

Definition of ready in practice

Why Agile Scrum teams use a definition of ready in practice and how they create it.

Mark van der Meer, 7672187

A thesis presented for the degree of
Master of Science

Utrecht University
Faculty of Science
Business Informatics
Gerard Wagenaar & Fabiano Dalpiaz

Abstract

This paper describes a case study on the rationale of using definition of ready at an Agile Scrum team, in combination with how teams create, maintain and apply their definition of ready. Conducted qualitative interviews showed that definition of ready can mitigate some Agile Scrum deviations. However, teams are inclined to use it to improve their process of refining requirements, by applying it to a template for their requirements. A definition of ready is often created through brainstorm sessions, and usually solely maintained when something went wrong during a sprint, with improvements being clarifications to existing criteria or additional criteria.

Contents

1	Introduction	4
1.1	Problem statement	5
2	Research Questions	6
2.1	Main research question	6
2.2	Sub-questions	6
3	Research Methods	8
3.1	Literature review	8
3.1.1	Literature review approach	8
3.1.2	Exploratory search	8
3.1.3	Initial search	9
3.1.4	Inclusion- and exclusion criteria	9
3.1.5	Analysis	9
3.2	Qualitative Case Study	10
3.2.1	Qualitative interview	10
3.2.2	Sampling strategies	10
3.2.3	Questionnaire	11
3.2.4	Phases of the interview	12
3.2.5	Analysis of the results	12
3.2.6	Transcribing	13
3.3	Validity threats	14
3.3.1	Interpretive validity	14
3.3.2	Descriptive validity	14
3.3.3	Theoretical validity	14
3.3.4	Generalizability	15
3.3.5	Evaluative validity	15
4	Results	16
4.1	Literature Review	16
4.2	Case Study	18
4.2.1	Rationale	18
4.2.2	Creating a definition of ready	20
4.2.3	Applying definition of ready	20
4.2.4	Mitigating common Agile Scrum deviations	21
4.2.5	maintaining definition of ready	22
5	Discussion	23
5.1	Application	23
5.2	Interpretation	23
5.3	Limitations	24
6	Conclusion	25
7	Acknowledgements	26
A	- Appendix: Analysed papers semi-systematic literature review	30
A.1	Agile Scrum deviations	30
A.2	ScrumBut	32
A.3	Agile Scrum variation	33
A.4	Scrum in practice	35

B	- Appendix: Found deviations	36
B.1	Estimation	36
B.2	Breakdown	36
B.3	Assignment	37
B.4	Sprint backlog creation	37
B.5	Product backlog creation	38
B.6	Product backlog refinement and prioritisation	39
C	- Appendix: Found deviations	40
C.1	Deviations	40
C.2	Application moment	40
C.3	Creating definition of ready	40
C.4	Organisations which do not use definition of ready	41
C.5	Other	41
C.6	Reason for using definition of ready	41
C.7	Reason for not using definition of ready while a team has one	42
C.8	Moment to reflect	42
C.9	Application DoR	42
C.10	Improvements to DoR	43
D	- Appendix: Interview Protocol	44
D.1	Introduction	44
D.2	Participation consent	44
D.3	Interview questions	44
D.3.1	Situational questions	44
D.3.2	Open questions	45
D.3.3	End of the interview	45

1 Introduction

The original project method for software system design originates from different domains, mainly industrial and civil engineering [12]. Therefore, it is not surprising that project management for developing software systems has changed [12]. The result were Agile project management methods. There are multiple types of Agile methods: Scrum, Extreme Project Management, Dynamic Project Management and Adaptive Project Management [12].

This research is performed in the context of the project management method Agile Scrum. The argument to use Agile Scrum in software development is that developing software is a complex process, where many factors can influence the final result [38], and it is therefore difficult to plan ahead [14, 38, 27]. Agile Scrum provides a solution to this complexity, by being an iterative and incremental process in which requirements and solutions evolve through collaboration between self-organizing and cross-functional teams [7, 29]. To adapt to these evolving requirements and solutions, Agile Scrum uses adaptive planning, evolutionary delivery, a time-boxed iterative approach, and flexible response to change [14].

Agile Scrum is performed by a scrum team, consisting of three roles: the scrum master, product owner and development Team [45]. An Agile Scrum concept related to this research is the concept of sprints. Sprints are iteration cycles of a predetermined length in which the functionality of a product is further developed [12]. The functionalities which the developers will work on in a sprint are planned at an Agile Scrum activity called the sprint planning. The sprint planning is held at the beginning of every sprint, in which the sprint goal will be determined. The sprint backlog is the formal outcome of a sprint and is a list of requirements which will be finished by the end of the sprint [12].

To verify if a requirement is ready to be placed into a sprint, the concept "definition of ready" was created, which is defined as a set of criteria adopted by a scrum team to help them remember things they need to do before work starts on a requirement [44, 16, 9, 42]. Definition of ready does not only have to be applied to requirements, as noticed by Power, who describes that definition of ready could also be used in sprints and software releases.

To provide an understanding of what criteria are used in a definition of ready, a few examples of these criteria, taken from [44], are: clear and testable acceptance criteria, the team understands how to demonstrate the finished user story at the sprint review, and the requirement is estimated small enough to comfortably be completed in one sprint.

1.1 Problem statement

Successful software projects are projects which meet their budget, delivery, and business objectives [32]. Successful software projects are however rare [11, 27]. In literature, different causes are given for this phenomenon [11, 29, 20, 32, 17, 27].

A project can be unsuccessful due to inadequate requirements [32, 11]. According to Cerpa and Verner, inadequate requirements have as consequence that customers have unrealistic expectations of the result. In addition, inadequate criteria could result in the underestimation of requirements, causing a project to fail [11, 20]. Criteria defined in a definition of ready can assist as a measure on whether a requirement is refined in an inadequate way, or that a requirement is sufficiently refined.

A vague project scope is another possible origin of a failing software project [11, 17]. The definition of ready can be proposed as a mitigating tool for a vague project scope, as complete requirements can provide project stakeholders with a sufficient overview to assess whether the project's scope is set correct or requires adjustments.

The rationales presented above are however not proven facts, an empirical study, or are results found during a review of existing literature. They are assumptions, as there is limited scientific literature available on definition of ready and why or how it is applied in practice.

This lack of scientific literature on why definition of ready is used, and how it is used by scrum teams, was the incentive for performing this research. Why do scrum teams use a definition of ready and how it is used are interesting to research, as different literature mentions that adjustments are made by scrum teams to the Agile Scrum process [22, 3]. It is for instance possible that scrum teams deviate from Agile Scrum by not using definition of ready as it was rarely consulted or was not actively maintained and therefore did not provide value for the scrum team.

The insights from this research could influence a scrum team on how they use Agile Scrum, as for instance, on a regular basis software projects fail because one of the aforementioned causes occurs, and the decision has to be made whether to use definition of ready to partly mitigate this issue. At this moment, a lack of scientific evidence is available to understand whether this could help the scrum team and how they could apply a definition of ready in their processes.

2 Research Questions

To resolve the problem statement, chapter 1.1, research questions are created. A main research question has an overarching function, with more detail in underlying sub-questions. Between two sub-questions is a dependency, sub-question 3 and 4, where the result of sub-question 3 is used to answer sub-question 4.

2.1 Main research question

Why do Agile Scrum teams use a definition of ready, and how do they create it?

The phrasing of this question could imply that Agile Scrum teams, which are using a definition of ready, are examined. However, Agile Scrum teams who are not familiar with definition of ready are also examined, to understand what alternatives they use, what challenges they possibly face by not using definition of ready, and whether they would be interested to use definition of ready and what their rationale would be for this decision.

2.2 Sub-questions

The main research question discussed in chapter 2.1 is divided into five sub-questions. These five sub-questions are stated below, including a description of what value the questions adds to the main research question.

RQ1. *What is the rationale behind using a definition of ready for an Agile Scrum team?*

The first research question has as goal to understand why Agile Scrum teams use a definition of ready, or why they do not. What issues an Agile Scrum team has when not using definition of ready, and how it helps them solve those issues.

On the contrary, Agile Scrum teams which are not using definition of ready could explain why they are not using it, and whether they have not encountered issues that could be solved by definition of ready, or if they have not reacted to those issues, yet.

RQ2. *How is a definition of ready created and maintained by an Agile Scrum team?*

The second research question's purpose is to understand the process of creating a definition of ready, and who's involved in its making. For instance, are product owners formulating a definition of ready by themselves, or does a scrum team have brainstorm meetings to produce one?

Furthermore, this research question includes whether they maintain their definition of ready, and if so, when they reflect on the current criteria and possibly update it according to new findings.

RQ3. *What Agile Scrum deviations exist at Agile Scrum teams according to scientific literature?*

Sub-question three is a question to explore what deviations in Agile Scrum exist in practice, through a scientific literature review. This question is not directly related to our main research question, and is a preparation for sub-question four, where discovered deviations are used to answer it.

RQ4. *Can definition of ready mitigate common Agile Scrum deviations related to the content of requirements?*

Sub question three has a list of common Agile Scrum deviations as result, which were filtered to only have ones which are related to the content of requirements, something which definition of ready has an influence on. Filtering deviations reduced the scope of this sub-question, to make it feasible for this research. By presenting these deviations as situations in interviews, it was examined whether these deviations could be solved, or were already solved by using a definition of ready.

RQ5. *When do members of an Agile Scrum team use their definition of ready in the development process?*

Research question 5 is used to define when a definition of ready is used in day-to-day processes. For instance, is a definition of ready consulted when creating a new user story, or during a sprint planning to assess if stories are ready to be planned for a new sprint?

Alternatives for Agile Scrum teams which are not using definition of ready are documented within this question, as overlap exists. Chapter [4.2](#) will provide a further elaboration on this phenomenon, along with the other results from the case study.

3 Research Methods

A mixed research method will be used, combining literature review and case study. This chapter describes both methods, how they will be executed, and the threats that can occur, including strategies to mitigate those threats.

