflow of the onboarding process design and the prototype supporting tool. The methods used to obtain the themes of onboarders' barriers, onboarding activities employed, and desired onboarding goals as well as categorization of the gaps discovered in onboarding literature are explained. Finally, the chapter explains how a personalized team onboarding design workflow could be designed by using a proposed onboarding artefact, which is the main purpose of this thesis.

3.1 Research Purpose and Questions

This study aims to show how a personalized onboarding program can be designed for a software development team, and illustrates the execution of the onboarding plan by proposing an onboarding artefact, and the monitoring of the progress and customization of onboarding when required. A systematic literature review method is the mechanism for gathering the research data from available relevant onboarding articles. Rather than generally reviewing these articles to understand the current onboarding process in software development teams, the five main aspects of the systematic literature review are the focus:

- 1) Investigate onboarding barriers, i.e., the challenges of onboarding programs when joining a new software development team.
- 2) Examine the onboarding activities that are conducted by onboarders as well as organizations over the entire period of onboarding program.
- 3) Classify a set of common desired onboarding goals which both onboarders and organizations wish to achieve.
- 4) Investigate existing onboarding process gaps based on reported evidence from the collected articles.
- 5) Design the workflow of a personalized team onboarding process in order to deal with the gaps discovered for future implementation of an onboarding program in software development teams.

In order to achieve the main aims of this research, the following research questions are answered through the process.

RQ1 How is team onboarding conceptualized?

Based on related research, the common desired outcomes/goals of team onboarding are summarized, categorized, and defined. The definition of common onboarding/off onboarding programs are described, including skills, and knowledge gaps (team/organization culture, employed technologies, supporting tools).

RQ2 How is team onboarding done currently?

Common team onboarding implementation approaches, techniques and tools are presented and how they are applied is described in order to achieve a list of desired onboarding goals/outcomes.

RQ3 What are main challenges of team onboarding and the difficulties/barriers to achieving the goals?

Onboarding barriers/difficulties/challenges are listed and categorized, and an explanation is given of how they prevent the team onboarding process from achieving its ultimate goals.

RQ4 What is a systematic process for designing a personalized team onboarding program?

A description of a personalized team onboarding design process is provided. This starts from collecting personal background information about newcomers and designating appropriate onboarding goals to be achieved by doing a set of related tasks.

RQ5 How can this design process be supported by a tool?

A working mockup/prototype is designed to illustrate the design workflow of a personalized team onboarding program, and an evaluation is conducted by seeking some feedback from industrial practitioners and the mockup enhanced accordingly.

3.2 Research Method

A Design Science Research method was used as the main umbrella to cover all discrete tasks in this study. The main justification for using this design science method is that it provides a mental model that fits the characteristics of the research outcomes which reviewers and editors could possibly use to build ideal information systems based on perception, imagination, or comprehensive conclusions after reviewing the key findings from the research (Hevner & Chatterjee, 2010, p. 28; Peffers et al., 2007, p. 52), and the main objective of this method is the

development of technology-based solutions in order to solve significant and related business problems (Hevner et al., 2004, p. 7). In addition, the design science method is widely known in software development industries, especially the development of information systems in order to address organizational problems as well as boosting revenue in the business (March & Storey, 2008, p. 726). Therefore, in the investigation of onboarding challenges and proposing an artefact for a software development team, the design science research method is suitable to ensure that onboarding gaps are addressed with a set of practical solutions.

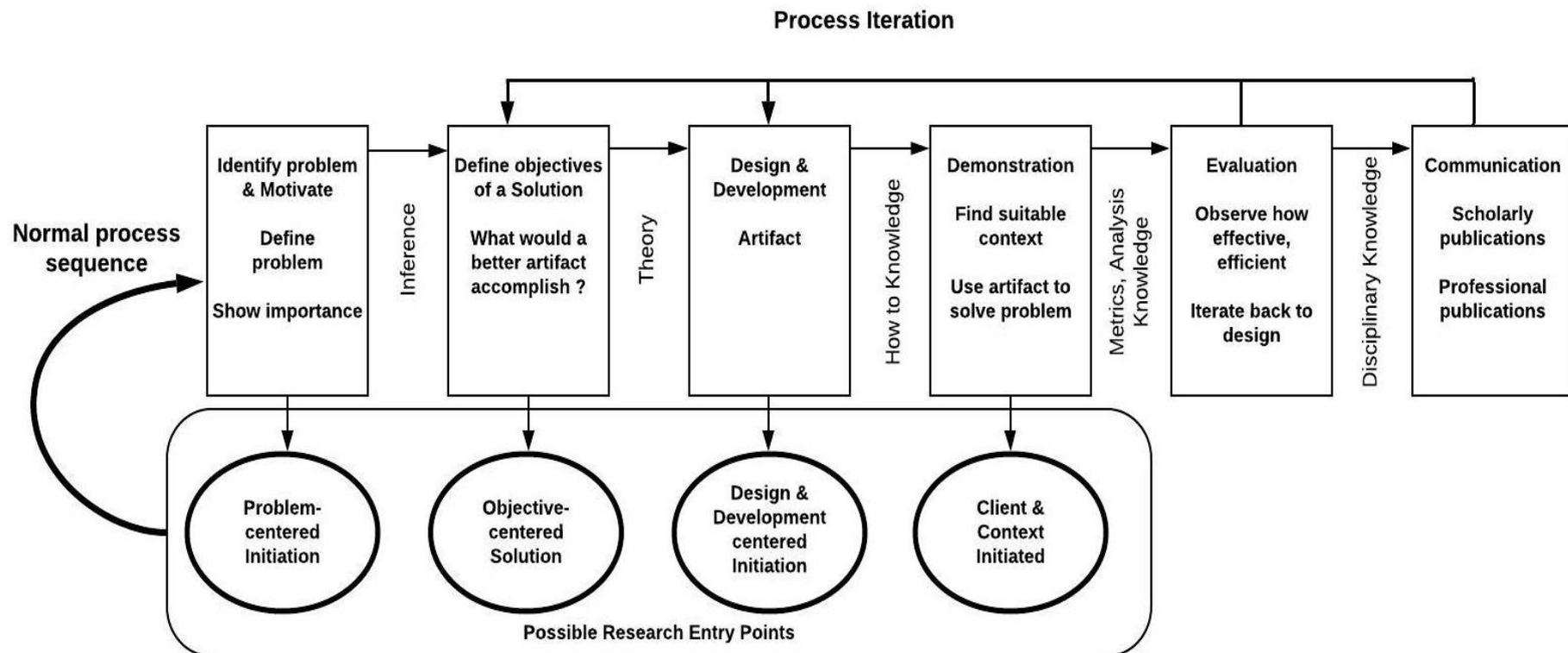
In Figure 4, the design science research concept consists of six essential activities where iterative processes could be consistently implemented until an ultimate outcome is satisfied (Hevner & Chatterjee, 2010, p. 28; Peffers et al., 2007). Three out of six steps from this design science concept adapted for this research, as follows:

- 1) Problem identification and motivation: This step was adapted to the initial step of the research. By identifying the research problems, the researcher retrieved related articles on organizational/team onboarding programs in the software development context. A systematic literature review was mainly conducted for data collection. Three main onboarding data components were extracted
- 2) Definition of the objectives for a solution: This step was adapted to define the main purpose of the research. Once some gaps/problems in a team onboarding program had been initially studied and possible solution could be developed to minimize the gaps, the researcher started drafting the aims of this thesis and these were iteratively updated
- 3) Design and development: This step was adapted to the design of a personalized team onboarding process which is the proposed solution of this thesis (see Chapter 5). Once all team on-boarding data components had been analyzed and categorized, the personalized team onboarding process was initially designed, and a list of input data defined (team requirements, onboarders' characteristics, and any related factors such as gaps in the comparison of teams, onboarder background experiences or personal risks).

To validate the designed process or new idea, the development of an onboarding artefact is offered. In this development step, the researcher adapted the design science method for iterative development in order to ensure that the tool that was developed met the process that had been designed.

Figure 4

DSR Process Model



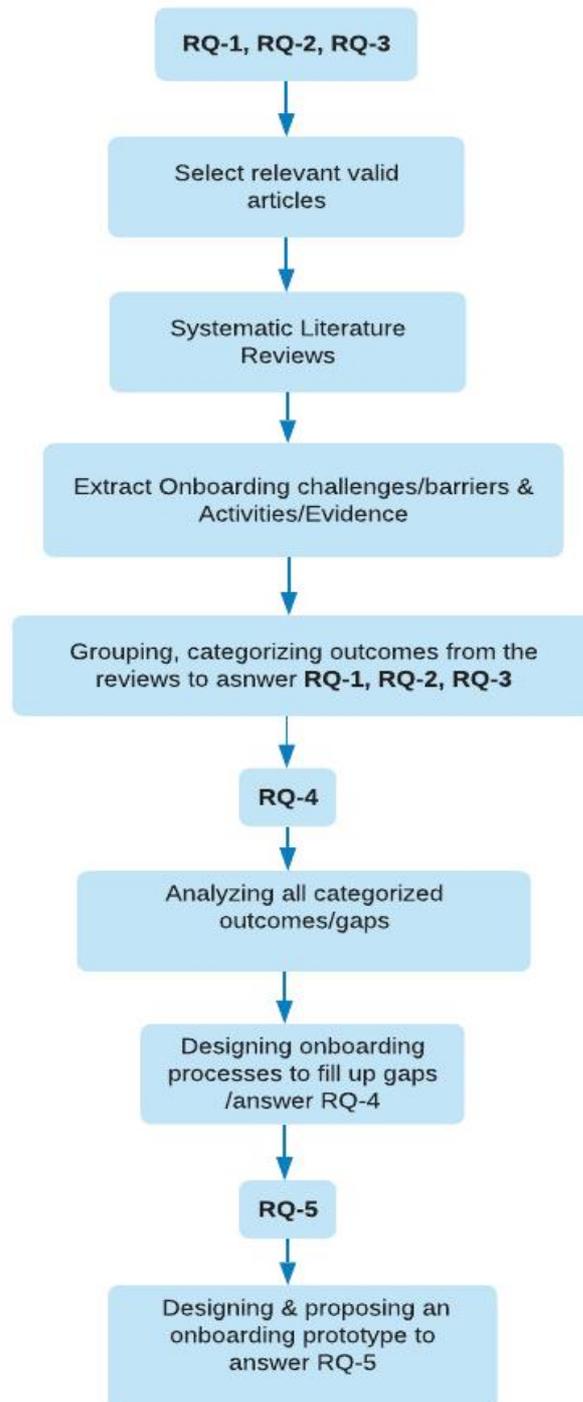
Note. Design Science Process Elements from IS and Other Disciplines and Synthesis Elements for a DSRM in IS. From “A Design Science Research Methodology for Information Systems Research” by Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). *Journal of Management Information Systems*, 24(3), 45-77.

3.3 Research Process

An overall process for this research is shown in Figure 5. All research questions are answered in different stages of the process. The diagram starts from collection of relevant articles related to the main purpose of the research in order to understand the conceptualization of team onboarding in a software development organization.

Figure 5

The Overall Research Process



RQ1 is answered by consolidating a subset of desired team onboarding goals, onboarding / off onboarding practices in software development organizations, describing team onboarding outcomes, expected knowledge, technologies and skills, and how onboarding difficulties are dealt with. RQ2 is answered by understanding common employed onboarding approaches/techniques in order to achieve the subset of desired onboarding outcomes, and the previous implementation of those techniques is explained. RQ3 is answered from the consolidation of team onboarding barriers/difficulties/challenges/concerns. These barriers are categorized/grouped and how different challenges delay the productiveness of on-boarders in a team onboarding process is described. RQ4 is answered by the design of a personalized team onboarding process where all the main steps of the design are described such as collecting newcomers' profiles, assigning goals and supporting tasks as well as the scenario for implementation and evaluation throughout the entire onboarding process. RQ5 is answered by offering a working artefact to support the design process and seeking ideas and feedback from my supervisor as a proxy to industry.

3.3.1 Systematic Collection of Relevant Articles

This research mainly relied on digital libraries for data collection. Three main online databases/portals were chosen for searching articles: SCOPUS, EBSCO, and Google Scholar. The main reasons for choosing these three databases is that a large number of academic journals, international conference reports, other books/sources are published on these databases especially in the field of software development teams. In addition, their advanced search functions are more constructive in customization, for instance, when working with both exclusive or inclusive criteria in SCOPUS/EBSCO databases. As studies on onboarding programs have been grown in the field of software development teams, they are still being published in different places. Hence, searching from all these three databases could ensure that relevant articles were picked up and analyzed for this research.

Based on the main purpose of this research, two main key words/phrases (“onboarding”, “software development team”) were used to search for relevant articles. The word “onboarding” was used to recover all research articles relating to onboarding studies from all three databases. The main reason to use the word “onboarding” as an included criterion was to collect all available articles related to onboarding terms which were directly associated with the purpose of this research. “Onboarding” is a specific process of welcoming newcomers to an organization or team as well as learning, understanding, and adjusting to their new responsibilities in a productive and efficient way (Ellis et al., 2017; Moe et al., 2020). For instance, in open-source software development and/or distributed global software projects, “onboarding” is the key word used to support those research projects, describing a recruitment process to get new developers to participate into the team (Fagerholm et al., 2014; Matturro et al., 2017; Moe et al., 2020), where practitioners need to pay critical attention in order to ensure not only that onboarders can be

productive and able to offer their first contribution into projects in a short time; it is also significant in retaining them for the organization (Deshpande & Gupta, 2019; Panichella, 2015)

Consequently, the combined phrase, “software development team”, was included with the selection of “all fields” in order to retrieve available articles that contain this phrase. The term “software development” itself is associated with “system development” or “application development” which are in the software engineering area. In the research by Guinan et al. (1998), the word “software development” was used as one of the key words to investigate the performance of software development teams by comparison of the teams’ behavioral and technical approaches. More than 66 software development teams from 15 companies participated, which were the main contribution to increase the evidence base of the findings of the research. Similarly, Sawyer (2001) studied 40 packaged software development teams in examining intragroup conflict, and the level of conflict management associated to the performance of teams. In this research, the word “team” is added to “software development” in order to be more specific and exclude articles which do not focus on working as a team.

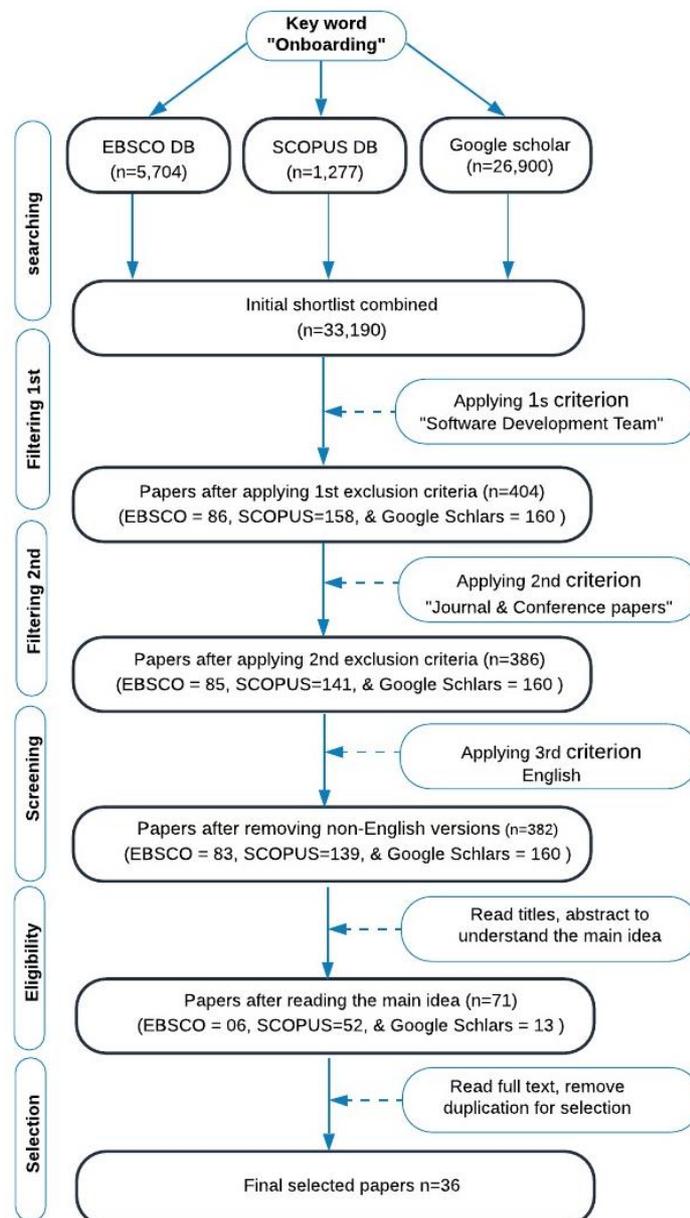
In the initial search with the key word “onboarding”, with the selection of “all fields”, the total result of the initial search was 33,190 articles (EBSCO=5,704; SCOPUS=1,277; Google Scholar=26,209). With the combination of the word “onboarding” and “software development team” using the Boolean operator “AND”, a number of articles associated with onboarding were excluded due to not being related to software engineering. It brought the total shortlist to 404 articles (EBSCO=86; SCOPUS=158; Google Scholar=160). In consequence, the academic journals (peer reviewed) and conference paper types only were chosen for inclusion, while other type of retrieved articles were filtered out and excluded. The total number of articles shortlisted was reduced to 386 articles (EBSCO=85; SCOPUS=141; Google Scholar=160). In addition, an English language only criterion was applied to bring the total number of articles to 382 (EBSCO=83; SCOPUS=139; Google Scholar=160).

The next step was that the researcher manually checked each article by reading its title and abstract to understand their main aims and then eliminate any irrelevant publications. The criteria for removing those papers were if they were similar to any of the following: a) did not explore onboarding in a software development team context, or information system development, or software engineering; b) did explore software development, information system development or software engineering, but did not investigate onboarding, or relocating the team members, or adding newcomers; or c) was published before the year 2000. The result was that 71 articles remained on the shortlist at this stage (EBSCO=06; SCOPUS=52; Google Scholar=13).

The final selection process was that the researcher critically read the full text of each article to understand the main purpose of the article, findings, discussions, and onboarding aspects based on the research questions and main objectives of the present study. A selected article had to meet at least one of these conditions: 1) provide some information about onboarding difficulties/barriers/concerns/obstacles in a section of the report; 2) provide onboarding goals or outcomes or results or expectations from conducting an onboarding activity; or 3) provide a list of onboarding tasks/activities that could be implemented by newcomers or teams or organizations or stakeholders. By applying these elements in the critical reviews as well as getting rid of some duplication, 36 articles were finally selected for future reviewing based on the research questions. The overall searching/selecting process of relevant articles is shown in Figure 6

Figure 6

A Workflow of Systematic Selection of Relevant Research Papers



Beside searching for articles for the systematic literature review, some other articles related to organizational onboarding and previous onboarding concepts and theories were retrieved and reviewed for an exploration of different perspectives. Some key words “onboarding”, “newcomers”, “new hires” were used to retrieve articles from the three main database with the use of operators “OR”, “AND”. The analysis of articles retrieved is described in Chapter 2 of this thesis.

3.3.2 Inclusion and Exclusion Criteria

Articles returned from the query were included if they met the following inclusion criteria. (1) The articles must be relevant to the area of onboarding. (2) The articles are documented as being peer reviewed. (3) Only articles that were published starting from the year 2000 to the present are included. (4) Only articles that are written in English language were selected for the review.

Exclusion criteria are opposite to the inclusion ones, and also eliminated duplicates

3.3.3 Systematic Literature Review

In order to address the research questions, the Systematic Literature Review research method was used to identify the gaps in and the goals and tasks of onboarding programme, as well as synthesizing all relevant onboarding components from different perspectives such as those of onboarders, mentors, the organization. The term ‘systematic literature review’ refers to a critical review of articles, meaning that researchers need to identify, evaluate, interpret and synthesize all available related articles, or topic areas, or phenomena of interest associated with specific research questions (Butler et al., 2016; Kitchenham, 2007; Verner et al., 2012). This systematic review methodology is aiming to minimize researchers’ subjectivity and bias in regard to the findings (Budgen & Brereton, 2006; Marques et al., 2012; Siddaway et al., 2019). Indeed, it could also result in a structure and patterns from the findings of existing studies and an ideal supporting tool could be developed to address a particular issue (Niazi, 2015).

A systematic literature review approach is widely used in different research fields (Borrego et al., 2014). It was originally used in the field of medicine which where it was intended to support health professionals, medical researchers, tertiary education, and government agencies in presenting evidence-based results in order to better inform policy makers (Borrego et al., 2014; Briner & Denyer, 2012; Kitchenham, 2007; Kitchenham et al., 2009). In consequence, systematic review approaches have gradually come to be used in software development fields where researchers have rigorously investigated problems, gaps, and requirements in empirical studies in order to design and develop tools to maximize benefits for organizations. For instances, Steinmacher et al (2014) conducted systematic reviews to identify onboarding barriers facing by

newcomers when they first provided their contribution to open-source software development projects. Moreover, in the study by Verner et al (2012) 24 global software development systematic literature review studies were reported.

In this research, the systematic literature review followed the guidelines from Keele (2007). One of the main reasons is that systematic literature reviews aim not just to be an accumulation of all related evidence associated to research questions, they are also intended to support the development of practitioners' evidence-based guidelines (Kitchenham et al., 2009; Siddaway et al., 2019), whereas a traditional literature review is likely inadequate as an in-depth investigation and focuses more on a comprehensive review instead (Snyder, 2019). In addition, the systematic literature review approach has a clear set of instructions and researchers are able to systematically follow them when conducting a review (Budgen & Brereton, 2006) to improve the quality of the review. Therefore, while the main purpose of this research is to develop a team onboarding design process tool (in the field of software engineering), it also suggests applying the evidence-based approach to improve the process of reviews (Kitchenham et al., 2009) such as the search process, applying inclusion and exclusion criteria, making a quality assessment of relevant articles, and using data extraction, and analysis.

3.3.4 Data Analysis

In the process of the selection of articles, EndNote X9 was used to consolidate all selected articles into one main library. By using EndNote X9, the researcher can screen and compare articles with the use of many segments in one place, such as sorting by title, versions, published date/time, and authors. EndNote is one of the most efficient citation management tools, especially for coding/formatting references for systematic literature reviews in a research project (Bramer et al., 2017; Peters, 2017). The data analysis of this research was managed in 3 main stages by following the recommended method of Smith and Firth (2011) and Williams and Moser (2019)

Stage 1: Data extraction

Based on the research questions, onboarding aspects were required to be extracted for further analysis. The extracted data allows researchers to view, interpret, and work on it (Butler et al., 2016). The researcher took out data and stored it in a data spreadsheet where every header described its purpose. This method is a common tool for managing extracted data (Carver et al., 2013), but the most important aspect is that it allow researchers to fundamentally organize the extracted data from different research resources to make it ready for coding, categorizing/theming, and synthesizing (Gibbs, 2007; Lockwood et al., 2015). In the data extraction process, the researcher first read the selected papers and listed them in the prepared spreadsheet based in nine categories as described on each header of the sheet: (a) Name of the

author(s) and year of publication; (b) Research aims; (c) Key findings; (d) Desired onboarding goals, outcome, expected results; (e) Mentioned onboarding barriers, challenges, concerns, difficulties; (f) Onboarding strategies, activities, tasks; (g) Discussed evidence from implemented onboarding tasks or onboarders' confirmation; (h) Monitoring metrics, feedback from newcomers; and (i) Mentioned onboarding challenges which are remaining barriers.

From the 36 reviewed articles, extracted datasets were recorded into spreadsheet (see Appendix A). This is the main table in which the researcher placed extracted data. The extracted data was ordered in the table based on the alphabet sorting of the main data library stored in EndNoteX9. The researcher selected three main attributes from this main extracted datasheet and then reproduced them in three individual figures to separate out these different attributes in order to store them and facilitate the further data analysis step. The three figures cover (i) outcomes/goals, (ii) challenges/barriers, and (iii) common tasks of onboarding, and are presented in Chapter 4 (see Figures 10, 11, and 12). The contents of these tables were listed according to similar meaning, characteristics and purpose, to facilitate categorizing and theming (Williams & Moser, 2019). A reference column was added to indicate the source articles.

Stage 2: Data Analysis

Over time, there have been many opinions published on qualitative data analysis to guide researchers. Those approaches provide some scenarios for completing qualitative data analysis: 1) assigning codes and themes from iteration of reading from extracted data, documents or visual data; 2) sorting and shifting throughout coded data items in order to search for similar descriptions, characteristics, meanings, relationships between variables, categories, patterns and common sequences; 3) separating/isolating some distinct patterns, processes, differences and then conducting the next wave of data collection; 4) reflection on remarks, jotting memos and/or journals; 5) elaboration on themed/categorized outcomes and generalizations which still remain consistent with the meaning of previous coded data; and 6) comparison/formalization of those elaborated body of knowledge in the form of concepts or theories (Bryne, 2001; Kawulich, 2004; Miles et al., 2018). One of the main purposes of data analysis is data interpretation, the conversion of data into stories which describe and enhance the phenomenon or participants' perspectives (Kawulich, 2004).

In order to analyze data for this research, the three attribute sets of data extracted from Appendix A were transferred and categorized into three tables, and the researcher adapted the instructions for implementing a coding/categorizing analysis (Dierckx de Casterlé et al., 2012; Gibbs, 2007; Kawulich, 2004; Pope et al., 2000). The main steps were systematically conducted according to the following procedure:

1) Iterative reading through all extracted items (onboarding goals, barriers, tasks) in the three figures (see Chapter 4, Figures 10, 11, and 12) which were transferred from the main table in Appendix A, in order to understand the meaning and core value of extracted phrases, gain familiarity, understand relationships, and ensure all extracted data items were captured in an organizable way. As one of the main aims of the research is to examine onboarding goals and barriers, and the tasks of onboarding processes, these three main aspects were extracted from the main data spreadsheet.

2) Examining, paraphrasing, and re-arranging data items for data coding. Throughout this process, the researcher thoroughly re-read the data items to understand the in-depth meaning of words and phrases in each single item from the tables (Dierckx de Casterlé et al., 2012; Gibbs, 2007), and manually marked similarities, patterns and relationships. Then, those items were interpreted and abstracted meaningfully, also known as “coding”. Coding is a method of categorizing or indexing qualitative data in order to create a group of similar patterns and establish a concept (Gibbs, 2007); it is not only minimizes a long list of data, and allows phrases to be more conceptual and meaningful, but it is also a more systematic way to organize data to be more manageable for further stages of data analysis (Williams & Moser, 2019). In the actual coding process, the researcher manually coded a phrase to represent data items from research findings. The similarity of words/phrases were combined, rephrased and abstracted for better conceptualization, but still being consistent to its original values. In order to avoid mistakes, such as missing any key values in the extracted data, the researcher sometimes double-checked meaning by re-reading the original article. If the data updating was required, the main extracted data spreadsheet as well as the figures were updated. Therefore, iterative processes were conducted throughout the entire process of data analysis.

3) After the data items were coded, descriptive basic themes were developed from reading through codes that have similar meanings, patterns, characteristics, and comparable values for the purposes of core thematic categories in the next levels of theming/categorizing (Bryne, 2001; Miles et al., 2018; Williams & Moser, 2019). An example theme is about “building confidence”. This category was formed from two codes from data items: getting constructive feedback and be encouraged to participate and speak their mind (see Figure 10). The reason for combining these codes was because they have comparable values relevant to the desired goals of the onboarding programs where the on-boarders are expected to gain coding knowledge. For those codes that were unable to be grouped into any category, they were placed into other category for later refining and allow them to be shifted to general category.

