Master thesis

for the study program MSc. Business Information Technology

DEVELOPING AN AGILE PERFORMANCE MANAGEMENT INFORMATION SYSTEM

GIJS T. STERKEN

August 2020

University of Twente

Developing an Agile Performance Management Information System

Master Thesis, University of Twente, August 2020

Keywords: Agile software development, Agile, SAFe 5.0, core competencies, development outcomes, performance management, performance system, information system

Author

Gijs T. Sterken

Study program: *MSc. Business Information Technology*

Specialisation: *IT Management and Enterprise Architecture*

Graduation Committee

Dr A.B.J.M. Wijnhoven (Faculty of BMS, University of Twente)
Dr Ir. M.J. van Sinderen (Faculty of EEMCS, University of Twente)

Summary

The digital transformation requires companies to work with a combination of Agile methodologies. However, organisations have difficulties to find a performance management information system to monitor those practices to maximize their benefits. The literature does not give a system that fits in the current software age. Most systems are only focused on a specific methodology or used in a specific company.

The goal of this research is to design a uniform Agile performance management information system to enable organisations to determine and improve their Agile performance. The design of the system is based on the results of a systematic literature review and evaluated by a panel of professionals in the field of Agile and the Scaled Agile Framework. This framework is an online knowledgebase for scaling Agile across the enterprise.

The literature review on Agile performance management systems resulted in eleven research papers which are selected for this research. The review made clear which systems are available, and which gaps still exist in these systems. First of all, teams on different maturity levels operate in different ways and there is a lack of guidance to define a growth path. Secondly, the systems do not provide clarity on how the performance of teams should be interpreted. Finally, there is a lack of evidence about systems for collecting the required information.

The developed Agile performance management system in this research assesses the core competencies of the participants and team performance indicators. The core competencies of SAFe 5.0 to achieve business agility and providing a superior product are included within the system. Assessments are used to provide more insights into these competencies. The performance indicators are divided into four development outcomes: productivity, time-to-market, quality, and engagement. For each outcome, there are indicators determined and visualized in an information system. Thereafter, the competencies are discussed in relation to the development outcomes to indicate which skills or practices are essential for each outcome.

The applicability of the system is evaluated in one of the largest insurance organisations in the Netherlands. The system is used and evaluated by six departments within the case study company, and a survey is used for collecting their feedback. Due to the global COVID-19 pandemic, it wasn't possible to do face-to-face meetings and for this reason, online questionnaires and virtual sessions are used to discuss the results.

The research resulted in a uniform Agile performance management information system which enables organisations to grow in their performance. However, the research has some limitations:

- The non-verbal validation session due to the COVID-19 pandemic might have given different results, but unfortunately, it was outside the scope of the research to postpone the sessions.
- Further case studies are required to assess the completeness of the model in different organisations and industries.
- The evaluation participants argued that the current system is still too rigid. It is suggested to add objective and Agile release train specific indicators to make it more appropriate in both business and information technology.
- An organisation should take the time and costs of implementing the system into account.

Content

List of figures

List of tables

1 Introduction

To remain competitive, organisations need to digitally transform their entire enterprise. This digital transformation might entail companies working with a combination of several types of Agile methodologies (Scaled Agile, 2020). Agile is a software development methodology to build software incrementally so that the development is aligned with the changing business needs (Kaushik, 2007). Not only the software development teams are working according to the Agile principles, but also operations and support teams are finding their way within the Agile environment (Radstaak, 2019; Scaled Agile, 2020). The Scaled Agile Framework (SAFe) is an online knowledge base to support these changes in software development of organisations. The framework provides patterns, principles and tools to successfully develop large-scale products. In January 2020, an updated version (5.0) of the framework is introduced, including updated competencies to enable business Agility within the whole organisation. This framework is currently the most widely adopted framework for scaling Agile in the software industry (Putta, Paasivaara, & Lassenius, 2019).

Agile software development approaches introduce practices and a gradual approach for establishing them. Organisations need to adopt and monitor those practices and processes to maximize their benefits (Patel & Ramachandran, 2009). Several Agile performance tools are designed to focus on specific elements of one of these methodologies. However, these tools create difficulties when an organisation combines multiple Agile methodologies.

An example of an organization combining multiple Agile methodologies can be found in this research's case study. The focal point of the case study is a large Dutch insurance company, working with both Scrum teams and implementing DevOps and SAFe principles since 2018. The organisation is aligned according to the business and technology missions, and divided into thirty-five Agile Release Trains (ARTs). An ART is a self-organising team of Agile teams. Each ART is led by a Release Train Engineer (RTE) who analyses the Agile team performance to develop improvement scenarios for each team. However, the Dutch insurance company has had found difficulties to draw company-wide conclusions and to establish the overall Agile performance.

The insurance company's difficulties are reflected in scientific literature. There is a lack of knowledge related to Agile performance management information systems: no existing system is a good fit for our current, ever-changing software age. This confirms Greening (2015) and Fagerholm et al. (2014) statement that it is difficult to define uniform metrics in changing environments. Both authors also mention a gap in knowledge regarding uniform performance information systems. Most systems focus on a specific methodology in a specific company. In this research, a more comprehensive system will be developed.

This research contributes in two ways to existing literature. First, a literature review on existing Agile maturity and performance systems will be presented. Second, the practical contribution of this research is the provision of an Agile performance information system. This system can be applied in organisations to achieve insight into their Agile performance, and to support the management to lead in the right direction.

Thesis structure

This thesis is organised in several chapters and adheres the following structure:

- Chapter 2 introduces the research design and research methodology, based on Peffers, Tuunanen, Rothenberger, and Chatterjee (2007). Research questions are introduced.
- Chapter 3 provides the theoretical background. Several issues regarding the distinction between performance and productivity, as well as problems related to practical implications of performance management, are described. This is followed by a more extensive literature review in section 3.4, where several Agile systems are discussed.
- Chapter 4 proposes a new Agile Performance management system.
- Chapter 5 discusses the proposed system by professionals and a case study evaluation.
- Chapter 6 answers the research questions as setting up in chapter 2
- Chapter 7 discusses the implications of the system.

2 Research Design

This section starts with defining the design problem and the supporting research questions. Furthermore, the Design Science Research Methodology is described.

2.1 Design problem

The core problem of this research is the lack of a multi-facet performance system, in both literature and practice, and is therefore a design problem. Wieringa (2014) gives a template for design problems in Figure 1.

Figure 1: Template for design problems.
Not all parts to be filled in may be clear at the start of the project

- Improve <a problem context>
- by <(re)designing an artefact>
- that satisfies <some requirements>
- in order to <help stakeholders achieve some goals>

Note. Reprinted from "Wieringa, R. J. (2014). Design science methodology: For information systems and software engineering, pp 16".

Problem Definition

As the popularity of Agile adaption increases, the questions organizations ask shifts from *why* to adapt Agile practices to *how* to implement and scale these practices (Stojanov, Turetken, & Trienekens, 2015). These concerns reach from team level to organisational level. However, it is difficult to find a sufficient performance management information system suitable for all levels. Most existing systems are criticized for applying primarily to small organisations rather than large organisations. This is the critique argued by our case study, the large Dutch insurance company. We will investigate this in more depth from chapter 5 onward.

Another problem is the benchmarking of Agile team performance, which lacks uniform metrics which can be used across Agile teams within organisations. Our case study company mentioned having difficulties getting insight into Agile team performance, and establishing Agile performance improvement scenarios. The company has different combinations of Agile methods and makes it difficult to create a uniform system. McMahon (2015) argued that in such a situation too little time is spent to help the development teams with situations where they need the most help.

Designing an artefact

An artefact is something created by people for some practical purpose (Wieringa, 2014). In software engineering research, there are several examples of artefacts: techniques, methods, systems, conceptual frameworks and even algorithms. Wieringa (2014) mentioned that an artefact is always used by people, and this means that the problem context, along with other things, should involve people. In this research, the artefact is an Agile performance management information system. This system will create more insight into the performance of teams, and will include a mechanism to collect the required information.

Satisfaction of requirements and assist stakeholders

Two groups of persons are affected by treating the problems with a performance information system. First, the development teams and operational teams need a better artefact to determine whether they improve their Agile performance. Additionally, they must learn from each other and be able to create a continuous learning environment. The second group of persons, management, needs an overall overview of teams' Agile transformation. This artefact can help to change the mind of the management and show where they should focus on from their perspective (McMahon, 2015).

The artefact we aim at should be useful for supporting organisations to indicate and improve their Agile performance. The designed artefact should provide a uniform system with metrics to analyse the Agile performance of the Agile Release Trains (ARTs). The use of metrics can only be successful when the organisation, management and the teams use it in a collaborative approach (Ertaban, Sarikaya, & Bagriyanik, 2018). Ertaban et al. (2018) argued that metrics become meaningless in case of misuse.

Figure 2 completes Wieringa's template (2014) with the design problem.

Figure 2: The design problem

- Improve < Agile team performance>
- by <designing a uniform information system for Agile performance management>
- that employs <uniform metrics and can be used across Agile teams within large organisations>
- in order for <development and operational teams to improve their performance and for management to help in decision making>

Note. Adapted from "Wieringa, R. J. (2014). Design science methodology: For information systems and software engineering, pp 16".

An artefact design goal, alternatively, a technical research goal, is to design an artefact that will improve a problem context (Wieringa, 2014). In this research, the artefact design goal is to design a uniform Agile performance management information system to enable organisations to determine and improve their Agile performance.

2.2 Research questions

Now that the design problem and overall goal of this research are clear, research questions to achieve that goal can be defined:

RQ: "*What is an Agile performance management information system that enables organisations to grow in their Agile performance?*"

To develop a suitable Agile performance management information system, it is necessary to divide the research questions into sub-questions. First, it is relevant to review the existing research in the design context. So, the first sub-question is:

SQ1: *"Which Agile performance management information systems are available in the literature?"*

The results of SQ1 will enable us to combine the most important characteristics from literature and to develop a first iteration of the system. This developed system will be used for answering the second sub-question of the usability of the system. When the system is user friendly the users are more likely to use it, resulting in more valuable results. The second sub-question is:

SQ2: "*How can the usability of Agile performance systems be improved?"*

The third question focuses on improving the system further by gathering the opinion from professionals. These professionals are for example Agile managers, coaches or SAFe Program Consultants. The system refinement by involving professionals in the development of the system is critical and can be considered as an important part of the evaluation (Helgesson, Höst, & Weyns, 2012). By asking professionals to imagine critical incidents which occur in practice and to predict what effects they think this would have, the system can be redesigned if required. The third sub-question is:

SQ3: *"How do professionals in the Agile field evaluate the developed Agile performance system and what improvements need to be made based on their evaluation?"*

The last sub-question is related to the adoption of the system within an Agile environment. In this study, research will be done into how stakeholders evaluate the designed system. The fourth question is:

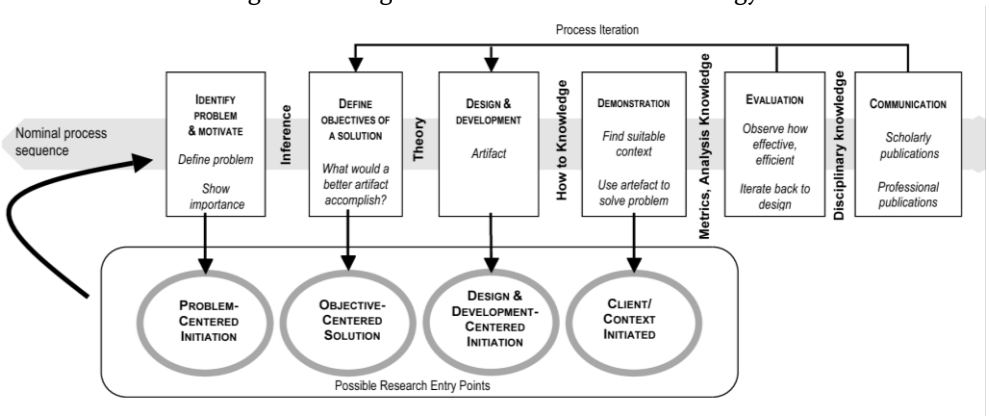
SQ4: *"How do the stakeholders evaluate the utility of the designed Agile performance management information system and what improvements need to be made based on their evaluation?"*

Answering the four sub-questions results in acquiring knowledge about the current systems within the literature and the development of a uniform Agile performance management information system.

2.3 Research methodology

This chapter describes the research design of this study following the Design Science Research Methodology (DSRM) of Peffers et al. (2007) (see Figure 3). The DSRM is divided into six processes: identify the problem and motivate, define objectives of a solution, design and development, demonstration, evaluation, and the communication of the developed performance system.

Figure 3: Design Science Research Methodology



Note. Reprinted from "Peffers, 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)".

- *Identify the problem and motivate:* in this activity, the specific research problem is defined. The design problem definition of this research is yet described in section 2.1.
- *Define objectives for a solution:* the second activity of the DSRM is about inferring the objectives of a solution. These objectives are defined based on the problem definition. Knowledge is required for the definition of objectives to know what the state of problems and current solutions is in the literature. As mentioned, extensive literature research will be done in chapter 3.
- *Design, development and validation:* create the artefact, which is potentially a technique, a method, a conceptual framework and an algorithm. A design research artefact can be any designed object in which the research contribution is embedded in the design. Chapter 4 describes the development of the performance management system of this research. Agile SAFe professionals are involved in the process to validate whether or not the design is appropriate and will perform as expected (Marwedel, 2018).
- *Demonstration:* in this activity, the artefact is used to demonstrate the artefact solves the problem(s) as mentioned in the first activity of the DSRM. This demonstration can be executed in several forms (e.g. case study, proof or experimentation). In this study, the system will be implemented in the earlier mentioned Dutch insurance company.

- *Evaluation:* in this activity, it will be observed to what extent the problem is solved by the artefact. The evaluation will be executed in which a selected panel of researchers gives feedback on the created artefact. When the professionals suggest further improvements of the artefact it can be decided to iterate to the 'design, development and evaluation' activity until the professionals agree. Besides, a user evaluation will be executed using a case study at a Dutch insurance company. The goal of this evaluation is to shape and reshape the artifact by the use context (Sein et al., 2011).
- *Communication*: this study will be published and the administrative mechanism within the Dutch insurance company is communicated.

3 Theoretical background

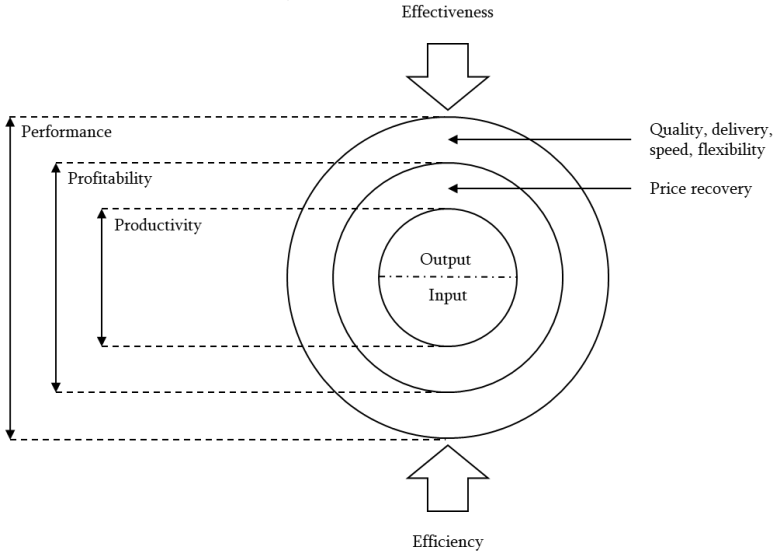
Two topics forming the foundation of this research are discussed. In creating a new performance management system, we must first start defining 'performance' before it is applied to Agile-specific systems. Problems arise regarding defining 'performance' versus 'productivity'. This shall be discussed using the triple-p model by Tangen (2005). Then, we'll look at common problems concerning existing Agile performance systems.

3.1 The triple-p model

The triple-p model by Tangen (2005) is a schematic view of common terms within the field of performance and productivity (Grünberg, 2004; Tangen, 2005) (see Figure 4). According to an older statement of Thomas & Baron (1994), many researchers claim to be discussing productivity but are looking at performance instead. The triple-p model resolves this issue by distinguishing the two terms. Even though productivity is a multidimensional term, it is the ratio between input and output (Tangen, 2005). Performance, on the other hand, is the umbrella term for all concepts related to the success of a company and its activities (Tangen, 2005). It includes almost any objective of competition related to flexibility, speed, dependability and quality (Tangen, 2005), and is represented in the outer edge of the triple-p model. Slack, Chambers, Johnston, & Betts (2012) offer the five external effects of performance objectives, as shown in Figure 5.

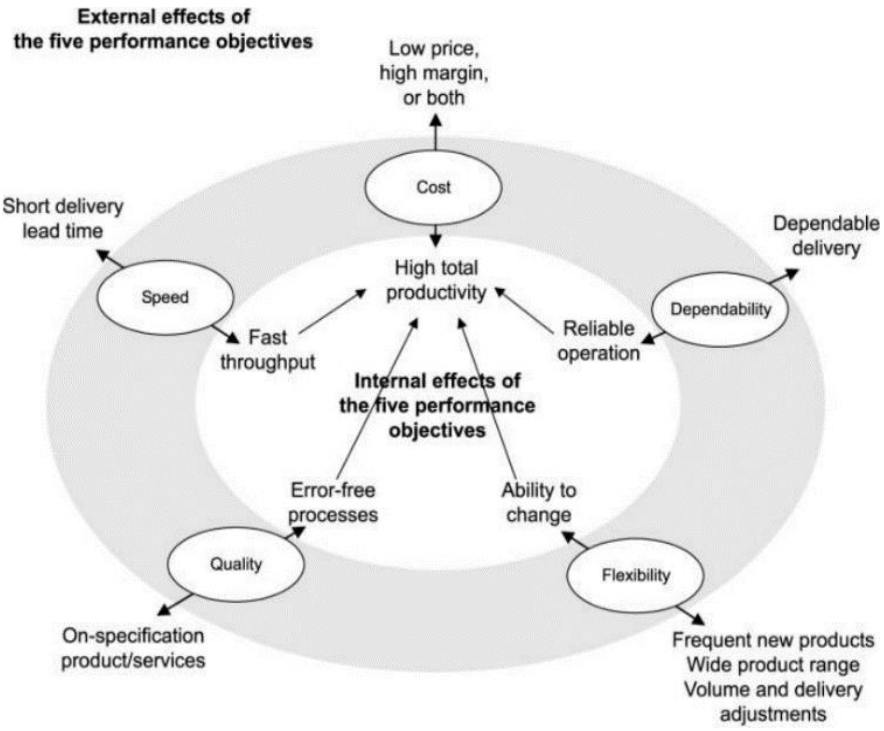
Tangen (2005) explains that these terms are commonly interpreted in various ways. A mistake often made by people is to interchange not only terms as performance and productivity but also terms such as effectivity, efficiency and profitability (Jackson & Petersson, 1999; Tangen, 2005).