3.1 Literature review

A literature review is conducted to retrieve deviations, present in scientific literature, which occur in Agile Scrum at software companies, to answer sub-question four. The approach of this literature review is based on a paper from [Snyder](#). This chapter covers the rationale for the chosen approach and explains the steps taken during the literature review.

3.1.1 Literature review approach

[Snyder](#), discussed three different approaches to a literature review. These are: systematic, semi-systematic and integrative literature review. The semi-systematic literature review approach will be used out of the three.

The semi-systematic literature approach allows for a broad research question [49], in contrast to a systematic literature review approach. The research question answered by the literature review is which deviations of Agile Scrum at software teams exist in scientific literature. This question could be seen as broad, as there are multiple possible answers to the question, making the semi-systematic approach desired for this research.

Furthermore, the semi-systematic literature review approach uses research articles, which corresponds to the 'scientific literature' mentioned in the research question. This is in contrast with the integrative literature review approach which allows for books, and other published texts besides research articles [49]. The systematic literature review approach contains a quantitative approach [49], which is not suitable for this literature review, because it is supposed to gather qualitative data on deviations in Agile Scrum, rather than quantitative data, as the literature review looks at experiences of deviations from Agile Scrum at software teams.

The semi-systematic literature review approach can be useful for detecting themes [49], which is in accordance with this literature review's goal to find themes in Agile Scrum deviations at software teams.

3.1.2 Exploratory search

An exploratory search is first performed to identify available documents and understand whether a sufficient number of documents are available to use for a literature review, using the search term 'Agile AND Scrum AND Deviations' in Scopus. The decision to go with Scopus for the initial and regular search is its database, which contains the titles and abstracts of major academic databases, like IEEEEXplore, Elsevier and Springer [34]. Furthermore, Scopus provides filters to be able to for instance show papers from a certain period, document type (e.g., conference paper, article) or subject area.

3.1.3 Initial search

After the exploratory search, a list of terms is formulated to use as queries in Scopus for the regular search. The queries are only applied to the title, abstract and keywords of a document, and are the following:

"Agile AND Scrum AND Deviation", "ScrumBut", "Agile Scrum Variation*", "{Scrum in Practice}"*

The AND-operator guarantees that only documents are found that contain all of the terms. The asterisk operator (*) can replace multiple characters in a word. For instance, with "deviation*", the singular and plural for the word "deviation" are covered. The curly braces operator ({}) assures that the sentence or combination of words has the same sequence as written down between the curly braces [1].

3.1.4 Inclusion- and exclusion criteria

Inclusion and exclusion criteria are used to identify useful papers for the literature review. A part of the inclusion criteria are applied through Scopus' filters, and another part are applied when screening found articles. The applied inclusion criteria are partly inspired by a research of [Mäntylä et al.](#), and are the following:

- Papers with the main focus on the Agile Scrum process.
- Papers with the main focus on deviations in Agile Scrum.
- Papers with the main focus on ScrumBut.
- Papers with the main focus on Agile Scrum by the book.

The exclusion criteria are also inspired by the research of [Mäntylä et al.](#), and are the following:

- Papers about an opinion with a lack of empirical data.
- Papers older than 10 years to use papers that are relatable to the current use of Agile Scrum at teams developing software.

3.1.5 Analysis

A selection of 11 papers for the literature review remained after applying inclusion and exclusion criteria, and these papers were scanned for deviations. Categories taken from [35] are used to group discovered deviations. When deviations between papers are similar, one formulation is taken, with both papers referenced as source. Reasons for each deviation are also scanned in the papers, to provide context, and verify whether the deviation is interpreted in a correct way. However, not every paper includes rationales as to why a deviation occurred, and therefore some are left blank in the tables of appendix B.

3.2 Qualitative Case Study

A qualitative case study involves analysis of persons, communities or organisations, to provide the opportunity to gain a deep view of the research problem, and could facilitate describing, understanding and explaining a research problem or situation [6]. This suits the research well, as organisations, in this case scrum teams, were analysed to understand how they use definition of ready, and what their rationale is for using or not using one. Sub-question one, two and five are answered by the case study.

3.2.1 Qualitative interview

Qualitative interviews are held as they are more flexible than other qualitative data collection techniques [4]. This provides flexibility to the interviews when a question is not prepared, but is nonetheless interesting to ask to the interviewee.

The flexibility of an interview can be three types: structured, semi-structured or unstructured [4]. Semi structured interviews are held for this research, as an overview of the definition of ready is known, and questions regarding the definition of ready can therefore be formulated. Moreover, semi-structured interviews provide the opportunity to increase the depth and richness of the interviewees' responds [4], as interviews could deviate from the prepared questions and let an interviewee elaborate further on their answers.

3.2.2 Sampling strategies

The scope of this research is software development teams working with Agile Scrum, with experts from these teams being interviewed. It is not required for the team or company, of which the expert is part, to be using a definition of ready, as it provides incentive why not using a definition of ready does not provide (additional) business value.

Two strategies are used during the course of this research. As first method, purposive sampling is used, where participants are selected based on their ability to suit the purpose of the study [24]. For this research, participants are required with a certain understanding of Agile Scrum, definition of ready and how definition of ready is used in their organisation.

For purposive sampling, criteria are set to verify whether a potential participant could be part of the research (table 1).

In addition to purposive sampling, snowball sampling is used. This method uses existing participants to recruit future participants from other experts they know [8]. Participants gained through snowball sampling were examined by the same criteria.

A criterion from [46] is used to determine when enough participants are interviewed, being saturation of information taking place, where results from interviews will become similar to what was discussed in previous interviews and new information is no longer gained from doing more interviews.

No.	Criteria	Rationale
1.	The participant is a software developer, tester, product owner, scrum master or a combination of the aforementioned roles.	Aforementioned roles can be involved in the Agile Scrum process, as they interact with requirements throughout their work. Testers are not specifically mentioned in scrum, but they have to understand what has to be tested to verify a requirement.
2.	The participant should at least have worked at the same software company for half a year, and four sprints in a team with the same members.	The participant has an understanding of how the organisation operates, the things related to the development of software that go well, and the things related to the development of software that could be improved upon.
3.	The software company mentioned in criteria 2 uses or used the Agile Scrum methodology during the time the participant was active at the company.	This provides the participant with an understanding of how Agile Scrum works and knowledge about potential improvements within the software development process of the company.
4.	The scrum team in which the participant works or has worked consisted of at least 5 members, including a scrum master and product owner, during a time span of at least 4 sprints.	This minimum is to ensure that it is a complete scrum team [40].
5.	Participant has at least one year of experience in their respective role.	The goal of this criteria is to ensure that participants have gained experiences to reflect upon during the interview and could give their view on the definition of ready in combination with those experiences.

Table 1: Criteria for purposive sampling

3.2.3 Questionnaire

Similar to a research by [Shastri et al.](#), this research includes a questionnaire prior to the interview, to be able to form a picture of the project(s) the interviewee is involved in and perhaps adjust interview questions accordingly.

Furthermore, the questionnaire saves time by having background knowledge on the interviewee. As a result, more time is reserved for deeply exploring the rational behind using a definition of ready and how it is used in the software company where they work. The data retrieved from this questionnaire will be quantitative. However, no quantitative analysis will be done during the research, as the data is solely used to supplement interviews.

Through the answers to the questionnaire is also verified whether an interviewee is eligible to participate according to the criteria set in table 1.

3.2.4 Phases of the interview

The structure of the interview is split into five phases, according to a recommendation made by [Robson](#):

1. Introduction: the interviewer and the aim of the interview are introduced.
2. Warm-up: begin the interview with easy questions to ease the situation from the beginning.
3. Main body: focus on the main topic of the study.
4. Cool-off: once more, easy questions which will conclude the interview.
5. Closure: the interviewee is thanked for his or her contribution to the research.

These recommendations are followed with the structure of the interview, by starting with situations based deviations found in the literature review. These were considered 'easy questions' for the warm-up, as the interviewee has to answer whether the situation has occurred or if it could occur in the future.

The main body of the interview contains open questions focused on definition of ready. Answers given at situational questions are reflected upon during these open questions, and a further analysis on how and why definition of ready is used and maintained, is performed.

The goal for the cool-off phase is to end the interview with a 'fun' question. Depending on whether an interviewee's team has a definition of ready, the question is how they could improve the usage of their definition of ready, or whether they are going to implement definition of ready in their current or future project(s) and how. The aim is that they use their imagination, and apply discussed topics from the interview in the context of their own team. The protocol for the interview, including questions, is available in [appendix D](#).

3.2.5 Analysis of the results

For analysing the interviews, thematic coding is used, as [\[4\]](#) states that 'the thematic coding approach is the most heavily employed analytical tool in social research', which shows that thematic coding has been used extensively and has proven itself. Furthermore, the method is best suiting to the research, as the thematic approach offers a way to systematically analysis qualitative data, which can be linked to conceptual issues [\[13\]](#). To assist with the thematic coding, Nvivo is used to code the qualitative data from the interviews.

Other common approaches would be a quasi-statistical approach, or a grounded theory approach [\[4\]](#). The quasi-statistical approach is described as qualitative coding where text is divided into categories and examined statistically to find connections among the categories [\[15\]](#). Discovering connections is not a goal in this research, and as a result, the quasi-statistical approach was not chosen to analyse the qualitative data from the interviews.

The grounded theory approach has as main goal to develop new theories from qualitative data [\[51\]](#). This research is not meant to create a new theory. It attempts to create a better understanding of an existing theory in practice, the definition of ready. Therefore, the grounded theory was likewise not chosen as method to analyse the data.

3.2.6 Transcribing

Interviews are transcribed, as transcribing improves the research's transparency [4, 25], and allows the data to be used for other research purposes [4]. Transcribing is performed by the researcher himself, because this provides coherence between the research approach and approach to transcribing, as well as the researcher's confidence that the written transcript is an accurate record of the interview that took place [37]. Transcribing furthermore helps to reflect on interviews.