4) From the list of basic categories, the researcher selected those categories with shared values and integrated them into a higher-order abstraction. The higher-order ones are repeatedly unifying

concepts or statements of characteristic evidence from the complete set of several studies (Cruzes & Dyba, 2011). This is also called a selective coding process that allows the researcher to combine cohesive and meaningful abstractions together in order to elaborate or formulate a story for the research findings; this could also be called a clustering theme (Williams & Moser, 2019). The “To fit in with the team” category is an obvious example that was abstracted as a bigger one formed from the following groups: a) be able to collaborate and cooperate with the team members; b) Fit in with team norm and culture; c) Trusting and being trusted by the team members ; d) building confidence (see Figure 10)

5) Finally, the researcher developed a highest-order group for this study by synthesizing all higher-order categories from the previous step into an overall theme, also known as a “global theme”. All the higher-order themes the emerged from the previous step were subsequently explored and interpreted for the construction of a model which represented the relationships among them (Cruzes & Dyba, 2011). As an example from the abstraction of desired onboarding goals, three higher-order categories were abstracted into a global theme which was described “To be productive and integrated team member contributing to the team as expected” (see Figure 10). This highest-order theme covered all essential desired goals of a team onboarding program in software development organization as follows: a) be able to fit in company culture, b) be able to fit in the team, c) be able to do the work as expected by the team for the role and at the level of quality expected. To make sure of consistency and coherence, or to avoid missing any crucial findings in the review, the researcher also conducted iterative processes to review the data from coding step up to the highest-order category creation in this final step.

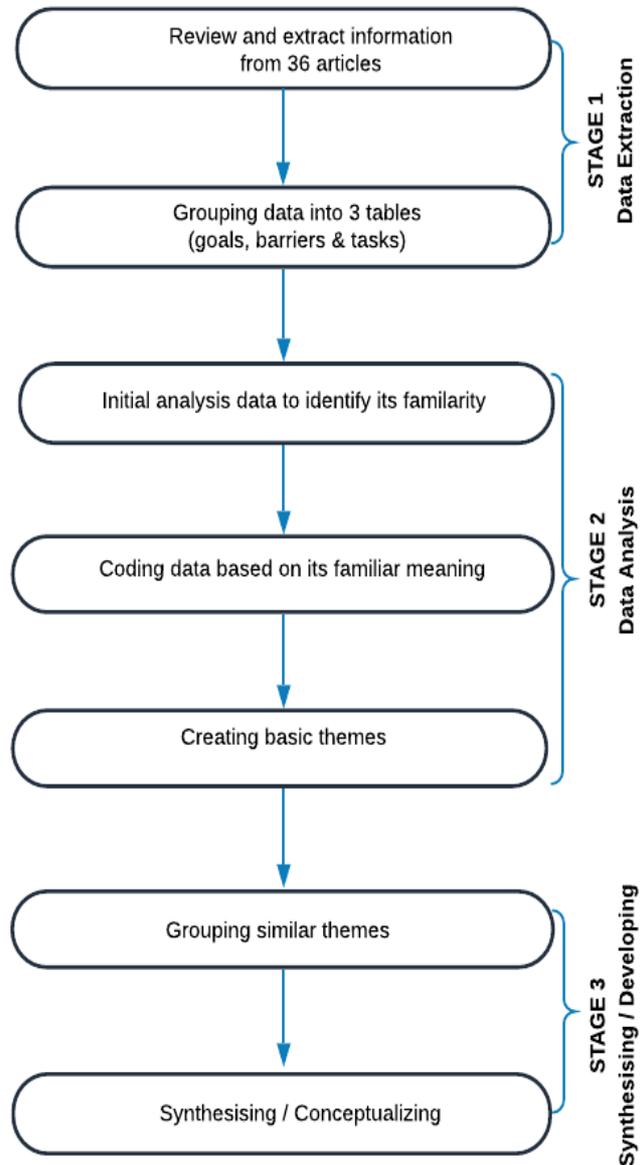
Stage 3: Multiple-study Synthesis

Once all selected articles were analyzed in the systematic literature review process, the synthesizing process was commenced. Thomas & Harden (2008) developed a data analysis approach called “thematic synthesis” and applied it in a series of systematic reviews in order to address research questions specifically on people’s viewpoints and experiences. Using the thematic synthesis method for qualitative data analysis can offer premium and insightful understandings of complex phenomena from different perspectives (Smith & Firth, 2011). The thematic synthesis method was also widely used to analyze human behavior in the software engineering fields. For instance, Cruzes and Dyba (2011) recommended the thematic synthesis method for systematic review in support of evidence-based software engineering. Thus, the thematic synthesis approach was adopted in the process of basic descriptive coding, basic theme, higher-order category and global theme processes as it offered an appropriate framework for theme synthesis in team onboarding in software development organizations (desired goals, barriers, and tasks). These three findings are elaborated in Chapter 4.

Throughout the entire process of coding and basic theme, higher-order theme and global theme creation, the researcher used an online tool “Lucidchart” to build charts associated with the extracted data table in Microsoft Word (see Figure 10, 11, 12 in chapter 4)

Figure 7

High-level Workflow of Data Analysis



3.3.5 A Design Workflow for a Personalized Team Onboarding Program

In the data analysis focusing on three main attributes of the extracted data, the researcher classified the potential onboarders from extracted data. Although some of the studies reviewed did not specify explicitly a type of onboarder to be onboarded, a list of potential onboarders with reference sources was summarized based on the type of newcomer described in the aims of the studies, such as new graduates, novices, experienced newcomers. These are presented in Table 3, below.

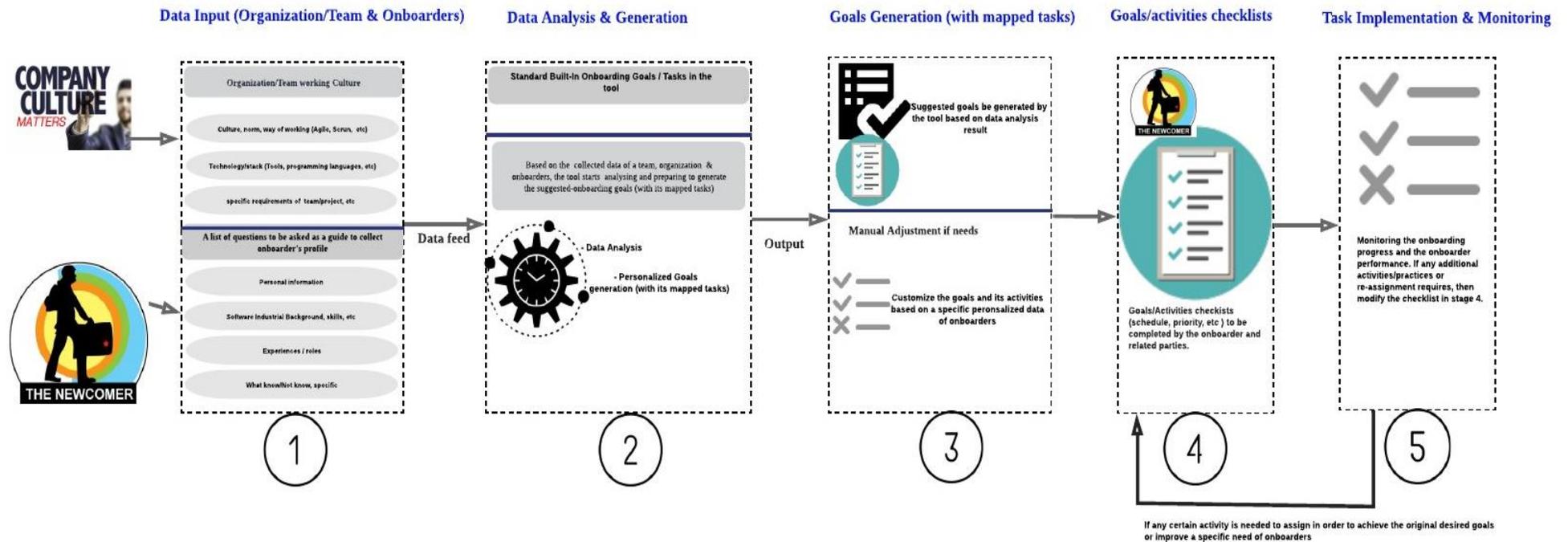
Table 3*A List of Potential Onboarders to be Onboarded in a Team Onboarding Program*

No	On-boarder	References
1	New graduate, novice	(Balali et al., 2018; Buchan et al., 2019a; Deshpande & Gupta, 2019; Diniz et al., 2017; Dominic et al., 2020; Fagerholm et al., 2014; Hoffström, 2019; Liu, 2019; Pham, 2014; Pham et al., 2017; Pham et al., 2015; Silva et al., 2020; Viana et al., 2014; Yang, 2017)
2	New experienced employee (local open source projects)	(Britto et al., 2018; Buchan et al., 2019a; Hannebauer & Gruhn, 2017; Kovalenko & Bacchelli, 2018; Panichella, 2015; Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015; Viviani & Murphy, 2019b; Wang, 2012; Yang, 2017)
3	New experienced employee to join distributed global projects (virtual team onboarding)	(Britto et al., 2020; Britto et al., 2018; Cunha et al., 2020; Fagerholm et al., 2013; Moe et al., 2020)
4	Older adult on-boarders	(Davidson et al., 2014)
5	New team member technically capable but new to the Agile way of working	(Buchan et al., 2019a; Yang, 2017)
6	Temporary team	(Buchan et al., 2019a; Yang, 2017)
7	New team member from another workstream	(Buchan et al., 2019a; Yang, 2017)
8	New team leader	(Buchan et al., 2019a; Yang, 2017)
9	New team, some from other roles, some new to the company	(Buchan et al., 2019a; Yang, 2017)
10	New experienced team taking over another team's work – requirements volatile and new to product	(Buchan et al., 2019a; Yang, 2017)

In accordance with the list of potential on-boarders, and the desired goals and tasks of a team onboarding program described in previous sections, the researcher started designing a systematic personalized team onboarding program in order to support onboarding practitioners dealing with team onboarding goals and the specific characteristics of different onboarders. Overall, an entire systematic design process for a personalized team onboarding program was designed which consisted of these main steps as shown in Figure 8.

Figure 8

High-level Design Workflow of a Systematic Personalized Team Onboarding Process



Step 1: Data Collection: Three main data components are considered to input into the process and drive outcomes of a personalized team onboarding model: 1) personality traits/characteristics of on-boarders, 2) team norm and project requirements, and 3) organizational culture/requirements. These three data components need to be collected in the early stage of a design process for an onboarding program in order to analyze them, identify desired goals, assign supporting tasks to each goal, and visualize any specific barriers for designated onboarders that could prevent the achievement of a goal in the onboarding executive process. The personality traits/characteristics of onboarders, such as proactive personality, extraversion, openness, or veteran employee and newcomer behaviors, such as information seeking, feedback seeking or relationship building, were related to the successful factors of newcomers becoming organizational insiders (Bauer & Erdogan, 2011).

Step 2: Collected Data Analysis: The second step is designed for data analysis. An onboarder would be identified as one of the types of potential onboarders listed in Table 3. Team, project and organization requirement lists would also be analyzed and synthesized in order to designate the necessary onboarding goals into the program. Each onboarder type would have a set of desired onboarding goals that need to be achieved so that practitioners could simply assign those goals to onboarders in the program. Consequently, a list of essential tasks with implementation schedules in order to support the achievement of each goal would be attached for onboarders to complete.

Step 3: Goals Generation/Customization: Following step 2, all appropriate goals and supporting tasks would be placed into the program in this step. In case of any particular requirements, either on-boarders' personalities or team/project/organization requirements, the practitioners could analyze those circumstances and customize a list of desired goals and supporting tasks appropriately in this step. For instance, if an onboarder has an English language barrier in communication, the programme designer could assign a specific English training program for that onboarder to attend in order to achieve the professional communication goal. An expected outcome from this step is a finalized checklist of desired goals associated with supporting tasks so that an onboarder would have a personalized onboarding plan to start work on.

Step 4: Implementation/Monitoring/Evaluation: Based on the finalized checklist of the personalized team onboarding program from step3, the onboarder can simply start the activities and be following up from time to time in the entire process. An iterative process is conducted to ensure all implemented tasks meet minimum indicators. A canvas concept is suggested for implementation, monitoring and evaluation of the progress of a personalized team onboarding program in this step. The canvas concept is a model that could be used to implement the transformation of different business contexts. It is a visual model method used to depict the business movement of an organization. This method consists of building blocks that represent the

components of each business stage. In a canvas model, creators will demonstrate the key elements of business stages with a fixed timeline. It is not only used for illustration of the current state of a business model, but also can be applied to forecast any business model innovation (Fritscher & Pigneur, 2014). In a sustainable project management study by Schipper & Silvius (2017), the canvas methodology was used in a practical way to assist project managers and project teams in the development process. The authors pointed out that this kind of business model canvas is widely used for business consultations as it is visualizable and easy to use to illustrate project components in any state. Likewise, Michelle and Chris (2020) proved that the creative business model canvas could help artists to propose products and their value to reach audiences/clients. In this personalized team onboarding process for a software development organization, the canvas concept is adopted as an onboarding canvas tool that iterates onboarding processes until desired onboarding goals are achieved and newcomers can be productive for the organization.

3.3.6 A Proposed Supporting Tool for a Design Workflow

According to a designed process of onboarding goals, a supporting tool could be developed in order to facilitate onboarding planners and better manage personalized onboarding goals and tasks and handle possible risks throughout the entire process of onboarding programs. To develop the tool, a research design science method (see Figure 4) is suggested as this onboarding tool is a type of information system where developers need to input team requirements, on-boarders' characteristics and iteratively analyze and develop the tool until the product requirements are satisfied. One of the most significant reasons for using this method is that the design science method is the systematic approach as well as the evidence-based practical principles in which all processes are iteratively implemented, and systematically studied, to address business concerns by relying on proven evidence (Hevner & Chatterjee, 2010; Hevner et al., 2004; Peffers et al., 2007). In addition, the quality of the information artefact could be secured by an iterative evaluation process and by enhancing related concerns accordingly (Pries-Heje et al., 2008) so that the onboarding tool could accurately design a personalized onboarding program for those who join a team.

CHAPTER 4: FINDINGS AND DISCUSSION

The results of the systematic literature review, including data analysis results, onboarding goals, suggested onboarding tasks, possible risks, and a proposed onboarding design workflow, as well as a possible tool to supporting the proposed design process, are presented in this chapter. There are two main sections. The first section presents the results of the systematic literature review, consisting of onboarding goals discovered, supporting tasks for those desired goals, and risks. The second section discusses a viable personalized design onboarding workflow based on teams and onboarders' characteristics, and this section includes the proposed artefact to facilitate the design workflow.

4.1 Systematic Literature Review

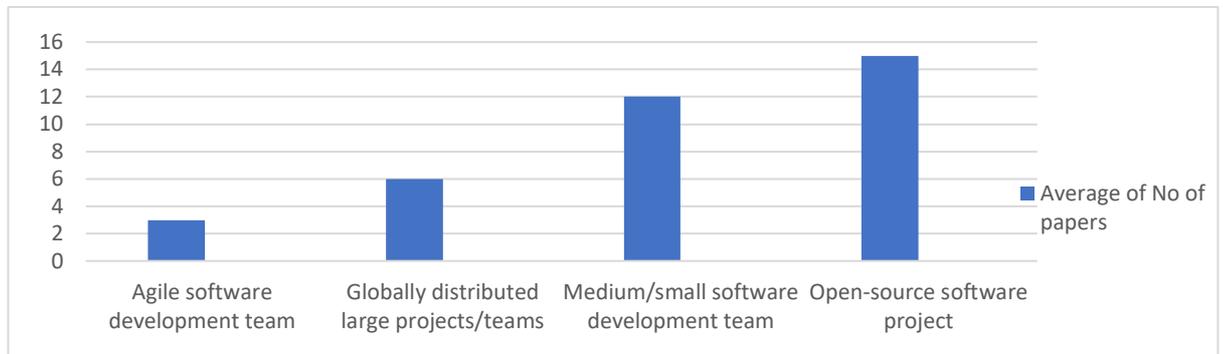
The 36 selected papers were classified into four main onboarding research categories based on their purposes. Appendix B listed three data attributes of each article, namely author(s) and year, purpose of article, and category. The first category is about onboarding research on open-source software development projects, the second is about globally distributed projects, the third is about onboarding in Agile, and the fourth is medium and small software development companies where specific aspects of onboarding programs were investigated and reported to better inform the future onboarding design program for practitioners. The researcher used a data-table spreadsheet to record the extracted data (see Appendix A) and then reorganized them into the Appendix B to summarize the four review categories.

The open-source software development project category gave the highest number of papers in Appendix B with 15 research papers categorized into this group. It is the closest to a team onboarding program because developers in open-source software development contribute their efforts to enhancing or adding more features into the products which were developed by other developers, or developers could build a completely new product with other developers. The second largest category is for onboarding research in the medium/small software development category, in which 12 articles are reported. Six articles are reported in the globally distributed large project/team category, and three articles are the Agile category. The research in all categories investigated onboarding issues in various situations when companies integrate newcomers into working for teams, covering such topics as barriers, possible activities, goals, employment of helping tools into the process of onboarding, and evaluation. For examples, Steinmacher et al. (2014 & 2015) investigated a set of onboarding barriers, evaluated social barriers, proposed a tool called "FLOSScoach" to support newcomers in addressing a range of revealed risks, and evaluated how the tool could possibly minimize those risks when executing an onboarding program. Balali et al. (2018) classified onboarding risks into 44 barriers, of which

10 items affect mentors and 25 affect onboarders and 9 barriers affected both. The extracted data (goals, tasks, challenges of onboarding) was analyzed and is reported in the following sections.

Figure 9

Overall Selected Onboarding Research Papers



4.2 Results of Data Analysis

Three main aspects of team onboarding data were analyzed and are discussed in this section: a) a list of desired team onboarding goals and its sub-categories; b) a list of suggested onboarding tasks/activities to supports companies, on-boarders, and related parties; and c) a list of possible onboarding challenges/risks/barriers which on-boarders or teams/companies could possibly encounter throughout the course of using onboarding programs.

4.2.1 Research Questions

RQ1 How is team onboarding conceptualized?. It can be seen that the list of onboarding goals (see Figure 10 at the end of this chapter) and suggested onboarding tasks (see Figure 12) are the main supporting information for team onboarding designers to analyze and use in planning a team onboarding program. To understand the insight into how goals, tasks, and barriers of onboarding relate each other from the Figure 10, it is necessary to discuss and show some evidence from the reviewed articles. This could explain how onboarding designers conceptualize a team onboarding program in software development contexts. However, this is insufficient for onboarding newcomers into the team because the conceptualization of the onboarding program must be more specific to the personal characteristics of onboarders to maximize the effectiveness of an individual onboarding program. That is why main purpose of this thesis is to investigate how to implement the systematic design of a personalized team onboarding program. See the proposed design process in Chapter 5.

RQ2 How is team onboarding done currently?. It can be seen from the literature review that the implementation of a team onboarding program has been done by simply designating a set of goals

and supporting tasks as well as organizational efforts to support onboarders. However, some gaps still require to be considered critically. For instance, designing the program is not a clear set of steps, and the onboarding implementation in medium- and small-sized software development company is informal, and some companies have no proper onboarding plan, but let newcomers learn from doing a real job with ad-hoc support from senior team members.

RQ3 What are main challenges of team onboarding and the difficulties/barriers to achieving the goals?. Figure 11 shows the results of analyzed data from the systematic review in terms of the list of challenges/barriers/difficulties. They are grouped into eight categories: 1) Personal barriers, 2) Interpersonal barriers, 3) Process barriers, 4) Technical barriers, 5) Contribution barriers 6) Testing barriers, 7) Reception Issues, and 8) Documentation problems. This research question is answered by these barriers which are consolidated from software development contexts and different perspectives.

RQ4 What is a systematic process for designing a personalized team onboarding program?. From the onboarding goals, tasks, and possible barriers discovered in this study (see Figures 10, 11 and 12), a personalized team onboarding design process could be designed using the analysis of team requirements and onboarders' characteristics, and then designating desired goals with supporting tasks. This research question is answered by addressing the gaps in existing team onboarding programs in the software development context. The details are discussed in Chapter 5.

RQ5 How can this design process be supported by a tool?. The question is answered based on the systematic process to design the personalized team onboarding program arising from the response to RQ4. A proposed working mockup/prototype was designed and developed, an iterative method suggested for implementing each of the designated tasks, and the canvas model could be used for evaluation. The iterative processes could be carried out in the tool development process as well as the onboarding process implementation, to ensure that both effectively serve the ultimate goals of the team personalized onboarding program.

4.2.2 Onboarding Goals, Challenges/Risks, and Tasks/Activities

The term 'desired onboarding goals' refers to results expected to be achieved by onboarders in the team onboarding programs by conducting a set of tasks and utilizing organizational efforts as well as senior team members. Based on the data extracted from Appendix A, the analysis show onboarding goals in all collected articles. There are a set of expected onboarding outcomes for different project/team situations when conducting onboarding programs. From those onboarding perspectives, the ultimate goal of a team onboarding program is an integration of onboarders so they are able to be productive and contribute to a team as expected (Britto et al., 2018; Buchan et

al., 2019a; Steinmacher et al., 2014). The results show that the desired goals were classified into three main categories as follows: a) be able to fit into the company/department, b) be able to fit into the team, c) be able to do work in the expected roles at the level of the team. Each level of goals consists of sub-categories to which onboarders are differently assigned based on his/her personalities, background, and experience. Figure 10 shows the overall team onboarding goals that were discovered, and the table 1 in Chapter 2 & Appendix C contain the original outcome from the review.

Barriers/challenges/risks refer to issues and difficulties that could delay onboarding programs, causing them to be late in being successful, or that could lead the programs to fail for some circumstances. Onboarders, related supporters, or companies may possibly encounter these barriers or take the risks throughout the ongoing processes of the programs (Balali et al., 2018; Steinmacher et al., 2014). From data extraction and analysis from the software development contexts that were discovered, team onboarding barriers are classified into different categories as follows: a) personal barriers, b) interpersonal barriers, c) process barriers, d) contribution barriers, e) testing barriers, f) reception barriers, and g) documentation barriers. Figure 11 shows the full set of possible barriers and their sub-categories, as well as Appendix E shows the detail and its sub-categories.

Tasks/activities refer to a set of supported activities/artefacts that onboarders, support persons or companies need to conduct or participate in throughout onboarding programs to overcome possible team onboarding barriers and achieve the desired team onboarding goals. Based on the desired goals, personal characteristics, or experience of onboarders, the tasks could be assigned and scheduled accordingly. For instance, onboarders could be mentored by senior team members, could receive peer support, or could join an online course. A mentoring program, or participating in online courses, are the types of activities that onboarders, mentors and peers need to work on together to drive the result. From data extraction and analysis, the tasks/activities are classified into three main categories: a) activities, b) artefacts, and c) people. Figure 12 shows the overall recommended tasks/activities and the relationships among them which originally from Appendix D and table 2 in Chapter 2.

4.2.2.1 Relationships Between Onboarding Goals, Risks, and Tasks

To achieve a team personalized onboarding goal, a set of activities/tasks must be implemented by on-boarders and related parties in an onboarding program. The following sections discuss the commonly expressed team onboarding desired goals and supporting tasks, and the possible risks/barriers that have been discovered and that could detract from the success of onboarding programs.

4.2.2.1.1 Company/Department Culture

For onboarders to fit with the company or department, they must understand company culture. The company's or the department's culture is the set of values, beliefs, ethics and goals of the organizations that employees need to understand and they must comply with such as a way of working, learning, supporting, or even promoting employees within the organization (Buchan et al., 2019b; Edmondson, 1999; Yang, 2017). The culture could be shared beliefs defined by the owners or managers of the company. Onboarders are required to learn and understand this culture in order to be productive when in charge of their roles in the company as well as avoiding producing any serious conflict. Culture could possibly influence a working style, internal processes and/or how management of departments or team in the companies could be achieved (Britto et al., 2018; Moe et al., 2020). Thus, expecting onboarders to be fit in company culture could be a high layer of goals where everyone who is willing to join any department or team in the company should learn about and understand the culture. For instance, in the study by Yang (2017), developers pointed out that *"We have an induction session for new staffs to get familiar with company structure, health and safety. We also have an orientation talking about the same things as the induction every year that called basic class which is for all employees of both new and old"* (p. 68).

Concurrently, onboarders also need to adapt themselves to support company values, beliefs, and goals despite the characteristics of software development companies being new or different from the onboarder's previous working experiences. Cooper-Thomas and Wilson (2011) pointed out that *"Changes occurring during this period of organizational socialization include learning the role, getting to know colleagues, and understanding the organization's culture and norms"* (p. ??). So onboarders are expected to change themselves to new ways of working.

To support onboarders achieving the company culture goals, several activities/tasks are suggested (see Figure 12). For instance, nominating a mentor for onboarders, and providing formal/informal sessions. Those tasks may consist of participation in induction sessions, joining regular company conferences, taking related courses on Agile, and reviewing the company's products. In the research by Britto et al. (2018), it was pointed out that coaching and support by a mentor can teach and provide advice to onboarders about the company's job instruction as well as the orientation session potentially helping newcomers to understand significant aspects of the company's culture and values.

Concurrently, it is important to keep in mind some risks/barriers that may be slow down the achievement of the onboarding program. Some critical risks are personal barriers, including cultural differences, or inability to improve upon criticism. For instance, in the study by Balali et al. (2018), a participant mentioned that *"in some cultures, people get more upset when people*

criticize their code ... which can be tough". Similarly, when newcomers and mentors live in different geographical locations and the face-to-face interaction is difficult, the mutual trust can be reduced (Balali et al., 2018; Steinmacher et al., 2014)

4.2.2.1.2 Being Able to Fit in With the Team

For an onboarder to fit in with a team, a set of desired goals must be achieved in a team onboarding program. The following points describe several different supporting goals and related supporting activities, as well as noteworthy risks/barriers.

Team norm/culture: To fit into the team, onboarders need to understand team norms/culture, characteristics, or any specific requirements. Team norms are a set of values, beliefs, and/or ethics that are influenced by the company culture and assigned roles where teams need to be responsible. The team norms could influence how team work patterns operate, such as communication, decision-making processes, or conflict solution methods (Yang, 2017), as well as other workflows, in order to fulfil business requirements and achieve the best results. For example,

"Team Lead is the person I will ask or discuss about team norms with. The ways of working, sometimes, are different from team to team. For example, when I came here, I had no idea about the meeting schedule and who should attend to the different meetings. Nobody but the person who in your team knows team norms, so the best way is to ask your team lead, or your colleagues" (Developer). (Yang, 2017, p. 71)

This quote reflects how to get final answers in the way that it is normally done in the team.

Team norms also relate to company culture, but they are more specific to roles of the team, and the team aims, their implications, and the business requirements which need to be achieved. In addition, the team norms also refer to how members in the team typically act, feel and think, and what is approved or disapproved of in general (Hoda et al., 2012; Lenberg & Feldt, 2018). A team member explores a common understanding of situation, and is aware of the quality of team work is the key to success in a team (Edmondson, 1999). In an Agile software development team, newcomers are expected to understand how to develop artefacts in Agile ways, and understand Agile mindsets (Buchan et al., 2019a; Yang, 2017). For example, onboarders need to learn about the understanding of a task shared between developers and project owners through team processes and mutual interactions (Yu & Petter, 2014).

To achieve these goals, a list of iterative sessions is suggested such as orientation, mentoring, training, participating in team stand-up meetings, and retrospectives, as well as support from

supervisors or peers (see Figure 12). A vast literature exists on the training and mentoring functions of onboarding in software industries, and has pointed out that training/mentoring programs could teach newcomers to understand and integrate into a team and the organizational culture (Britto et al., 2018; Buchan et al., 2019a; Fagerholm et al., 2014; G. G. Sharma & K. J. Stol, 2020).