Figure 4: Triple-p model



Note. Adapted from "Tangen, S. (2005). Demystifying productivity and performance. International Journal of Productivity and Performance Management, 54(1)".

Figure 5: External effects of the five performance objectives



Note. Reprinted from "Slack, N., Chambers, S., Johnston, R., & Betts, A. (2012). Operations and Process Management. Operations Management".

We will for now abandon the triple-p model and focus on this chapter's second topic: issues regarding Agile performance management.

3.2 Literature review on Agile performance management

This literature review shows that there are still scientific and practical problems of conceptualizing Agile team performance. Each of the following sub-sections categorizes a concept in which these difficulties occur: indicators, teamwork and improvement. These indicators were most commonly mentioned in the literature reviewed. The search protocol of this literature review is shown in Appendix A.

3.2.1 Indicators

Performance indicators can be defined as physical values to manage and measure organizational performance (Gosselin, 2005). Agile team performance indicators can only be used successfully when the organisation, management and the teams use it collaboratively (Ertaban et al., 2018). Ertaban et al. (2018) argued that indicators become meaningless in the case of miss-use. Some examples of miss-use are becoming too strict on minor changes, too much focus on the numbers of the indicators and finally, stakeholders inflating indicators by intuition (Ertaban et al., 2018).

Furthermore, defining and estimating these indicators in changing environments is a challenge itself. Once you have defined and estimated an indicator, the organization might have changed (Fagerholm et al., 2014; Greening, 2015).

If the objective of these indicators is to analyse the development teams, human factors should be included, like skills and how engaged the employees are with the organisation (Fagerholm et al., 2014).

Kupiainen, Mäntylä, & Itkonen (2015) literature review shows that Agile teams are most of all using indicators suggested by the Agile literature (e.g. velocity, effort estimate, work in progress). Software developers are using these indicators to plan and track the progress of their project, the teams care about quality and improve their processes (Kupiainen et al., 2015). They argue that indicators in Agile development are focused on the products and the processes, and not on people.

McMahon (2015) has a framework for development teams, regardless of the method, practice or lifecycle a team is using. McMahon (2015) argues that this framework is an essential tool for effective and efficient software engineering. This framework called "Essence" has seven key project success elements: opportunity, stakeholders, requirements, software system, work, way of working and team (McMahon, 2015). The problem commonly observed in past performance improvement systems is an over-focus on work product and causing to fall off their goals related to performance (McMahon, 2015).

So, instead of only looking at the product, organisations should also focus on the knowledge, skills and behaviour of the employees.

3.2.2 Teamwork

In most organisations, Agile performance management is focused on the individuals in a team (Ertaban et al., 2018). This is a common mistake because the main goal of Agile performance management information systems is on teamwork and teams' common results (Ertaban et al., 2018). The emergence of Agile methodologies created an interest in more team-related elements, such as communication, coordination and self-managing teams (Forsyth, 2006).

Overall project performance is related to the teamwork productivity in an Agile environment (Fatema & Sakib, 2018). Productive teamwork requires a certain unity in norms (Fagerholm et al., 2014). Norms are the standards that regulate team members behaviours (Forsyth, 2006). Forsyth (2006) supports the concept that norms enable team performance, but simultaneously other norms hinder team performance (Fagerholm et al., 2014). To encourage productive behaviour of team members, Fagerholm et al. (2014) suggested that teams should reflect on both norms: injunctive norms and descriptive norms. Injunctive norms are what is approved or disapproved behaviour and descriptive norms are what is commonly done.

3.2.3 Improvement

A component that applies to the improvement of team performance is the overall understanding of the experience of performance. To strive for the better, the teams need to make experiments, get feedback, act and improve (Ertaban et al., 2018). According to Fritze (2016) improvement is "anything that makes the current situation better and continuous improvement is making small change collaboratively to reach a more efficient and effective state". Agile methods provide frameworks to solve issues, but it is hard to identify where the teams should look for issues, and to monitor how teams perform in addressing issues and achieving their goals (McMahon, 2015).

3.3 Literature review on Agile performance management systems

In this section, unlike former sections, several scientific papers will be compared. Therefore, a systemic review according to the principles of the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) (Moher et al., 2016). This PRISMA-statement is based on four different principles: identification, screening, eligibility and finally inclusion and exclusion criteria. For the literature protocol of the literature review, see Appendix B. The literature review resulted in eleven research papers which were selected for this research. These eleven papers describe performance management systems and are summarized in Appendix C.

In Table 1 the systems proposed in the selected papers are analysed according to the capabilities presented by Maier, Moultrie and Clarkson (2012) for evaluating existing systems. These capabilities are related to a roadmap and can be used by researchers for evaluative purposes of existing systems.

Capabilities

When developing a system, the first important decision is to specify the audience. It refers to the stakeholders of the system, who will participate in the system. The goal of the design should be that some of these stakeholders are better off when the design problem is treated by developing an artefact (Wieringa, 2014). In some literature, it is specifically made to reach typical organizations (Ambler & Lines, 2016), while in other literature they intend to reach a specific team (Fontana, Reinehr, & Malucelli, 2015).

Besides, the author should be clear if the system is designed to be a generic system or more specific to a domain. After identifying the audience and scope, a system can be evaluated based on the 'improvement paradigms': analytic and benchmarking (Emam & Goldenson, 2000). The analytic paradigm aims to determine the required improvements and if the suggested improvements have been successful. In contradiction to benchmarking, which is aimed at identifying the best practices and organizations in which the practices are compared to.

Thereafter, a system can be evaluated based on the success criteria. These criteria are required to determine whether the development and application of the system were successful. These criteria appear in the form of requirements. The usability is an example of such criteria and addresses the degree whether the users of the system understand the concepts and processes of it.

Maier et al. (2012) defined the selection of the processes which are assessed in the system as the conceptual framework and this should be generated from the principle of good practice and established knowledge (Chiesa, Coughlan, & Voss, 1996). The starting point for defining the key processes can be based on different options. Some options are available to provide a theoretical starting point and justification of process areas, but it depends on the involved stakeholders (e.g. concepts in literature or interviewing professionals).

The next step in the evaluation of a system is based on the number of maturity levels which are selected and what it is based on. In the literature, a different number of maturity levels have been chosen by the authors of these systems. Maier et al. (2012) say: “levels need to be distinct, well defined, and need to show a logical progression as clear definition eases the interpretation of results”. To discriminate between these maturity levels, it is required to describe each of the process characteristics in each defined level. The formulation of these behavioural characteristics for capabilities or growth scenarios is on the important decisions in the development of a maturity system and is defined as the cell texts.

The last capability is related to the administration mechanism of the system. It is about the delivery method which is connected to the aim of the assessment. Approaches with an analytic paradigm more often focus on raising awareness and choose for interviews or group workshops. In contrast to benchmarking paradigms which prefer electronic system for questionnaires and focus on results.

Table 1: Capabilities of literature on maturity systems

Title short	Audience	Aim	Scope	Success criteria	Conceptual foundation	Maturity levels	Cell texts	Administration
Implementing CMMI Project Management … (Soares & De Lemos Meira, 2015)	Organizations with CMMI and Agile	Analysis	Agile project management	No evidence	CMMI and Agile.	5 levels	Processes briefly described.	No evidence
Scaling Agile Development (Stojanov et al., 2015)	Organizations scaling Agile	Analysis	SAFe	Delphi study	Sidky's model and SAFe.	5 levels	Online document	Exemplifies how to perform the assessment
Agile Compass (Fontana, Reinehr, et al., 2015)	Agile teams	Analysis	Agile in general	No evidence	Outcomes.	No levels	Checklist	A checklist as an assessment method
Maturing in Agile … (Fontana, Meyer, et al., 2015)	Agile teams	Analysis	Agile in general	No evidence	Practices and outcomes.	No levels	Briefly described	No evidence
A maturity model for IT-based case management (Koehler, Woodtly, & Hofstetter, 2015)	Organizations with CMS and Agile	Analysis	Agile case management	No evidence	CMMI and BPM	5 levels	Processes and levels	No evidence
DevOps (Mohamed, 2015)	DevOps organisations	Analysis	DevOps	No evidence.	CMII and the HP model.	5 levels	Processes and levels in detail	A transformation framework
Application Lifecycle Automation (Menzel, 2015)	DevOps organisations	Analysis	DevOps	No evidence.	No evidence.	5 levels	Processes and levels	No evidence
DevOps Quick Guides (Eficode, 2015)	DevOps organisations	Analysis	DevOps	No evidence.	No evidence.	4 levels	Levels briefly described	No evidence
An approach to Agile maturity (Ambler & Lines, 2016)	Organizations with CMMI and Agile	Analysis	Agile project management	Examples of practice	Empirical observations of Agile/Lean teams.	No levels	Three strategies.	No evidence.
DevOps Adoption (Bucena & Kirikova, 2017)	DevOps organisations	Analysis	DevOps	Tested at an international company.	Analysis of related work.	5 levels	Processes and levels in detail.	An online questionnaire is available.
DevOps competencies and maturity (de Feijter et al., 2018)	DevOps organisations	Analysis	DevOps	Executed at Centric.	Validated perspectives, focus areas and capabilities	10 levels	Processes and levels.	An assessment tool is developed.

*Note. A*dapted from "Maier, A. M., Moultrie, J., & Clarkson, P. J. (2012). Assessing organizational capabilities: Reviewing and guiding the development of maturity grids. IEEE Transactions on Engineering Management, 59(1)".

3.4 Agile systems within the literature

The selected papers provide multiple systems. Three systems are chosen for a further description because of several reasons.

First, these systems all have a different scope. It was mentioned in the introduction of this paper that organisations found difficulties to establish the overall Agile performance when they are working according to the principles of different methodologies. To get a broad picture of systems with a different scope it was chosen to describe three systems which are specified to different methodologies. Based on the literature review, the Agile Compass of Fontana, Reinehr, et al. (2015) was chosen, because the authors didn't define any levels and it was the only system of which the authors included an administration mechanism. Subsequently, only one system related to SAFe emerged, therefore the SAFe Maturity Model of Stojanov et al. (2015) was chosen. Finally, the choice fell on the DevOps Maturity Model of de Feijter et al. (2018) because this was the most recent system from the literature study and de Feijter et al. (2018) were the only authors who defined ten maturity levels.

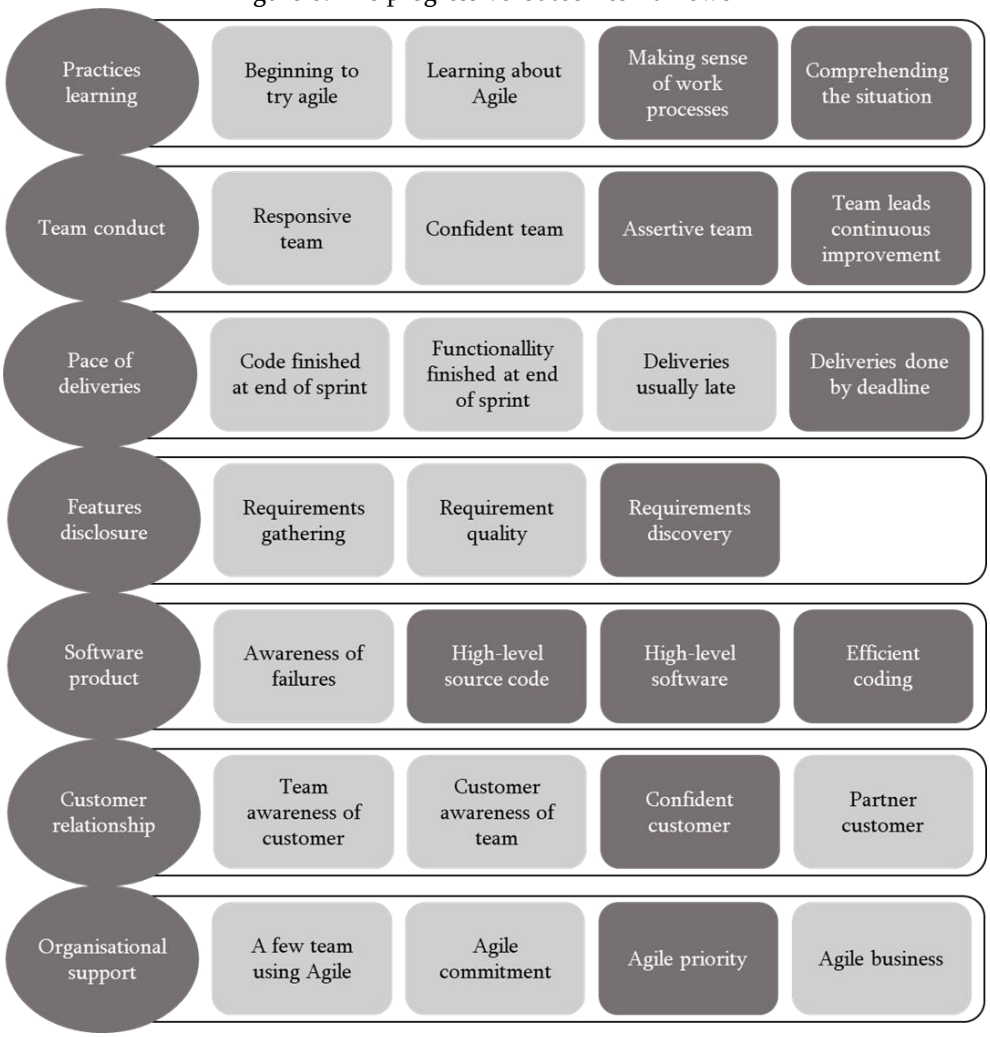
3.4.1 The Agile Compass

The Agile Compass written by Fontana et al. (2015) is a tool for identifying Maturity in Agile Software-Development Teams. Fontana et al. (2015) found that Agile teams accomplished their maturity via a dynamic evolution based on the pursuit of specific outcomes instead of reaching a specific maturity level.

Fontana et al. (2015) analysed nine software development teams' evolutions. The interview data of twenty-five Agile practitioners were analysed using content analysis and they created networks of codes identifying the practices applied in the past, present and future. They cross-analysed the network of codes which resulted in the Progressive-outcomes framework in Figure 6.

The model consists of seven outcome categories: practices learning, team conduct, the pace of deliveries, features disclosure, software product, customer relationship and organisational support. There wasn't enough evidence that teams either pursue all outcomes to follow a predefined sequence of outcomes in the maturing process. However, they found evidence of the achieved outcomes of mature teams, which appear in the coloured boxes. By using this framework, they don't guarantee a final, definitive picture of a software team's progress toward maturity because the situation can change over time. For example, when a team member is leaving, the team might have to start pursuing the progressive-outcome framework again. As the environment changes, the management observes the development teams and should make the necessary improvements to enable continuous improvement.

Figure 6: The progressive-outcomes framework



Note. Adapted from "Fontana, R. M., Reinehr, S., & Malucelli, A. (2015). Agile Compass: A Tool for Identifying Maturity in Agile Software-Development Teams. IEEE Software, 32(6)". The circles represent outcome categories. The boxes are development outcomes. The coloured boxes are development outcomes that indicate team maturity.

In the Agile Compass, Fontana et al. (2015) consider maturity from an Agile perspective. They do not prescribe any practice to reach maturity. The purpose of the analysis was to improve the processes of teamwork in Agile development teams. It seems to be a limited model in which the seven outcome categories are identified, and few details are given. Additionally, Fontana et al. (2015) mentioned that there is a lack of evidence that teams pursue all outcomes or follow a predefined sequence of outcomes to become mature. While at the same time, the framework seems to indicate a team must follow a certain flow.

3.4.2 A SAFe Maturity Model

In recent years the Agile software development approaches have gained wide acceptance in practice. But at the same time, challenges exist on the scalability and integration of Agile practices in large-scale software development environments (Stojanov et al., 2015). The Scaled Agile Framework (SAFe) is a solution to address some of these challenges, but despite some encouraging results, Stojanov et al. (2015) argued that there are still several challenges related to SAFe adoption. Stojanov et al. (2015) maturity model address the need for assessing the progress of SAFe adoption. The authors extended the Sidky Agile measurement index (SAMI) model by Sidky, Arthur, & Bohner (2007) with practices key to SAFe.

The SAMI model gives five levels from the four values of the Agile Manifesto (Hazzan & Dubinsky, 2014). The levels are defined as collaborative, evolutionary, effective, adaptive and encompassing. Additionally, the model clusters the Agile principles of the Manifesto into categories: embrace change to deliver customer value, plan and deliver software frequently, human centricity, technical excellence, customer collaboration (see Table 2). These categories group Agile practices in levels. The practices are techniques or methods which are used for developing software consistent with the principles of Agile (Stojanov et al., 2015). The model is developed in a way that organisations should implement the practices on lower levels first because the practices on a higher level are dependent on the practices at these lower levels. For example, within the Agile principle 'Plan and deliver software frequently': before an organisation can deliver continuously, collaborative planning is required.