An AI transcription tool could have also been used to automatically generate transcripts from audio recordings. It is however unknown how accurate these tools are able to transcribe recordings, which results in having to manually check transcripts for accuracy, and fill in missing details, or edit for context and readability [37]. Furthermore, most of these tools are used as a cloud based service, which creates ethical concerns about data protection and confidentiality [37].

To make a decision on how audio recordings are transcribed, three types of transcribing mentioned by McMullin were analysed: selective, full verbatim and intelligent verbatim [37]. Selective is used to pull out relevant quotes, full verbatim is used to create an exact replica of the dialogue from the audio recording [25], and intelligent verbatim is used to create a replica of the dialogue from the audio recording, while omitting occasions where for instance misspeaks, nonverbal cues, or grammar mistakes occur, unlike full verbatim where these occasions are written down.

Intelligent verbatim is used for transcribing interviews, due to it capturing all content from an audio recording. Full verbatim is deemed unnecessary as it does not answer the research questions, and provides data to make assumptions on how for instance interviewees perceive definition of ready by analysing their nonverbal cues. Moreover, by not providing misspeaks and self corrections in the transcript, intelligent verbatim allows to create a more accurate record to what was intended and how the interviewee might have portrayed themselves in a written form [37].

As the interpretation of recordings cannot be neutral, and always has a point of view [10], each interviewee receives a transcript of their interview with the opportunity to provide feedback. Providing feedback will be, like their participation, on a voluntary basis.

3.3 Validity threats

To perform a well-grounded research, a paper by Maxwell is used to recognise different types of validity threats. The types distinguished by Maxwell are: interpretive validity, descriptive validity, theoretical validity, generalizability, and evaluate validity. By analysing papers which use a similar research method [23, 19, 47] and papers which analysed the research method [4, 33], potential threats were located per validity type, including ways to mitigate the impact per threat. Every threat is split into a part for the qualitative case study and part about the semi-systematic literature review, except for the theoretical- and evaluative validity which share the same mitigation strategy.

3.3.1 Interpretive validity

Interpretive validity describes how well a researcher reports participants' meaning of events, objects and or behaviours [36]. It is important that interpretations are not based on the perspective of the researcher, but on the perspective of the participant [50]. An example given by Thomson is when an interviewee hits the top of the desk, it may be to punctuate a point or it may have been done in anger. This type of validity is not exclusive to visual cues, and could also be recorded through audio, for instance, when an interviewee raises his or her voice.

A paper by Alsaawi mentions the option of video recording interviews to capture visual cues. However, participants might become more aware that they are being recorded, and as result become cautious about what they disclose and share during interviews [2]. As a result, only audio cues were recorded.

The decisions to include or exclude a paper from the literature review are available in appendix A. This mitigation does not contribute to interpretive validity while performing the literature review, and is however used to provide transparency on decisions afterwards.

3.3.2 Descriptive validity

Descriptive validity is a threat to this research, as descriptive validity is concerned with the factual accuracy of statements [36]. This threat can for instance be related to interviews, as it can be questioned whether a statement was really made by an interviewee, or that a statement was possibly misheard, mistranscribed, or misremembered [36].

To mitigate this type of validity when performing interviews, the audio will be recorded. The statement in [2], mentioned in the interpretive validity about how recording influences the interviewee, implies that when performing an interview, recording audio has a negative effect on the results of the interview. Therefore, it could be argued that only notes should be taken during the interviews. However, the decision was made to record the audio during interviews nonetheless to prevent data missing during the analysis, besides having evidence for statements made by the interviewees. A lack of audio recordings becomes apparent in results, when some are marked as missing [19]. The audio recordings were transcribed, as this provides the opportunity to share the data obtained with other researchers who can evaluate the analytical process, to reduce claims of researcher bias, and to allow the data to be used for other research purposes [4].

For the literature review, all papers that are analysed are available to the reader, to provide the possibility to verify the statements that are made.

3.3.3 Theoretical validity

For theoretical validation, this can be divided into two aspects, the validity of the concepts themselves as they are applied to the phenomena and the validity of the postulated relationships among the concepts [36]. The only mitigation for this threat is the evaluation of the research by both supervisors, and whether the created theories based on the data are constructed in a sufficient manner.

3.3.4 Generalizability

The generalizability is the extent to which one can extend the account of a particular situation or population to other persons, times or settings than those directly studied [36]. The ambition is to be able to generalize the result for all scrum teams which are developing software, or in other words, internal generalizability [36].

To achieve this level of generalizability for the interviewees, purposive and snowball sampling are used. A disadvantage of these sampling techniques is that it can be prone to researcher bias [8]. To mitigate this bias, a set of criteria is set. In extend to mitigating the sampling bias, these criteria are created to provide transparency to the reader on how sampling was performed. Paragraph 3.2.2 goes into more detail on how purposive and snowball sampling are used, including the criteria used to prevent the aforementioned research bias.

For the semi-systematic literature review, synonyms of the word 'deviation' in scientific literature are searched when planning the literature review, to retrieve as many papers as possible. This is important, as most papers on deviations held their research at a team or organisation with specific ways of working, decreasing the generalizability of the results. Therefore, an increase in the number of papers to analyse improves the generalizability of the literature review.

3.3.5 Evaluative validity

Evaluative data deviates from the data itself and attempts to assess the evaluations drawn by the researchers [36]. For instance, claims drawn by the research based on own understandings, rather than basing it on gathered data, is a scenario where evaluative validity is not met [50].

As a result, the mitigation technique is to be able to provide evidence before the statement is formulated. In the context of this research, valid evidence is considered data from interviews, or found in literature which adheres to the inclusion criteria of the literature review. The decision to only create these statements when evidence is available, was made as otherwise the research could possibly contain bias as there is only searched for certain literature, or interview questions are shaped in a certain way to be able to support a statement. As mentioned in theoretical validation, the evaluation of the research by two supervisors will furthermore mitigate the threat of evaluative validity.

4 Results

The results are split into two sections, starting with the results from the literature study. This will include a summary of the results, with a more elaborate coverage of the results in appendix A and appendix B. The second section covers the results of the case study.

4.1 Literature Review

42 papers were analysed to discover what deviations on Agile Scrum are common in scientific literature. An overview of all analysed papers, including for each paper its author(s), whether it is used for the review, and the corresponding reason is available in appendix A. Every deviation discussed in the aforementioned papers is presented in appendix B; this chapter only provides a summary. The results are divided into categories found in a paper by Masood et al.

The first category of deviation is estimation, where scrum teams are meant to estimate requirements together [35]. It was discovered that efforts for a requirement are sometimes estimated individually, rather than in a team [18, 35, 21]. Masood et al. noticed team leads, experienced team developers or influential product owners making estimation themselves, which Diebold and Mayer also saw occurring during his case study. The lack of estimation for a requirement's effort is another deviation observed by Mortada et al., Jilani and Ikram in this category.

After estimation, breakdown is the second category of deviation, which entails the scrum team breaking down big requirements into smaller requirements to be able to finish them within a sprint [35]. A first deviation found is the breakdown of requirements is based on an individual basis [18, 35], as domain knowledge and expertise leads to better breakdown [35]. Another deviation is the breakdown of a requirement, after the sprint has already started [39], to save time and due to the difficulty of the project.

Assignment is the third category, described by Masood et al. as the self-organisation of a team to carry out work on requirements. Only the paper by Masood et al. reported on two deviations in this category. The first one is a lot of discussion about assignment of requirements in the team, and the second one is individual work is assigned through a manager or team lead. Both deviations were explained in the study to assist knowledge sharing across members, by assigning certain requirements to them. Assignment describes self-organisation of a team among other things, and therefore, a deviation described in [28] is also marked under this category, being the lack of self-organisation of a team.

For the fourth category, sprint backlog creation, eight deviations were found in literature. Three of those deviations were about the sprint goal, being: no sprint goal is created for every sprint, multiple sprint goals are created per sprint, and a sprint goal is created by only one person [35]. Other deviations found in this category are sprint plannings without a product owner [18], using under-defined development cycles instead of sprints [31, 28], having no difference between the sprint and product backlog [5], not knowing the team's capacity for the next sprint [39, 26] and not having a sprint backlog [28].

To create a sprint backlog, requirements are gathered on the product backlog. Requirements are prioritised on the product backlog, and are the responsibility of the product owner [35]. Product backlog creation is the fifth category in Masood et al.'s paper, with two deviations discovered in the literature review. The first is about requirements which are created by a different role than the product owner [35], caused by the structure of the organisation, or the direct submission of ideas by clients. The other deviations is that a team does not have a product backlog [39, 28, 21], where no cause was given by any of the papers the deviation was found in.

The last category is product backlog refinement and prioritisation. Refining adds additional information to a requirement to provide developers and testers with more context of what has to be developed and eventually tested for the requirement [35]. This category has a total of 11 discovered deviations. Three of those deviations related to the content of a requirement, with one being to not follow a prescribed requirement template [18, 39]. Both papers reporting on the deviation came to different conclusions as to why this deviation occurs. Diebold and Mayer said that the team was not used to using a prescribed template and it was not complete for them, while Mortada et al. mentions that a name and a description are sufficient for the participants of their study. The other deviations related to a requirement's content are not including a part on testing [31] and not writing down acceptance criteria [19]. Four deviations in this category were about prioritisation. One of those deviations was that the team would immediately react to when the customer would submit a request [31]. This

happened due to customers becoming accustomed to, as the examined organisation began as a start-up with the aim to quickly gain customers. The other three deviations are that requirements in the product backlog are not prioritised [18, 35, 21], the team prioritises the product backlog, rather than the product owner or business analyst [35], and the prioritisation of the sprint backlog is not consulted when making changes to the sprint [5]. The last four deviations found in this category are accepting changes when the sprint has already started [26, 5, 19], not writing a requirement for each task [5, 19], only refining requirements in the sprint planning and not in the product backlog [39], and not being able to complete requirements in one sprint [26].