Similar to the company culture goals, some risks/barriers could prevent the success of team norm goals, especially the personal/interpersonal barriers listed in Figure 11, for instance, the diversity of onboarders or the lack of patience with working as a team.

Collaboration and Cooperation: Being able to collaborate and cooperate with the team members is a desired goal for onboarders to be able to fit in with the team. This goal refers to the understanding of onboarders about how to cooperate among team members, or between virtual teams where newcomers join globally distributed software development projects (Britto et al., 2020; Britto et al., 2018; Fagerholm et al., 2014). Collaboration and coordination are fundamental aspects of software teams when solving complex problems and carrying out innovative tasks (Moe et al., 2015). In addition, they relate to the interpersonal/personal communication methods of onboarders, and how to overcome possible limitations or obstacles, especially in the case of virtual teams where newcomers are integrated from diverse cultures, languages, and time zones (Balali et al., 2018), and political contexts (Moe et al., 2020). Moreover, the utilization of some tools throughout the working processes is essential to facilitate interaction among team members. Thus, it is necessary for onboarders to have the ability to use management tools or any related portals used by the team.

Achieving these goals is like other goals in having onboarders participate in a set of sessions and complete a list of tasks such as induction sessions, a mentoring program, training courses, and other internal meetings like stand-up meetings, conferences, and retrospectives. Also, some risks to be considered are an on-boarder's lack of a project background, the lack of interpersonal skills, and the communication styles of onboarders (see Figure 11).

Trust from the Team: "Trust is defined as the expectation that others' future actions will be favourable to one's interests, such that one is willing to be vulnerable to those actions" (Mayeret al., 1995, and Robinson, 1996, as cited in Edmondson, 1999, p.7). Being trusted by the team in a team on-boarding program in a software development context is a desired goal that needs to be achieved (see Figure 10). It means that on-boarders need to understand what work to do and when, with a demonstration of achieving results that at least meet the minimum expectation of team standard quality. Along with this insight, onboarders are required to be able to understand the criteria of the team's work quality in order to ensure that he/she will become a productive team

member and contribute significant effort (Buchan et al., 2019a; Liu, 2019; Yang, 2017). This may cover, for example, understanding code standard, architectural patterns, team practices, and team services.

To achieve this goal, a list of activities and sessions could be conducted by onboarders and relevant company's resources. For instance, a mentor-assigned simple, small, and interesting task in the early stage for onboarders could motivate them to focus and generate productive results. In consequence, on-boarders need to share their work for getting feedback from others in the team (Balali et al., 2018). A participant in the study by Balali et al. (2018) said that

“We have sessions for newcomers to present their work. We also encourage them to write blog posts, so people know what they are doing” (p. 700).

This is a chance for onboarders to become familiar and gain trust from the team as well as themselves trusting the members of the team. Concurrently, some difficulties could be encountered, such as the lack of interpersonal skills, lack of communication skills, and the distribution of team members.

Be Able to Be Confident: A onboarder's participation is one of the main preferences in a software development team. Building confidence is a way to motivate onboarders to share their thoughts with the team. Thus, having confidence is a team onboarding desired goal such that onboarders are expected to confidently contribute their ideas into the team or be active throughout various tasks, such as building curiosity, seeking feedback, asking questions, and showing proactive and interactive behaviors when the onboarders are being onboarded (Britto et al., 2020; Deshpande & Gupta, 2019; Fagerholm et al., 2014). Other activities are communicating by different means that may or may not be publicly visible, and participating in the regular stakeholder meetings, team retrospectives, team stand-up meetings. These are not only strategies to strengthen the confidence of onboarders, but they are also well-known as a learning progression. The engagement in actions and interactions of cultural, historical and social context is also considered as a learning progression (Cauwelier et al., 2019).

4.2.2.1.3 Carrying Out Expected Roles and Meeting Team Quality Standard

Newcomers carrying out expected roles and generating results that meet the team's quality standard is also one of the main team onboarding goals. To succeed in this goal, some relevant desired sub-goals should be achieved. The following sections discuss these sub-goals as well as related supporting activities, and risks/barriers.

Understand Expectations of the Roles (Accountability and Responsibility): The term ‘expected roles’ refers to the position that an employee need to take in a team and the responsibilities are duties/tasks which employees have to implement based on their job descriptions, employing the team’s practices and ways of working in order to contribute to the team’s expected results (Ahmad & Kolla, 2012; Bauer et al., 2007; Hoda et al., 2012). It is essential to be able to understand the expectations of the roles in a team onboarding program in a software development context. Onboarders/newcomers need to understand their main duties and when to do them as well as be able to report the results (Balali et al., 2018; Buchan et al., 2019a; Liu, 2019; Moe et al., 2020; Yang, 2017). Concurrently, they must understand the roles of other team members. For instance, onboarders are expected to be aware of who is doing what in the team. This would not only help onboarders to know the relationships among their tasks and the tasks that are in the charge of another team member, but it could also offer a chance to thoroughly understand the core team’s roles.

To support this goal, a range of activities are suggested for different team onboarding situations. A case study by Britto (2017) on virtual distributed projects, for example, reported that newcomers could achieve these goals by joining a job review at the recruitment stage along with intensive support from senior developers, while in the circumstances of open-source and Agile projects, the implementation may include mentoring, inducting, and training activities as well as the utilization of tools such as internal documentation and a wiki page (Cunha et al., 2020; Steinmacher et al., 2016).

Some barriers/risks could impact the implementing stage of the program, such as the lack of pro-activeness, lack of commitment, and lack of interest. With the diverse characteristics of onboarders, it is essential to acknowledge possible risks/barriers that have been discovered, and which are shown in Figure 11.

Having the Skills and Capabilities to do the Work: Some of the most significant team onboarding goals in software development are newcomers having the skills and capabilities to do assigned tasks when they become a member of the team. To achieve these goals, several sub-goals are expected to be achieved, as follows:

- a) Be able to use the technology stacks used by the team. This consists of the main programming language (front end, back end, relevant supporting tools, frameworks, and all dependency libraries) (Kumar et al., 2016; Matturro et al., 2017; Yates et al., 2020).

- b) Be able to understand ways of working such as the Agile method, and any specific practices like test-driven development, user stories, storing mapping, or code review. This would help newcomers to gain the capability to work with these team practices.
- c) Be able to understand and modify existing code. It is essential for developers to understand the existing code especially products which are developed by the team. When a newcomer joins the team to replace a member who has resigned, the newcomer is required to thoroughly understand and know the overall code architecture, where to change or update the code in order to further development, and facilitate customization, add new features, respond to change requests, and/or support any defects.
- d) Be able to understand the product knowledge/domain. Having product knowledge is required for onboarders in order to support their coding capabilities. If newcomers know the product requirements, and understand the business workflow, it would help them to plan and program functions more effectively.

To support all these goals, several activities/tasks (see Figure 12 and Table 2) could be assigned to onboarders for them to conduct throughout the onboarding program, for example mentoring, training, stand-up meetings, team retrospectives, buddy programming, code reviews, and joining an Agile course or online courses.

“I was a newbie as a developer when I came to my company, I had a lot of to learn and have lots of problem, and nobody going to help me all the time. I always try to find solutions online first, and if I still cannot find the answer, I will ask my mentor or my team members.” (Developer) (Yang, 2017, p.73)

Despite a lot of motivation for those who have a strong passion for learning new skills with a new team, some challenges could cause onboarders to fail. A set of risks/barriers that has been discovered is summarized in Figure 11. These include, for instance, technical barriers such as learning a huge amount of legacy code, code characteristic issues (low quality standard), and lack of version control knowledge (Balali et al., 2018; Steinmacher et al., 2014; Yates et al., 2020). Other barriers related to process issues are the lack of documentation, outdated list of bugs, and difficulty in reproducing bugs. Onboarders may encounter these risks, and the onboarding planners need to carefully analyze gaps and then apply appropriate activities to fill the gaps accordingly.

Understand and Achieve the Team’s Expectations of Quality: This goal actually supports the team norm goal, but it is more related to the technical quality of the team. Being able to understand

and achieve the team's expectation of quality is about the team's needs, which require onboarders to have a comprehensive understanding and achieve the standard quality level of the team. For instance, when newcomers program a feature of a product in a functional or class component, in order to have quality assurance of the code standard, all the code should be either manually or automatically tested by using dependent packages or a related library (Pham, 2014; Pham et al., 2017). Along with this, the design of test cases is another essential task to filter code bugs. "More than the act of testing, the act of designing tests is one of the best bug preventers known" (Dooley, 2017, p. 253). All these are related to the team's expectations in term of quality.

A set of onboarding activities could support the achievement of this goal. Some common activities are mentoring, training sessions, and having onboarders participate in internal meetings such as stand-up meetings and team retrospectives. At the same time, the designers need to keep in mind the skills and how much of the gap level of new graduates. There is a gap between new graduates' skills and the requirements of software industries due to the lack of hands-on experience of graduates of educational institutions (Pham et al., 2017).

Figure 10

Discovered Team Onboarding Goals

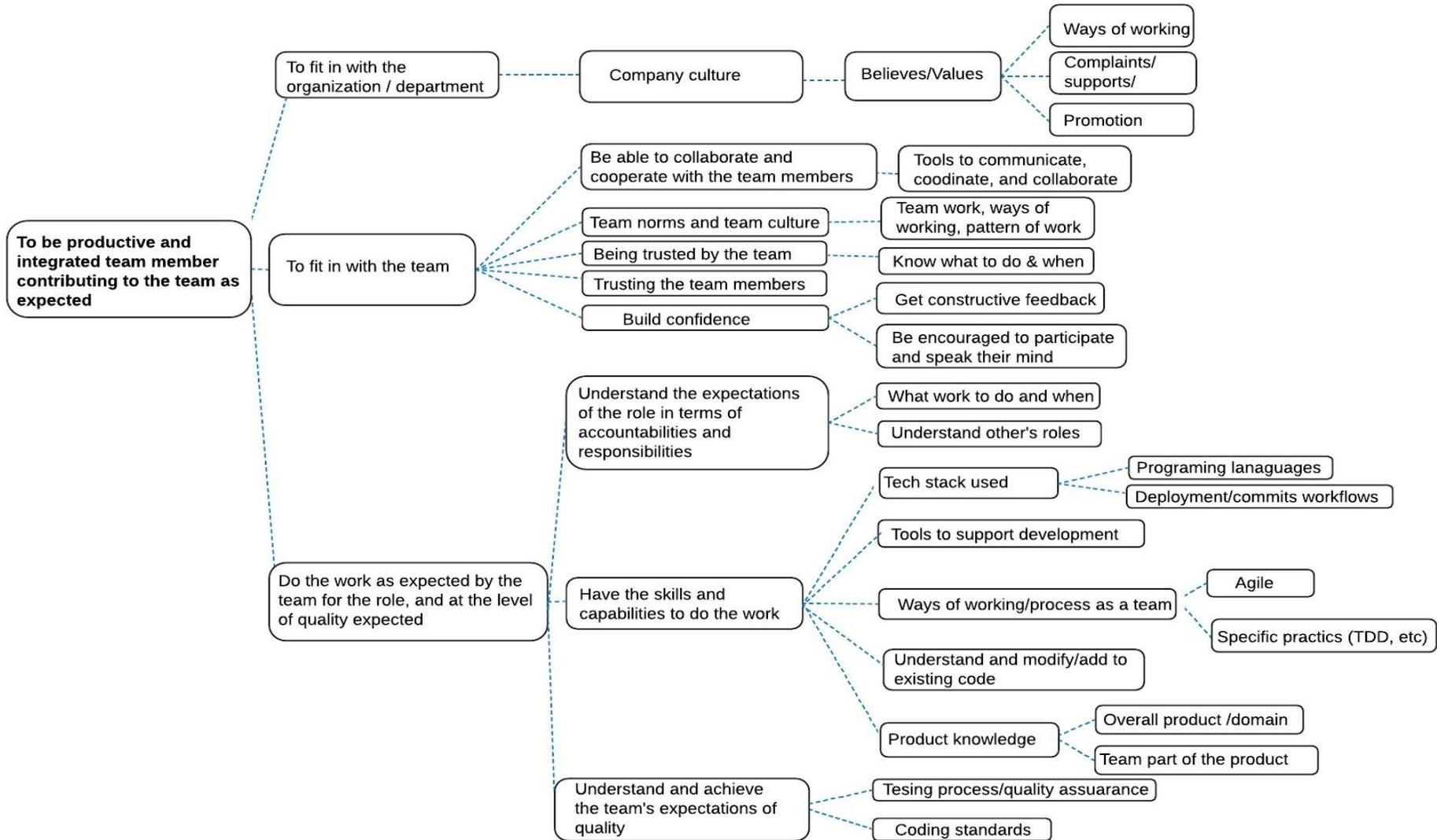


Figure 11

Discovered Team On-boarding Barriers/Challenges/Risks

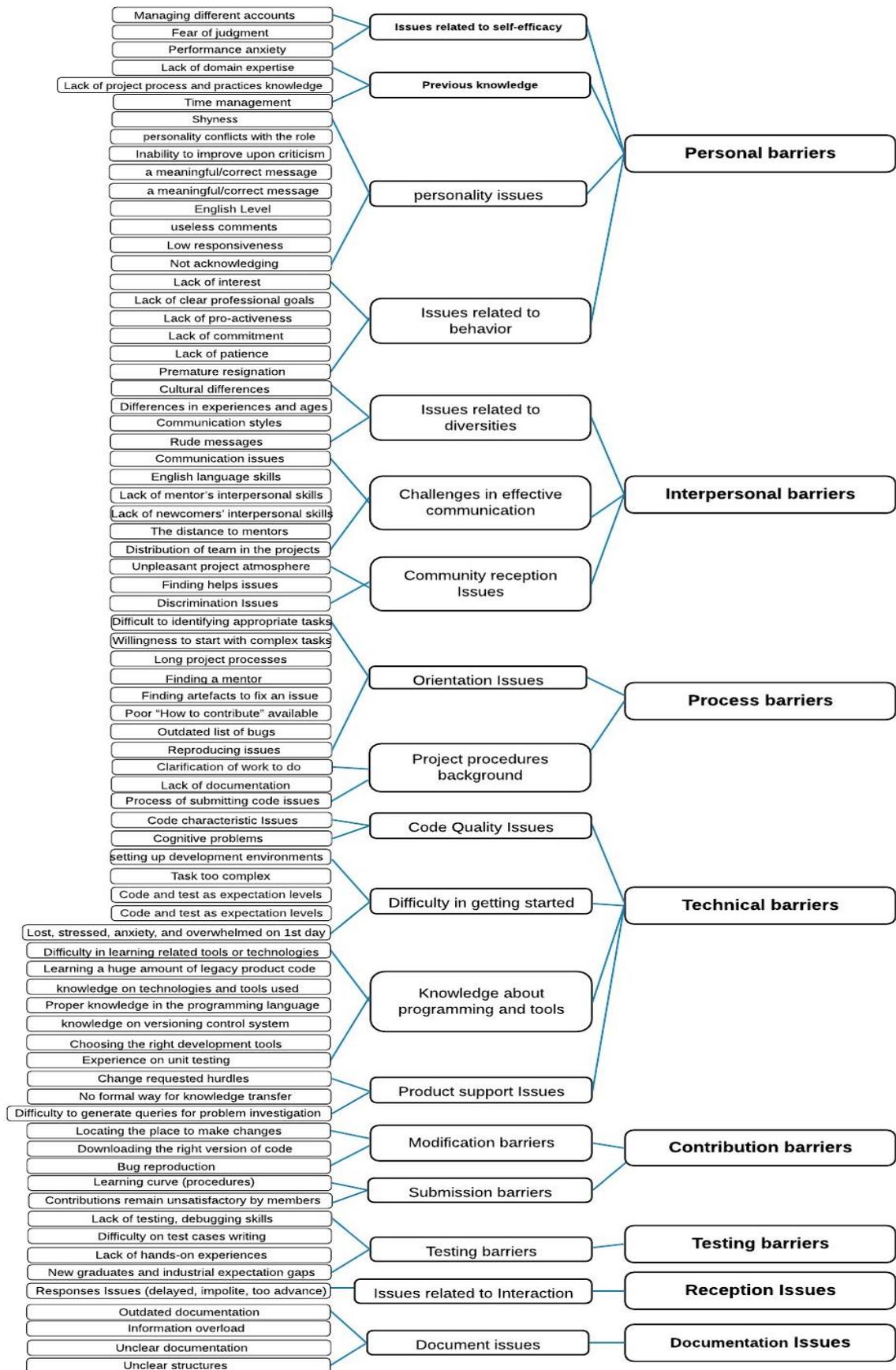
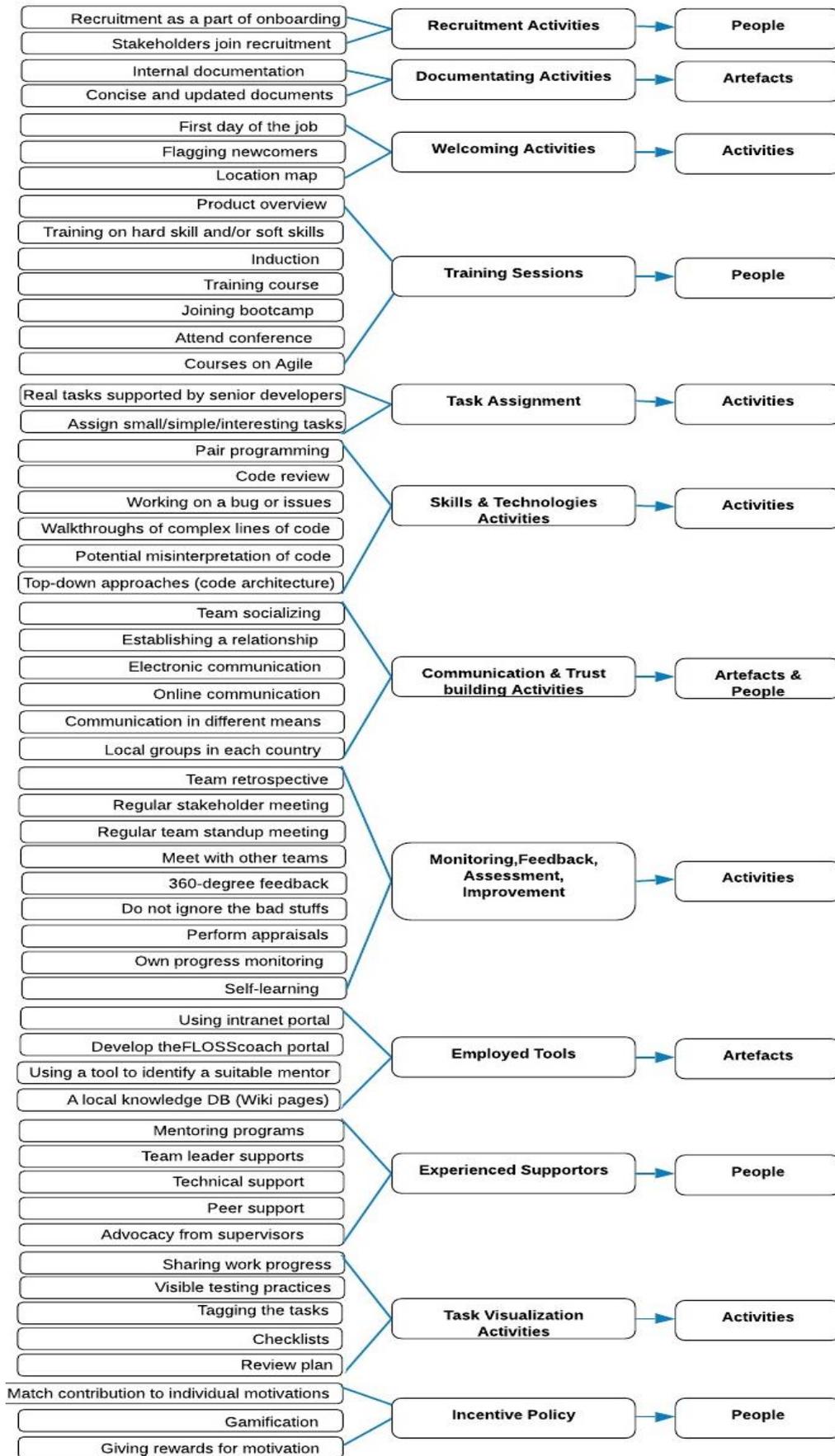


Figure 12

Discovered Team Onboarding Tasks/Activities



CHAPTER 5: PROPOSED ONBOARDING DESIGN PROCESS

This chapter presents a proposed onboarding design process based on the study results reported in the previous chapter. Two main sections are structured to explain the onboarding process and two simulation use cases. The first section contains the overall process diagram and the explanation of the design, related aspects such as team requirements, onboarders' characteristics, and gap distance identification. The second section explains the two simulation use cases for a new graduate and an experienced onboarder joining the company.

5.1 Process Design Overview

From the set of onboarding goals, suggested activities, and possible barriers, challenges and risks presented in Chapter 4 (see Figures 10, 11 and 12), it can be seen that a team personalized onboarding program is essentially required where an onboarding planner needs to filter a set of specific goals and tasks, and carefully consider and acknowledge how to overcome possible risks in different perspectives in order to achieve the program in a timely manner. An overall proposed onboarding design process diagram is shown in Figure 13. The design process consists of six main steps with a number of supporting elements.

STEP 1: Team Requirements, On-boarders' Personalities

This is the initial step of the design of a team personalized onboarding program. The main aim of this step is to collect preliminary data about the team and newcomers, in order to identify the size of the gap ('gap distance') between them by an analysis of team requirements against the personal background, characteristics, capabilities, skills of newcomers. The team requirements refer to architectural development patterns, development tools employed, tech stack, software development practices, code standard, deployment schemes, and/or observability method. The expected results of this step would be a listing of a set of personal gaps of onboarders, and how wide each gap is.

STEP 2: Onboarding Goals Selection

The purpose of this step is to select onboarding goals based on the gap distance discovered in the previous step, related to the team's working standards and expectations. The goals are classified into two categories for selection. The first category is common or generic onboarding goals that all type of onboarders need to be achieved in the onboarding implementation program. These goals are defined based on the company culture and team norms, such as being able to understand a way of working, being trusted by team members, being able to understand the core values of the company. The second goal category is a set of specific goals that are selected based on the

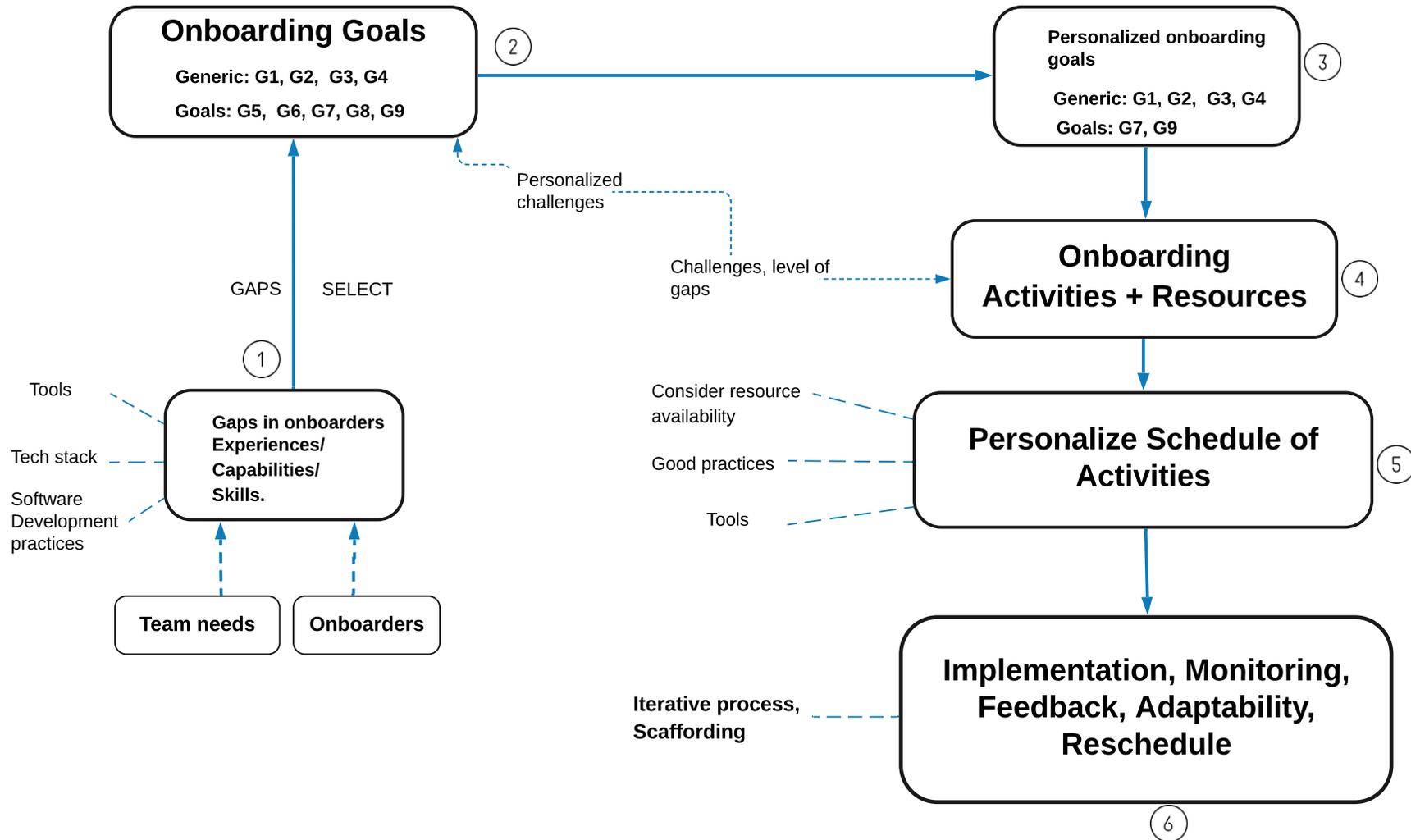
characteristics of on-boarders and the gaps discovered and the gap distance. Different onboarders may be designated with different sets of goals. In addition, the personalized challenges/barriers could also be an aspect of planning a specific goal or even defining a new specific goal that may be uniquely different from any other onboarding goals.

STEP 3: Finalizing a Set of Personalized Onboarding Goals

Based on the results of step 2, this step is for finalizing a complete set of team personalized onboarding goals, with details, that on-boarders are expected to achieve. The onboarding designer can review the list again and classify all desired goals to see how many personalized ones as well as generic ones, to make sure no appropriately personalized goals are missing.

Figure 13

An Overall Proposed Onboarding Design Process



STEP 4: Onboarding Activities & Resources

The main purpose of this step is to plan onboarding activities and resources to support the identified goals. With the finalized list of onboarding goals from the previous step, the onboarding designer will filter appropriate onboarding activities/tasks (see Figure 12) to support each desired goal. Some activities do not only support the single desired goal, and the planner could also schedule onboarders to iteratively implement those tasks to ensure all related goals are achieved. For instance, mentoring could not only support onboarders in being able to understand company culture and team norms but also could support tech stack goals such as being able to code and meet the standard of team expectations. Some significant factors for planners to consider in this step are relevant challenges and the extent of the gaps discovered in step 1. If there are any specific risks or if the gap distance is significant, more tasks may be assigned to onboarders or an appropriate resource nominated to constantly support the newcomer in order to minimize those risks and drive for satisfactory results.