Based on a design science research approach Stojanov et al. (2015) developed a new software engineering artefact, the SAFe maturity model (SAFe MM). By reviewing the Agile practices and evaluating this model in combination with the SAFe practices it resulted in an adaption of the Agile practices to address the team level practices.

Using a Delphi study, the initial SAFe MM model was reviewed and refined. Based on the proposed refinements and changes gathered in this study, several alternations were made in the model. By defining and populating the practices in the final version model a set of governing rules were applied. First, all the practices should contribute to the achievement of the maturity level in which they are positioned. Secondly, the relevancy of the practices concerning Agile principles should be ensured. The last rule is concerned with the positioning of the practices at a specific level. The SAFe MM is characterized as a descriptive model because it describes only at a level which Agile or SAFe practice is essential. Table 3 gives the levels, principles and practices of the final model.

Table 2: The SAMI Model

	Agile Principles				
Agile levels	**Embrace change to deliver customer value**	**Plan and deliver software frequently**	**Human Centricity**	**Technical Excellence**	**Customer collaboration**
Level 5 Encompassing	Low process ceremony	Agile project estimation	Ideal Agile physical setup	Test-driven development Paired programming No level -1 or 1b people in a team	Frequent face-to-face interaction between developers and users (collocated)
Level 4 Adaptive	Client driven iterations Continuous customer satisfaction	Smaller and more frequent releases Adaptive planning		Daily progress meetings User stories Agile documentation	Customer immediately accessible Customer contract around commitment of collaboration
Level 3 Effective		Risk driven iterations Plan features not tasks Backlog	Self-organizing teams Frequent face to face communication	Continuous integration Continuous improvement Unit tests 30% of level 2 and level 3 people	
Level 2 Evolutionary	Evolutionary requirements	Continuous delivery Planning at different levels		Software configuration management	Customer contract reflective of evolutionary development
Level 1 Collaborative	Reflect and tune the process	Collaborative planning	Empowered and motivated teams Collaborative teams	Coding standards	Customer commitment to work with the development team

Note. Adapted from "Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting Agile practices: The Agile adoption framework. Innovations in Systems and Software Engineering, 3(3)".

Table 3: The SAFe Maturity Model

	Agile Principles				
Agile levels	**Embrace change to deliver customer value**	**Plan and deliver software frequently**	**Human Centricity**	**Technical Excellence**	**Customer collaboration**
Level 5 Encompassing	L5P1: Low process ceremony L5P2: Continuous SAFe Capability Improvement	L5P3: Agile Project estimation	L5P4: Ideal Agile physical setup L5P5: Changing an organization	L5P6: Test-driven development L5P7: No/minimal number of level -1 or 1b people in a team L5P8: Concurrent testing	L5P9: Frequent face-to-face interaction between developers and user (collocated)
Level 4 Adaptive	L4P1: Client driver iterations L4P2: Continuous customer satisfaction L4P3: Lean requirements at scale	L4P4: Smaller and more frequent releases L4P5: Adaptive planning L4P6: Measuring business performance	L4P7: Managing highly distributed teams	L4P8: Intentional architecture L4P9: Daily progress tracking meetings	L4P10: CRACK customer immediately accessible L4P11: Customer contract revolves around the commitment of collaboration
Level 3 Effective	L3P1: Regular reflection and adaption	L3P2: Risk driven iterations L3P3: Plan features not tasks L3P4: Roadmap L3P5: Mastering the iteration L3P6: Software Kanban Systems L3P7: PSI/Release L3P8: Agile Release Train	L3P9: Self-organizing teams L3P10: Frequent face to face communication L3P11: Scrum of Scrum	L3P12: Continuous integration L3P13: Continuous improvement (refactoring) L3P14: Unit tests L3P15: 30% of level 2 and level 3 people	L3P16: DevOps (Integrated Development and Operations) L3P17: Vision, features L3P18: Impact on customers and operations
Level 2 Evolutionary	L2P1: Evolutionary requirements L2P2: Smaller, more frequent releases L2P3: Requirements discovery	L2P4: Continuous Delivery *L2P5: Two-level planning and tracking* L2P6: Agile Estimating and Velocity L2P7: Release planning	L2P8: Define/ Build/ Test team	L2P9: Software configuration management L2P10: Automated testing L2P11: Tracking iteration progress L2P12: No big design upfront (BDUF) *L2P13: Product Backlog*	L2P14: Customer contract reflective of evolutionary development
Level 1 Collaborative	L1P1: Reflect and tune the process	L1P2: Collaborative planning	L1P3: Empowered and motivated teams L1P4: Collaborative teams	L1P5: Coding standards L1P6: Knowledge sharing L1P7: Task volunteering L1P8: Acceptance testing	L1P9: User stories L1P10: Customer commitment to work with development teams

Note. Adapted from "Stojanov, I., Turetken, O., & Trienekens, J. J. M. (2015). A Maturity Model for Scaling Agile Development". The underlined practices are introduced and evaluated with the Delphi study. The italic practices are altered in the current model.

The SAFe MM is based on the SAMI model which is fully constructed according to the Agile levels, principles, practices and concepts, which makes the model more valuable and applicable on a larger scale. The authors indicated the dependencies between practices of different levels. They assume that it is a linear process in achieving the highest level. A limitation of this research is that it is related to a single case organisation, further case studies are required to assess the completeness and effectiveness of the model. However, SAFe 5.0 is published and this requires that this model should be updated with the current version of SAFe.

3.4.3 A DevOps Maturity Model

De Feijter et al., (2018) constructed a DevOps competence model (see Figure 7). This model is an improved model of an earlier version. To collect the drivers and practices of DevOps, de Feijter et al. (2018) initiated a literature review and semi-structured interviews were held at Centric, ICTU and Microsoft. Through different rounds with professionals, the focus areas and capabilities were validated. The model is dived into two components: the DevOps Competence model and the DevOps maturity model.

3.4.3.1 DevOps Competence model

From interview data and literature de Feijter et al. (2017) detected six DevOps drivers which are essential for aiming to adopt DevOps to a mature extent.

The first driver is *a culture of collaborating*. In traditional organizations, departments used to work separately ('silos'), which resulted in little to no communication between stakeholders (Sydor, 2014). DevOps aims to diminish these silos and promotes communication, collaboration and integration among the parties engaged.

The second driver is *Agility and process alignment*. DevOps creates an environment in which the stakeholders are under the same process to improve process alignment. Additionally, the alignment with the customers is an area which DevOps focuses on.

The third driver; *Automation*. DevOps drives the automation of tasks in for example building, testing and deployment. A reason for automating tasks is diminishing the error rate when compared to manually performing tasks.

Higher quality is the fourth driver emerging from the interviews and literature. DevOps contributes to enhancing process quality to detect errors early in the process and aims to complies with the customers' needs.

Development and deployment of cloud-based applications is the fifth driver of the model. Organisations are looking for possibilities to migrate to Software as a Service (SaaS). SaaS often encompasses a web-based delivery and is an environment in which the software provider runs and maintains the hardware and software and the customer make use of it through the internet (Choudhary, 2007).

The last driver detected is *Continuous improvement*. It aims at integrating measurement and monitoring to release faster and by identifying the performance bottlenecks.

The DevOps competence model consists of three perspectives (Rico de Feijter, Overbeek, van Vliet, Jagroep, & Brinkkemper, 2017). Continuous improvement is presented in all perspectives since the aim of the content within these perspectives is to improve (Rico de Feijter et al., 2017). The three perspectives:

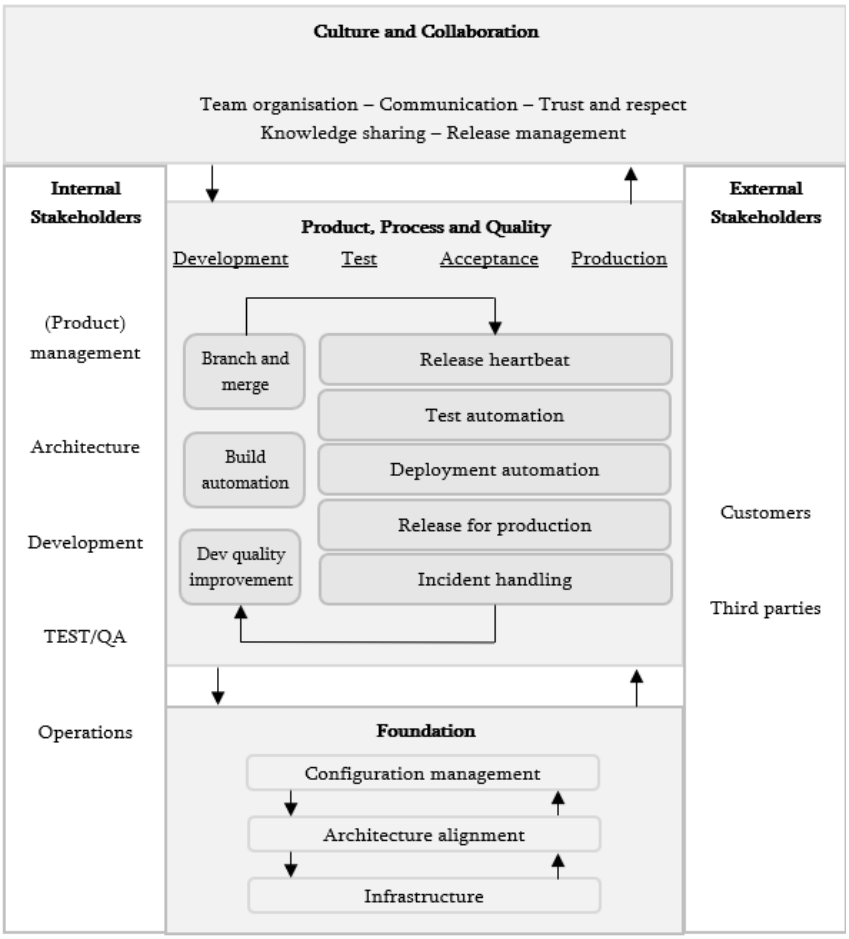
- Culture and Collaboration: organisations should be able to perform and the interdisciplinary people within this organisation should communicate and collaborate to deploy the developed product on time.
- Product, Process and Quality: the process of releasing a product, and the feedback loop for improving the quality. It is related to the value chain for operating in an industry to deliver a valuable product.
- Foundation: the perspective which is focused on the process from development and production and the aim of supporting the process of releasing a product.

The culture and collaboration perspective is focussed on knowledge sharing, trust, respect and communication. These areas mostly aim to create a culture of collaboration (Lwakatare, Kuvaja & Oivo, 2015). Additionally, it is important for agility and process alignment but there is a lack of evidence in the paper of de Feijter et al. (2017) to argue this.

The focus of the product, process and quality perspective is related to agility and process alignment, automation, higher quality and continuous improvement (de Feijter et al., 2017). Build automation is aimed at automation of processes and higher quality to build software as frequently as possible. Quality improvement includes higher quality and continuous improvement of the whole process to release a software product. Additionally, deployment automation includes the development of cloud-based applications. The continuous deployment phenomenon is often seen with software as a service (Bosch, 2014).

The foundation perspective includes agility and process alignment, automation, higher quality, development and deployment of cloud-based applications. Because the perspective is focused on a wide range of focus areas from the development of a product until the release of a product it's affected by such a wide group of drivers. Based on the perspectives and drivers, a DevOps competence model was constructed by de Feijter et al. (2018) in Figure 7.

Figure 7: The DevOps competence model



Note. Adapted from "de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018)".

3.4.3.2 Maturity model

The culture and collaboration perspective covers the software part of DevOps. An organisation itself should be in place to perform the work and for this, it is required that interdisciplinary professionals communicate, share knowledge, have trust and respect for one another, work in teams, and there should be some form of alignment (Rico de Feijter et al., 2018).

The product, process and quality perspective is about the process of releasing a product and feedback loops. The perspective is representing the DTAP-street: development, testing, acceptance and production (Heitlager, Jansen, Helms, & Brinkkemper, 2006). Within this section, there are multiple focus areas represented in the maturity model (see Figure 8). The relations in the competence model shows that this perspective relates in different ways to the culture and collaboration perspective and the other way around. When for example an incident comes in, different stakeholders within the company (e.g. product management, testers, operations) should communicate such that the fix can be put into production again.

The last perspective is about the foundation and encompasses the configuration management, architecture alignment and infrastructure as focus areas. These areas stretch from development to production and aim to support the process depicted in the product, process and quality perspective. It has been chosen to stretch these because all environment configuration items (e.g. middleware, database versions etc.) should be managed. Additionally, the software architecture is required for each environment and is concerned into this focus area. Thirdly, the infrastructure is an important focus area of the foundation because it mirrors all environments (Humble & Farley, 2010).

Figure 8: The DevOps maturity model

Focus area \ Level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Culture and collaboration (CC)											
Communication		A				B	C			D	E
Knowledge sharing				A		B	C				D
Trust and respect							A	B	C		
Team organisation		A	B						C	D	
Release alignment				A					B	C	
Product, Process and Quality (PPQ)											
Release heartbeat		A				B	C		D	E	F
Branch and merge			A	B		C		D			
Build automation			A	B		C					
Develop quality improvement			A			B		C	D	E	
Test automation				A	B	C			D		E
Deployment automation					A	B					D
Release for production					A			B	C	D	
Incident handling			A					B	C	D	
Foundation (F)											
Configuration management			A	B		C					
Architecture alignment			A						B		
Infrastructure				A				B	C	D	

Note. Reprinted from "de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018)". The letters (a, b, c and d) in the cells represent the capabilities detected in the study. The capabilities are positioned in increasing order of maturity.

In contradiction to the first two models, the model of de Feijter et al. (2018) is completely devoted to DevOps. The DevOps competence model drivers and perspectives are researched based on extensive literature research and interviews, but at the same time, there is a lack of evidence for the different focus areas. Also, the authors assume Agile growth looks the same across organisations, teams, and individuals. It looks like the context doesn't affect the focus areas, the phases or in what order they occur. The positioning of the capabilities and the determination of the ten levels shows that the authors consider the fact that not all focus areas reach a certain maturity level at the same time.

3.5 Discussion

The three models discussed in the previous sections approach the maturity of Agile methodologies from different dimensions, maturity levels and improvement scenarios.

3.5.1 Dimensions

The dimensions in which the practices, perspectives, principles or other synonyms used for the division of the systems are comparable. They can be subdivided into three categories: people, product and process, organisational culture. For example, the Agile Compass, the first two outcomes (Practices learning and team conduct) are related to people. Secondly, the pace of deliveries, features disclosure and software product are related to product and process. Finally, customer relationship and organizational support are related to the stakeholders of the organization. When looking at the DevOps maturity system the product and process part consists of the foundation, and the product, process and quality part. The third perspective, culture and collaboration, is focused on the stakeholders and the people. So, in terms of dimensions, the focus areas look at other practices or principles, but still, it is possible to categorize them.

3.5.2 Maturity levels

Even though the models have different stakeholders, who participate in various aspects of the assessment, the number of maturity levels does not seem to depend on it. Three models do not even define levels (Agile), one model has four levels (DevOps), six models have five levels (SAFe, Agile, DevOps) and finally, only one model has ten levels (DevOps). Organisations need to work out how much a possible improvement, to reach a higher level, will cost and what benefit this improvement will deliver the organisation (Humble & Farley, 2010).

3.5.3 Improvement scenarios

The maturity models are generic, and no step by step guide is incorporated that shows a maturity growth path (van Steenbergen, Bos, Brinkkemper, van de Weerd, & Bekkers, 2013). To overcome this, a model presented by van Steenbergen et al. (2013) consists of focus areas corresponding to the model by de Feijter et al. (2017). The Software Product Management (SPM) maturity model from Bekkers, van de Weerd, Spruit and Brinkkemper (2010) is a model which illustrates focus areas and capabilities. Such models provide better guidance on how to become more mature in a specific domain (Rico de Feijter et al., 2018). Additionally, maturity models with predefined levels imply that growth is a linear progression through some discrete phases (Verwijs, 2019). However, when you take a closer look at these maturity models, it is likely teams on different levels operate in different ways. The organisation first needs to improve the underdeveloped principles to reach a higher maturity level (Rico de Feijter et al., 2018). So, the current systems do not provide much clarity on how the maturity of such a team should be interpreted.

3.5.4 Information systems

An information system is as a set of components that work together to collect information to support decision making analysis and control (Bourgeois, Mortati, Wang, & Smith, 2019). However, within the maturity models, there is a lack of information systems. The models specify levels and the related practices or capabilities which are required to reach a level, but it isn't mentioned how they collect the information. In the capabilities overview of Maier et al. (2012), it is mentioned that the administrative mechanism of collecting the information is one of the capabilities for the development of a system. However, only five papers have mentioned a methodology for collecting the required information.

In conclusion, the literature review has made clear which systems are available within the literature, and which gaps still exist. To recap, the most relevant issues in regards to this research are:

- Teams on different levels operate in different ways and there is a lack of guidance to define a growth path.
- Current systems within the literature do not provide clarity on how the performance of teams should be interpreted.
- Lack of evidence about information systems for collecting the required information.

The next chapter will use these elements to develop a new Agile performance management system.

4 Development of the performance management information system

After the literature reviews in the previous chapter on performance management and the existing Agile systems within the literature, a starting point was created to develop a uniform performance management information system. The development of this performance management system will take the competencies of the participants and the outcomes of the work product as indicators.

4.1 Core competencies

As mentioned in the introduction of this chapter, the system should include core competencies. Core competencies are not seen as individual-based learning or skill but should be seen as collective learning in the organisation or project (Gallon, Stillman, & Coates, 1995). These competencies are reviewed in the system as leading indicators of the Agile team performance.

Leading indicators are prediction-based indicators and are considered drivers by Zheng et al. (2019). Prediction-based project performance is forward-looking, representing project expectations (Eilat, Golany, & Shtub, 2008). This type of indicators helps the management to focus on the right direction (Zheng et al., 2019).