4.2 Case Study

For the case study, 14 participants were interviewed. Of these participants, three are scrum master, three are product owner, five are software developer, two are tester, and one participant is both a scrum master and a developer. They are active in varying industries; education, public transport, IT security and software development. Their years of experience with Agile Scrum is also versatile, with most having three to five years of experience, while a few having one year of experience and some having 10 or more years. Time spent in their Agile Scrum team was approximately one year for most participants, with two exceptions; a team together for seven months, and a team together for almost five years. Participants were not required to work or have worked in an Agile Scrum team which uses definition of ready, which were five participants.

The results are structured following the structure of the sub-questions discussed in chapter 2.2. This structure was also used for the thematic coding, with a complete overview of all the codes being available in appendix C.

4.2.1 Rationale

Using definition of ready has shown to be a tool for discussion. A product owner noticed that it helped with preventing a hierarchy in the group, as junior developers would sometimes be silent in meetings. Using definition of ready, junior developers were able to more easily point to issues with requirements during refinements, as they are aware of the criteria that have to be met. In the past, they used to *"let the senior developer think about the issue and not think about it themselves"*. Furthermore, a scrum master from a different team mentioned that it could be a guide for discussions on the content of requirements, as *"changes [to the definition of ready] would be announced and actively discussed with all members of the team"*. This guide for discussions also assists with making the team work more autonomous or self-managing, as another scrum master noticed team members pro-actively addressing team members who are not adhering to defined criteria when creating requirements.

Definition of ready moreover provides the possibility to keep requirements small, dividing big requirements into smaller requirements, by having a criterion in the definition of ready, which states that a requirement cannot have a greater estimated effort than a discussed amount. In this case, the estimated effort is a predefined scale, which is used after a requirement has completely been refined. Keeping user stories small was only remarked by one product owner and one developer as rationale why their team uses definition of ready. However, multiple participants mentioned a similar criterion in their definition of ready, and could therefore be a significant reason to use definition of ready.

During the sprint, requirements transition from different states until they are finished. Definition of ready describes the state where a requirement is ready to be developed. A used follow-up state could be the definition of done, a list of criteria to check if a requirement is done [48], which was once mentioned in the context of being obsolete due to their definition of ready. One developer noted that they rarely use definition of done, because of how strict definition of ready is applied. They divide a requirement in sub-tasks, until everything that has to be done is a single sub-task. Therefore, if all sub-tasks are done, a requirement automatically adheres to their definition of done. Two examples of a sub-task given by the interviewee were a peer review, and changes to the user interface, which are created into a sub task. All sub-tasks will be checked at the review to verify whether everything has been done. Another participant, not using a definition of ready, said something similar, as he mentioned that *"there is a bigger necessity to check if a requirement adheres to definition of ready than definition of done. Definition of done is often whether the product owner is satisfied with the results. [...] The advantage of checking definition of ready is that when it goes wrong at that stage, it has an impact on the complete sprint. [...] You want to have the process correct at the beginning."* This answer was given when asked whether they should start using definition of ready, and what benefits it could provide.

Some teams were replacing an existing team at the same or an outsourced company. This meant that their initial reason to use definition of ready was the fact that it was already used. This was however not their only incentive to use definition of ready, as one developer mentioned that when new team members started at a transition period, they had an issue where *"at a certain moment, we did for instance not create tests, or documentation"*, as no definition of ready was there to impose it. Other developers also acknowledged the importance of definition of ready for the on boarding of new team members: *"for new members, I am very happy with definition of ready, as something can be normal for*

me, but is not a common practice for him or her” according to a scrum master. One developer whose team does not use a definition of ready also saw benefit in using it for new members, and in addition mentioned that *”the new member can scroll through it, when we are going to refine a requirement”*.

Failing to achieve sprint goals was mentioned in multiple interviews, with three interviewees making a connection with definition of ready as a way to partially mitigate this issue. One of these has been discussed earlier, where requirements are too big and should be divided into smaller requirements, with an example by an interviewed developer being: *”multiple elements are embedded in the acceptance criteria for one requirement, which are divided into independent requirements for each criteria, results in issues when testing. Testers will have to wait for requirements to be completed for development [...] and will receive a very big requirement to test.”*, making it difficult to finish on-time. This scrum team included a criteria where a user story cannot contain multiple elements.

A different reason for failing to achieve the sprint goal, was the clarity of user stories. A tester mentioned how unclear stories were included in the sprint, with *”such a task disrupting the whole sprint, as it takes forever to clarify it, and therefore we do not include those requirements anymore.”*. Several interviewees mentioned similar issues, with definition of ready being their tool to mitigate unclear requirements. For instance, a product owner mentioned that a criteria in their definition of ready is the analysis of risks that could be involved in the requirement: *”if you do not have any risks, you are not aware of what has to be accounted for. So in short, it is an extra check whether we are sure what the requirement is about and what we have to check”*. A developer had similar comments, with a view on requirements in general, rather than focusing on the criteria of their definition of ready: *”someone who refined a requirement knew what has to happen with the one liners, and could perform the task. However, when someone would get sick, or went on vacation, no one knew what should happen, and we could start refinement for the requirement again. Therefore, a definition of ready has been instated.”*.

Most participants mentioned that creating software development involves the risk of having unclear requirements. For instance, a scrum master discussed the possibility that *”after a refinement, we think that we have covered everything, however, the code is structured differently than anticipated when refining the requirement.”*. Other participants mentioned that some content could be missed during a refinement by accident, as a tester mentioned: *”we are all human, sometimes we will be missing something”*. However, all participants saw definition of ready as a way to increase the clarity of requirements.

For some participants, their application of definition of ready was also their rationale for using it. Chapter 4.2.3 elaborates further on how definition of ready is applied in practice, this paragraphs focuses on the rationale behind them. Two applications were described: a checklist to verify the readiness of requirements, and as a way to remember the criteria set for a refined requirement. The difference between the two is that the checklist is used by teams while refining requirements to verify whether a requirement adheres to the criteria. Memorizing the criteria is mostly used during a retrospective, as a way to remember what has been agreed upon, often when a sprint did not go according to plan. A developer told about a period in which they did not use definition of ready as checklist and they *”immediately noticed the consequences [that things were missing], even though our team had experienced developers. Checking the compliance with definition of ready is therefore a common procedure [while refining requirements]”*. Definition of ready as tool to memorize criteria for requirements has been described by another developer as *”definition of ready is nice as it remembers you to the set criteria. In case of a bad habit, you can point to our definition of ready how it is supposed to be. I suppose that we could be without one, but I would rather not have to.”*.

4.2.2 Creating a definition of ready

Where several different rationales were given for using definition of ready, creating a definition of ready is not as different between teams.

For one team, creating their definition of ready was done through their normal process of refining requirements, and developing them. The developer of this team described that: *"a requirement was refined with acceptance criteria to create a definition of ready. A couple of team members thought about the definition of ready's criteria, which were checked and demoed during the sprint review, providing other stakeholders with the opportunity to provide feedback. This resulted in a definition of ready."*

This pattern to brainstorm as a team about the content of their definition of ready can be seen in every interview, except for teams where a definition of ready is not used.

One team had meetings to reflect and improve their Agile Scrum process, called 'scrum of scrums'. During these meetings, they *"created criteria which they could come up with, and later verified with the whole team whether they agreed with them"*, according to a developer who was involved in the process. This developer also mentioned that the initial list of criteria became a *"big list, and if we would begin with this list, we would not use it, so every team was given the opportunity to chose items which were most important to them. This resulted in a list with the most important criteria, to make it as minimal as possible so to the would not forget them."* The team used a prioritisation technique like MoSCoW [30] where criteria were for instance marked 'nice to have' to help them find the most important criteria.

Teams which already had a definition of ready did not conduct extensive brainstorm sessions, as they retrieved previously made agreements and documented them as their definition of ready. Through reflection, discussed in chapter 4.2.5, those definition of readies adjusted to their specific needs.

4.2.3 Applying definition of ready

The application of a team's definition of ready has been shortly described in chapter 4.2.1, about the rationales to use definition of ready. This paragraph will further elaborate on how teams use their definition of ready, and what patterns were found between the different applications.

Two distinct ways of application were found for definition of ready. The first is using definition of ready as a checklist when refining requirements. The scrum master of a team which is using definition of ready as checklist explained their process: *"The team goes through the requirements [before the sprints start], we provide it with a status, which will become 'ready for sprint' if all team members reach a consensus."* A developer from a different team, which also applies definition of ready as checklist, mentioned as advantage that checking definition of ready helps them when *"our testers are for example on vacation, it is not only their responsibility to ensure that the requirement is testable, and that tests are taken into account, but it is the responsibility of the whole team. Also that of the developers [...]"*. This type of application was seen at larger organisations where interviews were held.

Using definition of ready in the process is the second type of application, seen at interviews held at smaller organisations, where a definition of ready has been created. Team members hold each other accountable, and definition of ready is used to memorise the criteria for a requirement, as described in chapter 4.2.1 about rationales. This was not always the case for every interviewee, as one scrum master told about when they started using definition of ready, they would check every requirement with their definition of ready next to it, however, they are now aware of its content *"and it has become part of the process"*. Another developer from a different team mentioned that incorporating it in the process is a desired way, as *"checking definition of ready all the time could drive the team crazy"*, and that after a while, they all know it quite well by heart. This type of application is therefore reliant on the self-management of a scrum team, which is helped by using a template for requirements, consisting of different topics. A few topic examples given by participants are acceptance criteria, a description of the requirement, performance criteria, and what has to be tested. Most participants considered their template as how they applied their definition of ready, like one of the testers who was interviewed: *"we have a definition of ready, however, it is present in every requirement, but we do not check every criteria by checking them off. For every requirement, we have a default template that has to be filled in"*, otherwise *"if the content of the requirement does not adhere to the template, I will personally prevent it from being planned for the next sprint [...] as a developer will do its own thing, and consequently, it is not clear what I have to test"* added the tester about how this template assists with the application of definition of ready. A developer described how their requirements consisted of

different levels, going from an epic, to a feature and product backlog item. Each level therefore had their own template: *"Epic is an overarching template which provides more freedom, in contrast to the template of features which has: problem, goal, proposed solution and acceptance criteria, which are the most important to know when a feature is finished"*.