STEP 5: Personalized Activity Schedules

The purpose of this step is to create a personalized schedule of activities or grouping a list of tasks based on related factors. Existing team members have their regular duties, but they may be assigned to or paired with onboarders to support them, especially in a medium- or small- sized software development companies. The onboarding designer should consider resource availability in order to ensure that onboarders continuously receive assistance when they need it. In addition, good practices from previous onboarders also could be considered for new onboarders who have similar characteristics or the same gaps and gap distances. Along with this insight, using tools is a good alternative method that may possibly be employed throughout the entire process of the onboarding program.

STEP 6: Implementation, Monitoring, Feedback, and Adaptability

Once all the desired personalized onboarding goals and supporting tasks are scheduled along with relevant resources, the purpose of this step is implementation according to the plan, monitoring progress from time to time, collecting feedback from perspectives of participating parties, and adaptability in order to maximize the efficiency of the program. The iterative method is offered for the implementation of the entire process in order to scaffold the understanding of an onboarder. The use of artefacts is suggested to track the performance of onboarders in doing related activities as well as for systematic communication among team members.

5.2 Use Cases

To test the application of the proposed onboarding design process (see Figure 13) in software development companies, the researcher created two used cases for simulation and explanation for greater understanding of the process (a new graduate and A newcomer with some industrial experience). These two cases are the common onboarding situations of industrial software development teams.

5.2.1 Use Case 1

Onboarder Characteristics: A new graduate with no industrial experience is about to join a software development company in order to fill a developer position. Due to the lack of work/industrial experience, he has several personal barriers related to tech stack and efficient communication; more importantly he has issues related to behavior such as shyness, and a lack of confidence in sharing ideas, especially in front of many people.

Team Requirement/Characteristics Description: The team of the company that he will be onboarding has norms similar to the company's values. However, this team has specific characteristics and requirements for achievement in product development. For instance, the team uses the Agile method to manage the team members and product development, uses test-driven development (TDD) as the key practice, JavaScript as the code standard, Vscode as the main code editor, Github as code repositories, NodeJS as the runtime environment, and also uses an NoSql database. Any new team members should be able to understand and use these technologies and should be able to adapt themselves to the way of working and meet the expected standards of the team.

Process Design: According to the onboarder's characteristics, team requirements and the proposed design process in Figure 13, the design process for this case could be conducted as follows:

Step 1: Identifying the gap level of the onboarder by analyzing characteristics of the onboarder compared with team requirements.

- To list out all team requirements, onboarding designers need to collect them by interviewing the leader of the team which the new graduate will be integrated into. The list of team requirements could be set out in a table, as in Table 4.

Table 4*Gap Measurement of The Onboarder in Use Case 1.*

No	Team Requirements	Onboarder's score (0-5)	Gaps
1	Agile method	0	High
2	Test-driven development	0	High
3	Code standard-JavaScript	0	High
4	Vscode editor	5	Low
5	Github-repository	2	High
6	NodeJS runtime environment	3	Medium
7	NoSql database type	0	High

Note: Onboarder's score – 0-2 = High, 3-4 =Medium, 5 = Low.

- To assign the gap level of the onboarder based on team requirements, there are some questions that the on-boarder could be asked and scored on each item. The questions could be asked as follows:
 - a) How experienced and familiar are you with the Agile method?
 - b) How experienced and familiar are you with TDD practice?
 - c) How experienced and familiar are you with the JavaScript code standard?
 - d) How experienced and familiar are you with the VScode editor?
 - e) How experienced and familiar are you with the Github repository tool?
 - f) How experienced and familiar are you with the NodeJS?
 - g) How experienced and familiar are you with the NoSql database?

- The on-boarder has some challenges for consideration such shyness, lack of confidence.

Step 2: Due to the onboarder having no industrial experience, there are some significant gaps with regard to team requirements (see Table 4). The set of desired onboarding goals is in place. While some of them are the generic/common ones that all newcomers need to achieve, others are personalized goals that will be filtered based on the gaps discovered in the previous step and personal challenges/barriers. See Figure 10 in Chapter 4.

Step 3: Based on the analysis of gaps, the personal challenges/barriers of on-boarders, and the generic desired goals, a set of personalized onboarding goals for the onboarder could be finalized as shown in Table 5:

Table 5*A Final Set of Personalized Goals for Use Case 1.*

No	Personalized Desired Goals	Note
G1	Be able to understand company culture (values/beliefs)	Personalized based on personality and
	- Understand ways of working: Compliance, support, promotion	

No	Personalized Desired Goals	Note
G2	Be able to collaborate and cooperate with the team members - Be able to use communicating tools	industrial working experience on teams
G3	Team norms and team culture - Ways of team working: complaints/support/team member promotion	
G4	Being trusted by the team - Know what to do and when - Be able to complete the commitment	
G5	Build confidence - Be able to give constructive feedback - Be able to participate and speak out their mind	
G6	Understand the expectations of the role in terms of accountabilities and responsibilities - Understand other's roles	
G7	Have the skills and capabilities to do the work - Tech stack used - Tools to support development - Ways of working/process as a team: Agile, TDD - Understand and modify/add to existing code - Be able to understand product knowledge	
G8	Understand and achieve the team's expectations of quality - Be able to conduct testing defined processes/quality assurance - Be able to understand and code to team's code standard	

Note: These personalized onboarding goals are expected to achieve.

Steps 4 & 5: Based on the final set of personalized onboarding goals from step 3, the level of gaps discovered, and the challenges of the onboarder, the assignment of supported activities and appropriate resources for each goal could be done in this step. While some activities not only support a single personalized goal, a team member could possibly act as a mentor and/or a peer as well, to support the on-boarder so that the onboarding planners could possibly also offer more than one iterative implementation for some activities to ensure that relevant goals are supported. Once the supported activities and resources are planned, they need to be properly scheduled, relying on the availability of resources, employing some good practices and tools for systematic implementation in the final step – see Table 6.

Step 6: Since the plan has been prepared in the previous steps, it is time to execute it. As an iterative approach is offered for implementing the activities, they could be implemented more than once in order to ensure that knowledge/skill scaffolding reaches the expectations of team quality. The canvas method is offered for the iteration of implementation, monitoring, feedback on progress or supported rescheduling.

Based on the task lists in Tables 5 and 6, it could be assumed that the onboarder is still unable to perform well in terms of the expectation of team quality. Therefore, some tasks and their details could be customized or rescheduled, or a new personalized sub-goal might even be created for the onboarder to achieve in order to reach the main goal.

Table 6*The List of Suggested Activities With Desired Goals for Use Case 1.*

No	Activity Desc	Schedule	Resources	Supported Goals
1	Having a mentor	3 iterations in 2 weeks	Senior developers	G1, G2, G3, G4, G5
2	Join the induction session	2 iterations in 1 st week	Human resources	G1, G2, G3, G4, G5
3	Join a pair programming session	2 iterations in 2 nd week	Experienced developer	G7, G8, G9
4	Join a product overview session	2 iterations in 1 st week	Team leader	G1, G2, G3, G4, G5
5	Join a bootcamp	2 nd week of the program	Online course	G5, G6, G7, G8
6	Code review/walkthrough complex lines of code	2 iterations of a function	Senior developer	G5, G6, G7, G8
7	Regular team stand up meeting	At least 3 iterations	Team members	G1, G2, G3, G4, G5
8	Join team retrospectives	At least 2 iterations	Team members	G2, G4, G5, G8
9	Join a course on Agile	At least 1 iteration	Scrum master	G1, G2, G3, G4, G5
10	Having a peer support	At least 2 iterations	Experienced developer	G2, G4, G5, G8
11	Conduct a review plan	Every day of first week	Team leader	G1, G2, G3, G4, G5
12	Sharing work progress	Every single completed task	Local repository	G2, G4, G5
13	Having communication by different means	At least 2 different schemes	Social media, in person	G1, G2, G3, G5
14	Advocacy from supervisor	At least 3 iterations	Team leader	G1, G2, G3, G4, G5
15	Own progress monitoring	At least 2 iterations	Onboarder	G5, G6, G7, G8
16	Attend conference	At least 2 iterations	Weekly conference	G1, G2, G3, G4, G5
17	Self-learning	At least 2 iterations	Online course	G6, G7, G8
18	Share testing activities	Every single completed test	Internal tool	G6, G7, G8

5.2.2 Use Case 2

On-boarder Characteristics: A newcomer with some industrial experience is about to join a software development company in order to fill a developer position. The newcomer has some experience in coding, testing, and is able to understand the expected code standard. She has no personal barriers related to team working and is able to communicate well by different means and in different situations, especially in front of many people.

Team Characteristics: Assume that the team characteristics are the same as in used case 1.

Process Design: According to the onboarder's characteristics, team requirements and the proposed design process in Figure 13, the design process for the second use case could be designed as follows:

Step 1: Identifying the gap level of the onboarder by analyzing the characteristics of the onboarder compared with team requirements.

e) To list out all team requirements, onboarding designers need to collect them by interviewing the leader of the team which the newcomer will be integrating into. The list of team requirements could be set out in a table, as in Table 7 below:

Table 7

Gap Measurement of the Onboarder in Use Case 2.

No	Team Requirements	On-boarder's score (0-5)	Gaps
1	Agile method	4	Low
2	Test-driven development	5	Low
3	Code standard-JavaScript	5	Low
4	Vscode editor	5	Low
5	Github-repository	4	Low
6	NodeJS runtime environment	5	Low
7	NoSql database type	5	Low

Note: Onboarder's score – 0-2 = High, 3-4 =Medium, 5 = Low.

f) To generate the gap level of the onboarder according to the list of team requirements, the onboarder could be asked the same questions as use case 1 and scored on each item.

Step 2: Because the onboarder has some industrial experience, the gaps distance to the team requirements is very low (see Table 7). The set of desired onboarding goals is in place from the previous chapter. Some of them are the generic/common ones that all newcomers need to be

achieved and others are personalized goals that could be filtered based on the gaps discovered in the previous step, and personal challenges/barriers.

Step 3: Based on the analysis of gaps, the personal challenges/barriers of on-boarders, and the generic desired goals, a set of personalized onboarding goals of the on-boarder could be finalized as shown in Table 8:

Table 8

A Final Set of Personalized Goals for Use Case 2.

No	Personalized Desired Goals	Note
G1	Be able to understand company culture (values/beliefs)	Personalized based on personality and industrial working experience on teams
	- Understand ways of working: Compliance, support, promotion	
G2	Be able to collaborate and cooperate with the team members	
	- Be able to use communicating tools	
G3	Team norms and team culture	
	- Ways of team working: complaints/support/team member promotion	
G4	Being trusted by the team	
	- Know what to do and when	Personalized based on the team's work, ways of working, tools, expectations, and the capabilities of the onboarder.
	- Be able to complete the commitment	
G5	Understand the expectations of the role in terms of accountabilities and responsibilities	
	- Understand other's roles	
G6	Have the skills and capabilities to do the work	
	- Tech stack used	
	- Tools to support development	
	- Ways of working/process as a team: Agile, TDD	
	- Understand and modify/add to existing code	
	- Be able to understand product knowledge	
G7	Understand and achieve the teams' expectations of quality	
	- Be able to conduct testing defined processes/quality assurance	
	- Be able to understand and code to team's code standard	

Note: These personalized onboarding goals are expected to achieve.

Steps 4 & 5: Similar to use case 1, based on the level of gaps discovered, and the challenges/barriers of the onboarder, the assignment of supported activities and appropriate resources for each goal could be done in this step. They need to be scheduled relying on the availability of resources, employing some good practices and tools for systematic implementation in the final step – see Table 9.

Step 6: Based on Table 9, similar to the use case 1, an iterative approach is offered for implementing the activities. They could be implemented more than once to make sure that knowledge/skill scaffolding meets the expectation of the team's work quality. The canvas method is offered for the monitoring step of implementation. Pending tasks could be carrying forward to implement in the next sprint.

Based on the task lists in Tables 8 and 9, it could be assumed that the onboarder could implement the plan well. For instance, she will be able to meet the team's work quality quickly and gain team knowledge after finishing some tasks while other related tasks are yet to begin. In this case, the onboarder could skip some of the tasks that have not yet started, and which support similar outcomes, in order to save time. All task implementation depends on the monitoring progress and results.

Table 9*The List of Suggested Activities With Desired Goals for Use Case 2.*

No	Activity Desc	Schedule	Resources	Supported Goals
1	Having a mentor	2 iterations of 3 weeks	Senior developers	G1, G2, G3, G4, G5
2	Join the induction session	One time in 1 st week	Human resources	G1, G2, G3, G4, G5
3	Join a pair programming session	One time in 1 st week	Experienced developer	G5, G6, G7
4	Join a product overview session	One time in 2 nd week	Team leader	G1, G2, G3, G4
5	Code review/walkthrough complex lines of codes	2 iterations of a function	Senior developer	G5, G6, G7
6	Regular team stand up meeting	At least one time	Team members	G1, G2, G3, G4, G5
7	Join team retrospectives	At least one time	Team members	G2, G4, G5, G7
8	Join a course on Agile	At least one time	Scrum master	G1, G2, G3, G4, G5
9	Having a peer support	At least one time	Experienced developer	G2, G4, G5, G7
10	Conduct a review plan	At least 2 iterations	Team leader	G1, G2, G3, G4, G5
11	Sharing work progress	All key tasks	Local repository	G2, G4, G5
12	Having communication in different means	At least 2 different schemes	Social media, in person	G1, G2, G3, G5
13	Advocacy from supervisor	At least 2 iterations	Team leader	G1, G2, G3, G4, G5
14	Own progress monitoring	At least 2 iterations	On-boarder	G5, G6, G7
15	Attend conference	At least 2 iterations	Weekly conference	G1, G2, G3, G4, G5
16	Share testing activities	Every single of completed test	Internal tool	G5, G6, G7

5.3 Comparison of Use Cases 1 and 2

It can be seen that although the team requirements are the same, the difference in the onboarders' characteristics in use cases 1 and 2 is a factor which influences the finalization of the onboarding goal lists, supporting tasks, and implementation schedules and time.

Generic onboarding goals are equally assigned to the onboarders in use cases 1 and 2. This is because they are common goals that everyone who joins the team needs to be able to understand and adapt themselves to, for instance, being able to understand the values or beliefs of the company and understand the team's working standards and expectations.

Personalized onboarding goals: Use case 1 had no industrial experience and some personal barriers while the onboarding designer in use case 2 had some industrial experience as well as no barriers in communicating skills. The gap distances clearly show the differences (see Table 4 for case 1 and Table 7 for case 2) so that the onboarding designer planned and finalized the set of desired goals differently. For instance, in building confidence, the onboarding designer in use case 1 is expected to be able to confide in order to communicate to the team members and contribute ideas confidently to the team, while the onboarding designer in use case 2 already has these skills from her previous work experiences.

5.4 A Proposed Tool

A tool could be developed to facilitate onboarding designers in designing personalized team onboarding programs following the designed process (see Figure 13). The main functions of the tool proposed in this prototype are team setting, gathering onboarders' characteristics, and new design.

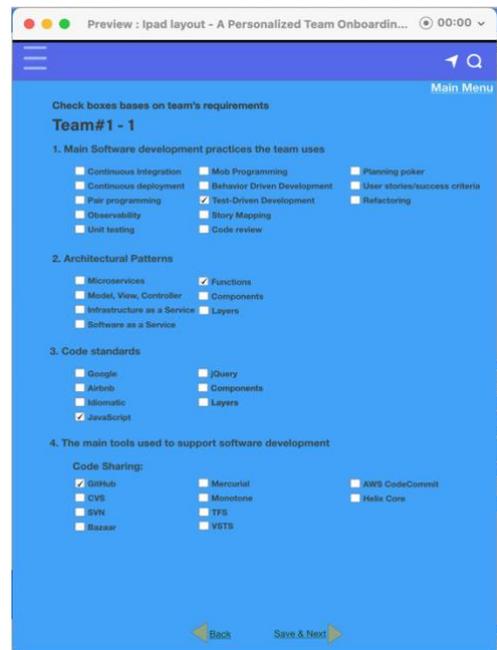
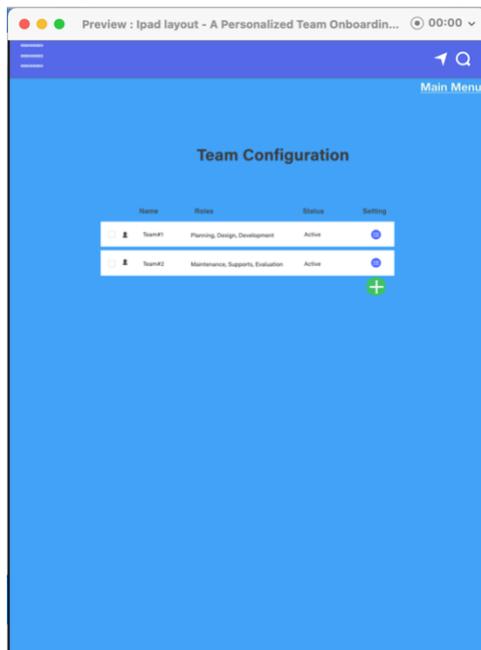
5.4.1 Team Setting

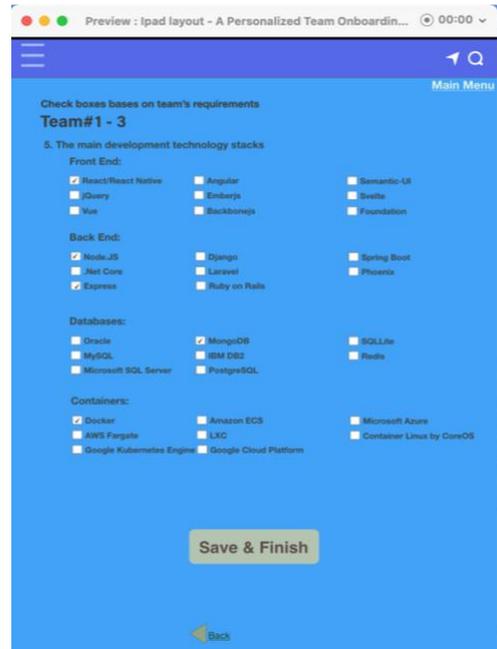
A software development company could have more than one team with different characteristics or requirements. The gathering of team requirements is needed and can be configured into the tool. An onboarding designer could ask a team leader a set of questions in order to collect the team's characteristics, with the questions potentially as follows:

- a) What are the main software development practices the team uses?
- b) What are the architectural patterns the team uses?
- c) What are the code standards the team uses?
- d) What are the main tools used to support the development?
- e) What are the main tools used for planning and monitoring the work board?

- f) What are the main coding IDEs the team uses?
- g) What are the testing frameworks/libraries the team uses?
- h) What are the deployment repositories the team uses?
- i) What are the observability tools the team uses?
- j) What are the main development stacks the team uses? Front end, back end, databases, containers.

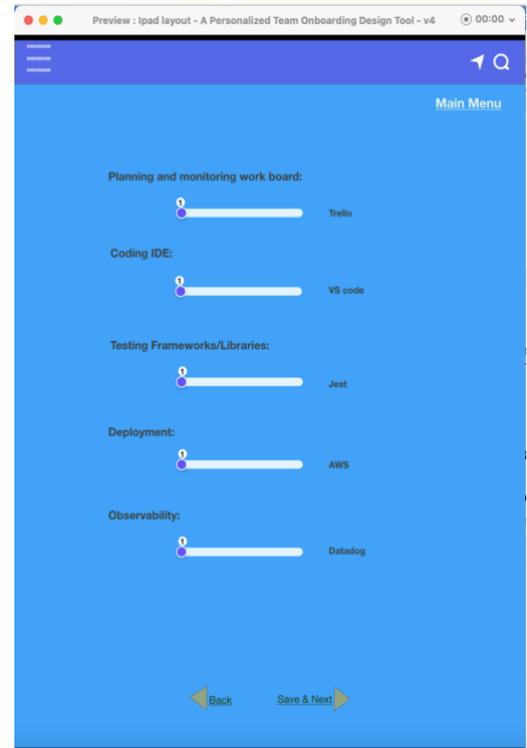
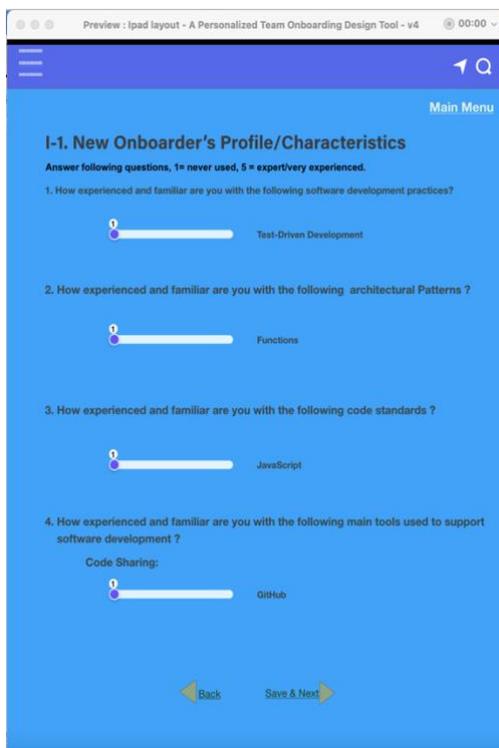
These questions could be customized to ensure all essential team requirements are captured. The following screens are samples of the proposed prototype. To capture the team's characteristics, start clicking the list of a team in order to configure it, then follow with the questions asked and check the boxes that are applicable to the team. Then click the Save & Finish button to record team requirements.

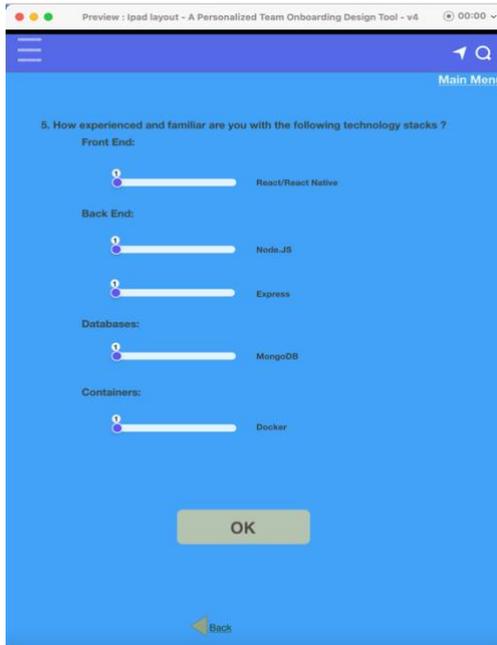




5.4.2 Onboarder's Characteristics

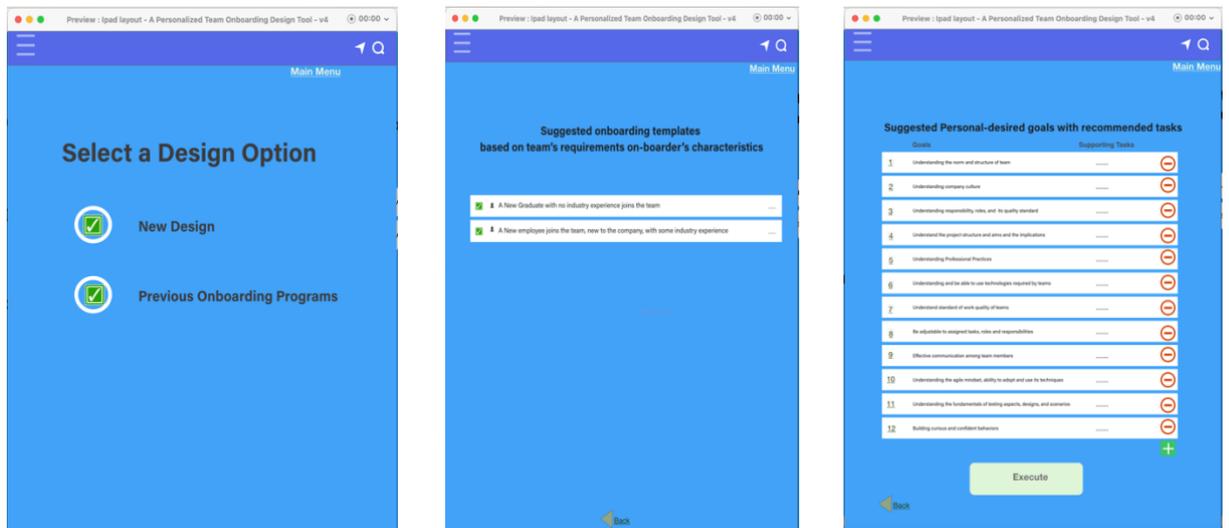
Based on the team configurations and responses to the questions asked, a set of questions will be shown on the screens to ask about the onboarder and capture the onboarder's characteristics. The user can score the onboarder's familiarity with each aspect of the team, as shown on the following screens.



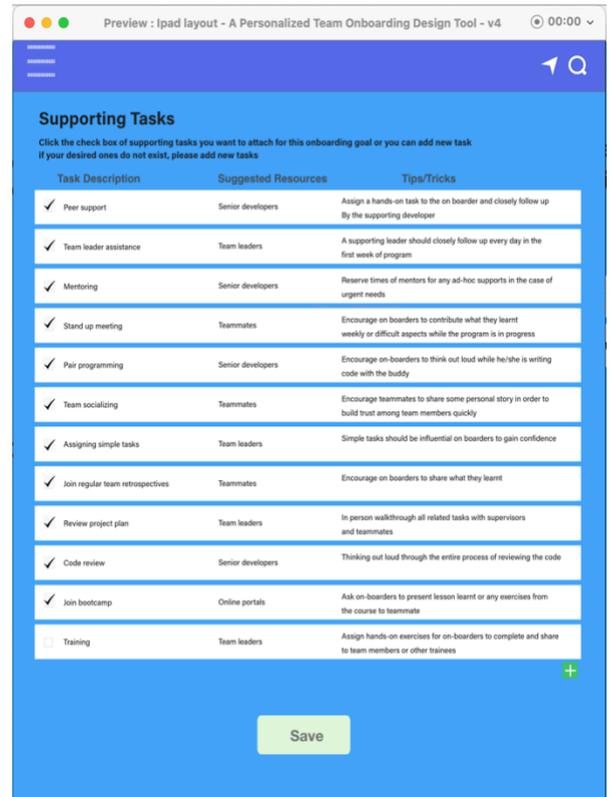
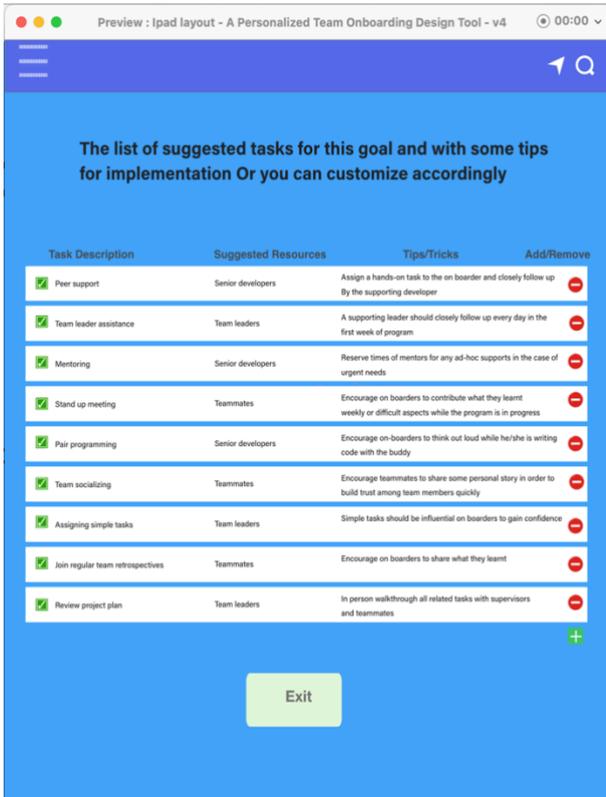


5.4.3 A Set of Onboarding Goals and Supporting Tasks

Once the tool has recorded the familiarity of onboarders with the team's characteristics, a set of team personalized onboarding goals is suggested with supporting tasks

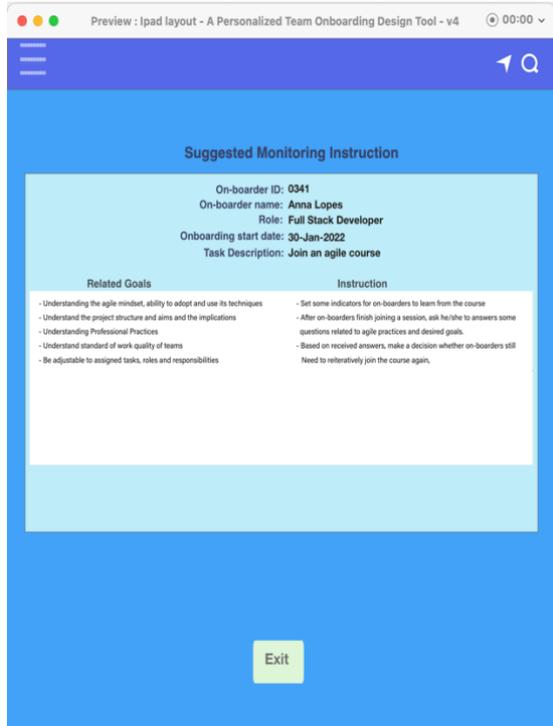


but the onboarding designer can customize this based on other personal assessments of the onboarder, these below screen are consequently linked from previous ones



A set of suggested tasks for each desired goal is listed for the onboarder to implement, as shown in the following screens of the tool.