4.1.1 SAFe 5.0 core competencies

Within Agile methodologies, the emphasis is mainly on personal factors such as talent and skills. The individual development of the participants within an Agile project is important because this ensures that people can add more value in the future (Cockburn & Highsmith, 2001). If the people on the project are functioning to a certain standard, they can use almost any process and deliver (Cockburn & Highsmith, 2001).

As mentioned in the introduction there is a lack of systems in which competencies are considered. This is also reflected in the SAFe Maturity Model of Stojanov et al. (2015), which is focused on Agile and SAFe practices. Competencies are not mentioned. However, the SAFe 5.0 framework is built around core competencies which are essential for developing products that deliver value to the customer (Danilovic & Leisner, 2007). A competency is a combination of complementary skills and knowledge embedded in a group to achieve business agility and providing a superior product (Coyne, Hall, & Clifford, 1997). The core competencies, based on the SAFe 5.0 framework, which are included:

- Agile Product Delivery: related to defining, building and releasing a continuous flow of products and services to the market. It enables the organisation to provide solutions for the market with lower development costs and to delight the customer.

- Lean-Agile Leadership: how leaders within a Lean-Agile organisation can drive organisational change and how to get the most out of the potential of others. The leaders do this by adopting an Agile mindset and to take a leadership role in changing the organisation in a new way of working.
- Team and technical agility: focus on the Agile practices and skills which are essential to delivering high-quality solutions for the customers of the organisation.
- Continuous learning culture: describes values and practices to continually increase the knowledge, competence, performance and innovation within the organisation. This is in line with the purpose of the performance management information system to get insight into the performance of teams.

When these core competencies are applied to the organisation it results in increased productivity, predictable delivery of value, faster time-to-market, and engaged employees (Scaled Agile, 2020) (see Appendix D).

4.1.2 Competencies assessments

As mentioned, the core competencies are sets of related knowledge, skills, and behaviours which are essential for the participants within Agile projects. To assess those competencies, Scaled Agile provides assessments of statements which represents those practices, skills and principles related to each core competencies. By using a Likert scale the assessments provide a great volume of reliable data into those practices and skills (Cooper & Schindler, 2014). Based on the results of these assessments the organisations can look for recommended improvement opportunities that support the mastery in each competency.

Thus, when a team wants to obtain insights into the core competencies, the statements within those assessments can be used to gain insight into the leading indicators of the performance management system.

4.2 Performance indicators

To efficiently and effectively analyse the Agile team performance, organisations should make use of performance indicators which are easy to measure, understandable, and which are the most relevant for achieving the business objectives (Choong, 2013). This type of indicators is known as lagging indicators.

In business performance management lagging indicators communicate the outcome of a past action of practice. A balance between the leading and lagging indicators results in enhanced business performance overall. If there exists such a balance between these indicators, the practices are in place and this results in the right outcomes (Zheng et al., 2019). The indicators to measure these outcomes are divided into the development outcomes of SAFe.

4.2.1 SAFe 5.0 development outcomes

The Scaled Agile framework executed fifty-one case studies which identify the most common development outcomes by implementing Agile methodologies. In Appendix D the development outcomes of four case studies are mentioned. The outcomes can be categorized into an increase in productivity, a faster time-to-market, improved quality and more engaged employees. These outcomes are also recognized in the research of Rico (2008). Rico (2008) points out that on average, the studies of several Agile methodologies have reported 91% better schedule, 97% improved productivity, quality improvement of 50% and four times better satisfaction of employees. These outcomes correspond to the triple-p model of Tangen (2005) and the external effects of Slack et al. (2012) in which those four outcomes are represented as the components of performance.

One of the key questions when establishing lagging indicators for a project performance management information system is to decide *what* will be measured (Zheng et al., 2019). It has been recognized that inadequate project performance indicators provide inappropriate information for decision making, resulting in poor project results (Kawalek & Wastell, 2008). In the following section, the indicators for each development outcome are explained.

4.2.1.1 Productivity

One of the development outcomes of SAFe 5.0 is an increase in productivity by adopting SAFe and applying the practices related to the Core Competencies of the Lean Enterprise. As mentioned by Tangen (2005), the definitions of terms frequently used within the field of productivity and performance management is confusing. In the triple-p model of Tangen (2005), productivity is defined as the relation between correctly produced products which fulfil their specifications and the resources that are consumed in the transformation process. This definition corresponds to the definition of Moseng and Rolstadas (2001) who define productivity as the ability to satisfy the market's need for goods and services with a minimum of total resource consumption.

To measure the productivity in an Agile environment it is chosen to make use of business value points compared to the Agile Release Train (ART) capacity. This is in line with the definition of Tangen (2005) in which the business value points are the output-quantity, and the ART capacity is a resource which is used to deliver these business value points. These points are estimated at the start of the development process, the most common is the start of the program increment (PI). Which is a timeboxed planning interval during which an ART plans and delivers. The capacity of the ART is the costs related to the ART during the coming PI. This is not only costs related to their salary, but for example also the costs of certifications.

By looking at the growth in the ratio between these components, it becomes clear if the ART can deliver more value with fewer resources in a PI, and so the ART grows in their productivity.

4.2.1.2 Time-to-market

The second development outcome in the SAFe Framework is the time-to-market and this refers to the time from which a company initially conceives a product or service idea to the point when the actual product or service is accessible to buyers in the market (Afonso, Nunes, Paisana, & Braga, 2008).

To get more insight in the time-to-market, the indicators are the lead and cycle time of the development process. First, the lead time measures the time it takes from receiving the order to develop the feature to the final release of the product or service. The cycle time, on the other hand, is focussed on the start of coding the software to the release of the software and therefore is a subset of the lead time.

4.2.1.3 Quality

According to the IEEE Standard Glossary of Software Engineering Terminology, software quality is defined as: "the degree to which a system, component or process meets specified requirements" (P. Miguel, Mauricio, & Rodríguez, 2014).

To measure the quality of the product or services delivered by the ART, the number of incidents during the PI are counted and a ratio compared to the total number of features is calculated. With the indicators, the management gets insight into what degree ARTs can deliver products or services which meet specified requirements by the different stakeholders of the product. However, only counting is limited and for this reason, the impact of the incidents are also taken into account by looking at the priority of the incident (i.e. first priority incidents are more impacting the customer than second or third priority incidents).

4.2.1.4 Engagement

Engagement is the fourth development outcome mentioned within the SAFe framework and it relates to the employee willingness to invest a high level of physical, cognitive and emotional resources on the work tasks associated with the job (Christian, Garza, & Slaughter, 2011).
As mentioned by Fagerholm et al. (2014) performance management information systems should not only focus on the products or services but also human factors such as skills and behaviour evaluating the performance of ARTs. So, for this reason, it is decided to define the engagement of employees by using a questionnaire for the ARTs. This is the most common data collection instrument in business research according to Cooper and Schindler (2014). In this questionnaire, the ARTs are asked to rate the engagement during the last PI.

4.3 The core competencies related to the development outcomes

In the developed performance management information system the core competencies are identified as the leading indicators and the development outcomes as the lagging indicators. To achieve a balance between leading and lagging indicators, those concepts are discussed in relation to each other. Because the goal of the performance management information system is to be uniformly applied in practical situations, Agile SAFe professionals are involved in the process to relate them to each other. This is because these professionals are familiar with the application of the core competencies and development outcomes within practical situations.

The core competencies are associated with the skills, practices and required behaviour as suggested in the self-assessments of SAFe. These assessments are organised following a standard process pattern of running the assessment, analysing the results and acting. These competencies are essential to achieve the development outcomes: productivity, time-to-market, quality and engagement. However, there is a lack of specific reactive indicators to indicate to which development outcome an improvement of the skills, principles or practises contributes. For this reason, it is decided to relate the core competencies with the four development outcomes.

These professionals are asked to individually verify to which development outcome(s) the several statements are related. For each SAFe competencies Scaled Agile developed a list of statements which represents those practices, skills and principles of each core competency:

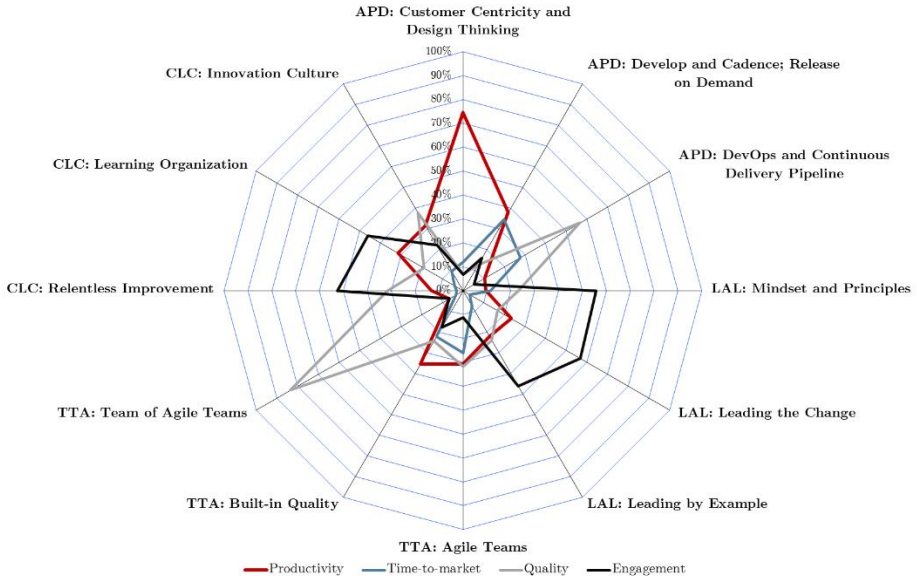
1. Lean-Agile Leadership
2. Team and technical agility
3. Agile product delivery
4. Continuous learning culture

All of these statements are individually discussed by the four professionals in relation to the development outcomes. All the individual results are shown in Appendix E. The results are combined and discussed during four sessions to understand each other's motivation. The result of this discussion is summarized in a radar chart (see Figure 9). For example, the statements of the core competency Agile Product Delivery contributes almost 40% to productivity, 30% to time-to-market, 30% to quality and not to engagement.

Figure 9: Relation between the development outcomes and competencies

Each core competency has dimensions. The Agile Product Delivery includes 'Customer Centricity and Design Thinking', 'Develop and Cadences; Release on-demand' and 'DevOps and Continuous Delivery Pipeline'. Figure 10 summarizes the association of the development outcomes with the dimension of each competency. In Appendix F the statements are categorized according to the development outcomes.

Figure 10: Relation between development outcomes and competency dimensions



These radar charts show to what extent a development outcome is represented in each of the core competencies. The statements of these competencies are the leading indicators to achieve a better outcome on the four development outcomes.

Growth stages

To give teams scope and direction, the ARTs should focus on relevant practices that will increase their Agile performance in each growth stage. The stages of the system are in line with the core competencies of NODA (NODA, 2016). Additionally, the naming of the stages is similar to the SAFe growth stages. The growth stages of the developed performance management information system:

- The *core* stage supports the basic understanding of each competency and the practices which are essential for becoming Agile are categorized in this stage;
- The *intermediate* stage provide growth opportunities to further development of skills and re-enforce existing knowledge and practices;
- The *advanced* stage supports the ARTs in developing expertise on the competencies and dimensions which are related to this. When all statements are implemented in the organisation and teams can focus on optimizing them.

This system is as a road to Agile in which the ART has to pass different stations by implementing practices and developing their knowledge. Each station represents a growth stage and it can be validated by the lagging indicators on productivity, time-to-market, quality and engagement if the desired outcome is achieved. In Appendix F the statements of each competency are associated with the growth stages.

4.4 The information system

In the analysis of current literature about Agile performance systems, there is a lack of administration mechanism to analyse and track the progress of these systems, see Table 1. It wasn't mentioned how organisations can collect the information to establish if an ART has reached a certain degree of performance. In this research, an information system is considered to indicate how teams perform.

4.4.1 Assessment results

The assessments gave insight into the most relevant practises executing when ARTs wants to grow on a certain competency or development outcome. Using multiple assessments across time improves the prediction and saves time in data analysis of the results (Cooper & Schindler, 2014).

In the system for core competencies, the diagrams gave insight into the results of the assessments (see Figure 11 and Figure 12). The RTE can click through the dashboard and filter on development outcomes, growth stages, competencies or dimensions. The results can then be discussed with the professionals within the organisations for evaluating whether the RTE have the correct interpretation of the statements. Thereafter, they can establish the next steps to develop the core competencies. By looking at the lagging indicators, good consideration can be made to improve the right competencies to achieve a better result on the performance indicators.

Figure 11: Screenshot of the information system - Assessment dashboard

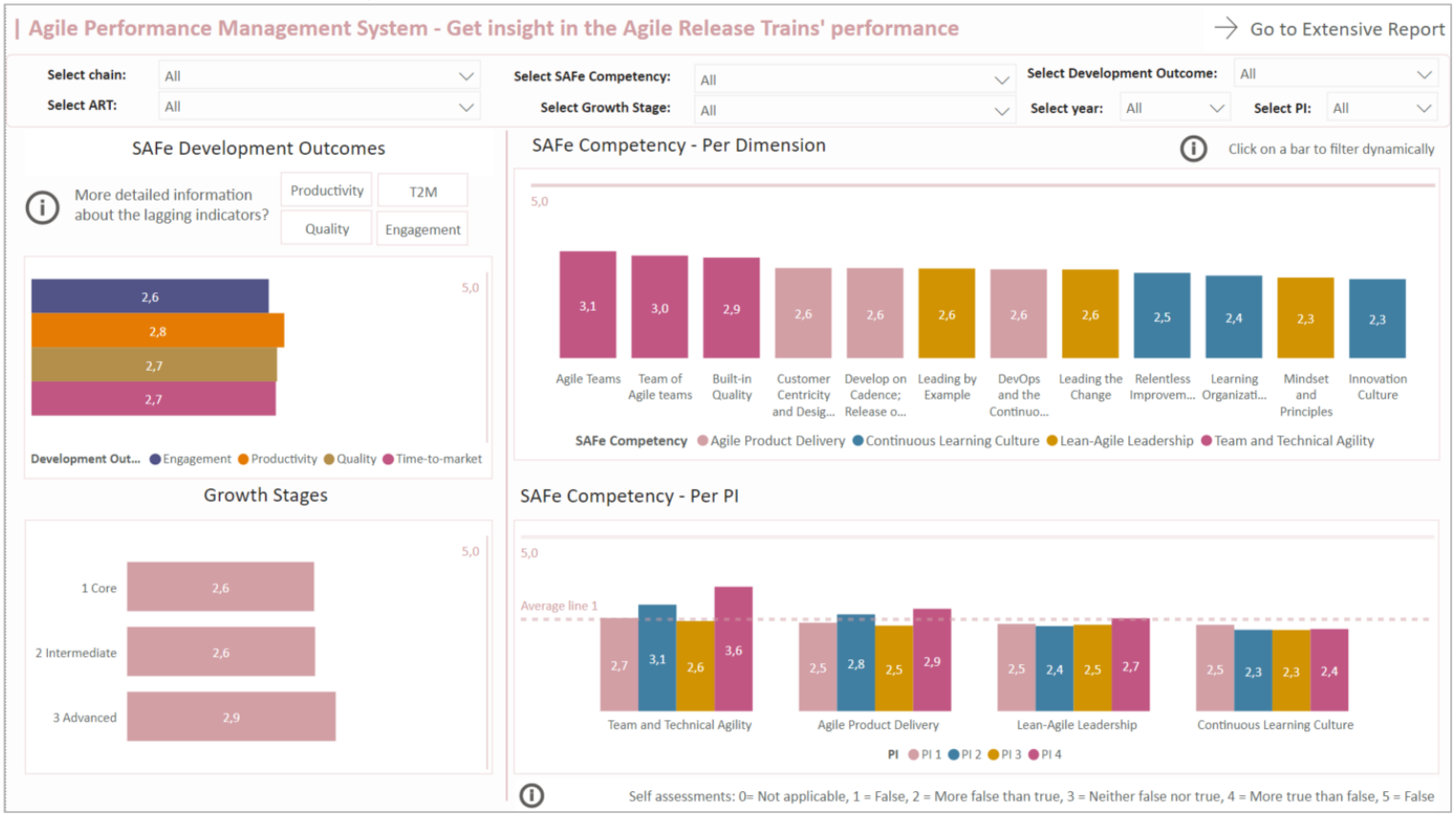
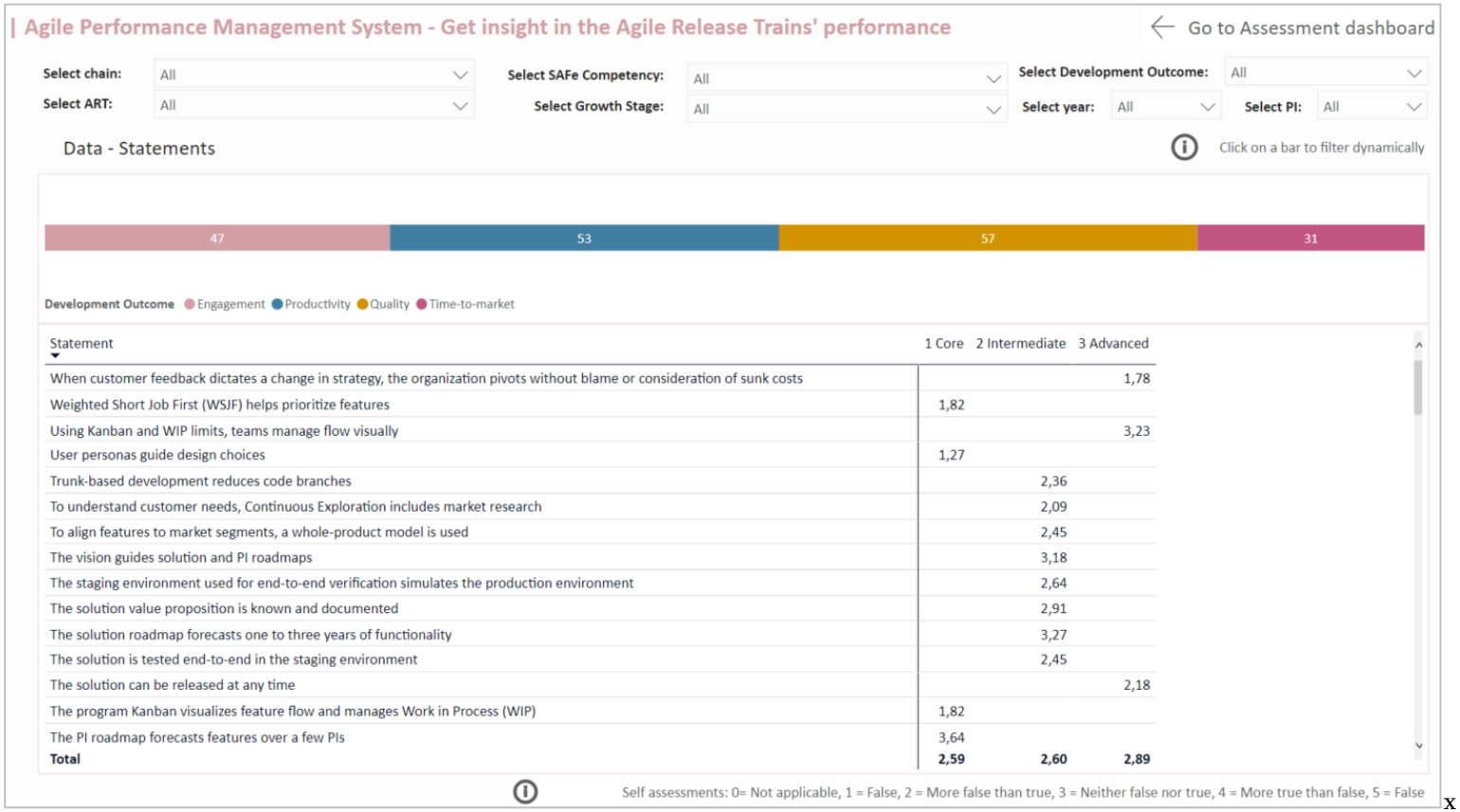