It became clear after the interviews with all participants, that all, but one team of the participants interviewed, use a template for their requirements, even teams which do not have a definition of ready defined.

4.2.4 Mitigating common Agile Scrum deviations

Seven deviations found in the literature review were discussed during interviews, and related to the content or structure of requirements. Accepting changes when the sprint has started was one, which rarely happens according to the interviews, as one participant mentioned that some changes are made while the sprint has started. This has, in their experience, applied to small tasks, and can be done as short lines of communication are present between them and the product owner. The use of definition of ready does not seem to have an influence on this deviation, as most participants mentioned during their interview that there is a risk of missing something when refining requirements, even when using definition of ready.

Related to this deviation is another deviation: breakdown of requirements after the sprint has begun, where breakdown describes dividing a (big) requirement into smaller requirements. Similar to the previous deviation, this also rarely occurs, and is not mitigated using a definition of ready. A tester mentioned: *"that there will always be things forgotten, as I said, when you will implement a requirement, you can encounter things that are different. We could improve this by taking more time to refine, however, we do not have that time."*

A deviation which can possibly be mitigated by definition of ready is not writing a requirement for each task. A developer mentioned this as one of their rationales for using definition of ready, as through one of the criteria, a sub-task to create documentation became compulsory for requirements, as *"documentation was one of the most discussed topics, because team members did not know whether they had to make changes to the documentation"*. A scrum master from a different team mentioned a similar approach where dividing tasks proves to be beneficial, especially for testing purposes. However, dividing tasks was not part of their definition of ready.

Part of their definition of ready however was the criteria that a user story can not be higher than a certain effort value, to be able to complete it in one sprint. Participants from other teams reported a similar criteria in their definition of ready. However, one developer stated that it can be difficult to adhere to this criteria: *"some items are just not small"*, but continued with that their definition of ready is a tool to help them remember to keep requirements small when focus on this topic is lost.

The last three deviations focused on the content of a requirement, and are: not following the prescribed template, not including acceptance criteria and not estimating the effort required to complete a requirement. Almost all participants using definition of ready were united in that these deviations do not occur. One developer did mention that some requirements do not follow the prescribed template initially, as sometimes requirements are refined at the end of the refinement, when moral is low and requirements are quickly written. They however solved this by going through the to be planned requirements during the sprint planning as well to ensure that the prescribed template is followed. Participants, from teams which are not using definition of ready, indicated that not following the or a prescribed template is an occurring deviation. For one of the teams, customers provide information for the requirement, where some adhere to a template, and others do not. No template is defined by the team. Providing their customers with a definition of ready for them to adhere to, was seen as overkill by the interviewed developer. However, giving customers a template which they can fill in, including active coaching on how to apply it in practice was seen as beneficial. Another team not using definition of ready also encounters this deviation. They have a template for requirements, which is not always filled in from top to bottom, leaving topics empty.

4.2.5 maintaining definition of ready

Maintaining a definition of ready is to have a moment to reflect on whether it is still up-to-date. For some of the participants, this moment of reflection could happen ad-hoc, during a daily stand-up, during a refinement, scrum of scrums or a retrospective. It depends on what would be the best option, and the quickest way to deal with shortcomings or issues of their definition of ready. Most interviews had the same conclusion, there is not a planned moment to reflect on definition of ready, and reflection will happen when an issue occurs in the sprint where definition of ready could be a solution to, or is related to said issue.

One of the interviewed teams had a reflection moment for the definition of ready. Every sprint, they would go through the content of refined requirements, and *"have everyone on the team be able to reflect on it"*. Based on the remarks made during those reflection moments, improvements or a retrospective point could be made to improve their definition of ready.

Another way to improve definition of ready was brought up by two scrum masters from different teams, labelled as 'reset and discover' during the analysis phase. It is important to mark that both have not used this yet, and is a method they would like to try. As explained by one of them during the interview: *"if I had the opportunity, I would quit using definition of ready. I would say that for two sprints, we would not use it, to hear from the team what they actually need. Would they want definition of ready to return, and what its content should be, so I am sure that it would be the team's definition of ready."* Both scrum masters came into the team when a definition of ready was already present, which could relate to this idea of reflection, to make it the team's own definition of ready, rather than using an already existing list.

Most teams did not change their definition of ready a lot after the initial criteria were agreed upon. However, slight adjustments were present, and the ones discussed by interviewees are dependencies, balance, and specific points.

Including the dependencies that a requirement has on other requirements was very important for the team who added this to their definition of ready, as the team had split up into two teams, making it essential to understand on what components each team would be working on.

Finding a good balance between making definition of ready a gateway and a tool to discuss, was important for a scrum master to reflect upon. According to him, *"the team could say that a requirement does not adhere to their definition of ready, and it will not be planned for the next sprint, turning definition of ready into a contract"*, which is something the scrum master wanted to avoid. He likes the conversation that their definition of ready starts, not the contract it entails.

Making criteria more concrete was another improvement mentioned by several interviewees. An example is improving the clarity of used words, like the term 'clear', which was used in their definition of ready. They improved this by specifying what was meant by the term 'clear'. Another improvement taken from an interview is to further specify performance criteria, by including the performance tests that should be performed.

5 Discussion

This chapter provides an interpretation to the result, how it provides value to the scientific community, and how it provides value to society. Limitations are furthermore discussed, to show what can be improved upon or be further explored for future work.

5.1 Application

This research adds knowledge on how definition of ready is applied in practice for the academic world. Different contexts were studied, from teams working in small companies to teams in bigger companies, and teams who are using definition of ready to teams who are not using definition of ready.

Software teams can use this research as a start for their implementation of definition of ready: how they will implement it, what criteria could possibly be added to their definition of ready, and how they will maintain it. Furthermore, this research shows what potential benefits definition of ready brings to a team, and can therefore be used as a tool to discuss the application of definition of ready in their own team.

The results can also be used as a way to improve an already existing implementation of definition of ready. Most interviewees, who are or were in a team using definition of ready, mentioned that they were interested to read how other teams have applied it, and could change their team's approach accordingly.

5.2 Interpretation

Through 14 interviews, it has become clear that every team, and every role within a team has a different view on why to use definition of ready. For instance, some see it as a tool to assist new members of their team, some use it as a tool for discussion. This tool for discussion mostly exists in the initial phase, where teams are newly assembled, which became clear in the interviews as some participants forgot about its content, and retrieved it from an archive for their interview. Something which shows maturity of the team, as definition of ready becomes part of the process, and teams become self-managing to indicate when a requirement is not sufficiently refined. This seems to be the overarching rationale for teams to adopt definition of ready. At a certain moment, definition of ready has become integrated into the process and is therefore not actively checked during refinements. Furthermore, its content is not directly reflected upon, but rather the process to refine a requirement, or the template of requirements.

5.3 Limitations

Participants from different teams and organisations were interviewed, with a variation in organisation sizes. A pattern was discovered between the size of an organisation and how teams apply definition of ready, as was noted in the results on application of definition of ready. However, only two participants in the research were from larger organisations, making it difficult to generalize the result to all software teams active in a large organisation. For future research, it could be interesting to further explore this pattern, by having more participants from large organisations. In addition, participants from the same organisation, but from different teams, could be interviewed to discover whether those use a similar approach to definition of ready.

The interviews were an eye-opener for some of the participants, who indicated that they had not opened their definition of ready for a long period. They indicated that they would reflect or reflected upon it to possibly make changes to its content, or how it is used. Furthermore, participants operating in a team which does not use definition of ready became interested in using one. The results on reflection have been quite limited, as most teams did not make significant changes after the initial implementation of their definition of ready. Conducting follow-up interviews could therefore have been interesting to gather additional data on reflection and possibly how teams which did not use a definition of ready have adopted it.

Sub-question 4, whether definition of ready can mitigate Agile Scrum deviations related to the content of a requirement, was added after the data gathering phase, as data to answer the question was available. However, the case study was not designed with this question in mind, and could therefore lack information. A change for future research could be to include more deviations in the interview, as only 2 deviations were used in interview questions.

6 Conclusion

Definition of ready has proven to be a tool for discussion, providing team members with agreed upon criteria to point towards when a requirement is not according to the standard procedure. Criteria can be set by a team themselves to suit their desires. New members of the team can become familiar with these criteria to get acquainted with the team's way of working. Brainstorm sessions are the origin of most definition of readies to give everyone on the team the ability to provide input. After this phase, small adjustments are made to the definition of ready, while big changes are uncommon. Definition of ready is applied through a checklist when refining requirements, or embedded in the process by the team's self-management, and through a prescribed requirement template, also used by teams who have not defined a definition of ready. In essence, definition of ready is a checklist with criteria, which depending on how strict the team uses it, can assist them become more mature as scrum team.

7 Acknowledgements

I would like to thank my first supervisor, Gerard Wagenaar, for his patience, guidance and feedback during my thesis. It was not always easy sailing on my part due to circumstances and procrastination, which Wagenaar helped me through in an understanding and pleasant way. I would also like to thank my second supervisor, Fabiano Dalpiaz, for his feedback on my proposal, concepts and eventual end result. As last, I would like to thank all participants of the research, who shall remain anonymous. I am very grateful for all of their time and input.