CHAPTER 6: CONCLUSION AND FUTURE WORK

This chapter summarizes the key aspects of the thesis. The first section describes the conclusion of thesis starting from purposes, methodology, result of data analysis, and ending up with the proposed solution for a personalized team onboarding design process. Some threats to validity are explained in section two. The third section present the possible aspects for future research related to the personalized team onboarding design process.

6.1 Conclusion.

The purpose of this thesis was to understanding existing team onboarding practices in software development contexts and proposing a design process for a personalized on-boarding design process. Five research questions are created to guide the investigation and find answers to achieve the purpose and the goal on the research.

The research has been achieved its key aims throughout the investigation of common identifying desired goals, supporting activities, challenges/difficulties/risks, and other relevant factors. Four main onboarding contexts were explored in the existing literature to provide conceptualisation of the personalisation of the onboarding design problem. This provided a research framework to guide the research areas to consider

A Design Science Research approach was adopted in the research because it suited the type of research leading to building an artefact (the design process), which was the goal. Most of the work was done n the first three phases: the problem identification, objectives of a proposed solution and build phases. A Systematic Literature Review was used in each of the phases to collect and analyse the relevant information from academic literature and work out how to build the personalised onboarding design process and a prototype supporting software tool.

To answer all five research questions in this research, 36 related articles are reviewed systematically and studied on related work of onboarding in software development contexts. A definition of team onboarding in software contexts are summarized based on the consolidated results, the figure 3 in Chapter 2 shown the summary. Three main on-boarding extracted data components were analyzed and categorized (a common desired on-boarding goals, supported activities, and possible risks). With the result of analyzed qualitative data (from iterative reading, assigning code, grouping, categorizing, synthesizing), three main onboarding data aspects are found (goals, tasks, challenges). Each of them has more than one category reported (figure 10,11&12).

The main outcome of the research is a team personalized onboarding process design. From the main findings and in-depth understanding in different facets of team onboarding in software development, a design process of personalized team onboarding is designed as well as explanation of all related elements factors that could influence how the desired goals, tasks, and implementation could be planned. A tool was also designed and a prototype built to support the designed process of the personal team onboarding in software development.

6.2 Limitations and threats to validity

To collect research data, relevant journal articles were retrieved from three biggest online databases by having some main key words “onboarding”, and “software development” as well as some fundamental criterions were being proceeded. However, it’s quite challenges to ensure the output of searching query displaying all related articles due to the term of onboarding could be represented by other phrases and words such as organizational socialization, employee integration. To reduce this threat, some experimentation was initially done with key words to identify the ones most commonly used in literature. Also, a “snowballing” techniques was used where the reference lists of relevant articles were checked for other relevant articles.

Ideas from contexts other than onboarding to Agile software development teams (e.g. onboarding in open source development) were used with the assumption that some of these ideas would be applicable to Agile team onboarding also. There is a threat to validity since some of these assumptions may be wrong. I tried to make some reasoning and say the assumptions to help with this.

Thematic analysis was used for analyzing and categorizing the extracted data which relies on the researcher’s thinking in the data interpretation when theming data (goals, tasks, and challenges of onboarding). These themes and codes can be quite subjective and someone else may come up with other themes. To reduce this threat, my supervisor also coded some of the things and we then discussed and agreed on themes.

Another threat to validity could be used cases for testing the proposed onboarding design process. The simulated onboarders were defined for two cases, a new graduate and an experienced person. The onboarding goals and supported activities were assigned for both of them similarly while the implementing schedule of the new graduate was longer. The more various cases and different characteristics of onboarders should be tested under this designed process for validation the process and findings. The lack of real-world evaluation of the design process and the implementation of the designed personalized onboarding process is a threat that is future work.

6.3 Future Research

The study on personalised team onboarding is mainly from the view of the designer and the onboarder. There are other perspectives of personalized team onboarding that should be investigated in the future, such as mentor's viewpoints, virtual teams' leaders, and the other team members. In addition, a tool could be developed more and used by some software development companies in order to validate the proposed design process of onboarding.

While the onboarding designers nominate a set of personalized desired goals and supported tasks by filtered questions, mentors could possibly have a different viewpoint to pursue how an onboarders could possibly succeed. The mentors could identify a list of good practices where the proposed designed process may be unable to address, and some ad-hoc solution may be required. Many onboarders shared their personal thoughts to mentors where they feel safe. Along with the mentor's perspectives, it could be significant to further personalise an onboarding program by considering the work experience and characteristics of mentors. With the diverse culture and beliefs, mentors may consider a situation of onboarders in different ways and use a unique solution to support onboarders. Therefore, it would be essential to know the mentor's perspectives and expectation to make the onboarding design more effective.

In the era of working outside the office due to the pandemic, the situation for onboarding in face-to-face may not be possible. The virtual personalized onboarding in distributed software development teams could increase in any types of development projects. To understand how the leaders of virtual team perspectives or project stakeholders on a personalized onboarding design process could be useful. While the team leaders and project stakeholders may work closely with onboarders when they fully join the team, it is important to have inputs from these group of people to feedback on a personalized onboarding program or how the process could be designed in order to ensure the newcomers would be productive faster and fully contribute their effort into the team. Or hearing from the team leaders, stakeholders, mentors, and/or human resource manager could be an asset for mapping a set of tasks to appropriate onboarders or overcomes any critical risks.

It is important that the design process is low effort, and a supporting tool would help with this. While this research proposed the personalized onboarding design process and a prototype, the further viewpoints on the process could be valuable as well as the actual tool development to support the process.

REFERENCES

- Abdulkadir, Ş., Banu, D., & Halil, A. (2021). New developer metrics for open source software development challenges: An empirical study of project recommendation systems. *Applied Sciences*, *11*(920), 920-920. <https://doi.org/10.3390/app11030920>
- Ahmad, A., & Kolla, S. J. (2012). *Effective Distribution of Roles and Responsibilities in Global Software Development Teams* [Student thesis, Blekinge Institute of Technology]. DiVA. <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-4701>
- Allison, I. (2010). Organizational factors shaping software process improvement in small-medium sized software teams: A multi-case analysis. *Proceedings of the 2010 Seventh International Conference on the Quality of Information and Communications Technology* (pp. 418-423). IEEE. <https://doi.org/10.1109/QUATIC.2010.81>
- Babajide, B., & yagoub, H. A. A. (2019). Exploring the role of employers' onboarding programs in preparing early-career engineers [Conference]. 1-6. <https://doi.org/10.1109/FIE43999.2019.9028654>
- Balali, S., Steinmacher, I., Annamalai, U., Sarma, A., & Gerosa, M. A. (2018). Newcomers' barriers... Is that all? An analysis of mentors' and newcomers' barriers in OSS projects. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, *27*(3-6), 679.
- Batistič, S., & Kaše, R. (2015). The organizational socialization field fragmentation: A bibliometric review. *Scientometrics*, *104*(1), 121.
- Batistič, S., & Kenda, R. (2018). Toward a model of socializing project team members: An integrative approach. *International Journal of Project Management*, *36*(5), 687-700. <https://doi.org/10.1016/j.ijproman.2018.03.003>
- Bauer, T., & Erdogan, B. (2011). Organizational socialization: The effective onboarding of new employees. In S. Zedeck (Ed.), *APA handbook of industrial and organizational psychology, Vol. 3. Maintaining, expanding, and contracting the organization* (pp. 51-64). <https://doi.org/10.1037/12171-002>
- Bauer, T. N., Bodner, T., Erdogan, B., Truxillo, D. M., & Tucker, J. S. (2007). Newcomer adjustment during organizational socialization: A meta-analytic review of antecedents, outcomes, and methods. *Journal of Applied Psychology*, *92*(3), 707.
- Borrego, M., Foster, M. J., & Froyd, J. E. (2014). Systematic literature reviews in engineering education and other developing interdisciplinary fields. *Journal of Engineering Education*, *103*(1), 45-76.
- Bramer, W. M., Milic, J., & Mast, F. (2017). Reviewing retrieved references for inclusion in systematic reviews using EndNote. *Journal of the Medical Library Association: JMLA*, *105*(1), 84.
- Briner, R. B., & Denyer, D. (2012). Systematic review and evidence synthesis as a practice and scholarship tool. In D. M. Rousseau (Ed.), *Handbook of evidence-based management: Companies, classrooms and research* (pp. 112-129). Oxford Handbooks. <https://doi.org/10.1093/oxfordhb/9780199763986.013.0007>
- Britto, R., Cruzes, D. S., Smite, D., & Sablis, A. (2018). Onboarding software developers and teams in three globally distributed legacy projects: A multi-case study. *Journal of Software: Evolution & Process*, *30*(4), 1.
- Britto, R., Damm, L. O., Smite, D., & Borstler, J. (2019, 2019 / 05 / 01 /). Performance Evolution of Newcomers in Large-Scale Distributed Software Projects: An Industrial Case Study [Conference Paper]. 1-11. <https://doi.org/10.1109/ICGSE.2019.00000>

- Britto, R., Smite, D., Damm, L.-O., & Börstler, J. (2020). Evaluating and strategizing the onboarding of software developers in large-scale globally distributed projects. *Journal of Systems & Software, 169*, 110699. <https://doi.org/10.1016/j.jss.2020.110699>
- Bryne, M. (2001). Data analysis strategies for qualitative research. (Research Corner). *AORN Journal, 74*(6), 904+.
- Buchan, J., MacDonell, S. G., & Yang, J. (2019). Effective team onboarding in Agile software development: techniques and goals [Conference]. 1-11. <https://doi.org/10.1109/ESEM.2019.8870189>
- Budgen, D., & Brereton, P. (2006). Performing systematic literature reviews in software engineering. *Proceedings of the 28th International Conference on Software Engineering* (pp. 1051-1052). ACM. <https://doi.org/10.1145/1134285.1134500>
- Butler, A., Hall, H., & Copnell, B. (2016). A guide to writing a qualitative systematic review protocol to enhance evidence-based practice in nursing and health care. *Worldviews on Evidence-Based Nursing, 13*(3), 241-249. <https://doi.org/10.1111/wvn.12134>
- Canfora, G., Penta, M. D., Oliveto, R., & Panichella, S. (2012). Who is going to mentor newcomers in open source projects? *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, Cary, North Carolina* (Article No. 44, pp. 1–11). ACM. <https://doi.org/10.1145/2393596.2393647>
- Carter, M., & Carter, C. (2020). The creative business model canvas. *Social Enterprise Journal, 16*(2), 141-158. <https://doi.org/10.1108/SEJ-03-2019-0018>
- Carver, J. C., Hassler, E., Hernandez, E., & Kraft, N. A. (2013). Identifying barriers to the systematic literature review process. *Proceedings of the 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 203-212). IEEE. <https://doi.org/10.1109/ESEM.2013.28>
- Cauwelier, P. (2016). *The influence of team psychological safety on team knowledge creation: a comparative study between Thai, French and American engineering teams* [Doctor of Philosophy in Knowledge and Innovation Management, The Institute for Knowledge and Innovation Southeast Asia (IKI-SEA)]. <http://dspace.bu.ac.th/jspui/handle/123456789/1992>
- Cauwelier, P. (2019). Building high-performance teams through action learning. *Action Learning: Research and Practice, 16*(1), 68-76. <https://doi.org/10.1080/14767333.2019.1562693>
- Cauwelier, P., Ribiere, V. M., & Bennet, A. (2019). The influence of team psychological safety on team knowledge creation: a study with French and American engineering teams. *Journal of Knowledge Management, 23*(6). <https://doi.org/10.1108/JKM-07-2018-0420>
- Cooper-Thomas, H. D., & Wilson, M. G. (2011). Influences on newcomers' adjustment tactic use. *International Journal of Selection & Assessment, 19*(4), 388. <https://doi.org/10.1111/j.1468-2389.2011.00567.x>
- Cruzes, D. S., & Dyba, T. (2011). Recommended steps for thematic synthesis in software engineering. *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement* (pp. 275-284). IEEE. <https://doi.org/10.1109/ESEM.2011.36>
- Cunha, R. F. V., Souza, F. B. S., Lima, F. O., & Bonifacio, B. A. (2020). How do newcomers learn work process in global software development (GSD)? A survey study from the perspective of newly project leaders. *Proceedings of 2020 ACM/IEEE 15th International Conference on Global Software Engineering, ICGSE 2020* (pp. 71–75). ACM. <https://doi.org/10.1145/3372787.3389302>

- Davidson, J. L., Naik, R., Mannan, U. A., Azarbakht, A., & Jensen, C. (2014). On older adults in free/open source software: Reflections of contributors and community leaders. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* (pp. 93-100). IEEE. <https://doi.org/10.1109/VLHCC.2014.6883029>
- Deshpande, A., & Gupta, R. (2019). Newcomer retention – An investigation on Indian IT industry. *Management Decision*, 59(4), 787-800. <https://doi.org/10.1108/MD-02-2019-0233>
- Dierckx de Casterlé, B., Gastmans, C., Bryon, E., & Denier, Y. (2012). QUAGOL: A guide for qualitative data analysis. *International Journal of Nursing Studies*, 49(3), 360-371. <https://doi.org/10.1016/j.ijnurstu.2011.09.012>
- Diniz, G. C., Silva, M. A. G., Gerosa, M. A., & Steinmacher, I. (2017). Using gamification to orient and motivate students to contribute to oss projects. *Proceedings of the 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2017* (pp. 36-42). ACM. <https://doi.org/10.1109/CHASE.2017.7>
- Dominic, J., Houser, J., Steinmacher, I., Ritter, C., & Rodeghero, P. (2020). Conversational bot for newcomers onboarding to open source projects. *Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020* (pp. 46-50). ACM. <https://doi.org/10.1145/3387940.3391534>
- Dooley, J. F. (2017). *Software development, design and coding: With patterns, debugging, unit testing, and refactoring* (2nd ed.). Apress.
- Edmondson, A. (1999). Psychological safety and learning behavior in work teams. *Administrative Science Quarterly*, 44(2), 350-383.
- Ellis, A. M., Nifadkar, S. S., Bauer, T. N., & Erdogan, B. (2017). Newcomer adjustment: Examining the role of managers' perception of newcomer proactive behavior during organizational socialization. *Journal of Applied Psychology*, 102(6), 993-1001. <https://doi.org/10.1037/apl0000201>
- Escobar-Sarmiento, V., & Linares-Vásquez, M. (2012). A model for measuring agility in small and medium software development enterprises. *Proceedings of the 2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)* (pp. 1-10). IEEE. <https://doi.org/10.1109/CLEI.2012.6427226>
- Fagerholm, F., Johnson, P., Sanchez Guinea, A., Borenstein, J., & Münch, J. (2013). Onboarding in open source software projects: A preliminary analysis. *Proceedings of the 2013 IEEE 8th International Conference on Global Software Engineering Workshops, ICGSEW 2013* (pp. 5-10). ACM. <https://doi.org/10.1109/ICGSEW.2013.8>
- Fagerholm, F., Sanchez Guinea, A., Borenstein, J., & Munch, J. (2014). Onboarding in open source projects. *IEEE Software*, 31(6), 54.
- Fagerholm, F., Sanchez Guinea, A., Münch, J., & Borenstein, J. (2014). The role of mentoring and project characteristics for onboarding in open source software projects. *International Symposium on Empirical Software Engineering and Measurement*,
- Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science*, 46(12), 1554-1568.

- Finn, P. (2006). *The importance of being onboarded*. Northwestern University School of Education and Social Policy. Retrieved October, 16, 2009, from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.496.6193&rep=rep1&type=pdf>
- Fritscher, B., & Pigneur, Y. (2014). *Visualizing business model evolution with the business model canvas: concept and tool* [Conference]. 1, 151-158. <https://doi.org/10.1109/CBI.2014.9>
- Gibbs, G. R. (2007). *Analyzing qualitative data*. Sage. <https://doi.org/10.4135/9781849208574>
- Griffin, A. E., Colella, A., & Goparaju, S. (2000). Newcomer and organizational socialization tactics: An interactionist perspective. *Human Resource Management Review, 10*(4), 453-474.
- Gruman, J. A., Saks, A. M., & Zweig, D. I. (2006). Organizational socialization tactics and newcomer proactive behaviors: An integrative study. *Journal of Vocational Behavior, 69*(1), 90-104. <https://doi.org/10.1016/j.jvb.2006.03.001>
- Guinan, P. J., Coopridge J. G., & Faraj, S. (1998). Enabling software development team performance during requirements definition: A behavioral versus technical approach. *Information Systems Research, 9*(2), 101-125. <https://doi.org/10.1287/isre.9.2.101>
- Hannebauer, C., & Gruhn, V. (2017). On the relationship between newcomer motivations and contribution barriers in open source projects. *Proceedings of the 13th International Symposium on Open Collaboration, OpenSym 2017* (pp. 1–10). ACM. <https://doi.org/10.1145/3125433.3125446>
- Hatmaker, D. M. (2015). Bringing networks in: A model of organizational socialization in the public sector. *Public Management Review, 17*(8), 1146-1164.
- Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In A. Hevner & S. Chatterjee (Eds.), *Design research in information systems* (pp. 9-22). Springer.
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science research in information systems. *MIS Quarterly, 28*(1), 75-105.
- Hillebregt, J. M. (2019). Creating a cultural onboarding experience. *Policy & Practice, 77*(1), 8-11.
- Hoda, R., Noble, J., & Marshall, S. (2012). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering, 39*(3), 422-444.
- Hoffström, A. (2019). Product-related learning activities and their impact on the effectiveness of the onboarding process in a software development team. *Bachelor of Science Thesis in Software Engineering and Management, University of GTHENBURG, Sweden*.
- Jensen, C., King, S., & Kuechler, V. (2011). Joining free/open source software communities: An analysis of newbies' first interactions on project mailing lists. *Proceedings of the 2011 44th Hawaii International Conference on System Sciences* (pp. 1-10). IEEE. <https://doi.org/10.1109/HICSS.2011.264>
- Jones, G. R. (1986). Socialization tactics, self-efficacy, and newcomers' adjustments to organizations. *Academy of Management Journal, 29*(2), 262-279.
- Judith, A. H. (2001). Building trust and collaboration in a virtual team. *Team Performance Management, 7*(3/4), 36-47. <https://doi.org/10.1108/13527590110395621>

- Kawulich, B. B. (2004). Data analysis techniques in qualitative research. *Journal of Research in Education, 14*(1), 96-113.
- Keele University. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. [EBSE Technical Report. Ver 2.3].
https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf
- Kim, M., Chai, D. S., Kim, S., & Park, S. (2015). New employee orientation: cases of Korean corporations. *Human Resource Development International, 18*(5), 481-498.
<https://doi.org/10.1080/13678868.2015.1079294>
- Kirkpatrick, S. A. (2017). Understanding the role of vision, mission, and values in the HPT Model. *Performance Improvement, 56*(3), 6-14. <https://doi.org/10.1002/pfi.21689>
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology, 51*(1), 7-15. <https://doi.org/10.1016/j.infsof.2008.09.009>
- Klein, H. J., Polin, B., & Leigh Sutton, K. (2015). Specific onboarding practices for the socialization of new employees. *International Journal of Selection and Assessment, 23*(3), 263-283.
<https://doi.org/10.1111/ijsa.12113>
- Korte, R., & Lin, S. (2013). Getting on board: Organizational socialization and the contribution of social capital. *Human Relations, 66*(3), 407-428.
- Kovalenko, V., & Bacchelli, A. (2018). Code review for newcomers: Is it different? *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE '18*. (pp. 29-32). ACM. <https://doi.org/10.1145/3195836.3195842>
- Kowtha, N. R. (2018). Organizational socialization of newcomers: The role of professional socialization. *International Journal of Training & Development, 22*(2), 87-106.
<https://doi.org/10.1111/ijtd.12120>
- Kumar, S., Wallace, C., & Young, M. (2016). Mentoring trajectories in an evolving agile workplace. *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)* (pp. 142-151). IEEE. <https://ieeexplore.ieee.org/document/7883298>
- Lenberg, P., & Feldt, R. (2018). Psychological safety and norm clarity in software engineering teams. *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*. (pp. 79-86). ACM. <https://doi.org/10.1145/3195836.3195847>
- Liu, C., Yang, D., Zhang, X., Ray, B., & Rahman, M. M. (2018). Recommending GitHub projects for developer onboarding. *IEEE Access, 6*, 52082-52094, Article 8458104.
<https://doi.org/10.1109/ACCESS.2018.2869207>
- Liu, J. (2019). *Students' expectations of team onboarding support in Agile software development* [Master's thesis, Auckland University of Technology]. Tuwhera.
<http://hdl.handle.net/10292/13008>
- Lockwood, C., Porritt, K., & Munn, Z. (2015). Qualitative research synthesis: Methodological guidance for systematic reviewers utilizing meta-aggregation. *International Journal of Evidence-Based Healthcare, 13*(3), 179-187. <https://doi.org/10.1097/XEB.0000000000000062>

- Major, D. A., Davis, D. D., Germano, L. M., Fletcher, T. D., Sanchez-Hucles, J., & Mann, J. (2007). Managing human resources in information technology: Best practices of high performing supervisors. *Human Resource Management, 46*(3), 411-427. <https://doi.org/10.1002/hrm.20171>
- March, S. T., & Storey, V. C. (2008). Design science in the information systems discipline: An introduction to the special issue on design science research. *MIS Quarterly, 725-730*.
- Marques, A. B., Rodrigues, R., & Conte, T. (2012). Systematic literature reviews in distributed software development: A tertiary study. *Proceedings of the 2012 IEEE Seventh International Conference on Global Software Engineering* (pp. 134-143). IEEE. <https://doi.org/10.1109/ICGSE.2012.29>
- Matturro, G., Barrella, K., & Benitez, P. (2017). Difficulties of newcomers joining software projects already in execution. *Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 993-998). IEEE. <https://doi.org/10.1109/CSCI.2017.171>
- Miles, M. B., Huberman, A. M., & Saldaña, J. (2018). *Qualitative data analysis: A methods sourcebook*. Sage.
- Milligan, C., Margaryan, A., & Littlejohn, A. (2013). Learning at transition for new and experienced staff. *Journal of Workplace Learning, 25*(4), 217-230. <https://doi.org/10.1108/13665621311316410>
- Minghui, Z., & Mockus, A. (2011). Does the initial environment impact the future of developers [Conference]. 271-280. <https://doi.org/10.1145/1985793.1985831>
- Mishra, A., & Mishra, D. (2006). Software quality assurance models in small and medium organisations: A comparison. *International Journal of Information Technology and Management, 5*(1), 4-20.
- Moe, N. B., Cruzes, D. S., Dybå, T., & Engebretsen, E. (2015). Coaching a global agile virtual team. *Proceedings of the 2015 IEEE 10th International Conference on Global Software Engineering* (pp. 33-37). IEEE. <https://doi.org/10.1109/ICGSE.2015.26>
- Moe, N. B., Stray, V., & Goplen, M. R. (2020). Studying onboarding in distributed software teams: A case study and guidelines. *Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway*. ACM. <https://doi.org/10.1145/3383219.3383235>
- Morgan, P., Sanchez, M., Anglin, L., Rana, R., Butterfield, R., & Everett, C. M. (2020). Emerging practices in onboarding programs for PAs and NPs. *Journal of the American Academy of PAs, 33*(3), 40-46.
- Morrison, E. W. (2002). Information seeking within organizations. *Human Communication Research, 28*(2), 229-242.
- Muethel, M., Siebrat, F., & Hoegl, M. (2012). When do we really need interpersonal trust in globally dispersed new product development teams? *R&D Management, 42*(1), 31-46.
- Niazi, M. (2015). Do systematic literature reviews outperform informal literature reviews in the software engineering domain? An initial case study. *Arabian Journal for Science and Engineering, 40*(3), 845-855.

- Nolte, A., Hayden, L. B., & Herbsleb, J. D. (2020). How to support newcomers in scientific hackathons - An action research study on expert mentoring. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1), Article 25. <https://doi.org/10.1145/3392830>
- Oldham, G. R., & Cummings, A. (1996). Employee creativity: Personal and contextual factors at work. *Academy of Management Journal*, 39(3), 607-634. <https://doi.org/10.2307/256657>
- Panichella, S. (2015). Supporting newcomers in software development projects. *Proceedings of the 2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015* (pp. 586-589). IEEE. <https://doi.org/10.1109/ICSM.2015.7332519>
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77.
- Perrot, S., Bauer, T. N., Abonneau, D., Campoy, E., Erdogan, B., & Liden, R. C. (2014). Organizational socialization tactics and newcomer adjustment: The moderating role of perceived organizational support. *Group & Organization Management*, 39(3), 247-273. <https://doi.org/10.1177/1059601114535469>
- Peters, M. D. (2017). Managing and coding references for systematic reviews and scoping reviews in EndNote. *Medical Reference Services Quarterly*, 36(1), 19-31.
- Pham, R. (2014). Improving the software testing skills of novices during onboarding through social transparency. *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering* (pp. 803-806). ACM. <https://doi.org/10.1145/2635868.2666604>
- Pham, R., Kiesling, S., Singer, L., & Schneider, K. (2017). Onboarding inexperienced developers: struggles and perceptions regarding automated testing. *Software Quality Journal*, 25(4), 1239-1268. <https://doi.org/10.1007/s11219-016-9333-7>
- Pham, R., Mörschbach, J., & Schneider, K. (2015). Communicating software testing culture through visualizing testing activity. *7th International Workshop on Social Software Engineering, SSE 2015 - Proceedings* (pp. 1-8). ACM. <https://doi.org/10.1145/2804381.2804382>
- Pinto, G., Steinmacher, I., Dias, L. F., & Gerosa, M. (2018). On the challenges of open-sourcing proprietary software projects. *Empirical Software Engineering*, 23(6), 3221.
- Pope, C., Ziebland, S., & Mays, N. (2000). Qualitative research in health care. Analysing qualitative data. *BMJ (Clinical Research ed.)*, 320(7227), 114-116. <https://doi.org/10.1136/bmj.320.7227.114>
- Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for design science research evaluation. *European Conference on Information Systems. ECIS 2008 Proceedings*. 87. <https://aisel.aisnet.org/ecis2008/87>
- Rabel, K., & Stefaniak, J. (2018). The onboarding of instructional designers in the workplace. *Performance Improvement*, 57(9), 48-60.
- Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: An empirical study. *Information Systems Journal*, 20(5), 449-480.
- Ross, W. E., Huang, K. H., & Jones, G. H. (2014). Executive onboarding: Ensuring the success of the newly hired department chair. *Academic Medicine*, 89(5), 728-733.