Figure 12: Screenshot of the information system - Extensive report



4.4.2 Performance indicators

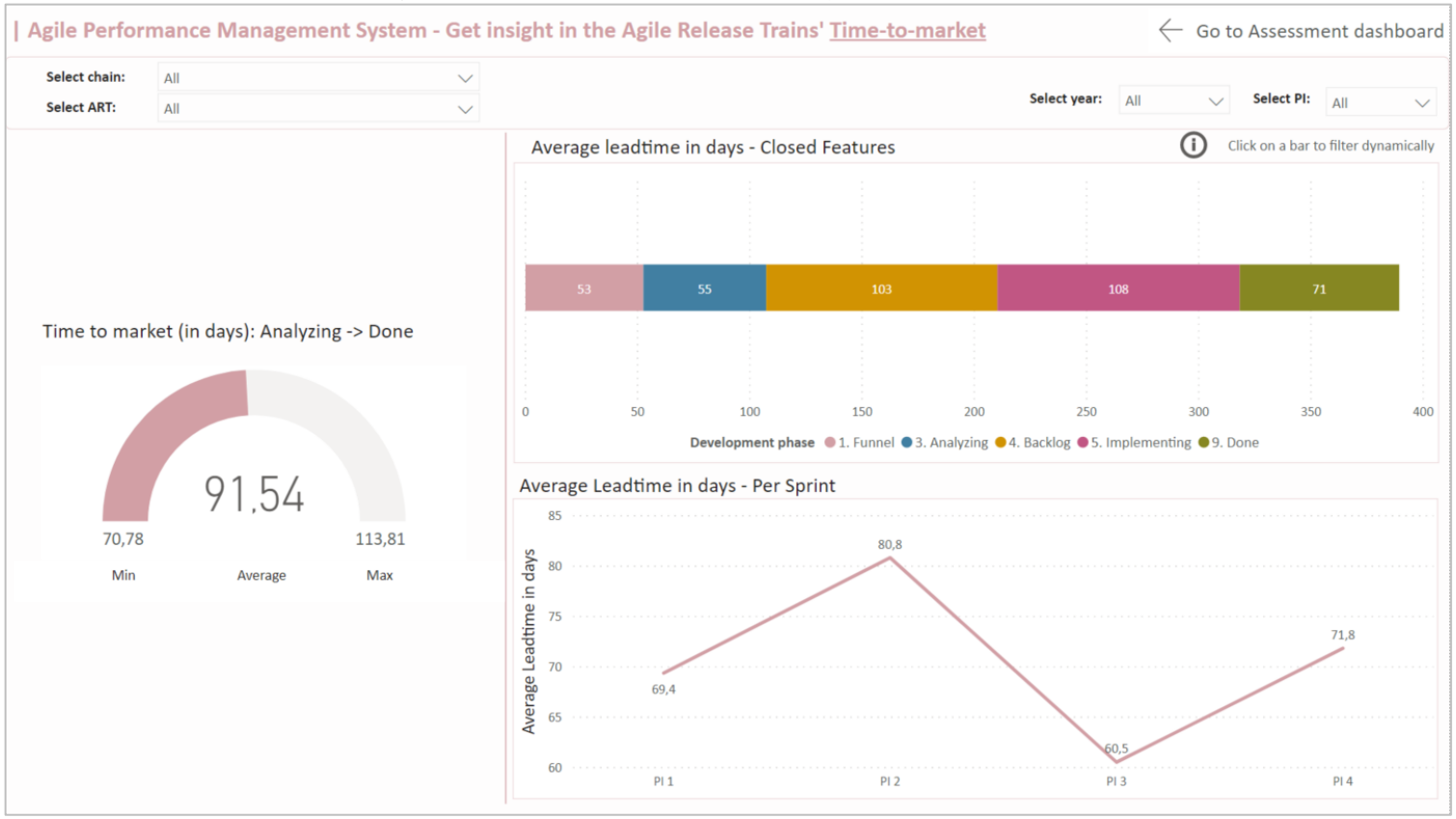
The developed dashboard of the lagging indicators gives insight into the performance of the ARTs on the development outcomes. The data is automatically updated from the tooling which is used by the case study company. Instead of reviewing their performance quarterly, bi-annually or once per year, this system makes it possible to review the result of their applied practices and knowledge at any time. Although the information system is designed for a specific business environment, the content does reflect the possibilities to get insight into the different lagging indicators for each organisation.

The system for the development outcomes is based on the indicators defined in section 4.2.1. For each development outcome the following insights are relevant:

- *Productivity:* the earned business value points related to the ART capacity gives insight into the productivity of the ART. As mentioned in section 4.2.1, by looking at the growth in the ratio between these components, it becomes clear if the ART can deliver more value with fewer resources in a PI, and so the ART grows in their productivity.
- *Quality:* the data is based on the registered incidents within the development system of the organisation. The dashboard gives insight into the number of incidents for each date and the meantime to recovery (MTTR). In several visualisations, the company get insight into the quality and can filter on the relevant characteristics (see Figure 13).
- *Time-to-market:* the indicators of time-to-market are focussed on days in transition for each development phase and how the ARTs perform over time (see Figure 14).
- *Engagement:* the development outcome focussed on engagement is focussed on the PI grade for each sprint. The ARTs should give the engagement a grade which represented the employees' involvement and willingness to invest a high level of physical, cognitive and emotional resources on the work tasks associated with the job.

Figure 13: Screenshot of the information system - Quality

Figure 14: Screenshot of the information system - Time to market



The information system for productivity and engagement is being implemented within the Dutch insurance company at this moment and therefore screenshots are not yet available to be published.

4.4.3 Information system architecture

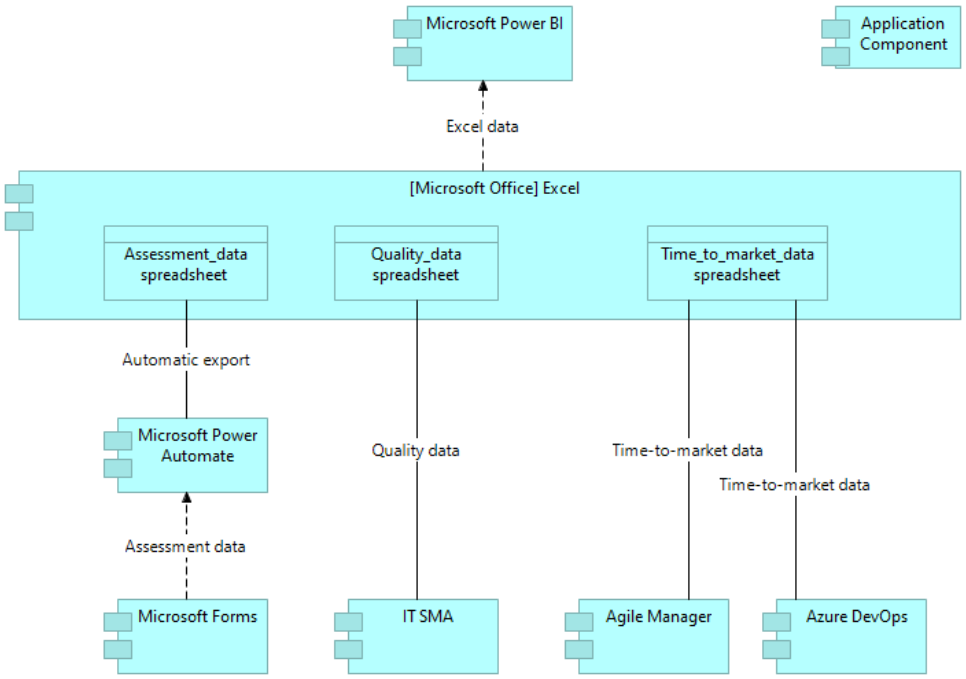
The architecture of the system is visualized in an ArchiMate figure (see Figure 15). In this figure, the Application Co-operation viewpoint shows an overview of the application landscape of the information system.

The data of the information system is retrieved from four different applications which are used by the case study company.

- Microsoft Forms: assessment data
- IT SMA: quality data
- Agile Manager: time-to-market data
- Azure DevOps time-to-market data

The data of these applications is automatically exported to different spreadsheets in Microsoft Office Excel. The spreadsheets are connected to Microsoft Power BI which is used as an interactive data visualisation application.

Figure 15: Architecture of the Information System



4.4.4 Recommendations for implementation

A fundamental aspect of all Agile methodologies is the ability for an ART to self-organise their way of working and to establish a growth path. This system gives the teams and management a system to get more insight into the performance in the current software age.

Two scenarios could be established, to discover a) which of the four development outcomes could prove problematic and b) which is of highest priority to improve Agile performance. First, the ARTs themselves can look at their performance on the provided information system and can establish which outcome is the most underperforming and should be improved. Secondly, the management has insight into the performance of each ART and can give priority to an outcome which is essential to achieve the upcoming goals.

To indicate the baseline, the ARTs should perform the assessment with statements for a specific outcome on the core growth stage. These assessments will be conducted in meetings with a SAFe Program Consultant (SPC) and the Release Train Engineer (RTE). The SPC is included because an SPC has experience in the application of SAFe practices and can assess whether the ART is correctly implementing the Agile practices. Together with the RTE, who is responsible for the ART events and process and assisting the teams in delivering value, the Agile performance will be evaluated.

Based on this baseline the ART can recognize a growth opportunity and identify a root cause. Thereafter, the ART can learn from best practices within the organisation to develop and implement the required improvement. By looking at the information system for the lagging indicators, the ART can evaluate how this improvement influenced the result on the lagging indicators. With this, the designed performance management information system provides a continuous feedback loop to improve Agile performance. When the expected result is achieved the ART should celebrate their success and adjust their standards to continuously improve their performance and by doing this they can look for the next improvement. If the ART executes all practices of the core stage, the statements of the intermediate are the following growth opportunities to focus on and thereafter the advanced statements.

5 Evaluation

To evaluate the system (Yin, 2018) encourages in his book the use of triangulation. This is the use of different kinds of sources to confirm relevant information (Abdul Rahim & Wan Daud, 2015) and builds on an older statement that triangulation provides an important way of ensuring the validity of a case study (Jick, 1979). There are several types of triangulation: data, investigator, theoretical, methodological or perceptual (Groza & Ragland, 2016). In evaluating the system at the case study's company, data triangulation is used. This refers to data collected from different sources (Cometto, Meunier-fitzhugh, Labadie, & Roux, 2016; Yin, 2018).

5.1 Design study evaluation

A design study is a study in which a visualization system is designed to solve a real-world problem which is faced by professionals in a specific domain (Sedlmair, Meyer, & Munzner, 2012). A crucial aspect of a design study is the evaluation of the system to assess whether the intended outcomes were realized (Sein et al., 2011). In this research, the system is evaluated by a selected panel of professionals and they are selected based on their theoretical and practical perspectives (Sedlmair et al., 2012). Involving professionals in the development of the system is critical and can be considered as an important part of the evaluation (Helgesson et al., 2012).

The panel of this evaluation consists of several roles within the field of Agile and SAFe: lean-Agile advisers, performance improvement managers, Agile coaches, IT managers and release train engineers. The initial system was presented to the panel and the development choices are both verbally and textually explained. The professionals were able to use the system and to experience all the functionalities of it. Afterwards, the professionals reviewed each part of the system using an online questionnaire (due to the global COVID-19 pandemic), with questions formulated as hypotheses. The professionals should answer when they think the hypotheses will be fulfilled. The method of Stojanov et al. (2015) was used, which consists of a rating system based on statements. In cases when the professionals disagree with the initial system, they were asked to elaborate on their responses. The results are shown in Table 4.

Table 4: Results design study evaluation (n=10)

To what extent do you agree with	1	2	3	4	5	Average
The use of predication-based and outcome-based indicators	0	1	2	4	3	3,9
The use of the SAFe Development outcomes	0	0	1	4	5	4,4
The use of the SAFe assessments	0	0	3	4	3	4,0
Business value as part of productivity	1	2	1	3	3	3,5
Capacity as part of productivity	3	1	0	1	5	3,4
The indicators of time-to-market	0	0	2	4	3	4,1
The indicators of quality	0	1	4	3	2	3,6
The indicators of engagement	1	1	1	6	1	3,5

The results of the questionnaire show that the panel of the design study seems to agree on a certain point, but that are also major differences in some other points. The degree to what extent the panel agrees on the use of prediction-based and outcome-based indicators varies. People argue that leading indicators should be more objective measures instead of subjective self-assessment measures. It isn't that they disagree with the SAFe assessment, but they prefer a combination of objective and subjective measures. The development outcomes are recognized in the current outcomes of their Agile approach and that's the main reason why the panel agrees with them. On the other hand, they do not fully agree with the lagging indicators. This is mainly because the panel has different interpretations of those concepts. At the same time, this was also known from the triple-p model of Tangen (2005) the author explains that people should accept the fact these terms are commonly interpreted in various ways.

5.2 Case study

Although the Agile performance management information system has been developed as a joint work of professionals in the field of Agile and SAFe, its applicability of the design artefact should be evaluated in a business environment (Stojanov et al., 2015). An evaluation of the artifact reflects not only the theoretical foundation but also the influence of users and use in context (Sein et al., 2011).

For this reason, the designed system will be applied to one of the largest insurance organisations in the Netherlands. This organisation is in the transition of taking the Agile performance to a higher level within the whole organisation. The case study has as objective to investigate the applicability of the performance management information system in practice.

The system is used and evaluated by six departments within the case study company. Based on the available assessments, the information system, informal conversations and observations the departments created a comprehensive image of the system. A complete overview of these departments is provided in Table 5.

Table 5: Overview of the cases

Case ID	Department (Contact person)	Case description	Agile performance challenges
C1	Value stream support (Manager/ Scrum master)	The Value stream support focuses on the support to provide a continuous flow of value to the customers.	The department is experiencing difficulties with to get insight into the amount of customer value, and thereafter to manage and growth based on this value.
C2	Continuous improvement department I (Continuous Improvement advisor)	The department advises the ARTs on making choices at organisational, ART a team level in terms of growth opportunities.	The problem within the department is the Agile Mindset. The department sometimes lapses into Agile project-based thinking and working. A new ART will soon start and there is a lack of tools to gain insight into the growth opportunities of this ART.
C3	ART – 6 teams (RTE)	The ART is responsible for the realization of IT adjustments for target architecture and legacy.	The further development of DevOps brings several implications. The ART is looking for methods to improve their monitoring about attitude, behaviour and their way of working, and to visualize metrics.
C4	Continuous improvement department II (Agile and Lean Advisor)	Continuous improvement focuses on the challenge of keeping bottom-up and top-down connected and in balance. To amplify the transformation and to move in the same direction.	The department is looking for the right way to give each ART freedom for their growth, while at the same time keeping sufficient insight into the Agile growth to test whether the Agile transformation contributes to realizing the operational and strategic goals.
C5	ART – 10 teams (RTE)	The ART work on the systems as a product division. The ART has a white label, intermediary and health declaration portal for various channels.	Implementing Agile is not only a way of working but also an attitude and behaviour. How to get a new working method implemented in a department with employees who have had various reorganisations and different way of working.
C6	IT Debtor management and payment transactions (Team manager)	Responsible for digitizing the payments of customers, to make it easier to pay, and to better control the debtor management process.	Within the department there is a desire to move towards DevOps, however, they found difficulties to take the next step towards this way of working.

5.2.1 Results

The six departments from the case study company have used and evaluated the system. A complete explanation has been given to each department about the purpose, design and insights of the system. Besides, documentation of the system has been published. Afterwards, the department members were able to use the system and they get the ability to try the functionalities. All members shared their findings and improvement suggestions and a contact person filled in a survey about several aspects of the system. Survey research is used for collecting information by asking the same survey questions, and to make it possible to qualify answers. The following section is focused on the questions proposed in the online questionnaire.

To what extent do the leading indicators contribute to determine and improve the Agile performance of the ARTs?