References

- [1] Scopus search guide. URL <http://schema.elsevier.com/dtds/document/bkapi/search/SCOPUSSearchTips.htm>.
- [2] Nabeel Al-Yateem. The effect of interview recording on quality of data obtained: A methodological reflection. *Nurse researcher*, 19(4), 2012.
- [3] Scrum Alliance. State of scrum 2017-2018, 2017.
- [4] Ali Alsaawi. A critical review of qualitative interviews. *European Journal of Business and Social Sciences*, 3(4), 2014.
- [5] Leanah Alsaber, Ebtessam Al Elsheikh, Sarah Aljumah, and NS Mohd Jamail. Perspectives on the adherence to scrum rules in software project management. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1):360–366, 2021.
- [6] Sasa Baskarada. Qualitative case study guidelines. *Başkarada, S.(2014). Qualitative case studies guidelines. The Qualitative Report*, 19(40):1–25, 2014.
- [7] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001.
- [8] Andrea E Berndt. Sampling methods. *Journal of Human Lactation*, 36(2):224–226, 2020.
- [9] Mariya Breyter. Agile estimation and planning. In *Agile Product and Project Management: A Step-by-Step Guide to Building the Right Products Right*, pages 161–185. Springer, 2022.
- [10] Mary Bucholtz. The politics of transcription. *Journal of pragmatics*, 32(10):1439–1465, 2000.
- [11] Narciso Cerpa and June M Verner. Why did your project fail? *Communications of the ACM*, 52(12):130–134, 2009.
- [12] H Frank Cervone. Understanding agile project management methods using scrum. *OCLC Systems & Services: International digital library perspectives*, 2011.
- [13] Victoria Clarke, Virginia Braun, and Nikki Hayfield. Thematic analysis. *Qualitative psychology: A practical guide to research methods*, 3:222–248, 2015.
- [14] Robert G Cooper. Agile–stage-gate hybrids: The next stage for product development blending agile and stage-gate methods can provide flexibility, speed, and improved communication in new-product development. *Research-Technology Management*, 59(1):21–29, 2016.
- [15] Benjamin F Crabtree. *Doing qualitative research*. sage, 1999.
- [16] Jeff Dalton and Jeff Dalton. Definition of ready. *Great Big Agile: An OS for Agile Leaders*, pages 163–164, 2019.
- [17] Saru Dhir, Deepak Kumar, and VB Singh. Success and failure factors that impact on project implementation using agile software development methodology. In *Software engineering*, pages 647–654. Springer, 2019.
- [18] Philipp Diebold and Udo Mayer. On the usage and benefits of agile methods & practices. *Agile Processes in Software Engineering and Extreme Programming*, page 243, 2017.
- [19] Philipp Diebold, Jan-Peter Ostberg, Stefan Wagner, and Ulrich Zandler. What do practitioners vary in using scrum? In *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings 16*, pages 40–51. Springer, 2015.
- [20] Yogesh K Dwivedi, David Wastell, Sven Laumer, Helle Zinner Henriksen, Michael D Myers, Deborah Bunker, Amany Elbanna, MN Ravishankar, and Shirish C Srivastava. Research on information systems failures and successes: Status update and future directions. *Information systems frontiers*, 17:143–157, 2015.

- [21] Veli-Pekka Eloranta, Kai Koskimies, Tommi Mikkonen, and Jyri Vuorinen. Scrum anti-patterns—an empirical study. In *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, volume 1, pages 503–510. IEEE, 2013.
- [22] Veli-Pekka Eloranta, Kai Koskimies, and Tommi Mikkonen. Exploring scrumbut—an empirical study of scrum anti-patterns. *Information and Software Technology*, 74:194–203, 2016.
- [23] Zoran Ereiz and Denis Mušić. Scrum without a scrum master. In *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*, pages 325–328. IEEE, 2019.
- [24] Ilker Etikan, Sulaiman Abubakar Musa, Rukayya Sunusi Alkassim, et al. Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, 5(1):1–4, 2016.
- [25] Elizabeth J Halcomb and Patricia M Davidson. Is verbatim transcription of interview data always necessary? *Applied nursing research*, 19(1):38–42, 2006.
- [26] Ville T Heikkilä, Maria Paasivaara, and Casper Lassenius. Scrumbut, but does it matter? a mixed-method study of the planning process of a multi-team scrum organization. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 85–94. IEEE, 2013.
- [27] Watts S Humphrey. Why big software projects fail: The 12 key questions. *Software Management*, 3, 2005.
- [28] Muneeba Jilani and Naveed Ikram. Investigating the gap between scrum theory and practice in pakistan. In *ICSOFT*, pages 180–186, 2020.
- [29] Magne Jørgensen. Do agile methods work for large software projects? In *Agile Processes in Software Engineering and Extreme Programming: 19th International Conference, XP 2018, Porto, Portugal, May 21–25, 2018, Proceedings 19*, pages 179–190. Springer, 2018.
- [30] Javed Ali Khan, Izaz Ur Rehman, Yawar Hayat Khan, Iftikhar Javed Khan, and Salman Rashid. Comparison of requirement prioritization techniques to find best prioritization technique. *International Journal of Modern Education and Computer Science*, 7(11):53, 2015.
- [31] Jacinto Ramirez Lahti, Antti-Pekka Tuovinen, Tommi Mikkonen, and Rafael Capilla. Scrumbut as an indicator of process debt. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 318–321. IEEE, 2022.
- [32] Kurt R Linberg. Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2-3):177–192, 1999.
- [33] Steve Mann. A critical review of qualitative interviews in applied linguistics. *Applied linguistics*, 32(1):6–24, 2011.
- [34] Mika V Mäntylä, Bram Adams, Foutse Khomh, Emelie Engström, and Kai Petersen. On rapid releases and software testing: a case study and a semi-systematic literature review. *Empirical Software Engineering*, 20:1384–1425, 2015.
- [35] Zainab Masood, Rashina Hoda, and Kelly Blincoe. Real world scrum a grounded theory of variations in practice. *IEEE Transactions on Software Engineering*, 48(5):1579–1591, 2020.
- [36] Joseph Maxwell. Understanding and validity in qualitative research. *Harvard educational review*, 62(3):279–301, 1992.
- [37] Caitlin McMullin. Transcription and qualitative methods: Implications for third sector research. *VOLUNTAS: International journal of voluntary and nonprofit organizations*, 34(1):140–153, 2023.
- [38] Nils Brede Moe and Torgeir Dingsøy. Scrum and team effectiveness: Theory and practice. In *Agile Processes in Software Engineering and Extreme Programming: 9th International Conference, XP 2008, Limerick, Ireland, June 10-14, 2008. Proceedings 9*, pages 11–20. Springer, 2008.

- [39] Mohamad Mortada, Hamdy Michael Ayas, and Regina Hebig. Why do software teams deviate from scrum? reasons and implications. In *Proceedings of the International Conference on Software and System Processes*, pages 71–80, 2020.
- [40] Ashish Mundra, Sanjay Misra, and Chitra A Dhawale. Practical scrum-scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications*, pages 119–123. IEEE, 2013.
- [41] Ken Power. Definition of ready: An experience report from teams at cisco. In *International Conference on Agile Software Development*, pages 312–319. Springer, 2014.
- [42] Ken Power. Social contracts, simple rules and self-organization: a perspective on agile development. In *International Conference on Agile Software Development*, pages 277–284. Springer, 2014.
- [43] C Robson. Resource for users of social research methods in applied settings, 2011.
- [44] Kenneth S Rubin. *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.
- [45] Ken Schwaber and Jeff Sutherland. The scrum guide. *Scrum Alliance*, 21(1):1–38, 2011.
- [46] Irving Seidman. *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press, 2006.
- [47] Yogeshwar Shastri, Rashina Hoda, and Robert Amor. Spearheading agile: the role of the scrum master in agile projects. *Empirical Software Engineering*, 26:1–31, 2021.
- [48] Ana Silva, Thalles Araújo, João Nunes, Mirko Perkusich, Ednaldo Dilorenzo, Hyggo Almeida, and Angelo Perkusich. A systematic review on the use of definition of done on agile software development projects. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 364–373, 2017.
- [49] Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of business research*, 104:333–339, 2019.
- [50] S Bruce Thomson. Qualitative research: validity. 2011.
- [51] Diane Walker and Florence Myrick. Grounded theory: An exploration of process and procedure. *Qualitative health research*, 16(4):547–559, 2006.

A - Appendix: Analysed papers semi-systematic literature review

A.1 Agile Scrum deviations

Search term in Scopus: “agile AND scrum AND deviation*”.

ID	Title	Author(s)	Usable	Reason
a1.	ScrumLint: Identifying violations of agile practices using development artifacts	Matthies C., Kowark T., Richly K., Uflacker M., Plattner H,	No	Focusses on the code, with a tool that can be applied in Agile Scrum, called ScrumLint. Hit on “deviations” for this query was the deviation this tool introduces to Agile Scrum.
a2.	Integrating quality assessment through metrics into scrum software development	Zaouali S., Ayachi Ghannouchi S.	No	Focusses on how the Agile Scrum method is used in a company, and how deviations can be corrected, rather than showing what deviations exist.
a3.	On the usage and benefits of agile methods & practices: A case study at Bosch Chassis systems control	Diebold P., Mayer U.	Yes	Mentions how Agile Scrum is used and adapted at Bosch Automotive, which can be seen as deviations.
a4.	Real World Scrum A Grounded Theory of Variations in Practice	Masood Z., Hoda R., Blincoe K.	Yes	Discusses how Scrum is used in practice, including “significant variations in these practices”.
a5.	Agile methods: The gap between theory and practice	Conboy K.	No	This paper has a scope which is too broad. It focusses on all Agile methods, rather than Agile Scrum and does therefore not adhere to the inclusion criteria.
a6.	Prototype design in agile product development processes for technical systems	Riesener M., Doelle C., Schloesser S., Schuh G.	No	The deviations discussed in this paper are how Agile Scrum should deviate when used in the industrial industry.