- Saks, A. M., & Gruman, J. A. (2011). Organizational socialization and positive organizational behaviour: Implications for theory, research, and practice. *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, 28(1), 14-26.
- Sawyer, S. (2001). Effects of intra-group conflict on packaged software development team performance. *Information Systems Journal*, 11(2), 155. <https://doi.org/10.1046/j.1365-2575.2001.00100.x>
- Schipper, R., & Silvius, G. (2017). The sustainable project management canvas. *Journal of Modern Project Management*, 50-59. <https://doi.org/10.19225/JMPM01206>
- Sharma, G. G., & Stol, K. J. (2020). Exploring onboarding success, organizational fit, and turnover intention of software professionals. *Journal of Systems and Software*, 159, Article 110442. <https://doi.org/10.1016/j.jss.2019.110442>
- Siddaway, A. P., Wood, A. M., & Hedges, L. V. (2019). How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses. *Annual Review of Psychology*, 70, 747-770.
- Silva, J. O., Wiese, I., German, D. M., Treude, C., Gerosa, M. A., & Steinmacher, I. (2020). Google summer of code: Student motivations and contributions. *Journal of Systems and Software*, 162, Article 110487. <https://doi.org/10.1016/j.jss.2019.110487>
- Singh, J. P., Chand, P. K., Mittal, A., & Aggarwal, A. (2020). High-performance work system and organizational citizenship behaviour at the shop floor. *Benchmarking*, 27(4), 1369.
- Smite, D., Kuhrmann, M., & Keil, P. (2014). Virtual teams [Guest editors' introduction]. *IEEE Software*, 31(6), 41.
- Smith, J., & Firth, J. (2011). Qualitative data analysis: The framework approach. *Nurse Researcher*, 18(2), 52-62.
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333-339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Stein, M., & Christiansen, L. (2010). *Successful onboarding*. McGraw-Hill Professional Publishing.
- Steinmacher, I., Chaves, A. P., Conte, T. U., & Gerosa, M. A. (2014). Preliminary empirical identification of barriers faced by newcomers to open source software projects. Proceedings of the 28th Brazilian Symposium on Software Engineering, SBES 2014 (pp. 51-60). IEEE. <https://doi.org/10.1109/SBES.2014.9>
- Steinmacher, I., Conte, T. U., Gerosa, M. A., & Redmiles, D. F. (2015). Social barriers faced by newcomers placing their first contribution in open source software projects. *CSCW 2015 - Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing* (pp. 1379-1392). ACM. <https://doi.org/10.1145/2675133.2675215>
- Steinmacher, I., Conte, T. U., Treude, C., & Gerosa, M. A. (2016). Overcoming open source project entry barriers with a portal for newcomers. *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)* (pp. 273-284). IEEE. <https://doi.org/10.1145/2884781.2884806>

- Steinmacher, I., Gerosa, M., Conte, T. U., & Redmiles, D. F. (2019). Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work: CSCW*, 28(1-2), 247-290. <https://doi.org/10.1007/s10606-018-9335-z>
- Steinmacher, I., Wiese, I., Conte, T. U., & Gerosa, M. A. (2015). Increasing the self-efficacy of newcomers to open source software projects. *Proceedings of the 29th Brazilian Symposium on Software Engineering, SBES 2015* (pp. 160–169). ACM. <https://doi.org/10.1109/SBES.2015.10>
- Steinmacher, I., Wiese, I. S., Conte, T., Gerosa, M. A., & Redmiles, D. (2014). The hard life of open source software project newcomers. *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, Hyderabad, India*. <https://doi.org/10.1145/2593702.2593704>
- Steinmacher, I., Wiese, I. S., & Gerosa, M. A. (2012). Recommending mentors to software project newcomers. *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)* (pp. 63–67). ACM.
- The Society For Human Resource Management. (2020). *Developing and sustaining high-performance work teams. SHRM*. <https://www.shrm.org/resourcesandtools/tools-and-samples/toolkits/pages/developingandsustaininghigh-performanceworkteams.aspx>
- Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. *BMC Medical Research Methodology*, 8(1), 1-10.
- Thompson, J. A. (2005). Proactive personality and job performance: A social capital perspective. *Journal of Applied Psychology*, 90(5), 1011-1017. <https://doi.org/10.1037/0021-9010.90.5.1011>
- Van Maanen, J. (1978). People processing: Strategies of organizational socialization. *Organizational Dynamics*, 7(1), 19-36.
- Van Maanen, J. E., & Schein, E. H. (1977). Toward a theory of organizational socialization. *Massachusetts Institute of Technology (MIT)*. <https://dspace.mit.edu/bitstream/handle/1721.1/1934/?sequence=1>
- Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2012, 14-15 May 2012). *Systematic literature reviews in global software development: a tertiary study* 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), Ciudad Real, Spain
- Viana, D., Conte, T., & De Souza, C. (2014). Knowledge transfer between senior and novice software engineers: A qualitative analysis. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE* (pp. 235-240). KSI Research. https://ksiresearch.org/seke/seke14paper/seke14paper_160.pdf
- Viviani, G., & Murphy, G. C. (2019). Reflections on onboarding practices in mid-sized companies. *Proceedings of the 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2019* (pp. 83-84). IEEE. <https://doi.org/10.1109/CHASE.2019.00027>
- Wanberg, C. R. (2012). *The Oxford handbook of organizational socialization*. Oxford University Press.
- Wang, J. (2012). Supporting developer-onboarding with enhanced resource finding and visual exploration. *CSE Technical Reports*.146. <http://digitalcommons.unl.edu/csetechreports/146>

- Williams, M., & Moser, T. (2019). The art of coding and thematic exploration in qualitative research. *International Management Review*, 15(1), 45-55.
<https://ezproxy.aut.ac.nz/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=bth&AN=135847332>
- Yang, J. (2017). *Onboarding process in Agile software development team: An empirical study* [Master's thesis, Auckland University of Technology]. Tuwhera. <http://hdl.handle.net/10292/11450>
- Yates, R., Power, N., & Buckley, J. (2020). Characterizing the transfer of program comprehension in onboarding: an information-push perspective. *Empirical Software Engineering*, 25(1), 940-995.
<https://doi.org/10.1007/s10664-019-09741-6>
- Yates, R. Y. (2014). *Onboarding in software engineering* [Doctoral thesis, University of Limerick]. University of Limerick Institutional Repository. <http://hdl.handle.net/10344/4272>
- Yu, X., & Petter, S. (2014). Understanding agile software development practices using shared mental models theory. *Information and Software Technology*, 56(8), 911-921.

APPENDICES

Appendix A - Extracted data from systematic literature review

Author(s) & Year	Purpose of articles	Key Findings	Discussed Onboarding goals/ Expected onboarding outcomes	On-boarders' challenges/barriers are discussed	Onboarding activities/practices/strategies are discussed to address mentioned on-boarders' challenges	Mentioned evidences from implemented onboarding activities or on-boarders said that they are improved	Monitor/Evaluate Feedback the onboarding newcomers (metrics)	On-boarders' challenges but not be addressed by onboarding practices/strategies
(Balali et al., 2018)	To investigate the barriers that affect mentors and newcomer as mentees in Open-source software projects	44 barriers: 19 that affect mentors; and 34 that affect newcomers (9 affect both newcomers and mentors)	<ul style="list-style-type: none"> - Completion of assigned tasks without giving up (Self-efficacy) - Personality adaptation of newcomers to team and project environments. - Clear professional goals and responsibility of newcomers - Communication process among mentors and mentees - Constant motivation of newcomers - Contribution of newcomers to the team - Time management 	<p>34 newcomers' barriers as follow:</p> <p>Personal barriers:</p> <ul style="list-style-type: none"> - Difficult in managing different accounts - Issue related to self-efficacy: <ol style="list-style-type: none"> 1) Fear of judgment, 2) low self-efficacy, 3) performance anxiety - Newcomer personality issues: 1) shyness to ask question, 2) Newcomers' personality conflicts with the role, 3) Newcomers' inability to improve upon criticism - Lack of management's skills: <ol style="list-style-type: none"> 1) Difficult in time management - Issues related to newcomer's behavior: 1) Lack of interest, 2) Lack of 	<ul style="list-style-type: none"> - Working on a bug or issue together with mentee could overcome "high code complexity" and "lack of newcomer's background knowledge" barriers. - Holding training sessions for newcomers could overcome "lack of newcomer's background knowledge" and "difficulty in learning related tools or technologies" barriers - Flagging newcomers so others are welcoming to them could overcome "low response rate" 	<p>P10 mentioned that it helps people understand how the project works, understand the type of problems, mentees can later work on their own.</p> <p>P7 mentioned that this strategy could help newcomers to overcome technical barriers</p> <p>P9 described that holding training sessions helps newcomers to overcome most technical barriers.</p> <p>P6 stated that flagging newcomers helps others to recognize them, be more patient,</p>		<ul style="list-style-type: none"> - Difficult in managing different accounts - Shyness to ask question - Newcomers' personality conflicts with the role - Newcomers' inability to improve upon criticism - Difficult in time management - Lack of clear professional goals - Lack of proactiveness - Differences in work experience and age - Lack of mentor's interpersonal skills - Lack of newcomers' interpersonal skills - Difficulty in finding help in the community - Difficulty in choosing a

				<p>clear professional goals, 3) Lack of pro-activeness.</p> <p>Interpersonal barriers:</p> <ul style="list-style-type: none"> - Issue related to diversity: 1) Cultural differences, 2) Differences in work experience and age - Challenges in communicating effectively: 1) Communication issues related to time zone and place, 2) Lack of English language skills, 3) Lack of mentor’s interpersonal skills, 4) Lack of newcomers’ interpersonal skills. - Issues with community reception: 1) Harsh project atmosphere, 2) Low response rate, 3) Difficulty in finding help in the community <p>Process barriers:</p> <ul style="list-style-type: none"> - Issue with newcomers’ orientation: 1) Difficult to identifying appropriate tasks for 	<p>and “harsh project atmosphere” barriers</p>	<p>welcoming, and responsive.</p>		<p>newcomers-friendly project</p>
					<ul style="list-style-type: none"> - Communication through different means could overcome “cultural differences”, “Communication issues related to time zone and place”, and “Lack of English language skills” barriers 	<p>P4 informed that communication through emails or IRC helps him to overcome language and time zone barriers.</p>		<p>Some specific barriers for women newcomers:</p> <ul style="list-style-type: none"> - women seemed to have lower self-efficacy - women contributors feel less comfortable with and accepted by their counterparts who are men - Differences in motivation - style of communication
					<ul style="list-style-type: none"> - Giving the newcomers small/interesting tasks could overcome “lack of interest” and “Performance anxiety” 	<p>P9 said that if high expected contributing works assigned to newcomers, that was not work. P3 stated that in order to keep newcomers engaging, the community need to pick an appropriate task for them.</p>		<p>Stakeholder Impact</p> <p>The study pointed out that these barriers are interesting for future research in term of influence to stakeholders of the projects.</p>
					<ul style="list-style-type: none"> - Giving newcomers rewards to keep them motivated could overcome “lack of interest” barriers 	<p>P2 mentioned that giving rewards to newcomers while getting through their guides could keep them be motivated.</p>		
					<ul style="list-style-type: none"> - Having newcomers share their work to have more exposure could overcome “Fear of judgment”, “low self-efficacy”, 	<p>P9 stated that encouragement of newcomers to write blog post helps other people knows on what they are doing.</p>		

				<p>newcomers, 2) Difficulty in choosing a newcomers-friendly project, 3) Willingness to start with a complex task.</p> <ul style="list-style-type: none"> - Issues with project schedules: 1) Issues with project microclimate, 2) Long project processes - Lack of background about project procedures: 1) Problem with the process of submitting code, 2) Lack of knowledge about procedures and conventions, 3) Lack of documentation <p>Technical barriers:</p> <ul style="list-style-type: none"> - Differences in the devices that mentors and mentees use - High code complexity - Difficulty in getting started: 1) Difficulty in setting up development environment, 2) Task too complex for newcomers - Lack of newcomers' knowledge about 	<p>and “performance anxiety” barriers</p>			
					<ul style="list-style-type: none"> - Tagging the tasks according to their complexity could overcome “difficulty to identifying appropriate tasks for newcomers” barriers 	<p>P5 said that things had gotten much easier from he started as no documentation, guidelines, and mentors tagged unsuitable bugs to newcomers. Things have changed and become easier. This strategy used by many big project Apache, Mozilla, Gnome, and KDE, P10 mentioned that some large projects consume a lot time to identify the appropriate tasks for newcomers.</p>		
					<ul style="list-style-type: none"> - Having local groups in each country could overcome “cultural differences” and “lack of English language skills” barriers 	<p>P9 explained that starting alone is harder than having a local group. Lots of local groups in China, India, US, and Korea, having those group is significant to welcome newcomers to free software communities, talking in your mother language, similar culture.</p>		

				<p>programming and tools: 1) Lack of newcomers' background knowledge, 2) Difficulty in learning related tools or technologies ----- Some specific barriers for women newcomers:</p> <ul style="list-style-type: none"> - Women seemed to have lower self-efficacy - Women contributors feel less comfortable with and accepted by their counterparts who are men - Differences in motivation - Style of communication 	<ul style="list-style-type: none"> - Keeping documentation concise and updated could overcome "cultural differences", "harsh project atmosphere", "Long project processes", "Willingness to start with a complex task", "Issues with project micro-climate", "Difficulty in choosing a newcomer-friendly project", "Lack of knowledge about procedures and conventions", "Lack of documentation", "Problem with the process of submitting code", "Difficulty in identifying appropriate tasks for newcomers", and all technical barriers. 	<p>Interviewees indicated that this strategy helps newcomers overcome a bunch of barriers. P5 mentioned that newcomers' guidelines make things easier and help people get along, do not need to teach rules, they are there for newcomers. In addition, P8 stated that there shouldn't be too many since we document everything</p>		
--	--	--	--	--	--	---	--	--

(Britto et al., 2018)	To investigate the onboarding of software developers/teams, associated challenges, and areas for further improvement in 3 globally distributed legacy projects by doing multi-case study (3 cases studies).	<ul style="list-style-type: none"> - Distributed team in projects and legacy products are significant challenges for onboarding of software developers, especially the considerable amount of legacy code that newcomers need to learn due to difficulty to connect newcomers to original developers - Some onboarding functions/strategies were not applied among different sites; some were centrally implemented while some were only in local. - Studies suggested that legacy projects or co-located ones needs hands-on training and longer mentoring period than new projects. 	<ul style="list-style-type: none"> - Socialization and performance expectation of newcomers - Building cooperate behaviors of newcomers - Building interaction between newcomers and senior developer - Gaining product and legacy code knowledge - Understanding in the work environments and company structure 	<ul style="list-style-type: none"> - Distribution of team in a project. - Learning a huge amount of legacy product codes by newcomers. 	<p>Recruitment:</p> <ul style="list-style-type: none"> - Recruitment integrated with onboarding - Realistic job previews for newcomers - Stakeholder involvement in recruitment 	<p>Case 1 proved that a successful attendee who joined the summer training sessions with actual software developer teams, received a job offers after the recruitment was conducted with the involvement of senior developers to interview/evaluate the candidate's technical skills. Case 2 and Case 3 also gained benefits from this strategy. Overall, recruitment is partially integrated with the onboarding process.</p>	<p>Feedback during the face-to-face meeting and code review from mentors is mainly used to evaluate whether newcomer need more supports</p>
					<p>Orientation:</p> <ul style="list-style-type: none"> - First day of the job is special - Using intranet portal to centrally share all documents 	<ul style="list-style-type: none"> - No formal orientation. However, a bootcamp 3-month long program is conducted as a session for training, coaching and orientation for case 1. In the actual onboarding, newcomers dedicate the whole first week to familiarize with the environment in Case 1 & 3. 	

						Case 2, newcomers dedicated the whole first month to learn on the job		
						Coaching and support: - Assign real tasks to newcomers under careful support by senior developer - Mentoring programs	All 3 cases applied mentors to newcomers - Case 1: assigned a mentor to new developer inside the bootcamp. - Case 2: One or several mentors assigned to new developer in the team. Case 3: A mentor assigned to a new developer or a group of new developer.	
						Training: - Formal training on hard skill and/or soft skills	Case 1: provided 3-month program in bootcamp, focusing on technical and methodological knowledge. Case 3: If many newcomers, then provide long training by focusing on technical, methodological, product knowledge.	
						Support tools and processes: - Onboarding plans - Regular stakeholder meetings - Own progress monitoring	All cases provide onboarding plan, use an intranet with useful company materials,	

						face to face, video conference for meeting. Case 3 used spreadsheet for monitoring the progresses		
					Feedback: - Perform appraisals - 360-degree feedback	- All cases evaluated the performance during face-to-face meeting with mentors and immediate managers. - Obtaining feedback from mentors from doing code reviews.		
(Britto et al., 2020)	To investigate the relationship between onboarding strategies and the performance evolution of newcomers in large-scale, globally distributed projects	Some factors to reduce the performance of onboarding outcomes are found (distance to mentors, formal training is not fit to social cultural background, too many large and distributed tasks assigned to on-boarders in early stages, and instable team) - Onboarding in globally distributed projects which involved a huge amount of legacy codes needs to be well planned in advance and be ready	Based on the previous study in 2018 (Britto et al., 2018)	- The distance to mentors - Social cultural background (formal training method is not fit) - Allocation of large/distributed tasks in early stages of the onboarding process. - Difficulty to learn the legacy code - Instability of team	- Integrate recruitment process (technical interview) - Orientation (formal training if many newcomers, otherwise 1-week informal training is conducted for making familiarization to new environment, key persons/coworkers) - In the formal training, senior developers are virtually assigned as mentors - Coaching and support. - Use Excel spreadsheet to track		The productivity, autonomy goals are set by senior developers. - Newcomers' performance monitoring is mainly focus on productivity of customization tasks, and autonomous learning during the onboarding periods. - Both results are less than 50% compare to desired goals.	The study does not explicitly state the main

		to expand the mentoring period.			entire progression of newcomers. (Britto et al., 2018)			
(Buchan et al., 2019a)	To investigate how Agile software development team addresses the challenges of team onboarding in order to suggest future onboarding practitioners.	<ul style="list-style-type: none"> - 11 onboarding goals, with 25 practices are discovered. - High contribution onboarding techniques are mapped to desired goals. 	<ul style="list-style-type: none"> - Understand the company culture - Understand the team norm - Understand and meet others' expectation on your own responsibilities. - Understand the responsibilities, expertise, and authority of other team members - Understand tasks to do and timeline. - Understand how to code and test as expectation level of team members - Understand the standard of work quality of teams. - Understand and adopt Agile mindset. - Know how to use Agile artefacts and techniques used by the rest of the team - Understand structures of teamwork (short, medium, long), aims and implications. - Understand the product/project 	<ul style="list-style-type: none"> - Company culture - Team norm - Meet other's expectation on newcomers' responsibility - Responsibility, expertise, and authority of other team members - Be aware what to do and its timeline - Code and test as expectation level of team members. - Work standard of team quality - Agile mindset adaptation - Using Agile artefacts and techniques - Structure of teamwork (short, medium, long), aims and implications - Products/projects knowledge 	<ul style="list-style-type: none"> - Mentoring - Online communication - Peer support - Team socializing - Training course- - Code review - Internal documentation - Product overview - Pair programming - Regular team standup meeting - Simple task - Self-learning - Induction - Access and contribute to a local knowledge DB (Wiki pages) - Team leader supports - Course on Agile - Team retrospective - Review plan - Attend conference - Set expectation - Electronic communication - Meet with other teams - Location map - Checklists 	All techniques are reported from interviewees and then categorized by the researcher of this study	All discussed challenges are associated to desired onboarding goals and are mapped to the mentioned techniques in the study.	

			domain knowledge and terminology.					
(Cunha et al., 2020)	To investigate the difficulties of newcomers using a tool (Wiki) in global software development specially knowledge transfer to project leaders as the Wiki's content has a large amount of information, lack of information process, and structural knowledge sharing concern.	Study revealed that the difficulty of Wiki usage, roughly 90% of newcomers encountered searching difficulty. 1) Search information about work process, 2) difficult to define what to search, 3) difficult to search team responsibility.	- Organizational structure - Company culture	- Globally distributed collaborating teams caused a strong difficulty for knowledge transfers. - Working process, - Organization structure, - Company cooperate culture - Supporting tool like Wiki page is still required for improvement.	- No clear structure of employed onboarding process. However, the research describes that online tool like Wiki page is used to sharing information for newcomers to study while onboarding session is implementing.		As this research evaluated the use of Wiki page from newly project leaders. Some feedbacks are - Difficulty of using the Wiki page like usability problem. - The desire of newcomers such as 1) learning from experienced project leaders, 2) learning from other resources like tutorials, training.	
(Davidson et al., 2014)	To investigate the lack of diversity in Free/Open source Software (FOSS) communities.	- Motivations of older contributions, - Benefits and challenges to contribution, - Older adults' view on discrimination in FOSS, - Ways in which older adults enrich FOSS communities	- Efficient communication although diversity of team members in term of different ages.	- Communication issues - Discrimination concerns	According to the findings, the research suggested three main focus for onboarding older adults into the FOSS - Focus on social aspects: educate newcomers to understand how to communicate effectively in order to building community, friends. - Match contribution efforts to individual motivations: what	N/A	N/A	N/A

					<p>participants are contributing to the team should be well matched to what he/she is motivated.</p> <p>- Don't ignore the bad stuff (Discrimination)</p>			
(Deshpan de & Gupta, 2019)	To investigate factors which play important role in retention of newly hired employees to work in technological industry of India.	<ul style="list-style-type: none"> - Supportive behavior of supervisor encourages newcomers to seeking feedback and improve their commitment toward the organization, - Supervisors are a significant source of information for newcomers 	<ul style="list-style-type: none"> - Building feedback seeking behaviors to new hires with the supports from supervisors. 	<ul style="list-style-type: none"> - Socialization and culture norm of organization - Clarity of the expected tasks that required newcomers to perform. - Newcomers intention to quit working with organization. 	<p>Suggestion:</p> <ul style="list-style-type: none"> - Advocacy of supervisors could encourage newcomers to find feedback and improve their commitment. 	The SEM path analysis results reveal that supervisor's support behavior significantly influence newcomers whether or not to contribute and keep continue working with company or to quit.	N/A	N/A
(Diniz et al., 2017)	To investigate the use of games to advocate and encourage new graduates to overcome onboarding barriers and contribute to OSS projects.	The study reveals that gaming environments not only can be used as an onboarding strategy for orientation, but also to motivate their engagement in order to contribute efforts into the OSS projects	<ul style="list-style-type: none"> - Encourage contribution from newcomers into the project. 	<ul style="list-style-type: none"> - Motivate newcomers into an academic environment - Codebase issues 	<p>Suggestion:</p> <ul style="list-style-type: none"> - Gamification - Giving a reward can encourage constant motivation of students based on points and/or ranking from the gaming results. 	<ul style="list-style-type: none"> - Quest elements from games are assignments/tasks which have a clear instruction, well-defined goals, and they make students focus based on the instruction. - Points elements from result of games keep students motivated and can providing feedback from their performance 	N/A	N/A
(Dominic et al., 2020)	To propose a tool (conversational bot), in order to address	<ul style="list-style-type: none"> - The study comes up with the conversational bot 	<ul style="list-style-type: none"> - Improve newcomers experience in order 	<ul style="list-style-type: none"> - Low retention rate of newcomers in OSS projects 	<ul style="list-style-type: none"> - Using an ideal conversational tool 	N/A	N/A	N/A

	<p>facing problems of an onboarding process, reduce the low retention rate of newcomers, suggests projects to newcomers on their onboarding program in Open-source communities.</p>	<p>tool with a set of steps for its development</p> <ul style="list-style-type: none"> - The tool workflow how newcomers interact with the tool when joining the organization. 	<p>to support their first contribution into the project.</p>	<ul style="list-style-type: none"> - Low contribution effort to projects before moving to others company - Premature leaving the project (leaving project quickly) 	<p>called “Bot” to help newcomers.</p> <p>Starting from 1) collection of newcomer’s experience/interests/p referred languages by Bot, 2) Bot finds and advice projects, 3) Newcomers select projects, 4) Bot finds and summarizes issues then provide information to newcomers, 5) Newcomers select issues, 6) Bot process to pulls Stack Overflow data which relates to issues, then provides to newcomers, 7) Bot recommends project contributor to assist newcomers.</p> <p>- Mentorships.</p>			
<p>(Fagerholm et al., 2014)</p>	<p>To examine how mentoring and project characteristic influence the effectiveness and efficiency of the onboarding process in order to increase the performance of open-source software development projects.</p>	<ul style="list-style-type: none"> - New developers who receive supports from mentors are more active at the early stage than those who just follow through the available means. - Size and lifetime of projects influences the outcome of onboarding programs. 	<ul style="list-style-type: none"> - Proactive participation of newcomers - Increase the degree of collaborative activities between developers - Increase a number of commits into the project. 	<ul style="list-style-type: none"> - Need a significant investment of time and effort from mentors 	<ul style="list-style-type: none"> - Mentor: recommending, helping to clarify the tasks, explain the overall software architecture and technical development details. Helping newcomers to focus on specific activities 		<ul style="list-style-type: none"> - Onboarding support could be implemented through communication channels that are not publicly visible. - Mentoring may need private space for mentors and newcomers in order to avoid interferences from the rest of the 	

					- Establishing a relationship with developers		project, while onboarding managers must trust mentors to perform mentorship	
(Fagerholm et al., 2013)	To present the preliminary observations and results of in-progress research that studies the process of onboarding into virtual OSS teams.	Mentoring is the key factor for productive onboarding in OSS projects, influencing the cohesion among virtual teams and maintaining their appropriate working pace	- Contribution of newcomers to the projects		<p>- Kickoff session (getting to know virtual team members, meet mentors). During the sessions, mentors and students work together in a single location. The mentors assigned hands-on, practical training to developers</p> <p>- Continuous mentoring to support newcomers by mentors such as: participating in online forum and mailing list discussions, joining online video conferences, helping newcomer to understand tasks, reviewing code contribution, and providing feedback, and helping them coordinate through issue tracking systems.</p> <p>- Developers were free to work on any relevant task to the</p>			

					<p>projects. Mentors assigned small/greater tasks which were suitable to the developers based on their level.</p> <ul style="list-style-type: none"> - Developers integrated into the projects and community and follow a regular procedure of that team. Then they applied their experience into the team along with obtaining support with their local and remote team members. 			
(Hannebauer & Gruhn, 2017)	To find the relation between motivation and contribution barriers of newcomers.	<p>The results confirm that the newcomer's motivations reflect their mental models of the OSS project which they will contribute to</p> <p>And the mental model causes the impact of contribution barriers</p>	- Contribution of newcomers to the OSS projects	<p>Contribution barriers distinguished into two main different types:</p> <ol style="list-style-type: none"> 1. Modification barriers: <ul style="list-style-type: none"> - Find the code (difficulty on locating the right place to make specific change) - Difficulty on setting up development environment 	N/A	N/A	N/A	N/A