On a scale from 1 (not likely at all) to 10 (very likely). The professionals are asked to judge to what extent the self-assessment contribute to determine and improve the Agile performance of the ARTs. The results:

- C1 (Score: 6) – *"Look ok in itself, however, I find it peculiar that this compares ARTs in a certain way. I think that you will find people are more likely to give a socially desirable answer, to get to a higher level."*
- C2 (Score: 7) – *"The information system gives enough leads, but maybe there are too many indicators and the focus disappears."*
- C3 (Score: 6) – *"The self-assessments will help the ART only if this is filled in by different roles in the ART. If only the RTE has filled the self-assessment, it is difficult to start a conversation about it. When several roles in the ART have filled the assessment, the conversation can be conducted well. There is often a story behind the assessments and it is interesting to start a conversation about the differences on the same competencies."*
- C4 (Score: 8) – *"In my opinion not all the statements in the assessments point to behaviour, but the whole set of statements for each competency does give the ART a lot of guidance about what kind of behaviour is required and desired."*
- C5 (Score: 8) – *"Using the self-assessments alone provides already more insights than we have into behaviour at this moment. Using the entire information system, even more extensive insights can be obtained."*
- C6 (Score: 8) – *"It mainly indicates the next step in improvement potential. Perhaps the system can be improved by identifying which ART could be a good source of information in a specific competency of dimension. Some more metrics from the ART itself are needed to determine the performance."*

The six cases are all fairly positive about using the leading indicators to get more insight into the Agile performance of the ARTs. As mentioned by the ART of C5, Agile is not only a way of working but the attitude and behaviour is also an important aspect, these leading indicators contribute to the understanding of the Agile performance. It gives a good starting point to start a conversation with the stakeholders. The criticism of the continuous improvement department (C1) has been included in the recommendations for implementation (section 4.4.4), which indicate that the assessments will be conducted in meetings with a SAFe Program Consultant (SPC). This should prevent that RTEs give socially desirable answers.

To what extent do the lagging indicators contribute to determine and improve the Agile performance of the ARTs?

On a scale from 1 (not likely at all) to 10 (very likely). The professionals are asked to judge to what extent the performance indicators contribute to determine and improve the Agile performance of the ARTs. The results:

- C1 (Score: 5) – *"For Quality I would also look at the reason for closing the incidents. Was it correct? What caused the incident? For example, we often receive incorrect deliveries from external parties. Our ART is not to blame, but it still creates an incident."*
- C2 (Score: 5) – *"I think there are more factors that influence the quality instead of only looking at the number of indicators."*
- C3 (Score: 6) – *"At this moment I don't have a sufficient picture about how the data of these are imported from the systems."*
- C4 (Score: 7) – *"Continuous use of the system in practice will show which indicators the ARTs are still missing and whether it is the case for all ARTs. Not all ARTs wants to measure the same things or have the same assumptions even though they use the same definition."*
- C5 (Score: 6) – *"It is difficult to measure these indicators for all ARTs."*
- C6 (Score: 8) – *"It is still difficult to judge if this system is complete. In potentially useful, but practice have to show the value of using those indicators."*

The two continuous improvement departments (C2 and C4) are responsible to give each ART freedom for their growth opportunities, however, the departments are critical about the indicators of the information system because the ARTs within the case study company are very different. Some ARTs are completely focussed on the business, while there are also ARTs completely focussed on IT components. It is doubtful if these indicators give enough insights into the performance on those development outcomes.

To what extent does the system contribute to determine the growth opportunities of the ARTs?

On a scale from 1 (not likely at all) to 10 (very likely). The professionals are asked to judge to what extent the system contribute to determine the growth opportunities of the ARTs. The results:

- C1 (Score: 5) – *"In my view, you will not necessarily see growth opportunities in this system. It is more a picture of a certain moment and growth opportunities are utopias in my opinion."*
- C2 (Score: 6) – *"The system helps the ART to get insight into the growth opportunities, however, it helps if each ART can fill in standards. This gives a better indication of what an ART wants to achieve."*
- C3 (Score: 7) – *"The system forces you to think and carry on the conversation with each other."*

- C4 (Score: 10) – *"I think the assessments together with the development outcome give a very good picture of what's is going well in our organisation and what can be improved."*
- C5 (Score: 8) – *"With the SAFe concepts you will touch on the most important parts of Agile methodologies."*
- C6 (Score: 9) – *"The assessments are the basis and for this reason, the growth potential is the biggest gain of this performance system."*

The professionals vary in their opinions on whether the system provides sufficient growth opportunities. As the expert of the case C1 argued that growth opportunities are utopias which is the opposite of the professional of C4 who mentioned that the system gives a very good picture of the growth opportunities. However, for the new ART which will soon start in the Continuous Improvement department of C2, it helps the ART to get insight into the growth opportunities. As an improvement department, they are responsible for advising ARTs on making the right choices, and this system can help the department to give the ART a tooling method. The RTEs of C3 and C5 agree that a system based on SAFe helps to carry on the conversation.

How likely is that you are going to use the system?

On a scale from 1 (not likely at all) to 10 (very likely). The professionals are asked to judge how likely the use of the system within the organisation is. The results:

- C1 (Score: 4) – *"The statements within the SAFe assessment do not appeal to me, because I didn't recognize the skills which are prerequisites for working Agile. These questions need to be improved before I will start using the system within my department."*
- C2 (Score: 9) – *The system helps to communicate with teams, scrum masters and RTEs where improvements can be achieved and how to define strategies to achieve it."*
- C3 (Score: 8) – *"The assessments with assessments give good opportunities to start the conversation with the team. Also, the option to look into historical data with the information system makes this conversation easier. I also find the relation with the lagging indicators useful, because it makes it easier to improve on a specific part."*
- C4 (Score: 9) – *"I am confident that this is a good step towards increasing the self-learning ability for the ARTs. An information system is missing in the current way of working and this system fills this gap. However, the guidance during the implementation of the system looks difficult."*
- C5 (Score: 8) – *"In combination with other measurements we use in our ART like critical control points, we can use these tools to 'measure' and learn from each other to create a learning culture in the organisation."*
- C6 (Score: 8) – *"The performance management system is useful for both managers and ARTs to get insight into the performance. However, the information system is an automated process and I am curious whether this can run fully automated within the whole organisation without problems."*

Looking at the practical experiences of the professionals of using the system, there are mainly criticisms about how this will perform in the organisation. For a single-use, some professionals experienced some difficulties to use the information system. A detailed implementation plan of the system within the organisation can partly solve this problem. The Continuous Improvement department (C4) emphasise that the system gives the ARTs a self-learning ability and this is in line with their challenges to give each ART freedom for their growth. The ART (C5) confirms that this contributes to create a learning culture and to make the department members aware of not only a new way of working but also the related attitude and behaviour. The professionals motivate that it is primarily a useful means of conducting the conversation. However, the theoretical set-up of the whole system seems to be confirmed by practice.

5.2.2 Suggestions for improvement

In this section, the possible suggestions for improvement of the performance management information system will be discussed based on the results. Cooper and Schindler (2014) have set three criteria for evaluating such a system: reliability, validity and practicality.

To achieve these criteria in the system, the first suggestion is about the self-assessments. According to the different cases, these assessments are helpful to determine what is going well in the organisation. However, these measurements are subjective and it's questionable if it supplies consistent results (Cooper & Schindler, 2014). The continuous improvement departments indicate that it will be valuable to add objective measures to the core competencies to improve the reliability of the system. This will help to solve their challenges to give ARTs insights into their growth opportunities and to test whether the Agile transformation contributes to realizing their goals. Secondly, the performance indicators are too generic and the case studies are critical if the system measures what they wish to measure. The ARTs within a large organisation are very different and it would be valuable if the system is flexible to add ART-specific indicators. Finally, the departments indicate that the system should be evaluated more in practice to judge the practicality of the system.

6 Conclusion

This research aimed to develop a performance management information system to improve the uniformity of Agile team performance analysis to enable organisations to determine and improve their Agile performance. To achieve this, the main research question and four sub-questions were proposed. These four sub-questions will be answered before discussing the answer on the main research question: "What is an Agile performance management information system that fits in the current software age and enables companies to grow in their performance?"

SQ1: *"Which Agile performance management information systems are available in the literature?"*

To answer this sub-question a literature review resulted in eleven systems. These systems were analysed according to the capabilities of Maier et al. (2012) for designing systems. It became clear that first of all the systems have three dimensions: people, product and process, and organisational culture. Secondly, even though the systems have different stakeholders, the number of levels does not seem to depend on it. Finally, there is a lack of administrative mechanisms to collect the data. A full overview of the current systems within the literature is shown in Table 1.

SQ2: *"How can the usability of the performance system be improved?"*

According to the literature, there is a lack of information systems to collect the required information. The usability of the system is improved by designing such an information system in which the assessment results and performance indicators are summarized. By monitoring the results the Agile Release Trains (ARTs) can establish if a team reaches a certain degree of competence on the different core competencies. In addition, the system provides insight into the leading and lagging indicators of each development outcome.

SQ3: *"How do professionals in the Agile field evaluate the developed Agile performance system and what improvements need to be made based on their evaluation?"*

Answering this question was done through a design study evaluation. The system was evaluated by a selected panel of professionals with theoretical and practical perspectives. The professionals were able to use the system and the functionalities of the system to evaluate the applicability of the system in a business environment (Stojanov et al., 2015). One of the suggested improvements is adding objective indicators to the core competencies to prevent different interpretations of the subjective statements of the assessments. Secondly, the lagging indicators are too generic and the professionals suggest adding ART-specific indicators.

SQ4: *"How do the stakeholders evaluate the utility of the designed Agile performance management information system and what improvements need to be made based on their evaluation?"*

Based on practical experience the utility critique is mainly given on how the information system will perform in a large organisation and will lead to desirable outcomes. To determine whether the system leads to these outcomes, the professionals expect difficulties to get the required data. An implementation plan could solve this, however, the challenge will remain, according to the professionals.

RQ: *"What is an Agile performance management information system that enables organisations to grow in their Agile performance?"*

The answers on the sub-questions resulted in a uniform Agile performance management information system which enables organisations to grow in their performance. The system consists of both the core competencies of the team members and development outcomes of the work product. By using self-assessments the ARTs can assess themselves on the core competencies. The development outcomes are measured by different performance indicators and are monitored in the information system.

7 Discussion and future work

It became clear from the literature review, the current performance management systems are mostly focussed on one Agile method (e.g. scrum). Therefore, it was aimed in this research to create a uniform system which can be used in organisations with different combinations of Agile methods. The main contribution is to fill the gap within the literature and practice, however it doesn't come without limitations.

COVID-19 pandemic

Due to the outbreak of global pandemic COVID-19 the original validations plans had to be changed. It was outside the scope of this research to postpone, so it was decided to focus on virtual evaluation and validation methods. Due to restrictions of the case study company, it wasn't possible to do face-to-face meetings, so the non-verbal communication can't be observed during the meetings. With the use of online questionnaires for evaluating the system and virtual sessions to discuss the results, the results might have given different results.

Case study

The case study was performed to evaluate the applicability of the system in a business environment (Stojanov et al., 2015). A limitation of this case study was the number of participants, more participants participate could provide more input to evaluate the system. Also, the system could be evaluated in future work by different organisations with different combinations of Agile methods. This shows a limitation on the uniformity of methodologies in which the system is validated. Furthermore, the industry of the case study might have an impact on the results of the system. The implementation of the system within a manufacturer organisation might give significantly different results, and this should be validated in a follow-up study of this system. Each industry has different processes and this follow-up study should make clear whether these organisations are able to implement the developed system. Given these points, further case studies are required to assess the completeness and effectiveness of the model.

Flexibility

Although this research set a baseline, further research can be focused on the flexibility of the system. The evaluation participants argued that the current system is still too rigid. It is doubtful if the current performance indicators contribute to the understanding of the performance of the development outcomes. They suggest adding objective indicators of the core competencies to prevent different interpretations of the subjective statements of the assessments. The second suggestion is to add more ART-specific indicators because ARTs within the case study company are sometimes completely different. For example, some ARTs are completely focussed on the business, while others are focussed on information technology.

Management implications

To implement the system within an organisation the management should take into account the time and cost involved by implementing the system. First of all, to implement the information system the data should be available to collect with this system. It takes time to set up the work processes correctly to ensure the relevant information can be imported within the information system. The implementation time depends on the extent to which these processes are designed for the required data of the system.

Secondly, the teams should execute and discuss the results of the assessments. The time to execute is reduced by the information system due to automation of the results: teams no longer need to manually export the data and summarize the data in visualisations. The management still considers whether the invested time justifies the outcomes of using the performance management information system. In addition, as recommended, the assessments should be conducted in meetings with a SAFe Program Consultant (SPC). So, it is required that there are people within the organisation certified as SPC.

Bibliography

Abdul Rahim, M., & Wan Daud, W. N. (2015). Scholars Journal of Arts, Humanities and Social Sciences The Case Study Method in Business. *Sch. J. Arts. Humanit. Soc. Sci*, *3*(1B).

Afonso, P., Nunes, M., Paisana, A., & Braga, A. (2008). The influence of time-to-market and target costing in the new product development success. *International Journal of Production Economics*, *115*(2). https://doi.org/10.1016/j.ijpe.2008.07.003

Ambler, S. W., & Lines, M. (2016). The disciplined agile framework: A pragmatic approach to agile maturity. *CrossTalk*, *29*(4).

Bekkers, W., Van De Weerd, I., Spruit, M., & Brinkkemper, S. (2010). A framework for process improvement in software product management. In *Communications in Computer and Information Science* (Vol. 99 CCIS). https://doi.org/10.1007/978-3-642-15666-3_1

Bosch, J. (2014). *Continuous software engineering*. *Continuous software engineering* (Vol. 9783319112831). https://doi.org/10.1007/978-3-319-11283-1

Bourgeois, D., Mortati, J., Wang, S., & Smith, J. (2019). *Information Systems for Business and Beyond (2019)*. *Bioinformatics: Tools and Applications*. https://doi.org/10.1007/978-0-387-92738-1_18

Bucena, I., & Kirikova, M. (2017). Simplifying the devops adoption process. In *CEUR Workshop Proceedings* (Vol. 1898).

Chiesa, V., Coughlan, P., & Voss, C. A. (1996). Development of a technical innovation audit. *Journal of Product Innovation Management*, *13*(2). https://doi.org/10.1016/0737-6782(95)00109-3

Choong, K. K. (2013). Understanding the features of performance measurement system: A literature review. *Measuring Business Excellence*. https://doi.org/10.1108/MBE-05-2012-0031

Choudhary, V. (2007). Software as a Service : Implications for Investment in Software Development The Paul Merage School of Business. *Proceedings of the 40th Hawaii International Conference on System Sciences*, *40*(7). https://doi.org/10.1109/HICSS.2007.493

Christian, M. S., Garza, A. S., & Slaughter, J. E. (2011). Work engagement: A quantitative review and test of its relations with task and contextual performance. *Personnel Psychology*, *64*(1). https://doi.org/10.1111/j.1744-6570.2010.01203.x

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer*, *34*(11). https://doi.org/10.1109/2.963450

Cometto, T., Meunier-fitzhugh, K. Le, Labadie, G. J., & Roux, F. (2016). Marketing Challenges in a Turbulent Business Environment. *Proceedings of the Academy of Marketing Science*. https://doi.org/10.1007/978-3-319-19428-8

Cooper, D. R., & Schindler, P. S. (2014). *Business Research Methods 12th Edition*. *Business Research Methods*.

Coyne, K., Hall, S., & Clifford, P. (1997). Is Your Core Competence a Mirage? *The McKinsey Quarterly*, (1).

Danilovic, M., & Leisner, P. (2007). Analyzing Core Competence and Core Products for Developing Agile and Adaptable Corporation. Retrieved from https://www.designsociety.org/download-publication/27692/analyzing_core_competence_and_core_products_for_developing_Agile_and_adaptable_corporation

de Feijter, Rico, Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2017). Towards the adoption of DevOps in software product organizations: A maturity model approach. *Technical Report Series*, (UU-CS-2017-009).

de Feijter, Rico, Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018). DevOps competences and maturity for software producing organizations. In *Lecture Notes in Business Information Processing* (Vol. 318). https://doi.org/10.1007/978-3-319-91704-7_16

Eficode. (2015). *Eficode Quick Guide - DevOps*. Retrieved from https://www.eficode.com/hubfs/documents/Eficode-English-Devops-Guide.pdf?hsLang=en

Eilat, H., Golany, B., & Shtub, A. (2008). R&D project evaluation: An integrated DEA and balanced scorecard approach. *Omega*, *36*(5). https://doi.org/10.1016/j.omega.2006.05.002

Emam, K. El, & Goldenson, D. R. (2000). An empirical review of software process assessments. *Advances in Computers*. https://doi.org/10.1016/S0065-2458(00)80008-X

Ertaban, C., Sarikaya, E., & Bagriyanik, S. (2018). Agile performance indicators for team performance evaluation in a corporate environment. In *ACM International Conference Proceeding Series* (Vol. Part F147763). https://doi.org/10.1145/3234152.3234156

Fagerholm, F., Ikonen, M., Kettunen, P., Münch, J., Roto, V., & Abrahamsson, P. (2014). How do software developers experience team performance in Lean and Agile environments? In *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2601248.2601285

Fatema, I., & Sakib, K. (2018). Factors Influencing Productivity of Agile Software Development Teamwork: A Qualitative System Dynamics Approach. In *Proceedings - Asia-Pacific Software Engineering Conference, APSEC* (Vol. 2017-December). https://doi.org/10.1109/APSEC.2017.95

Feijter, Rico˜de, Vliet, R., Jagroep, E., Overbeek, S., & Brinkkemper, S. (2017). Towards the adoption of DevOps in software product organizations: A maturity model approach. *Technical Report Series*, (UU-CS-2017-009).