a7.	The impact of agile methodologies and cost management success factors: An empirical study	Iqbal J., Omar M., Yasin A.	No	This paper discusses the cost management of software projects and mentions deviation as ‘standard deviation’ for their statistical analysis.
a8.	Using agile methods for product software development	Zaretskiy D., Serbin V.	No	The paper is too old.
a9.	ScrumBut as an Indicator of Process Debt	Lahti J.R., Tuovinen A.-P., Mikkonen T., Capilla R.	Yes	Agile Scrum deviations are mentioned in this paper, with the focus being on the influence the deviations have on technical debt.
a10.	Shock therapy a bootstrap for hyper-productive scrum	Sutherland J., Downey S., Granvik B.	No	The paper is too old.
a11.	Perspectives on the adherence to scrum rules in software project management	Alsaber L., Al Elsheikh E., Aljumah S., Mohd Jamail N.S.	Yes	This paper presents qualitative and quantitative data on the experiences of professionals who are using Agile Scrum, and how it is applied in their organization.
a12.	Making agile retrospectives more awesome	Przybylek A., Kotecka D.	No	Focusses on Agile Scrum processes, but only on improving the retrospective activity.
a13.	What do practitioners vary in using scrum?	Diebold P., Ostberg J.-P., Wagner S., Zendler U.	Yes	Describes how Agile Scrum is adapted to fit different contexts, and what the reasons are for the changes.
a14.	Exploring Scrum-But - An empirical study of Scrum anti-patterns.	Eloranta V.-P., Koskimies K., Mikkonen T.	Yes	Paper mentions deviations in Agile Scrum, or in the words of the paper: ScrumBut, and the possible consequence which occurs because of them.
a15.	Why do software teams deviate from scrum?: Reasons and implications	Mortada M., Ayas H.M.	Yes	Studies Agile Scrum deviations, the reasons behind them and the consequences of those deviations.

a16.	Doing scrum rather than being Agile: A case study on actual nearshoring practices	Zieris F., Salinger S.	Yes	Deviations in Agile Scrum in a nearshoring setting will be discussed in this paper.
a17.	Investigating the gap between scrum theory and practice in Pakistan	Jilani M., Ikram N.	Yes	The paper investigates gaps between theory and practice, possibly highlighting Agile Scrum deviations which occur.
a18.	Scrum ANTI-patterns - An empirical study	Eloranta V.-P., Koskimies K., Mikkonen T., Vuorinen J.	Yes	Potentially harmful ways, in other words: deviations textbook ideals, to handling Agile Scrum, based on a survey, are described in this paper.

A.2 ScrumBut

Search term in Scopus: “ScrumBut”.

ID.	Title	Author(s)	Usable	Reason
b1.	Evolving to agile: A story of agile adoption at a small SaaS company	Block M.	No	The paper is too old.
b2.	ScrumBut as an Indicator of Process Debt	Lahti J.R., Tuovinen A.-P., Mikkonen T., Capilla R.	No	Duplicate
b3.	Exploring ScrumBut - An empirical study of Scrum anti-patterns	Eloranta V.-P., Koskimies K., Mikkonen T.	No	Duplicate
b4.	ScrumBut, but does it matter? A mixed-method study of the planning process of a multi-team scrum organization	Heikkila V.T., Paasi-vaara M., Lassenius C.	Yes	Paper about what changes have taken place for large software development organisations, and whether the changes were perceived harmful.
b5.	Blended Scrum model for software development organizations	Çetin E., Onay Durdu P.	No	Focuses on the transition to Agile Scrum, and what changes could facilitate this transition, rather than how Agile Scrum is used and the changes made to it, due to experiences.

A.3 Agile Scrum variation

ID.	Title	Author(s)	Usable	Reason
c1.	A dynamic variability management approach working with agile product line engineering practices for reusing features	Kiani A.A., Hafeez Y., Imran M., Ali S.	No	This paper proposes a hybrid development method, using the Agile Software Development method and the Software Product Line method.
c2.	Maturing the Scrum Framework for Software Projects Portfolio Management: A Case Study-Oriented Methodology	Vazifeh-Noshafagh S., Hajipour V., Jalali S., Di Caprio D., Santos-Arteaga F.J.	No	Focuses on making changes to Agile Scrum based on a case study.
c3.	Agile development of an attitude-behaviour driven simulation of alcohol consumption dynamics	Moyo D., Ally A.K., Brennan A., Norman P., Purshouse R.C., Strong M.	No	Combines Agile Scrum together with the UML method.
c4.	Real World Scrum A Grounded Theory of Variations in Practice	Masood Z., Hoda R., Blincoe K.	No	Duplicate
c5.	The State of Agile Software Development in the Czech Republic: Preliminary Findings Indicate the Dominance of "Abridged" Scrum	Dolezel M., Buchalceva A., Mencik M.	No	Partly focuses on Agile Scrum being used at organisations in an incorrect manner, but it only looks for incentive on why those companies would even use Agile Scrum. E.g., more of a show-off mechanism.
c6.	Agility in Telecommunication Product Development, Challenges and the Way Forward	Ameen Ishaq M.P., Prakash N.	No	Discusses the challenges that have to be overcome when implementing Agile methods, making the scope too broad and the subject not related.
c7.	Agile methods: The gap between theory and practice	Conboy K.	No	Duplicate
c8.	Face-to-face, email and instant messaging in distributed agile software development project	Niinimäki T.	No	Discusses the way the type of communication influences the amount of communication and the communication patterns at teams which use Agile Scrum.

c9.	The need to adopt agile methodology in the development of Cyber Forensics Tools	Balan C., Dija S.	No	Determines whether cyber forensics should develop using Agile Scrum to prevent their products from becoming obsolete.
c10.	Model of learning management system based on artificial intelligence in team-based learning framework	Pardamean B., Suparyanto T., Cenggoro T.W., Sudigyo D., Anugrahana A., Anugraheni I.	No	Topic is related to AI and, in the research, Agile Scrum was used as base for the application design.
c11.	An empirical study to design an effective agile knowledge management framework	Singh A., Kukreja V., Kumar M.	No	Focuses on knowledge management in an organization, which is not the scope of this literature review.
c12.	What do practitioners vary in using scrum?	Diebold P., Ostberg J.-P., Wagner S., Zendler U.	No	Duplicate
c13.	Cost estimate in scrum project with the decision-based effort estimation technique	Alshammari F.H.	No	Describes the difficulty of estimating size, cost and time-frame, when using Agile methods, as they have 'variable' requirements.
c14.	Definition of software process lines for integration of Scrum and CMMI	Dias De Carvalho D., Chagas L.F., Reis C.A.L.	No	It proposes to run a maturity model together with Agile Scrum.
c15.	Scrum metaprocess: a process line approach for customizing Scrum	Agh H., Ramsin R.	No	Identifies variables in Agile Scrum to configure or reconfigure Agile Scrum.

A.4 Scrum in practice

Search term in Scopus: “Scrum in Practice”

ID	Title	Author(s)	Usable	Reason
d1.	Bringing Agility to the Classroom: Integrating Professional Scrum Trainings to an Undergraduate IT Project Management Course	Neumann M., Baumann L.	No	Evaluates how Agile Scrum courses would be received by students and gives recommendations to other lecturers.
d2.	Using gamification for adopting scrum in practice	Marques R., Costa G., da Silva M.M., Gonçalves D., Gonçalves P.	No	Present a technique to make Agile more fun to use, using an app that is integrated with Jira.
d3.	Real World Scrum A Grounded Theory of Variations in Practice	Masood Z., Hoda R., Blincoe K.	No	Duplicate
d4.	Respond to Change or Die: An Educational Scrum Simulation for Distributed Teams	Laue Christensen E., Paasivaara M.	No	Presents an online Scrum simulation in a multiplayer game called ‘Don’t starve together’, where scrum teams can experience how to work with Agile Scrum.

B - Appendix: Found deviations

The different deviations have been split into different categories, inspired by a paper from Masood et al. on Scrum deviations. The categories are: estimation, breakdown, assignment, sprint backlog creation, product backlog creation, and product backlog refinement and prioritisation. Every category is accompanied with a short explanation, the deviations found in literature, including (a) rationale(s) why the deviation occurs, if available, and the paper where the deviation originates from. All papers used for the literature review are available in appendix A.

B.1 Estimation

Estimation involves predicting the effort required to carry out a requirement. Scrum teams are meant to estimate requirements collectively [35].

No.	Deviation	Rationale(s)	Source(s)
1.	Individual estimations are performed, rather than estimating in a team.	Team leads or experienced developers use their domain knowledge to ensure accurate estimations. Influential product owner made the estimation themselves [35]. Senior developer with experience or the one responsible for the feature [18].	[18, 35, 21]
2.	Stories are not estimated.	Due to the complexity of the product [39].	[39, 28]

Table 2: Deviations in the category estimation.

B.2 Breakdown

Requirements are of varying size and complexity. Therefore, the scrum team decomposes requirements with a big size into smaller tasks to be able to finish them within the sprint [35].

No.	Deviation	Rationale(s)	Source(s)
1.	Work breakdown is performed independently, not in a team.	Domain knowledge and expertise leads to better breakdown [35].	[18, 35]
2.	Large stories were broken down in smaller stories after the sprint planning.	To save time and due to the difficulty of the product [39].	[39]

Table 3: Deviations in the category breakdown.

B.3 Assignment

The scrum team is meant to self-organise to carry out work on requirements. Team members volunteer for a task, one at a time, based on the prioritisation of the sprint backlog [35].

No.	Deviation	Rationale(s)	Source(s)
1.	Lots of discussions about the assignment in team, where members were seen collaborating when making assignment decisions.	Held team members accountable collectively, and supported knowledge sharing. Advice new team members on tickets [35].	[35]
2.	Individual work is assigned through a manager or team lead.	Less experienced team members who lacked domain knowledge [35].	[35]
3.	Development team is not self-organizing.	-	[28]

Table 4: Deviations in the category assignment

B.4 Sprint backlog creation

The scrum team defines a sprint backlog during the sprint planning to set the objective of the requirements which the team is going to finish in the upcoming sprint [35].