				<ul style="list-style-type: none"> - Bug reproduction - Difficulty on downloading the right version of source code. <p>2. Submission barriers</p> <ul style="list-style-type: none"> - Submission procedure - Documentation “Instruction on home page” - Issue tracker 				
(Hoffström, 2019)	To investigate the current onboarding process in Ericsson in order to find gaps for improvement	Two main areas are found for improvement <ul style="list-style-type: none"> - Technical knowledge - Organizational knowledge 	- Understanding the products of the organization	<p>To understand products: some barriers:</p> <ul style="list-style-type: none"> - Comprehension of the codes - Development process - Know who to contact for what issues - Which approach is normally applied to deal with a specific issue. 	<ul style="list-style-type: none"> - No existing formulated onboarding procedure in the team, - Newcomers just attended a presentation session about company goals, products. <p>Suggestion from the study:</p> <ul style="list-style-type: none"> - Team building learning activities could be applied to improve the product-related learning by organizing workshop into two sessions 	N/A	N/A	N/A

					<p>1) Using a tool to help newcomers to learn about products, projects, tools which are developed by other team within the organization</p> <p>2) Organizational knowledge: technical skills and team knowledge in the organization.</p>			
(Kovalenko & Bacchelli, 2018)	To investigate how low experienced developers' contribution in a project receive a different support while doing code review. By doing the comparison of reviewer's experience, metrics of reviewers' attention, and change merge rate between changes from newcomers and from more experienced authors in 60 active open source projects	- No significant difference of experienced reviewers' method in projects. - The changes from newcomers are less reviewed by experienced developers, while the changes from not new developers are more reviewed.	- Gaining technical knowledge in the team - Contribution of newcomers	- Lack of familiarity with the codebase - High bug density in codes of newcomers - Learning curve	- Conducting code review by using the Gerrit tools for learning technical knowledge within team, and review each other's codes	N/A	N/A	N/A
(Kumar et al., 2016)	To report the evolving nature of onboarding and mentoring at a mature software development company which applied a Scrum method for a decade.	The community of practice is a significant tool to push the essence of the onboarding and mentoring process. A mix of participant and observer accounts can help	- Understanding new tools and systems of the company	- Feel lost, stressed, anxiety, and overwhelmed during the first few day of participation the onboarding program in the company.	- Mentoring by experienced developers - pair programming in the first few days of joining the company	One of the participants described that personal sitting and working with different members of team was extremely helpful where she learned a lot about	N/A	Mentoring practices: Mentoring styles are particularly suited for certain personalities, or specific stages of the onboarding process, but what

	To follow onboarding processes in every stage of the company through the perspective of employees, and their development experience from novice to mentor.	determine the forming of reification and participation in order to be integrating newcomers into the community.			- Putting newcomers into the real programming work in the team	code base as they shared different perspectives based on their expertise.		degree can an onboarding session requires mentors?
(Liu et al., 2018)	To propose a model called neural network for list-wise ranking (NNLRank) “” for project selection of developers before making a decision to join an open-source software development project.	We evaluate NNLRank with 2044 successful onboarding decisions from GitHub and compare it with three standard learning-to-rank models and a prior onboarding tool. Experimental results show that NNLRank can provide effective and efficient onboarding recommendation to developers, substantially outperforming the previous models.	- Effective and efficient onboarding advocacy to developers to choose the right project for participation.	- Social and technical barriers - Developers’ costly effort in contributing numerous commits - Contributions remain unsatisfactory by the core project members (newcomers need to rewrite, review, test new features over again)	N/A	N/A	Recommending projects for onboarding of developers is still challenging due to the complex interactions among social and technical aspects influence the onboarding process.	N/A
(Liu, 2019)	To gain understanding on student’s expectation of onboarding in Agile software development team (such as onboarding activities, resources, and duration) based	Three main factors influence the onboarding outcomes 1) New employees’ characteristics 2) New employees’ behaviors, 3) organizational efforts	Some onboarding goals refer to the research by Yang, 2017. Apart from that there are some onboarding expected goals by students in this research as below:	- Mentors’ characteristics may be not met the expectation of new graduates so that it may delay the integration progress into the team.	Expected onboarding activities: - Getting help from team member - Getting help from training/workshop	N/A	N/A	N/A

	<p>on the perception of students.</p> <p>To discover differences and similarities expectations of onboarding process between students and practitioners</p>		<ul style="list-style-type: none"> - Be guided and led to familiarize the new work environment - Be guided to work in the right direction - Be fully given help - Be faster to find solution to the problems - Gain a wide understanding of professional knowledge - Slowly catch up with pace of the team - Create a long-term relationship with team members - Feel free to question team member when needs help - Improve team productivity - Build up a social connection - More opportunities to communicate with team members 		<ul style="list-style-type: none"> - Support from mentor - Access to all documentation from previous projects - Introduction/ Explanation about organizational profile - Introduction to/about team members and leaders - Role clarity and prefer essential skills relate to their position. - Help from supervisors - Help from daily stand-up meeting - Desired working environment Other preference which has expected proportion less than 10% from participants of the study. 			
--	---	--	--	--	---	--	--	--

			<ul style="list-style-type: none"> - Expect more advocacy and sharing personal experience from team members - Be familiar with project and understand company culture - Learn software and essential skills associated to the work 					
(Maturro et al., 2017)	To report the results of a research aiming to identify newcomers' difficulties when start joining an in-progress executing software development projects. Then classifying actions which are applied to mitigate those concerns.	<p>The lack of documentation, and the need to know the in-progress development product are reported as main difficulties for newcomers.</p> <p>To mitigate mentioned problems, assignment of an experienced team member to coach newcomers, provide training are mainly applied by the organization</p>	<ul style="list-style-type: none"> - Product domain/knowledge - Understanding team working environment - Understanding company culture - Understanding project solutions - Desired tool and technology skills as requires using in the project -Effective communication 	<p>Difficulties mentioned by team members and team leaders</p> <ul style="list-style-type: none"> - Understanding of solution/knowing the product - Work methodology - Knowing the company - Integration to the team - Understanding the project - Documentation - Tool and technology in use - Communication 	<ul style="list-style-type: none"> - Assignment of an experienced team member to guide, supervise, - Monitor/follow up the assigned tasks of newcomers - Provide training and documentation of in-progress construction projects to newcomers - Give him "freedom" - Define an integration plan - Pair programming 	N/A	N/A	N/A

(Moe et al., 2020)	To investigate how a bank implemented onboarding developers for distributed team in global project, and report experiences of different onboarding practices, techniques.		<ul style="list-style-type: none"> - Social integration - Understanding company culture - Learn the expectation of new job quickly and smoothly - Effective communication among team members - Understand what to do and when 	<p>Four main mentioned challenges:</p> <ol style="list-style-type: none"> 1) Missing domain knowledge 2) Communication tools 3) Unclear tasks 4) Language barriers <p>Other barriers as follow:</p> <ul style="list-style-type: none"> - Cultural differences - Different time zones - Different values and norms - Lack of trust - Learning curve 	<ul style="list-style-type: none"> - The concepts of Bauer's onboarding model are applied into this distributed globally team research (Six activities affect the onboarding as follow <ol style="list-style-type: none"> 1) Recruitment 2) Orientation, 3) Training, 4) Coaching and support, 5) Support tools and processes, 6) Feedback, <p>As well as the four adjustments:</p> <ol style="list-style-type: none"> 1) Self-efficacy, 2) Role clarity, 3) Social integration, 4) knowledge of the culture 	<ul style="list-style-type: none"> - One of the banks (Norbank from the research) covered all activities of Bauer indicates that the Bauer's onboarding model is suited for globally distributed teams. <p>It shown that Norbank conducted both face to face and slack interviews as a recruitment process</p> <p>Developers participated in various course and social activities for 3 weeks. The results are positive as they learnt a lot of aspects and face to face meet each other.</p>	N/A	Mentoring is benefit to the newcomers, but sometime mentors feel frustrated as they lost some working time to perform their own tasks.
(Nolte et al., 2020)	Main aim of the research is to support scientific communities to conduct hackathons which grow interest in their community, allow newcomers to developer a useful technical artefact for their community.	- The study revealed that different approaches of each mentor might have contributed to perceived differences in accomplished learning between participating teams. - It is important for mentors to	N/A	<ul style="list-style-type: none"> - How mentors approach their roles to support newcomers? - Different approaches of mentors associated to the perceptions of newcomers/participants 	<p>Some main approaches how mentors support teams</p> <ul style="list-style-type: none"> - Initial goal setting: define specific goals and direction based on the background and purpose of the project 	N/A	N/A	N/A

	To investigate gaps of mentoring when team members were assigned as a mentor to perform their roles in order to support newcomers/participants in the hackathon events. How different approaches related to the perception of mentors/participants	understand newcomers' capabilities and assist them to plan tasks which fit to their abilities rather than let them choose what to do and learn new technical skills.			<ul style="list-style-type: none"> - Goal assessment and adaptation: monitor the progress and adjust accordingly to avoid the duplication of issues, as well as speed up the progress to reach original goals. - Technical support: all participants require support related to technologies they use. - Mentoring focus: provide ideas to help participants develop as defined plan as well as helping on particular technical issues 			
(Panichella, 2015)	<p>To investigate problems arising when newcomers join software project.</p> <p>Study how newcomers interact with mentors/other developers during the onboarding, then develop a viable tool to support them.</p>	<ul style="list-style-type: none"> - Collaboration /communication of analyzing developers through a specific channel could provide only a partial view of reality. They should rely on more than one communication channel. - Issue trackers and mailing lists are suitable sources to visual the key project 	<ul style="list-style-type: none"> - Understanding technical and organizational information - Gaining familiarity with source code and related documents in order to perform first maintenance/development tasks. 	<ul style="list-style-type: none"> - Written source code by other software developers - Lack of comment in source code - Team communication/ collaboration 	<ul style="list-style-type: none"> - Mentoring activities performed by experienced developers from the team - Training - Using a tool called YODA to identify a suitable mentor in order to pair with newcomers. 	- More than 70% of accurate rate to pair a mentor to newcomers.	N/A	N/A

		<p>roles for the newcomers</p> <ul style="list-style-type: none"> - Newcomers generally start their tasks from source code or from design documents then checking back and forth between source code and diagrams. - Source code identifiers are very important to newcomers for understanding the source code and perform any maintenance/development activities. - YODA (a tool to identify mentors based on historical data of the project) identifies candidate pairs of mentor-newcomer with a precision in most cases higher than 80%, and has the recommending precision greater than 70%. 						
(Pham, 2014)	To propose the onboarding strategies with particular traits of social transparency in order	Social transparency has been shown to influence the testing behaviour of development teams	- Understand the basic testing techniques	- Trouble understanding and basic testing techniques	- Social transparency mechanisms to adapt testing skills	N/A	N/A	- Mentor may not have time to do their own job

	<p>to improve the specific testing issues of newcomers</p> <p>The goal of the thesis is to provide effective means to systematically improve the onboarding process of inexperienced new hires regarding the lack of testing skills through the use of social transparency.</p>	<p>on a social coding site. An environment that is open for discussion helps newcomers to understand and adapt a team's testing culture.</p>		<p>- Lack of testing skills, debugging skills</p>	<p>- Pair experienced developers to newcomers</p>			
(Pham et al., 2017)	<p>To explore the view and concerns of practitioners dealing with onboarding phase of inexperienced software engineers.</p>	<p>- New comers seems to be able to pick up systematic testing quickly when they join a team that is already in practices.</p> <p>- Practitioners see many problems of education in software testing. The education has inadequate for real world industrial needs and standards.</p> <p>- Practitioners blame educational institutions that a software testing is not taught at university.</p>	<p>- Understanding of general systematic of testing</p> <p>- Knowledge to design test cases and a particular one for a certain situation.</p> <p>- Raising observability of newcomers [can be achieved by communicating the testing culture efficiently to newcomers]</p> <p>- Lowering complexity, raising trialability [can be achieved by strategically providing technical</p>	<p>- Viewing automate testing as a waste of time by inexperienced software developers.</p> <p>- Newcomers have difficulties applying their theoretical testing knowledge to work and implement actual tests.</p> <p>- Newcomers have hard time to struggle overcome the technical barriers of writing tests in an actual environment</p> <p>- A lack of hands-on experience and training</p>	<p>- Practitioners' experiences for dealing with new hires' testing skills 1) Mentoring, 2) Joining bootcamps.</p> <p>- Semi-active strategies as follow: Provide some support from senior developers as ad hoc so that new hires need to ask questions when they need help while doing their assigned tasks by mentor.</p> <p>- Active strategies as follow: Senior developers or practitioners need to teach new hires</p>	N/A	N/A	N/A

			knowledge to newcomers especially how to perform testing during the onboarding stages]		actively either putting newcomers join bootcamp or assign a small project for new hires to work on. The learning process more depend on practitioners' side to drive the process. - Work around together new hires and practitioners: Encourage new hires to perform manually test during their onboarding phases as this could give them a chance to learn and gradually advance a systematic testing skill			
(Pham et al., 2015)	To make the team's testing culture visible during the onboarding phases by strategically employing traits of social transparency. To increase testing activities in inexperienced new hires while keeping minimize the instructional efforts from practitioners' side	- Visualization of testing activities of team members encourage newcomers to perform their own test cases.	- Understanding in systematic and automat testing activities - Ability to assume testing activities (simulate test cases)	- Lack of hands-on experience in systematic testing - Difficulties in systematic and automate testing in software development.	- Visible testing cultures of team members in order to influence new hires aware of testing efforts were surrounded. - Show the testing signals in a dashboard screen where all other teams' testing activities display on the screen or this could be an	The newcomer can develop his/her own accountability once seeing others' testing performance against his/her testing ones.	N/A	The goals are not fully addressed, and further evaluation is need

					accountable way for everyone in the team. Six testing signals are introduced: 1) Test code Explorer, 2) Test code documentation, 3) Latest test code commits, 4) Test code ratio, 5) Test code coverage, 6) Using test services			
(G. G. Sharma & K. J. Stol, 2020)	To explore the relationship among onboarding new hires and turnover intention of these professionals. Develop a theoretical model to distinguish onboarding activities and associate them to the success of onboarding.	- Support from organizations play a significant role to the success of onboarding, while training is not that high important. - Job satisfaction is linked to the relationship between onboarding success and employees' retention in an organization, while workplace relationship is not an aspect to be incentive the employees.	- Newcomers feel comfortable to new position and workplace - knowledge of organizational culture, - Role clarity - Self-efficacy - Social integration	- Learning new development environments, tools and stacks (technical problems) - Product knowledge (including design, technologies) - Organization processes and practices (including roles, expertise, inter-team organization) - Lack of documentation - A skill gap between new graduates and industrial expectations relating to testing skills - Understanding legacy codes which	- Orientation: contents are oriented for newcomers which included 1) attendance of orientation program, 2) awareness of organizational rules and policies, 3) assignment of a "buddy" or mentor, 4) assignment of items related to organization to facilitate new colleagues and team activities - Training: focus on specific activities to ensure newcomers are able to handle their tasks (job role, internal system and operational practices, learn specific tools and methods)	- The study indicated that support from organization is positively associated to the success of onboarding program.	- Support is found the largest and most significant factor associated with onboarding success, with a standardized path coefficient of 0.58, and an effect size of 0.64. - Onboarding success has a considerable and statistically significant positive association with job satisfaction. - The orientation program does contribute to onboarding success, though the effect size is low (0.06).	N/A

				were originally written by developers who are not onsite	- Support: Organization helps newcomers in the onboarding process (including availability of senior/mentor when newcomers need help on given tasks, asking embarrassed questions, obtaining constructive feedback or discuss personal issues)			
(Silva et al., 2020)	To investigate how well the Google Summer of Code (GSOC) engagement program in 3 months with providing stipends and mentorships to students that later contribute to Open-source software development projects.	<p>- Participated students joining the Google Summer of Code could enrich their experience although the participation is not necessary become more frequent contributors.</p> <p>- Apart from stipends are incentive to motivate student's participation, they also can improve their ability to update their resumes</p>	<p>- Newcomers' self-guided involvement in OSS projects.</p> <p>- Familiarity on codebase so that students can start project early.</p> <p>-----</p> <p>Goals of joining the GSOC:</p> <ol style="list-style-type: none"> 1) Learning motivation 2) Career building 3) Technical challenges 4) Become frequent contributors to OSS projects 5) Essence of stipends for future tuition fee of students. 	- Difficulty to go into the OSS community	- Mentors were assigned to support students to work on given tasks.	- Participating the Google Summer of Code offered a chance for students to interact with OSS mentors and other members.	- Male mentors were suggested by respondents for supervision during the, and most of them should be more than 10 years of development experiences.	N/A

(Steinmacher et al., 2014)	To identify barriers that prevent newcomers when they are onboarding to OSS projects	<p>58 barriers were revealed, and they are organized into 6 categories:</p> <ul style="list-style-type: none"> - Reception issues - Newcomers' characteristic - Newcomers need orientation - Documentation problem - Cultural differences - Technical hurdles 	- Contribution of newcomers to the OSS project	<p>Reception issues:</p> <ul style="list-style-type: none"> - Not receiving an answer - Delayed answers - Impolite answers - Receiving answers with too advanced/complex contents <p>Newcomers' characteristic</p> <p>- Newcomers' behavior:</p> <ul style="list-style-type: none"> - Lack of pro-activity - Lack of commitment - Underestimating the Challenge - Lack of patience <p>- Newcomers' communication:</p> <ul style="list-style-type: none"> - Not sending a meaningful/correct message - English level - Shyness - Making useless comments in the mailing list/forum - Low responsiveness <ul style="list-style-type: none"> - Not acknowledging / thanking answers <p>- Newcomers' previous knowledge</p> <ul style="list-style-type: none"> - Lack of domain expertise 	N/A	N/A	N/A	N/A
----------------------------	--	---	--	--	-----	-----	-----	-----

				<ul style="list-style-type: none"> - Lack of knowledge in project process and practices - Lack of technical background: - knowledge on technologies and tools used - Proper knowledge in the programming language - knowledge on versioning control system - Choosing the right development tools - Experience on unit testing <p>Newcomers need orientation</p> <ul style="list-style-type: none"> - Finding a task to start with - Finding a mentor - Finding the correct artefacts to fix an issue - Poor “How to contribute” available - Outdated list of bugs - Reproducing issues - Don’t know what is the contribution flow? <p>Documentation problem</p> <ul style="list-style-type: none"> - Outdated documentation 				
--	--	--	--	--	--	--	--	--

				<ul style="list-style-type: none"> - Information overload - Unclear documentation - Spread documentation - Code comments not clear - Lack of documentation: <ul style="list-style-type: none"> - documentation in general - Design documents - Documentation on setting up workspace - Code comments - Code documentation - Documentation on project structure <p>Cultural difference</p> <ul style="list-style-type: none"> - Some newcomers need to contact in real person - Message is consider rude <p>Technical hurdles</p> <p>Code/Architecture hurdles:</p> <ul style="list-style-type: none"> - Code characteristics: <ul style="list-style-type: none"> - Bad code quality - Code complexity/instability - Codebase size 				
--	--	--	--	---	--	--	--	--

				<ul style="list-style-type: none"> - Bad design quality - Lack of code standard - Outdated code - Understanding the code <p>Cognitive problems:</p> <ul style="list-style-type: none"> - Understanding Architecture/ code structure - Understanding flow of information <p>Change request hurdles:</p> <ul style="list-style-type: none"> - Lack of information how to send a contribution - Delay to get contribution accepted/ reviewed - Getting contribution accepted - Issue to create a patch <p>Local environment setup</p> <ul style="list-style-type: none"> - Building workspace locally - Platform dependency - Library dependency - Finding the correct sources 				
(Steinmacher, Conte, et al., 2015)	Based on the previous study on critical barriers newcomers face	Empirical evidence of the barriers faced by newcomers in OSS projects are	- Contribution of newcomers to the OSS project	<p>Reception issues</p> <ul style="list-style-type: none"> - Not receiving an answer - Delay answers 	N/A	N/A	N/A	N/A

	when joining the Open Source Software (OSS). This research focussed on the social barrier category which is one of the revealed barriers from previous study.	discovered / and reported when they provide the first contribution to the project/team.		<ul style="list-style-type: none"> - Impolite answers - Receiving answers with too advanced/complex contents <p>Newcomers' characteristic</p> <p>Newcomers' communication:</p> <ul style="list-style-type: none"> - Not sending a meaningful/correct message - English level - Shyness - Making useless comments in the mailing list/forums - Low responsiveness - Not acknowledging/thanking answers <p>Newcomers need orientation</p> <ul style="list-style-type: none"> - Finding a mentor <p>Cultural differences</p> <ul style="list-style-type: none"> - Some newcomers need to contact a real person - Message is consider rude 				
(Steinmacher et al., 2016)	To proposed and evaluated a portal (called FLOSScoach) to support	The results show that the supported portal played an important role to guide	- First contribution of newcomers to the OSS project	Based on the previous research of Steinmacher, the lists of newcomers'	- Develop the FLOSScoach portal to support newcomers	- In case of students/newcomers took wrong path in the analysis of	- FLOSScoach portal constantly provide accessibility of observers for viewing	Technical hurdles: - workspace setup issues

	<p>newcomers, to OSS projects.</p> <p>This portal was designed based on a conceptual model of barriers created in our previous work.</p>	<p>newcomers and in reducing the level of oriented barriers and contribution process. However, the tool was still not able to address technical barriers. In addition, the portal is useful, simply used, and increased newcomers' confidence to contribute.</p>	<p>- Self-efficacy</p>	<p>barriers to OSS project mapping into the FLOSScoach portal as following:</p> <p>Newcomers need orientation</p> <p>Newcomers' characteristic behavior</p> <p>- Newcomers' communication</p> <p>Reception issues</p> <p>Technical hurdles</p> <p>Documentation problem</p>	<p>- Newcomers use the web portal in diary in order to track their behaviour on using the portal, the participants documented what they have done and shared to the researchers. The researchers then interact and provide feedback accordingly.</p>	<p>codebase (e.g cloning the wrong repository and trying to set up). The FLOSScoach could provide a guidance to avoid this situation.</p> <p>Participant 1 – 05 said that the tool helped a lot, the outstanding guidance as what I need was provided and I could save time and made me more confident.</p>	<p>what participants do and then the observers provide feedback appropriately.</p>	<p>- difficulty understanding the architecture/code structure, understanding the code, - problems finding the correct artefact to fix an issue</p>
<p>(Steinmacher et al., 2019)</p>	<p>To identify and better understand social barriers that prevent newcomers' first contribution to the OSS project.</p> <p>To develop FLOSScoach, in order to use for supporting the first contribution of newcomers</p>	<p>- The FLOSScoach tool can save/reduce required time of newcomers' communication</p> <p>The newcomers who reported receiving responses in their diaries did not mention any cases of receiving improper answers.</p> <p>Providing such a tool like FLOSScoach could help newcomers to avoid</p>	<p>- First contribution of newcomers to the OSS project</p> <p>- Self-efficacy</p>	<p>Based on the previous research of Steinmacher, 13 Social barriers to prevent the first contribution of newcomers:</p> <p>Reception issues</p> <p>- Not receiving an answer</p> <p>- Delay answers</p> <p>- Impolite answers</p> <p>- Receiving answers with too advance/complex contents</p> <p>Communication</p>	<p>- Develop the FLOSScoach portal to guide and support OSS project newcomers (their first contribution)</p> <p>- Mapping each category of barriers into the FLOSScoach's functions, and show descriptive guidance how to start each function.</p> <p>- Using robots to automated answers/feedback to newcomers'</p>	<p>- One participant mentioned that message templates from the FLOSScoach were helpful. The participants can present their faced problems concisely/clearly and this could help to reduce shyness in communication.</p>	<p>N/A</p>	<p>N/A</p>

		or overcome social barriers		<ul style="list-style-type: none"> - Not sending a meaningful / correct message - English level - Shyness - Making useless comments in the mailing list/forum - Low responsiveness - Not acknowledging / thanking answers <p>Finding a mentor</p> <p>Cultural differences</p> <ul style="list-style-type: none"> - Some newcomers need to contact a real person - Message is considered rude 	<p>contribution was recommended to handle the delay responsiveness barriers, but still need further investigation to see its effectiveness</p> <ul style="list-style-type: none"> - Automatic greeting was also suggested to handle shyness barriers when newcomer join the team. - Using a tool to help newcomers to find the appropriate mentor was also suggested to alleviate the difficulty finding mentors barriers. 			
(Steinmacher, Wiese, et al., 2015)	To evaluate how the self-efficacy of newcomers is influenced by the use of an environment that organizes the project information for developers who first contribute their efforts to an OSS project.	The study revealed that presenting the information according to the model of barriers throughout the portal had a positive influence on newcomers' self-efficacy, strengthening the newcomers, increase their confident and comfortable during the contribution process.	<ul style="list-style-type: none"> - First contribution of newcomers to the OSS project - Self-efficacy 	The list of barriers are presented based on the previous research in (Steinmacher et al., 2014)	- Develop the FLOSScoach, a portal intending to support newcomers. The tool presented/organized information according to the barrier models.	- Most used features of the contribution flow was "How to Contribute " page.	- The portal was assessed by students who participated to answers questionnaires before and after completing assignments throughout FLOSScoach	FLOSScoach portal did not lower the technical barriers for the newcomers when placing their first contribution to the OSS project.