Fontana, R. M., Meyer, V., Reinehr, S., & Malucelli, A. (2015). Progressive Outcomes: A framework for maturing in agile software development. *Journal of Systems and Software*, *102*. https://doi.org/10.1016/j.jss.2014.12.032

Fontana, R. M., Reinehr, S., & Malucelli, A. (2015). Agile Compass: A Tool for Identifying Maturity in Agile Software-Development Teams. *IEEE Software*, *32*(6). https://doi.org/10.1109/MS.2015.135

Forsyth, D. R. (2006). *Group dynamics 4th edition*. *Thomson* (Vol. 19). https://doi.org/10.1177/1046496404263771

Fritze, C. (2016). The Toyota Production System - The Key Elements and the Role of Kaizen within the System. Retrieved from https://www.researchgate.net/publication/289519018_The_Toyota_Production_System_-_The_Key_Elements_and_the_Role_of_Kaizen_within_the_System

Gallon, M. R., Stillman, H. M., & Coates, D. (1995). Putting Core Competency Thinking into Practice. *Research-Technology Management*, *38*(3). https://doi.org/10.1080/08956308.1995.11674264

Gosselin, M. (2005). An empirical study of performance measurement in manufacturing firms. *International Journal of Productivity and Performance Management*, *54*(5–6). https://doi.org/10.1108/17410400510604566

Greening, D. R. (2015). Agile enterprise metrics. In *Proceedings of the Annual Hawaii International Conference on System Sciences* (Vol. 2015-March). https://doi.org/10.1109/HICSS.2015.597

Grünberg, T. (2004). Performance improvement: Towards a method for finding and prioritising potential performance improvement areas in manufacturing operations. *International Journal of Productivity and Performance Management*, *53*(1). https://doi.org/10.1108/17410400410509969

Hazzan, O., & Dubinsky, Y. (2014). The Agile Manifesto. In *SpringerBriefs in Computer Science* (pp. 9–14). Springer. https://doi.org/10.1007/978-3-319-10157-6_3

Heitlager, I., Jansen, S., Helms, R., & Brinkkemper, S. (2006). Understanding the dynamics of product software development using the concept of coevolution. In *Proceedings - Second International IEEE Workshop on Software Evolvability, SE'06*. https://doi.org/10.1109/SOFTWARE-EVOLVABILITY.2006.17

Helgesson, Y. Y. L., Höst, M., & Weyns, K. (2012). A review of methods for evaluation of maturity models for process improvement. *Journal of Software: Evolution and Process*. https://doi.org/10.1002/smr.560

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. *Continuous delivery*. https://doi.org/10.1007/s13398-014-0173-7.2

Jackson, M., & Petersson, P. (1999). Productivity--an overall measure of competitiveness. In *Proceedings of the Second Workshop on Intelligent Manufacturing Systems Leuven Belgium*.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly*, *24*(4). https://doi.org/10.2307/2392366

Kaushik, A. (2007). A Literature Review on Agile Software Development. *International Journal of Advanced Research in Computer and Communication Engineering ISO*, *3297*(9). https://doi.org/10.17148/IJARCCE.2016.5971

Kawalek, P., & Wastell, D. (2008). Strategic alignment and a culture innovation: Using the SPRINT methodology to meet two challenges of information age government. In *Evaluating Information Systems: Public and Private Sector* (pp. 218–235). https://doi.org/10.4324/9780080570105

Koehler, J., Woodtly, R., & Hofstetter, J. (2015). An impact-oriented maturity model for IT-based case management. *Information Systems*, *47*. https://doi.org/10.1016/j.is.2013.12.003

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean software development - A systematic literature review of industrial studies. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2015.02.005

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of devOps. In *Lecture Notes in Business Information Processing* (Vol. 212). https://doi.org/10.1007/978-3-319-18612-2_19

Maier, A. M., Moultrie, J., & Clarkson, P. J. (2012). Assessing organizational capabilities: Reviewing and guiding the development of maturity grids. *IEEE Transactions on Engineering Management*, *59*(1). https://doi.org/10.1109/TEM.2010.2077289

Marwedel, P. (2018). Evaluation and Validation. https://doi.org/10.1007/978-3-319-56045-8_5

McMahon, P. E. (2015). A "thinking framework" to power software development team performance. *CrossTalk*, *28*(1).

Menzel, G. (2015). DevOps - The Future of Application Lifecycle Automation. *A Capgemini Architecture Whitepaper*, *2nd*, 1–24. Retrieved from https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2016/03/devops-the-future-of-application-lifecycle-automation.pdf

Mohamed, S. I. (2015). DevOps Shifting Software Engineering Strategy Value Based Perspective. *IOSR Journal of Computer Engineering Ver. IV*, *17*(2). https://doi.org/10.9790/0661-17245157

Moher, D., Shamseer, L., Clarke, M., Ghersi, D., Liberati, A., Petticrew, M., … Whitlock, E. (2016). Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Revista Espanola de Nutricion Humana y Dietetica*, *20*(2). https://doi.org/10.1186/2046-4053-4-1

Moseng, B., & Rolstadas, A. (2001). Success factors in the productivity process. Retrieved from www.catriona.napier.ac.uk/resource/wpc10th/%0Amoseng.htm

NODA. (2016). *NODA Core Competencies*. Retrieved from https://cdn.ymaws.com/www.nodaweb.org/resource/resmgr/docs/competenciesbooklet.pdf

P. Miguel, J., Mauricio, D., & Rodríguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Products. *International Journal of Software Engineering & Applications*, *5*(6). https://doi.org/10.5121/ijsea.2014.5603

Patel, C., & Ramachandran, M. (2009). Agile Maturity Model (AMM): A software process improvement framework for agile software development practices. *Int. J. of Software Engineering, IJSE*, *2*(I). https://doi.org/10.4304/jsw.4.5.422-435

Peffers, 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). https://doi.org/10.2753/MIS0742-1222240302

Putta, A., Paasivaara, M., & Lassenius, C. (2019). How are agile release trains formed in practice? A case study in a large financial corporation. In *Lecture Notes in Business Information Processing* (Vol. 355). https://doi.org/10.1007/978-3-030-19034-7_10

Radstaak, J. (2019). *Developing A DevOps Maturity Model*. University of Twente. Retrieved from https://essay.utwente.nl/77808/1/Thesis Jeroen Radstaak.pdf

Rico, D. F. (2008). What is the ROI of Agile vs. Traditional Methods? An analysis of XP, TDD, Pair Programming, and Scrum (Using Real Options). *TickIT International*, *10*(Cmmi). https://doi.org/10.1007/BF00419774

Scaled Agile. (2020). What's new in SAFe 5.0. Retrieved from https://www.scaledagileframework.com/whats-new-in-safe-5-0/

Sedlmair, M., Meyer, M., & Munzner, T. (2012). Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12). https://doi.org/10.1109/TVCG.2012.213

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *MIS Quarterly: Management Information Systems*, *35*(1). https://doi.org/10.2307/23043488

Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: The agile adoption framework. *Innovations in Systems and Software Engineering*, *3*(3). https://doi.org/10.1007/s11334-007-0026-z

Slack, N., Chambers, S., Johnston, R., & Betts, A. (2012). Operations and Process Management. *Operations Management*.

Soares, F. S. F., & De Lemos Meira, S. R. (2015). An agile strategy for implementing CMMI project management practices in software organizations. In *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*. https://doi.org/10.1109/CISTI.2015.7170402

Stojanov, I., Turetken, O., & Trienekens, J. J. M. (2015). A Maturity Model for Scaling Agile Development. In *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*. https://doi.org/10.1109/SEAA.2015.29

Sydor, M. (2014). *Beyond Deployment Automation: Realizing DevOps Metrics and Collaboration through APM Visibility*.

Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and Performance Management*, *54*(1). https://doi.org/10.1108/17410400510571437

Thomas, B. E., & Baron, J. P. (1994). *Evaluating Knowledge Worker Productivity: Literature Review*. Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a283866.pdf

van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., & Bekkers, W. (2013). Improving IS Functions Step by Step: the Use of Focus Area Maturity Models. *Scandinavian Journal of Information Systems*, *25*(2).

Verwijs, C. (2019). Here's what's wrong with maturity models. Retrieved from https://medium.com/the-liberators/whats-wrong-with-maturity-models-abfb6dd97607

Wieringa, R. J. (2014). *Design science methodology: For information systems and software engineering*. https://doi.org/10.1007/978-3-662-43839-8

Yin, R. (2018). *Case Study Research and Applications Sixth Edition*. *SAGE Publications*. https://doi.org/10.1017/CBO9781107415324.004

Zheng, L., Baron, C., Esteban, P., Xue, R., Zhang, Q., & Yang, S. (2019). Using Leading Indicators to Improve Project Performance Measurement. *Journal of Systems Science and Systems Engineering*, *28*(5). https://doi.org/10.1007/s11518-019-5414-z

Appendix

A. Literature review on performance management

The first literature review is executed to get knowledge of the current problems related to performance management. To get the most relevant search results the following search query is used:

("Agile" OR "Scaled Agile Framework" OR "DevOps")
AND
("Team Performance" OR "Team Management" OR "Team Maturity")
AND
("Analysis" OR "Examination" OR "Investigation" OR "benchmarking" OR "comparison" OR "improvement" OR "improvement scenarios")

To search for the most relevant data the conditions of the literature review for the search queries is summarized in Table 6.

Table 6: Literature review protocol

Academic databases sources	Scopus
Search fields	Title
	Abstract
	Keywords
Language	English
Publication year	2015-2020

The first search based on the search query as mentioned at the beginning of the section resulted in 55 records. After using the inclusion and exclusion criteria, as mentioned in Table 7, 31 records are excluded from the literature review. After reading the abstract of these records, 17 records were selected for full-text screening. After reading those records, 11 papers were suitable research articles to use during this study. After this full-text reading, 4 papers are added, and this results in a total of 15 research articles for this review.

Table 7: Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Literature is written in English	Literature in all other languages
Literature published after 2014	Literature published before 2015
The publication stage of the literature is final	All other publication stages
Literature related to computer science	Literature not related to computer science

A complete overview of this literate review is shown in Table 8.

Table 8: A literature review on Agile team performance management

Date: 06-02-2020
Database: Scopus

Search filtering	**Number of papers**
# of records identified through database searching	55
# of records excluded based on inclusion/exclusion criteria	-31
# of records after inclusion/exclusion criteria	**25**
# of records screened on abstract	25
# of records excluded based on abstract	-8
# of records screened after abstract reading	**17**
# of records screened on the full text	17
# of records excluded based on full-text reading	-6
# of records added after full-text reading	+4
# of research articles after full-text reading	**15**

B. Literature review on existing performance systems

This literature review is executed to obtain more context about the current maturity systems. To get the most relevant search results the following search query is used:

("Agile" OR "Scaled Agile Framework" OR "DevOps")
AND
("Maturity model" OR "Growth model")

The other conditions of the literature review are summarized in Table 9.

Table 9: Literature review protocol

Academic databases sources	Scopus
Search fields	Title
	Abstract
	Keywords
Language	English
Publication year	2015-2020

The inclusion and exclusion criteria are shown in Table 10.

Table 10: Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Literature is written in English	Literature in all other languages
Literature published after 2014	Literature published before 2015
The publication stage of the literature is final	All other publication stages
Literature related to computer science	Literature not related to computer science

The first search based on the search query as mentioned at the beginning of the section resulted in 104 records. After using the inclusion and exclusion criteria, as mentioned in Table 10, 16 records are excluded from the literature review. After reading the abstract of these records, 23 records were selected for full-text screening. After reading those records only 5 papers were suitable research articles to use during this study. After this full-text reading, 6 papers are added, and this results in a total of 11 research articles for this review. A complete overview of this literate review is shown in Table 11.

Table 11: A literature review on Agile maturity

Date: 05-01-2020

Database: Scopus

Search filtering	**Number of papers**
# of records identified through database searching	104
# of records excluded based on inclusion/exclusion criteria	-16
# of records after inclusion/exclusion criteria	**88**
# of records screened on abstract	88
# of records excluded based on abstract	-55
# of records screened after abstract reading	**23**
# of records screened on the full text	23
# of records excluded based on full-text reading	-18
# of records added after full-text reading	+8
# of research articles after full-text reading	**11**

C. Summary of selected papers of the systematic literature review

The eleven research papers which were selected based on the literature review are summarized in Table 12.

Table 12: Selected papers of the systematic literature review

Title	Description
An Agile Strategy for Implementing CMMI Project Management Practices in Software Organizations (Soares & Meira, 2015)	This paper proposed a strategy for implementing Capability Maturity Model Integration (CMMI) by making use of the best practices of Agile Project Management. In conclusion, they proposed 84 ways to implement in an Agile way project management and 38 work products as part of the strategy.
A Maturity Model for Scaling Agile Development (Stojanov, Turetken, & Trienekens, 2015)	In this paper, they developed a maturity model for adopting Agile and SAFe practices. Taking an existing Agile maturity model as the basis, they extended the model with practices that are key to scaling Agile practices for the SAFe. The model is developed and refined using a Delphi study.
Agile Compass: A Tool for Identifying Maturity in Agile Software-Development Teams (Fontana, Reineher, & Malucelli, 2015a)	The authors of this paper found that teams accomplish Agile maturity via a dynamic evolution based on the pursuit of specific outcomes. Seven outcomes categories are identified and organized in a framework.
Progressive Outcomes: A framework for maturing in Agile software development (Fontana, Reineher, Meyer, & Malucelli, 2015b)	In this paper, a Progressive Outcomes framework is proposed to describe the Agile development maturing process. It is a framework in which people have the central role, ambidexterity is a key ability to maturity, and improvement is guided by outcomes Agile teams pursue, instead of prescribed practices
An impact-oriented maturity model for IT-based case management (Koehler, Woodtly, & Hofstetter, 2015)	A CRM maturity model for IT-based case management is proposed. The different maturity levels are related to sets of capabilities which are typical for case management in different sectors. The model focuses on the impact of technology and is related to a map of benefits and risks.
DevOps shifting software engineering strategy-value based perspective (Mohamed, 2015)	In this paper, a new DevOps maturity model is introduced and analysed how the model will impact Global Software Engineering practices and processes. Additionally, it introduces a transformation model that helps in adapting the DevOps strategy.

DevOps - The Future of Application Lifecycle Automation (Menzel, & Macaulay, 2015)	In this whitepaper, they introduced the Capgemini's DevOps maturity model (DMM) according to three dimensions of people, processes and tools. It enables a business to indicate their maturity level and how the business can improve it.
DevOps Quick Guides (Eficode Oy, 2015)	Oy introduces two different types of DevOps Quick Guides: management and professionals. In both guides, a maturity model is introduced which is based on five dimensions and four maturity levels.
The disciplined Agile framework: A pragmatic approach to Agile maturity (Ambler, & Lines, 2016)	A maturity-based approach was developed which describes four development stages of an industrial service company from several views. The maturity model makes it possible to develop a digital roadmap that is tailormade to each company.
Simplifying the DevOps Adoption Process (Bucena, & Kirikova, 2017)	The paper identified the challenges of DevOps adoption and proposed a method for it. Additionally, they developed a DevOps maturity model based on an analysis of related work. It includes five levels of maturity according to four areas: technology, process, people, and culture.
DevOps competencies and maturity for software producing organizations (Feijter, Overbeek, van Vliet, Jagroep, & Brinkkemper, 2018)	A DevOps Competence Model showing an overview of the areas to be considered in adopting DevOps is proposed in this article. Additionally, a Maturity Model is proposed that presents a growth path for software producing organisations.

D. Development outcomes of succeeding with Agile

The development outcomes of four case studies by Scaled Agile are mentioned in Table 13.

Table 13: Development outcomes of succeeding with Agile

Case study	Development Outcomes
Deutsche Bahn	• Lead time dropped from 12 months to 4 months • Raised employees' satisfaction levels and greater collaboration among teams • Test automation coverage improved from 30% to 85%
Center for Medicare & Medicaid Services	• The number of help desk tickets decreased by 55% • Employee satisfaction increased by 27% • The full budget from maintenance is divided into 40% maintenance and 60% innovation
Dutch Tax and Customs Administration	• Three times more often major releases • Reduction in technical debt with 80% • 50% of the managers moved into different roles • Greater collaboration in the whole organisation and an increase in employee engagement
Murex	• Production-like testing 10 times faster • Reduction in the user story cycle time with 85% • Internal test management system time to release dropped from 37 to two man-days.