No.	Deviation	Rationale(s)	Source(s)
1.	The sprint is created by a single person: the PO, business consultant, project manager or the client.	These roles have a high visibility on the sprint goals [35].	[35]
2.	No sprint goal set.	Lack of understanding on its purpose or difficulty in finding a common purpose due to scattered priorities [35].	[35]
3.	Setting multiple sprint goals.	Teams were working on many features and created goals to fit the items on the sprint backlog [35].	[35]
4.	Product owner is not present during the Sprint planning.	-	[18]
5.	No sprint, but rather development that advances in under-defined increments.	-	[31, 28]
6.	There is no differentiation between the product- and sprint backlog.	-	[5]
7.	No calculation for resources available for the upcoming sprint.	Too much freedom and projected absences for vacations [39].	[39, 26]
8.	Do not have a sprint backlog.	-	[28]

Table 5: Deviations in the category sprint backlog creation

B.5 Product backlog creation

Based on the needs of the stakeholders and the business strategy, the product owner is responsible for defining the product features as requirements. The product backlog is a prioritised list of requirements [35].

No.	Deviation	Rationale(s)	Source(s)
1.	Requirements were created by a different role than the product owner.	Organization structure, or because ideas would directly be submitted by clients [35].	[35]
2.	Does not have a product backlog.	-	[39, 28, 21]

Table 6: Deviations in the category product backlog creation

B.6 Product backlog refinement and prioritisation

The product owner is responsible for prioritising the requirements on the product backlog. When a requirement has a high priority on the product backlog, it can be refined to add additional details to allow the developers to execute the requirement with an understanding of what has to be implemented [35].

No.	Deviation	Rationale(s)	Source(s)
1.	The team sometimes prioritised, rather than the product owner or business analyst.	-	[35]
2.	Items are not prioritised on the product backlog.	Prioritisation was based on contextual factors, e.g., user requests [35].	[18, 35, 21]
3.	Not following the prescribed requirement template.	The team was not used to it, and it needed more information according to them [18]. The name and description are good enough for the teams [39].	[18, 39]
4.	The team would immediately react whenever the customer would request something.	Customers became accustomed to it, after the organisation started as a start-up with the aim to gain customers quickly [31].	[31]
5.	Testing is not required to describe when refining the requirement, as it will not be performed.	The project grew to such an extent that going back to creating tests for everything would not be feasible [31].	[31]
6.	Accepting changes when the sprint has started.	Sometimes, requirements necessitate special skills not every in the team possessed, which let to changing the sprint goal [26].	[26, 5, 19]
7.	Not consulting the prioritisation of the sprint backlog to apply the changes to the working sprint.	-	[5]
8.	Not writing a requirement for each task.	-	[5, 19]
9.	No acceptance criteria are written down for requirements.	Vaguely described requirements, and the missing product owner are the cause of this deviation [19].	[19]
10.	Stories are not refined in the product backlog, only during the sprint planning.	Due to no real product owner and ambiguity [39].	[39]
11.	Requirements could not be completed in one sprint.	Complexity of the product, and inter-team and external dependencies of requirements [26].	[26]

Table 7: Deviations in the category product backlog refinement and prioritisation

C - Appendix: Found deviations

This appendix contains all codes used during the thematic analysis, divided by top level codes. Three columns are present: name of the code itself, number of files a code was marked in, and the total amount of times a code was referenced.

C.1 Deviations

Deviations taken from the literature review, and used to answer sub-question four.

No.	Label	Files	References
1.	Accepting changes when the sprint has started	3	3
2.	Breakdown after the sprint started	3	5
3.	Not writing a requirement for each task	3	5
4.	No acceptance criteria	4	5
5.	Requirements could not be completed in one sprint	4	8
6.	Not following the prescribed template	7	20
7.	Requirement not estimated	10	18

C.2 Application moment

Activities described by participants when they use their definition of ready.

No.	Code	Files	References
1.	Sprint planning	2	3
2.	Estimating requirement's effort	4	4
3.	Refinement	5	7

C.3 Creating definition of ready

How teams created their definition of ready.

No.	Code	Files	References
1.	Prioritising technique	1	1
2.	Task in sprint	1	1
3.	Retrieving earlier created agreements	3	3
4.	Brainstorm sessions	5	7

C.4 Organisations which do not use definition of ready

How participants, who's team does not use definition of ready, would possibly use definition of ready after the interview.

No.	Code	Files	References
1.	Alternative	5	29
2.	Potential consequences	5	10
3.	Reason	5	15
4.	Possible application after the interview	5	16

C.5 Other

Codes not categorised under a top level code, with the ability to add context to other codes.

No.	Code	Files	References
1.	Steps after the interview	1	1
2.	Experience	2	2
3.	Risk of requirements lacking information	6	6
4.	Content of a definition of ready	5	16

C.6 Reason for using definition of ready

Codes for the rationales given by participants on why they use definition of ready.

No.	Code	Files	References
1.	Hierarchy	1	1
2.	Dividing requirements	1	1
3.	Definition of done	2	2
4.	DoR already existing	2	2
5.	Tool for discussions	2	2
6.	Improving scrum	2	2
7.	Maturity of the scrum team	2	2
8.	On boarding of new team members	3	5
9.	Sprint goals are not achieved	3	3
10.	Checklist	4	5
11.	Memorising agreements	4	7
12.	Clarifying requirements	4	5

C.7 Reason for not using definition of ready while a team has one

Describes why a scrum team using definition of ready, sometimes did not use it, or ignored criteria.

No.	Code	Files	References
1.	Just a list	1	1
2.	Short lines of communication	2	2
3.	Not possibly to apply	2	2
4.	Mature scrum team	2	2
5.	Value of requirement has been achieved	2	3

C.8 Moment to reflect

When a team reflects on the content of their definition of ready, and what possible actions are taken.

No.	Code	Files	References
1.	Ad hoc	1	3
2.	Refinement	1	1
3.	Stand up	1	2
4.	Reset and discover	2	2
5.	No specified moment	3	3
6.	scrum of scrums	3	4
7.	Retrospective	6	10

C.9 Application DoR

How teams apply criteria from their definition of ready in practice.

No.	Code	Files	References
1.	Partial guidance	1	1
2.	Going through definition of ready as a checklist	2	3
3.	Requirement template	7	8
4.	Process	8	24

C.10 Improvements to DoR

Improvements that teams made or would like to make to their definition of ready, in terms of content or how it is applied.

No.	Code	Files	References
1.	Dependencies	2	2
2.	Balance	2	2
3.	Interactive	2	3
4.	Specifying points	4	5

D - Appendix: Interview Protocol

A description on the process of conducting interviews, when a participant agrees to take part and adheres to the sampling criteria.

D.1 Introduction

Interviews start with an introduction to the structure of the interview; having three situational questions in the beginning, and continue with open questions. Interviewees are given the opportunity to ask questions, and their right to withdraw is repeated. Before the interview starts, interviewees are made aware that no wrong answers exist and their experience matters.

D.2 Participation consent

Consent is achieved through a pre-interview questionnaire, which in addition to consent, also includes background questions for the interviewee; their role, organisation's product, scrum team size and years of experience, used to prepare interviews.

D.3 Interview questions

Prepared situations and questions from the interview are presented here. It was common to deviate from this structure, in accordance with the semi-structured interview approach. The interview starts with situational questions, and continues with open questions.

D.3.1 Situational questions

For every situation, the interviewee is asked whether the situation occurs or could occur in their team,

1. User stories were planned which do not have an estimation for the effort that they will take to complete. During the sprint, it is realised that the sprint goal is not going to be met, as there are some stories which were not added to the total effort estimated for the sprint.
2. A user story is not written according to the prescribed template, resulting in a lack of information for the user story which makes it difficult to understand.
3. A user story in the sprint backlog does not cover the complete feature intended by the product owner, and therefore, the sprint goal will be adjusted to accommodate the complete feature.

D.3.2 Open questions

The open question part starts with a general question, to understand what possible changes can be done to their scrum process, and introduce open questions, without directly talking about definition of ready. The assumption is that interviewees therefore do not feel forced to mention definition of ready as a perfect solution, and would describe how it is actually used in remaining question. Some questions include an addition or additional question, in case the question is quickly answered with a yes or a no, recognizable by a "-" between the original question and the addition.

1. Do you think that you and your team are following the guidelines of Agile Scrum?
2. Are you familiar with a definition of ready and how would you describe it in your own words?
3. Does your team have a definition of ready? - Yes
 - (a) Were you involved in creating your definition of ready?
 - i. (Optional) What did you use to make the definition of ready? This can be any tool, e.g., using something to assess whether the criteria are sufficient, or a way to format them in a certain way.
 - (b) Why did your team adopt the definition of ready? - And do you think it benefits the team having a definition of ready? In what way(s) does it benefit the team? Do you think that your team could benefit more from using definition of ready in a certain way?
 - (c) Are there moments to reflect on the content/criteria of the definition of ready and were proposed changes made? - Or should have been made, or should have been reflected upon during the process, but was not?
 - (d) Do you occasionally consult the definition of ready? - An example of consulting the definition of ready is going through it and possibly furthermore assess whether a user story was applicable to be inserted into the sprint.
 - (e) Do you have other applications, besides requirements, where you use definition of ready for? - E.g., use it to assess whether a sprint or release is ready to be done.
 - (f) What would you like to change to the way definition of ready is applied in your team? - Or do you think that you team is currently applying definition in its most optimal way?
4. Does your team have a definition of ready? - No
 - (a) Was it a conscious decision to not use a definition of ready? - If it was a conscious decision, why did you decide to not have a definition of ready?
 - (b) Did you and your team experience sprints where user stories were not correctly defined? - If they did, did issues ever arise which could possibly have been solved by using a definition of ready?
 - (c) During refinements or sprint plannings, how did your team assess whether a user story was ready to be developed?
 - (d) Would you like to adopt definition of ready for your current or future projects? - Do you think that this is possibly, and in case it is not possible, why it would not be possible.

D.3.3 End of the interview

Interviewees have the opportunity to ask questions and make remarks. They are furthermore informed that a transcription will be sent for them to optionally go through. This will be the last contact towards the participants.