(Viana et al., 2014)	To study about knowledge transfer from senior software developers to novices in a small software organization. - Understanding how novice engineers gain knowledge when perform initiative activities in the software organization.	- The study shown that software developers had plenty of knowledge sources in order to introduce to newcomers. - Novice engineers could learn by observation the organizational procedures and the assigned tasks had detailed guidelines	- Gaining a domain knowledge (technologies) - Understanding software engineering method - Understanding the organization's internal process	- No formal way for knowledge transfer in small software organization - No well-defined schedule - Lack of defined software process	- Provide diverse trainings focusing on specific programming languages, database, and business practices. (basic required skills for novice engineer's assigned tasks) - Novice engineers received new assignments every day from senior engineers by emails	N/A	- Emails from senior engineers contained too much information about organizational assignments - Most detailed knowledge was provided through face-to-face conversation.	- Informal conversation was preferable by novice practitioners - Ignorance of software documentation by novice practitioners
(Viviani & Murphy, 2019b)	To study how onboarding implemented in a mid-sized software development company To achieve this purpose, a case study was conducted by interviewing eight developers from a local company with a development team consist of 100.	The study revealed that the most reliable practices to speed up newcomers' onboarding process were buddy (pair) programming and code reviews in the developers' perspectives.	N/A	- Overwhelming with the amount of received information. - Struggling to understand the appropriateness of information to their tasks/roles - Lack of the familiarity with the system.	- Buddy(pair) programming - Buddies act as mentors to new hires - Code reviews	N/A	N/A	N/A
(Wang, 2012)	To explore a supporting tool for developers onboarding in a software development project	- Some critical onboarding concerns were discovered - Discovered onboarding challenges were mapped to the built-	- Understanding to the basic structure of a team code base - Understanding the overall of project structures	Focus challenges on this research: - Finding a good starting point - Difficulty for newcomers to generate appropriate	Proposed an onboarding support tool (called Tesseract) with a number of features associate to the list of challenges. Significant features	N/A	The Tesseract was evaluated by user studies. - Effectiveness: Correctness rate of search result from the synonym-based search and similar-	N/A

		<p>in features of the Tesseract tool</p> <p>- Built-in features in the Tesseract tool are almost support the discovered onboarding challenges of this research.</p>	<p>- Understanding social structure and culture of the project</p>	<p>queries in order to investigate a problem.</p> <p>- Difficulty on understanding the overviews of project structure and how each different structure is associated.</p> <p>Other mentioned onboarding challenges:</p> <p>- Technical problem</p> <p>- Communication</p> <p>- Collaboration</p> <p>- Finding relevant resources</p>	<p>in the tool consisted of:</p> <p>- Synonym-based search and similar-bugs search: to find a starting point</p> <p>- Cross-linked displays across different project entities: to investigate all relevant projects in a single repository</p> <p>- Semantic relationship: to support investigation of semantic relationships and capture all relevant project resources.</p> <p>- Dependencies across files, bugs, developers, and communications: to explore socio, technical, social-technical dependencies</p> <p>- Network-centric views: to provide a high-level overview of project structures,</p> <p>- Explore issue and related resources; view project structures: To explore issues and related resources</p>		<p>bugs search function is slightly higher than experimental group.</p> <p>- Efficiency: Users could learn while they performed their tasks</p> <p>- User satisfactory: highly satisfied by users especially the synonym-based search and similar-bugs search function.</p>	
(Yang, 2017)	- To study the onboarding process		Refer to (Buchan et al., 2019a)	Refer to (Buchan et al., 2019a)	Refer to (Buchan et al., 2019a)	Refer to (Buchan et al., 2019a)	Refer to (Buchan et al., 2019a)	

	<p>in Agile software development team.</p> <ul style="list-style-type: none">- To develop a conceptual framework of onboarding goals, characteristics of an onboarded team member.- To identify common onboarding activities in Agile software development teams.- To map a list of onboarding tasks to each desired onboarding goal.							
--	---	--	--	--	--	--	--	--

(Yates et al., 2020)	<p>To study the onboarding program in the perspectives of newcomers and experts.</p> <p>Start onboarding session by experts who explain unfamiliar codebase of application to newcomers.</p>	<p>On the newcomer's perspectives regarding codebase knowledge transformation, experts are required to support as questionnaires as following:</p> <ul style="list-style-type: none"> - Experts need to describe the high-level architecture - Experts need to name chunk of code and probable misconceptions - Experts need to go through all line of simple/complex code 	<ul style="list-style-type: none"> - Understanding the code base which is written by other developers. 	<ul style="list-style-type: none"> - Difficulty to understand code written by other developers - Complexity of code 	<ul style="list-style-type: none"> - Focus on walkthroughs of complex lines of code (experts explain its workings and intention) - Describing potential misinterpretation (experts describe probable misinterpretation in codebase that could lead to various mistakes.) - Adopting top-down approach (experts start explaining from high-level code architecture and gradually move to more detail) 	<ul style="list-style-type: none"> - Request to describe the high-level architecture is the first highest agreement score from newcomers - Request to name the chunk of code is the second highest agreement score 	<p>Some feedback from on-boarders:</p> <p>It could be useful if experts could follow:</p> <ul style="list-style-type: none"> - Describe the high-level architecture - Point out design pattern in use - Names chunks of code - Points out simple, complex area of code - Points out which areas of code are off-limits. - Goes through line by line of complex code - Points out possible misconceptions - Points out temporary fixes in the code - Points out dead code 	
----------------------	--	---	---	---	---	--	---	--

Appendix B - A summary of three discovered categories of onboarding research contexts

No	Author(s) & Year	Purpose of articles	Onboarding contexts
1	(Balali et al., 2018)	- To investigate the barriers that affect mentors and newcomer as mentees	- Open-source software development
2	(Davidson et al., 2014)	- To investigate the lack of diversity in Free/Open-source Software (FOSS) communities.	- Open-source software development
3	(Diniz et al., 2017)	- To investigate the use of games to advocate and encourage new graduates to overcome onboarding barriers and contribute to OSS projects.	- Open-source software development
4	(Dominic et al., 2020)	- To propose a tool (conversational bot), in order to address facing problems of an onboarding process	- Open-source software development
5	(Fagerholm et al., 2014)	- To examine how mentoring and project characteristic influence the effectiveness and efficiency of the onboarding process	- Open-source software development
6	(Fagerholm et al., 2013)	- To present the preliminary observations and results of in-progress research that studies the process of onboarding into virtual OSS teams.	- Open-source software development
7	(Hannebauer & Gruhn, 2017)	- To find the relation between motivation and contribution barriers of newcomers.	- Open-source software development
8	(Kovalenko & Bacchelli, 2018)	- To investigate how low experienced developers' contribution in a project receive a different support while doing code review	- Open-source software development
9	(Liu et al., 2018)	- To propose a model called neural network for list-wise ranking (NNLR55ank) "" for project selection of developers before making a decision to join	- Open-source software development
10	(Silva et al., 2020)	- To investigate how well the Google Summer of Code (GSOC) engagement program in 3 months with providing stipends and mentorships to students	- Open-source software development
11	(Steinmacher et al., 2014)	- To identify barriers that prevent newcomers when are being onboarding to OSS projects	- Open-source software development
12	(Steinmacher, Conte, et al., 2015)	- To investigate how social barriers prevent newcomers when joining the Open-source software projects	- Open-source software development
13	(Steinmacher et al., 2016)	- To proposed and evaluated a portal (called FLOSScoach) to support newcomers, to OSS projects.	- Open-source software development
14	(Steinmacher et al., 2019)	- To identify and better understand social barriers of onboarding that prevent newcomers' first contribution	- Open-source software development
15	(Steinmacher, Wiese, et al., 2015)	- To evaluate the self-efficacy of newcomers	- Open-source software development
16	(Britto et al., 2018)	- To investigate the onboarding challenges, and areas for further improvement	- Globally distributed, large projects/teams

No	Author(s) & Year	Purpose of articles	Onboarding contexts
17	(Britto et al., 2020)	- To investigate the relationship between onboarding strategies and the performance evolution of newcomers	- Globally distributed large projects/teams
18	(Hoffström, 2019)	- To investigate gaps of the current onboarding process	- Globally distributed large projects/teams
19	(Kumar et al., 2016)	- To report the evolving nature of onboarding and mentoring	- Globally distributed large projects/teams
20	(Moe et al., 2020)	- To investigate how a bank implemented onboarding developers for distributed teams	- Globally distributed large projects/teams
21	(Cunha et al., 2020)	- To investigate the difficulties of newcomers using a tool (Wiki)	- Globally distributed large projects/teams
22	(Buchan et al., 2019a)	- To investigate how Agile software development team addresses the challenges of team onboarding	- Agile software development team
23	(Yang, 2017)	- To study the onboarding process in Agile software development team.	- Agile software development team
24	(Liu, 2019)	- To gain understanding on student's expectation of onboarding in Agile software development team	- Agile software development team
25	(Deshpande & Gupta, 2019)	- To investigate factors which play important role in retention of newly hired employees	- Medium/small software development team
26	(Matturro et al., 2017)	- To report the results of a research aiming to identify newcomers' difficulties when start joining an in-progress executing software development projects	- Medium/small software development team
27	(Nolte et al., 2020)	- To investigate the support scientific communities in the development of hackathons which grow interest in their community	- Medium/small software development team
28	(Panichella, 2015)	- To investigate problems arising when newcomers join software project. - Study how newcomers interact with mentors/other developers during the onboarding	- Medium/small software development team
29	(Pham, 2014)	- To investigate the onboarding strategies with particular traits of social transparency and addressing of testing issues - To provide effective means to systematically improve the onboarding process regarding the lack of testing skills	- Medium/small software development team
30	(Pham et al., 2017)	- To explore the view and concerns of practitioners dealing with onboarding phase	- Medium/small software development team
31	(Pham et al., 2015)	- To make the team's testing culture be visible during the onboarding phases by strategically employing traits of social transparency.	- Medium/small software development team
32	(G. G. Sharma & K. J. Stol, 2020)	- To explore the relationship among onboarding new hires and turnover intention of these professionals	- Medium/small software development team
33	(Viana et al., 2014)	- To study about knowledge transfer from senior software developers to novices	- Medium/small software development team

No	Author(s) & Year	Purpose of articles	Onboarding contexts
34	(Viviani & Murphy, 2019b)	- To study how onboarding implemented in a mid-sized software development company	- Medium/small software development team
35	(Wang, 2012)	- To explore a supporting tool for developers in an onboarding program	- Medium/small software development team
36	(Yates et al., 2020)	- To study the onboarding program in the perspectives of newcomers and experts.	- Medium/small software development team

Appendix C - A Summary of discovery-desired onboarding goals of onboarding program

Categories	Descriptive Onboarding Goals	References
Company culture	Understanding the organization's internal process	(Britto et al., 2020; Britto et al., 2018; Cunha et al., 2020; Panichella, 2015; Viana et al., 2014)
	Understanding company culture/structure	(Britto et al., 2020; Britto et al., 2018; Buchan et al., 2019a; Cunha et al., 2020; Liu, 2019; Matturro et al., 2017; Moe et al., 2020; G. G. Sharma & K. J. Stol, 2020; Wang, 2012; Yang, 2017)
Team Norm	Understand structures of teamwork, aims and implications	(Buchan et al., 2019a; Liu, 2019; Yang, 2017)
	Understanding software engineering method	(Viana et al., 2014)
	Understanding the standard of work quality of teams	(Buchan et al., 2019a; Liu, 2019; Yang, 2017)
Understanding responsibility, roles, and its quality standard	Understanding and meet others' expectation on your own responsibility	(Britto et al., 2020; Britto et al., 2018; Buchan et al., 2019a; Liu, 2019; Yang, 2017)
	Understand the responsibilities, expertise, and authority of other team members	(Buchan et al., 2019a; Liu, 2019; G. G. Sharma & K. J. Stol, 2020; Yang, 2017)
Curious and confident behaviors	Building feedback seeking behaviours	(Deshpande & Gupta, 2019)
	Build confidence of asking questions when need help	(Liu, 2019)
Participation behaviors	Be proactive participation of newcomers	(Deshpande & Gupta, 2019; Fagerholm et al., 2014)
	Building interaction with senior developers	(Britto et al., 2020; Britto et al., 2018)
Familiarity in workplace environments	Understanding work environment	(Britto et al., 2020; Britto et al., 2018; Liu, 2019; Matturro et al., 2017)
	Be feel comfortable to new position and workplace	(G. G. Sharma & K. J. Stol, 2020)
Understanding and be able to use technologies required by teams	Be able to understand new tools and systems of the company	(Kumar et al., 2016; Matturro et al., 2017)
	Be able to understand desired technologies by projects	(Matturro et al., 2017)
	Be faster to find solution to the problem	(Liu, 2019)
Understanding and be able to use source code	Gaining familiarity with source code and related documents	(Panichella, 2015)
	Understanding the code base which is developed by other developers	(Silva et al., 2020; Wang, 2012; Yates et al., 2020)

developed by other developers	Understand how to code and test as expectation level of team members	(Britto et al., 2020; Britto et al., 2018; Buchan et al., 2019a; Kovalenko & Bacchelli, 2018; Liu, 2019; Panichella, 2015; Yang, 2017)
Effective communication among team members	Be able to efficient communicate in term of different ages	(Davidson et al., 2014; Maturro et al., 2017)
	Be flexible to communicate in different means	(Balali et al., 2018; Davidson et al., 2014; Moe et al., 2020)
Contribution	Understand the first contribution to teams/projects	(Balali et al., 2018; Dominic et al., 2020; Fagerholm et al., 2013; Hannebauer & Gruhn, 2017; Kovalenko & Bacchelli, 2018; Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Understand what to commits and workflow	(Diniz et al., 2017; Fagerholm et al., 2014)
Adaptability to tasks, roles and responsibilities	Be adaptable to tasks, teams & project environments	(Balali et al., 2018; Liu, 2019)
	Be able to evaluate self-ability	(Balali et al., 2018; G. G. Sharma & K. J. Stol, 2020; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Be able to understand socialization and skills	(Britto et al., 2020; Britto et al., 2018; Liu, 2019; Moe et al., 2020; G. G. Sharma & K. J. Stol, 2020)
Relationship and Cooperation	Understand and be able to build cooperation behaviours	(Britto et al., 2020; Britto et al., 2018; Fagerholm et al., 2014)
	Understanding and be able to create long term relationship to teams	(Liu, 2019)
Design test cases, activities & scenarios	Understand the basic, systematic testing techniques, activities	(Pham, 2014; Pham et al., 2017; Pham et al., 2015)
	Knowledge to design test cases and a particular one for a certain situation	(Pham et al., 2017)
	Be able to assume testing activities	(Pham et al., 2017; Pham et al., 2015)
Understanding the agile mindset, ability to adopt and use its techniques	Know how to use Agile artefacts and techniques used by the rest of the team	(Buchan et al., 2019a; Liu, 2019; Yang, 2017)
	Be able to understand and adopt to Agile mindset	(Buchan et al., 2019a; Liu, 2019; Yang, 2017)
Product domain knowledge and terminology	Be able to understand the product domain knowledge and terminology	(Buchan et al., 2019a; Hoffström, 2019; Liu, 2019; Maturro et al., 2017; Viana et al., 2014; Wang, 2012; Yang, 2017)

Appendix D - A summary of discovery-suggested onboarding tasks to support desired goal

Themes	Techniques/Tasks	Explanation	Sources
Activity	- Working on a bug or issue together	Mentors could work closely with mentees on bugs or re-generate issues and investigate the root causes.	(Balali et al., 2018)
Activity	- Flagging newcomers so others are welcoming to them	Putting a label or flag for newcomers in to remind other team members be aware	(Balali et al., 2018)
Artefact	- Communication through different means	Use wide ranges of communicating channels	(Balali et al., 2018)
Activity	- Giving the newcomers simple/interesting tasks	Assign interesting/small/simple tasks	(Balali et al., 2018; Buchan et al., 2019a; Fagerholm et al., 2013)
Activity	- Giving newcomers rewards for keeping motivation	Give rewards in order to motivate newcomers be active and focus on assigned tasks	(Diniz et al., 2017)
Activity	- Having newcomers share work progress	Let newcomers to share their tasks' progress to team members for transparency and accountability.	(Balali et al., 2018)
Activity	- Tagging the tasks according to their complexity	Tagging the tasks could possibly identify appropriate tasks for newcomers based on its complexity	(Balali et al., 2018)
People	- Having local groups in each country	Set a local group for exchanging experiences, challenges, solutions and address communication issues	(Balali et al., 2018)
Activity	- Keeping documentation concise and updated	Organization need to provide the simple, up to date documents to newcomers in order to avoid any confusion	(Balali et al., 2018)

Activity	- Recruitment integrated with onboarding	Having a senior developer to participate the recruitment process for prior assessment of candidates	(Britto et al., 2020; Britto et al., 2018; Moe et al., 2020)
Activity	- Stakeholder involvement in recruitment	Having a representative from stakeholder to participate the recruitment process for prior assessment of candidates	(Britto et al., 2020; Britto et al., 2018)
People	- First day of the job is special	Create a welcome atmosphere as much as possible in the first working day, or create a good memorable experience for newcomers	(Britto et al., 2020; Britto et al., 2018)
Artefact	- Using intranet portal to centrally share all documents	An intranet portal should be accessible for getting all up to date documents, resources	(Britto et al., 2020; Britto et al., 2018; Dominic et al., 2020; Hoffström, 2019)
Activity	- Assign real tasks under carefully support by senior developer	Refers to assigning hands-on tasks for newcomers rather than just theory	(Britto et al., 2020; Britto et al., 2018; Kumar et al., 2016; Viana et al., 2014)
People	- Mentoring programs	Assign a mentor to support newcomers throughout the onboarding process	(Britto et al., 2020; Britto et al., 2018; Buchan et al., 2019a; Dominic et al., 2020; Fagerholm et al., 2014; Fagerholm et al., 2013; Kumar et al., 2016; Liu, 2019; Nolte et al., 2020; Panichella, 2015; Pham et al., 2017; G. G. Sharma & K. J. Stol, 2020; Silva et al., 2020)
Activity	- Formal training on hard skill and/or soft skills	Organize formal training programs for newcomers. Hard skills refer to how to do actual project tasks while soft skills refer to how to work along with team members, communicating.	(Britto et al., 2020; Britto et al., 2018; Liu, 2019; Matturro et al., 2017; G. G. Sharma & K. J. Stol, 2020; Viana et al., 2014)
Activity	- Regular stakeholder meetings	Organize a regular meeting with stakeholders for sharing progress and exchange experiences.	(Britto et al., 2020; Britto et al., 2018)
Activity	- Own progress monitoring	Newcomers should review their own work progress	(Britto et al., 2020; Britto et al., 2018; Fagerholm et al., 2013; Matturro et al., 2017)
Activity	- Perform appraisals	Refers to assessment of newcomers' performance	(Britto et al., 2020; Britto et al., 2018)

Activity	- 360-degree feedback	Refers to giving feedback by all related onboarding facilitators to newcomers	(Britto et al., 2020; Britto et al., 2018; Moe et al., 2020)
Artefact	- Online communication	Refers to using wide ranges of online tools for communication	(Buchan et al., 2019a; Davidson et al., 2014; Yang, 2017)
People	- Peer support	Refers to providing constantly support to newcomers by peers	(Buchan et al., 2019a; Maturro et al., 2017; Moe et al., 2020; Pham, 2014; Pham et al., 2017; Yang, 2017)
Activity	- Team socializing	Refers to socialisation with team members in order to influence trust as well as exchange experience	(Buchan et al., 2019a; Moe et al., 2020; Pham, 2014; Yang, 2017)
Activity	- Training course	Refers to provide a wide ranges of training courses to newcomers	(Balali et al., 2018; Buchan et al., 2019a; Moe et al., 2020; Panichella, 2015; Yang, 2017)
Activity	- Code review	Refers to reviewing written codes of different products or functions	(Buchan et al., 2019a; Kovalenko & Bacchelli, 2018; Viviani & Murphy, 2019b; Yang, 2017)
Activity	- Internal documentation	Refers to having organizational/team information to be shared for newcomers	(Buchan et al., 2019a; Yang, 2017)
Activity	- Product overview	Refers to reviewing overall products developed by the team or company	(Buchan et al., 2019a; Hoffström, 2019; Yang, 2017)
Activity	- Pair programming	Refers to pairing a newcomer with his/her buddy and work on writing code in the same workstation	(Buchan et al., 2019a; Kumar et al., 2016; Maturro et al., 2017; G. G. Sharma & K. J. Stol, 2020; Viviani & Murphy, 2019b; Yang, 2017)
Activity	- Regular team standup meeting	Refers to a meeting to exchange working experiences, improve among teams or members in a team.	(Buchan et al., 2019a; Fagerholm et al., 2013; Liu, 2019; Yang, 2017) (Britto et al., 2020; Britto et al., 2018)
Activity	- Self-learning	Refers to learning new skills by newcomers without tutors, it could be from videos, online courses	(Buchan et al., 2019a; Yang, 2017)
Activity	- Induction	Refers to giving introduction session, explaining the overall team/company structure, working environments	(Buchan et al., 2019a; Liu, 2019; G. G. Sharma & K. J. Stol, 2020; Yang, 2017)
Activity	- Access and contribute to a local	Refers to using internal wiki pages to share assigned tasks	(Buchan et al., 2019a; Cunha et al., 2020; Yang, 2017)

	knowledge DB (Wiki pages)		
People	- Team leader supports	Refers to supporting from team leaders	(Buchan et al., 2019a; Yang, 2017)
Activity	- Course on Agile	Refers to joining agile course to understand agile practices	(Buchan et al., 2019a; Yang, 2017)
Activity	- Team retrospective	Refers to team reflection particularly the way of team working, exchange experiences and improvement.	(Buchan et al., 2019a; Fagerholm et al., 2013; Yang, 2017)
Activity	- Review plan	Refers to reviewing the working plan	(Buchan et al., 2019a; Yang, 2017)
Activity	- Attend conference	Refers to joining conference of newcomers	(Buchan et al., 2019a; Yang, 2017)
Artefact	- Electronic communication	Refers to communicating through electronic channels	(Buchan et al., 2019a; Davidson et al., 2014; Yang, 2017)
Activity	- Meet with other teams	Refers to meeting with other teams for getting more understanding on team norms, exchange experiences	(Buchan et al., 2019a; Yang, 2017)
Activity	- Location map	Refers to labelling a map for navigation in the workplace	(Buchan et al., 2019a; Yang, 2017)
Activity	- Checklists	Refers to a list of tasks with its status	(Buchan et al., 2019a; Yang, 2017)
Activity	- Match contribution efforts to individual motivations	Refers to matching what newcomers contribute to the teams and his/her motivative aspects	(Davidson et al., 2014)
Activity	- Do not ignore the bad stuffs	Refers to taking a good analysis on any minor bad stuffs	(Davidson et al., 2014)
Activity	- Advocacy from supervisors	Refers to suggestion by a supervisor while assigned tasks are in progress	(Deshpande & Gupta, 2019)
Activity	- Gamification	Refers to using of gamification to orient/motivate new graduates to be more engage in OSS projects	(Diniz et al., 2017)
Activity	- Establishing a relationship	Refers to building a trust and networking	(Fagerholm et al., 2014)
Activity	- Technical support	Refers to a team member to support any technical issues to newcomers	(Nolte et al., 2020; G. G. Sharma & K. J. Stol, 2020)
Artefact	- Using a tool to identify a suitable mentor	Refers to using a tool to identify a suitable mentor for mentees	(Panichella, 2015)
Activity	- Joining bootcamp	Refers to participation on intensive courses	(Pham et al., 2017)

People	- Visible testing practices	Refers to visualization on what are being tested by newcomers	(Pham et al., 2015)
Artefact	- Develop the FLOSScoach portal to support newcomers	Refers to using an artefact to help newcomers to overcome some discovered onboarding problems	(Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
Artefact	- Using built-in features in the Tesseract tool to support onboarding	Refers to using the Tesseract tool to facilitate newcomers and companies to visualize an onboarding progress	(Wang, 2012)
Activity	- Focus on walkthroughs of complex lines of code	Refers to sitting with senior developers to go through complex lines of code	(Yates et al., 2020)
People	- Describing potential misinterpretation	Refers to pointing out and explanation some written codes that are possibly misunderstood	(Yates et al., 2020)
People	- Adopting top-down approach (code architecture)	Refers to explanation of code architecture from high to small layers.	(Yates et al., 2020)

Appendix E - A summary of discovered onboarding barriers/challenges/difficulties/risks from reviewed article

Categories	Sub-categories barriers	Challenges/Barriers/Difficulties/Risks	References
Personal barriers	Issues related to self-efficacy	- Difficulties in managing different account	(Balali et al., 2018)
		- Fear of judgment	(Balali et al., 2018)
		- Performance anxiety	(Balali et al., 2018)
	Newcomers' previous knowledge	- Lack of domain expertise	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015; Viviani & Murphy, 2019b)
		- Lack of knowledge in project process and practices	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Difficulties in time management	(Balali et al., 2018)
	Newcomer personality issues	- Newcomers' personality conflicts with the role	(Balali et al., 2018)
		- Newcomers' inability to improve upon criticism	(Balali et al., 2018)
		- Not sending a meaningful/correct message	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- English level	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Shyness	(Balali et al., 2018; Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Making useless comments in the mailing list/forum	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)

		- Low responsiveness	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Not acknowledging / thanking answers	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Issues related to newcomer's behavior	- Lack of interest	(Balali et al., 2018)
		- Lack of clear professional goals	(Balali et al., 2018)
		- Lack of pro-activeness	(Balali et al., 2018; Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Lack of commitment	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Lack of patience	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Premature resignation	(Deshpande & Gupta, 2019; Dominic et al., 2020)
	Interpersonal barriers	Issues related to diversities	- Cultural differences / Company culture
- Differences in work experience and age			(Balali et al., 2018)
- Communication styles (contact in real person for some people)			(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)

		- Message is consider rude	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Challenges in communicating effectively	- Communication issues related to time zone and geography	(Balali et al., 2018; Davidson et al., 2014; Maturro et al., 2017; Moe et al., 2020)
		- Lack of English language skills	(Balali et al., 2018; Davidson et al., 2014; Moe et al., 2020; Wang, 2012)
		- Lack of mentor's interpersonal skills	(Balali et al., 2018)
		- Lack of newcomers' interpersonal skills	(Balali et al., 2018)
		- The distance to mentors	(Britto et al., 2020)
		- Distribution of team in the projects	(Britto et al., 2018; Cunha et al., 2020; Dominic et al., 2020; Yang, 2017)
	Issues with community reception	- Unpleasant project atmosphere	(Balali et al., 2018)
		- Difficulty in finding helps in the community	(Balali et al., 2018)
		- Discrimination Issues	(Davidson et al., 2014)
Process barriers	Issue with newcomers' orientation	- Difficult to identifying appropriate tasks	(Balali et al., 2018)
		- Willingness to start with a complex task	(Balali et al., 2018)
		- Long project processes	(Balali et al., 2018)
		- Finding a mentor	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Finding the correct artefacts to fix an issue	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Poor "How to contribute" available	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Outdated list of bugs	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016;

			Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Reproducing issues	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Lack of background about project procedures	- Problem with the process of submitting code	(Balali et al., 2018)
		- Lack of documentation	(Balali et al., 2018; Hannebauer & Gruhn, 2017; G. G. Sharma & K. J. Stol, 2020)
		- Clarification of work to do	(Buchan et al., 2019a; Deshpande & Gupta, 2019; G. G. Sharma & K. J. Stol, 2020; Yang, 2017)
Technical barriers			
	Code Quality Issues	- High code complexity	(Balali et al., 2018)
		- Code characteristics: bad code quality, code complexity/instability, codebase size, lack of code standard, Outdated code	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015; Yates et al., 2020)
		- Cognitive problems: understanding architecture/code structure, understanding flow of information	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Difficulty in getting started	- Difficulty in setting up development environment,	(Balali et al., 2018; Hannebauer & Gruhn, 2017; Steinmacher et al., 2014)
		- Task too complex for newcomers	(Balali et al., 2018)
		- Code and test as expectation level of team members	(Buchan et al., 2019a; Diniz et al., 2017; Yang, 2017)
		- Feel lost, stressed, anxiety, and overwhelmed on the first few day of onboarding program	(Kumar et al., 2016)
	Lack of newcomers' knowledge about programming and tools	- Difficulty in learning related tools or technologies	(Balali et al., 2018; Liu et al., 2018; Maturro et al., 2017; G. G. Sharma & K. J. Stol, 2020; Wang, 2012)

		- Learning a huge amount of legacy product code	(Britto et al., 2018; Kovalenko & Bacchelli, 2018; Panichella, 2015; G. G. Sharma & K. J. Stol, 2020)
		- knowledge on technologies and tools used	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Proper knowledge in the programming language	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- knowledge on versioning control system	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Choosing the right development tools	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Experience on unit testing	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
	Product support Issues	Change requested hurdles	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- No formal way for knowledge transfer in small software organization	(Viana et al., 2014)
		- Difficulty for newcomers to generate appropriate queries in order to investigate a problem.	(Wang, 2012)
	Contribution barriers	Modification barriers	- Difficulty on locating the right place to make specific change

		- Difficulty on downloading the right version of source code.	(Hannebauer & Gruhn, 2017)
		- Bug reproduction	(Hannebauer & Gruhn, 2017)
	Submission barriers	- Learning curve	(Kovalenko & Bacchelli, 2018; Moe et al., 2020)
		- Contributions remain unsatisfactory by the core project members	(Liu et al., 2018)
Testing barriers	Testing	- Lack of testing, debugging skills	(Pham, 2014)
		- Difficulty on test cases writing	(Pham et al., 2017)
		- Lack of hands-on experiences	(Pham et al., 2017)
		- Gap between new graduates and industrial expectations relating to testing skills	(G. G. Sharma & K. J. Stol, 2020)
Reception Issues	Issues relate to interaction	- Answering Issues: Delayed, Not received, Impolite, too advance/complex answers	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
Documentation problem	Documenting issues relate to information, version and structure	- Outdated documentation	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Information overload	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015; Viana et al., 2014; Viviani & Murphy, 2019b)
		- Unclear documentation, too general	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)
		- Unclear structures	(Steinmacher et al., 2014; Steinmacher, Conte, et al., 2015; Steinmacher et al., 2016; Steinmacher et al., 2019; Steinmacher, Wiese, et al., 2015)