E. Individual results of the relation between development outcomes and competencies'

Figure 16: Professional 1 – Relation between development outcomes and competencies

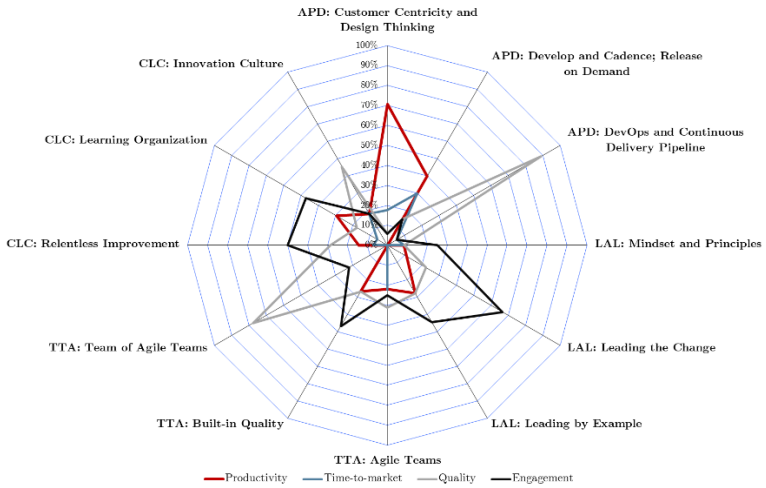


Figure 17: Professional 2 – Relation between development outcomes and competencies

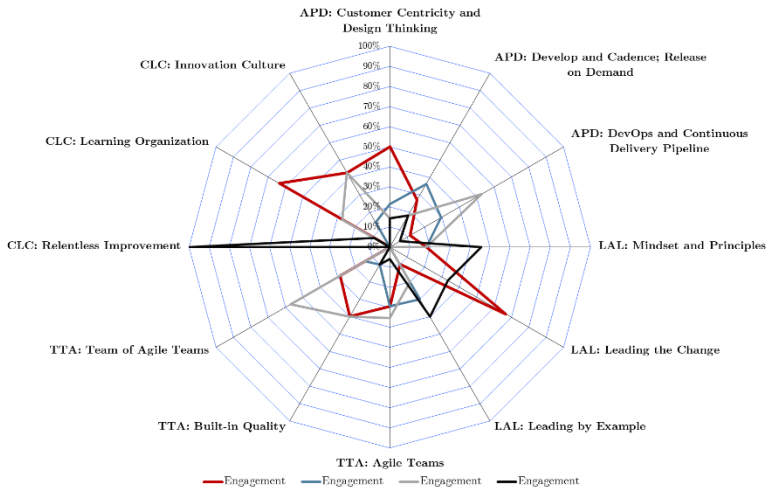


Figure 18: Professional 3 – Relation between development outcomes and competencies

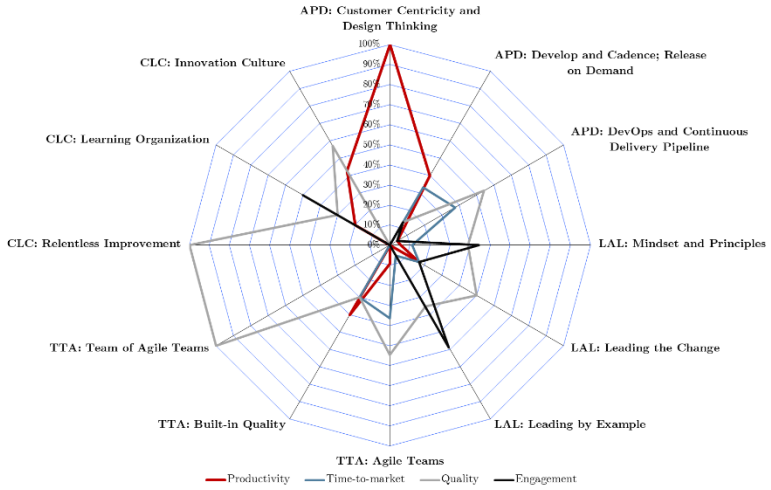
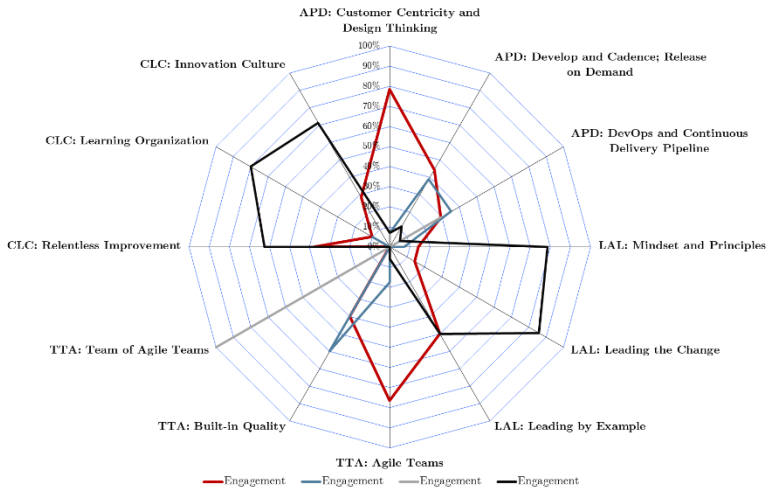


Figure 19: Professional 4 – Relation between development outcomes and competencies



F. The core competencies categorized by growth stage and development outcome

Agile Product Delivery

Statements	Dimension	Growth stage	Productivity	Time-to-market	Quality	Engagement
Capacity allocations balance investments in new features, architectural enablers, technical debt, and maintenance	Customer Centricity and Design Thinking	1. Core	x	x	x	
Market segments are clearly defined	Customer Centricity and Design Thinking	1. Core	x			
User personas guide design choices	Customer Centricity and Design Thinking	1. Core	x			
Story maps are used to design workflows	Customer Centricity and Design Thinking	1. Core	x			
The PI roadmap forecasts features over a few PIs	Customer Centricity and Design Thinking	1. Core		x		
The solution roadmap forecasts one to three years of functionality	Customer Centricity and Design Thinking	2. Intermediate	x	x		
To understand customer needs, Continuous Exploration includes market research	Customer Centricity and Design Thinking	2. Intermediate	x			
Gemba experiences provide direct insight into user needs	Customer Centricity and Design Thinking	2. Intermediate	x			
The vision guides solution and PI roadmaps	Customer Centricity and Design Thinking	2. Intermediate	x			
The solution value proposition is known and documented	Customer Centricity and Design Thinking	2. Intermediate	x			
Solution features are aligned with portfolio strategy	Customer Centricity and Design Thinking	2. Intermediate	x			
To align features to market segments, a whole-product model is used	Customer Centricity and Design Thinking	2. Intermediate	x			

Empathy maps help teams understand user needs	Customer Centricity and Design Thinking	2. Intermediate	x			
Journey maps describe customer experiences across the operational value stream	Customer Centricity and Design Thinking	2. Intermediate	x			
The program Kanban visualizes feature flow and manages Work in Process (WIP)	Develop on Cadence; Release on Demand	1. Core	x	x		
Weighted Short Job First (WSJF) helps prioritize features	Develop on Cadence; Release on Demand	1. Core	x	x		
Features are prioritized are socialized ahead of PI Planning	Develop on Cadence; Release on Demand	1. Core	x	x		
A program board is used to track feature planning and dependencies	Develop on Cadence; Release on Demand	1. Core	x	x		
Inspect & Adapt (I&A) identifies opportunities for relentless improvement at every PI	Develop on Cadence; Release on Demand	1. Core	x		x	
Feature benefits are clearly defined	Develop on Cadence; Release on Demand	1. Core	x			
Business owners accept and rank PI objectives by business value	Develop on Cadence; Release on Demand	1. Core	x			
Business value is assessed at the end of the PI	Develop on Cadence; Release on Demand	1. Core	x			
PI Planning is effective, routine and cadence based	Develop on Cadence; Release on Demand	1. Core		x		
Iterations are time-boxed to two weeks or less	Develop on Cadence; Release on Demand	1. Core		x		
A system demo is held at the end of every Iteration	Develop on Cadence; Release on Demand	1. Core			x	x
I&A improvement items are addressed	Develop on Cadence; Release on Demand	1. Core			x	
All stakeholders and team members participate in PI Planning personally, or through agreed proxies	Develop on Cadence; Release on Demand	1. Core				x
ART stakeholders, Business Owners, and shared services participate in PI Planning	Develop on Cadence; Release on Demand	1. Core				x
The IP iteration is dedicated to planning, innovation, and learning	Develop on Cadence; Release on Demand	2. Intermediate	x	x	x	x

Continuous exploration, continuous integration, and continuous deployment occur in all iterations	Develop on Cadence; Release on Demand	2. Intermediate	x	x	
The Agile Release Train (ART) consistently delivers between 80-100% of business value	Develop on Cadence; Release on Demand	2. Intermediate	x		
Releases exploit market rhythms and events	Develop on Cadence; Release on Demand	2. Intermediate		x	
The solution can be released at any time	Develop on Cadence; Release on Demand	3. Advanced		x	
Code is built automatically on every code commit	DevOps and the Continuous Delivery Pipeline	1. Core		x	x
Security considerations are included throughout the CD pipeline and release processes	DevOps and the Continuous Delivery Pipeline	1. Core			x
Build automation tools automate the compilation of all source code	DevOps and the Continuous Delivery Pipeline	1. Core			x
All changes to production systems are managed through version control	DevOps and the Continuous Delivery Pipeline	1. Core			x
At each step in the CDP, process time, lead time, delay time, and percent complete and accurate are measured	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	x
The CDP is continuously optimized	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	x
Recovery processes are operational and verified	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	x
Trunk-based development reduces code branches	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	x
Solutions can be deployed into production without being released to end users	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	
The Continuous Delivery Pipeline (CDP) workflow is clearly defined	DevOps and the Continuous Delivery Pipeline	2. Intermediate			x
Automated tests are used to minimize manual testing	DevOps and the Continuous Delivery Pipeline	2. Intermediate			x
Code analysis and inspection tools examine code and third-party packages for known security vulnerabilities	DevOps and the Continuous Delivery Pipeline	2. Intermediate			x

The solution is tested end-to-end in the staging environment	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	
The staging environment used for end-to-end verification simulates the production environment	DevOps and the Continuous Delivery Pipeline	2. Intermediate		x	
Teams have shared responsibility for development, deployment, and operations	DevOps and the Continuous Delivery Pipeline	2. Intermediate			x
Production code can be released to specific users	DevOps and the Continuous Delivery Pipeline	3. Advanced	x		
Solutions can be promoted continuously from staging into production	DevOps and the Continuous Delivery Pipeline	3. Advanced		x	

Statements	Dimension	Growth stage	Productivity	Time-to-market	Quality	Engagement
Leaders consistently invest in the education and professional growth of their teams	Leading by Example	1. Core	x			x
Leaders prepare their teams for increased decision-making authority, investing in their technical competence and providing organizational clarity	Leading by Example	1. Core		x	x	
Leaders openly admit and own their mistakes	Leading by Example	1. Core			x	x
Leaders continuously invest in their own learning	Leading by Example	1. Core			x	x
Leaders promote a positive, performance-oriented culture based on trust, respect, expertise, engagement, and commitment to organizational goals	Leading by Example	1. Core				x
Leaders act with honesty and integrity	Leading by Example	1. Core				x
Leaders are authentic – their words, actions and beliefs are aligned	Leading by Example	1. Core				x
Leaders demonstrate self-awareness and management of their emotions	Leading by Example	1. Core				x
Leaders show empathy towards others	Leading by Example	1. Core				x
Leaders manage intense emotional situations skillfully	Leading by Example	1. Core				x
Leaders consistently move decision authority to teams that have the best information and context	Leading by Example	2. Intermediate		x	x	
Leaders invest in the training necessary to build the knowledge and understanding teams and individuals need to perform Lean, Agile, and SAFe responsibilities	Leading the Change	1. Core	x		x	x
Leaders clearly communicate when and why change is needed	Leading the Change	1. Core	x			x

Leaders express the vision for change in ways that inspire, motivate, and engage employees/teams to make the change succeed	Leading the Change	1. Core	x		x
Leaders set the example by investing in their own training in Lean, Agile, and SAFe	Leading the Change	1. Core		x	x
Leaders use personal advocacy and drive to lead change versus positional authority	Leading the Change	1. Core			x
Leaders create a safe environment for change that supports risk-taking without fear of consequences to self-esteem, status, or career	Leading the Change	1. Core			x
Leaders use the Implementation Roadmap to guide the adoption of SAFe	Leading the Change	1. Core			x
Leaders provide sufficient SAFe Program Consultants (SPCs) to support the organization's SAFe implementation	Leading the Change	1. Core			x
Leaders form cross-domain guiding coalitions and empower them to plan and guide the change	Leading the Change	2. Intermediate	x		x
Leaders follow sound Organizational Change Management (OCM) practices	Leading the Change	2. Intermediate			
Leaders demonstrate a growth mindset by showing openness to new ideas, seeing challenges as a growth opportunity, and being receptive to feedback	Mindset and Principles	1. Core	x		x
Leaders identify and overcome existing fixed mindsets that need to be challenged in order to embrace new ways of working	Mindset and Principles	1. Core		x	x
Leaders fulfill their responsibilities by exemplifying the core values of alignment, transparency, built-in quality, and program execution	Mindset and Principles	1. Core		x	x
Leaders exemplify Lean principles by focusing on value, respect for people and culture, flow, innovation, and relentless improvement	Mindset and Principles	1. Core			x
Leaders live the values and principles of the Agile Manifesto	Mindset and Principles	1. Core			x
Leaders exhibit and teach the 10 SAFe Lean-Agile Principles	Mindset and Principles	2. Intermediate			x

Team and Technical Agility

Statements	Dimension	Growth stage	Productivity	Time-to-market	Quality	Engagement
Teams implement small, estimated, functional, vertical user stories that fit in an iteration	Agile Teams	1. Core	x	x		
Stories are completed throughout the iteration with multiple define-build-test cycles	Agile Teams	1. Core		x	x	
Teams reliably meet 80-100% of PI Objective business value	Agile Teams	2. Intermediate	x		x	
Business teams operate with specialized Lean-Agile practices	Agile Teams	2. Intermediate		x	x	
Teams have the cross-functional skills needed to define, build, test, and deploy value	Agile Teams	3. Advanced	x	x	x	
Scrum Masters reinforce Agile behaviors, facilitate effectively, help teams address challenges, and improve performance	Agile Teams	3. Advanced	x	x		
Agile technical teams are provisioned and trained	Agile Teams	3. Advanced	x		x	
Agile business teams are provisioned and trained	Agile Teams	3. Advanced	x		x	
Teams execute standard iteration events	Agile Teams	3. Advanced	x		x	
Product Owners facilitate user story development, prioritization, and acceptance criteria	Agile Teams	3. Advanced	x			
Teams plan, demonstrate, and deliver value in short iterations	Agile Teams	3. Advanced		x		
Using Kanban and WIP limits, teams manage flow visually	Agile Teams	3. Advanced		x		
Scrum Masters are provisioned and trained	Agile Teams	3. Advanced			x	
Product Owners are provisioned and trained	Agile Teams	3. Advanced			x	

Dedicated Product Owners support no more than two teams	Agile Teams	3. Advanced			x	
By implementing retrospective improvements, teams improve relentlessly	Agile Teams	3. Advanced			x	
Team members would recommend their team to a friend	Agile Teams	3. Advanced				x
Teams share responsibility for design	Built-in Quality	1. Core			x	
Teams foster cross-training and T-shaped skills	Built-in Quality	1. Core			x	
Continuous integration and automated tests run at team and system levels	Built-in Quality	1. Core			x	
Development teams use Agile technical practices: TDD, BDD, Agile architecture, refactoring, spikes	Built-in Quality	2. Intermediate			x	
Teams reduce technical debt in each iteration	Built-in Quality	2. Intermediate			x	
All Agile teams apply general quality practices: flow, peer review, collective ownership, standards, automation, Definition of Done (DoD)	Built-in Quality	3. Advanced			x	
Business teams apply domain-specific quality practices: marketing, finance, HR, etc.	Built-in Quality	3. Advanced			x	
ARTs have all the skills necessary to deliver business solutions	Team of Agile teams	1. Core	x	x		
Portfolio stakeholders provide constant engagement with their ARTs	Team of Agile teams	1. Core	x			x
ARTs are organized around value and cross-organizational silos	Team of Agile teams	1. Core	x			
System teams are effective in their roles	Team of Agile teams	1. Core			x	
ARTs include everyone required to deliver and support business solutions	Team of Agile teams	3. Advanced	x			
Business Owners provide vision and align the Agile Release Train (ART) to enterprise strategy	Team of Agile teams	3. Advanced	x			
Dedicated and trained Release Train Engineers (RTEs) program events effectively	Team of Agile teams	3. Advanced			x	
Dedicated and trained Product Management develop and prioritize the feature backlog	Team of Agile teams	3. Advanced			x	
Dedicated and trained System Architects work with teams to extend the architectural runway	Team of Agile teams	3. Advanced			x	
ARTs are aligned with portfolio strategy	Team of Agile teams	3. Advanced				

Continuous Learning Culture

Statements	Dimension	Growth stage	Productivity	Time-to-market	Quality	Engagement
The organization cultivates the courage and aptitude for innovation and encourages employee risk-taking	Innovation Culture	1. Core	x			x
The organization cultivates innovations by teams and Agile Release Trains (ARTs) as inputs to portfolio vision and strategy	Innovation Culture	1. Core			x	x
The organization provides clear paths for advancement to employees who demonstrate exceptional performance as innovation change agents	Innovation Culture	1. Core				
The organization provides physical spaces conducive to innovation activities	Innovation Culture	1. Core				x
Leaders train, encourage, and coach intrapreneurship and innovation	Innovation Culture	2. Intermediate	x			x
Teams have regular opportunities to see first-hand how customers interact with the organization's products and services	Innovation Culture	2. Intermediate			x	
The organization creates and protects regular time for employees to devote to creative, exploratory activities	Innovation Culture	3. Advanced	x			x
The organization promotes learning and exploration through experimentation without fear of failure	Innovation Culture	3. Advanced	x			x
When customer feedback dictates a change in strategy, the organization pivots without blame or consideration of sunk costs	Innovation Culture	3. Advanced	x			x
Leaders create an environment that supports creative thinking, curiosity, and challenging the status quo	Innovation Culture	3. Advanced				x

Teams work collectively to achieve common objectives by sharing knowledge, solving problems, and learning together	Learning Organization	1. Core			x	
The organization invites employees to share in and contribute to a common view of the future	Learning Organization	1. Core				x
Teams use the SAFe mental models to build a shared understanding of the Lean-Agile way of working	Learning Organization	1. Core				x
The organization encourages employees to challenge the status quo	Learning Organization	2. Intermediate			x	x
Teams defer group and personal goals for the greater good of the organization	Learning Organization	2. Intermediate				x
The organization empowers employees to gain knowledge and experience in multiple disciplines	Learning Organization	3. Advanced	x		x	x
The organization invests in the growth of employees	Learning Organization	3. Advanced			x	x
The organization continuously creates, acquires, shares, and transfers knowledge	Learning Organization	3. Advanced				x
Experiments routinely enable the organization to 'learn its way' to the most promising answers to problems	Relentless Improvement	1. Core	x			
Improvements optimize the end-to-end flow of value	Relentless Improvement	2. Intermediate	x	x		
Improvement efforts are based on facts and data over opinions and conjecture	Relentless Improvement	2. Intermediate			x	
The organization gives improvement activities priority, visibility, and resources	Relentless Improvement	3. Advanced	x			x
Individuals and teams are given the time and resources to identify and solve problems	Relentless Improvement	3. Advanced			x	
Teams at every level of the organization pause regularly to reflect and improve	Relentless Improvement	3. Advanced			x	
Problem-solving is engrained in the organizational culture	Relentless Improvement	3. Advanced				