



**Utrecht
University**

What is Ready in a Definition of Ready?

by

Tommy Versteeg

(6240887)

First Supervisor:

Dr. Gerard Wagenaar

Second Supervisor:

Dr. Fabiano Dalpiaz

A master thesis submitted in partial fulfillment for the
degree Master of Science in Business Informatics

June 30th, 2023

Abstract

The Definition of Ready (DoR) is a collection of criteria used in Agile Software Development (ASD) to determine when a product backlog item is ready in planning iterative development sprints. Although the DoR concept is applied with experienced benefits, there is a lack of insight into practitioner usage in the literature.

This study investigates how ASD teams apply the DoR and attempts to formulate a framework for its adoption. It aims to uncover the relatively low adoption rate of DoR despite insinuated benefits and prerequisite availability, and the usage of existing criteria from literature like INVEST. The framework likewise serves as an extension of current definitions of DoR, providing aspects relevant to practitioners.

In this research a Multivocal Literature Review is conducted to index available grey- and scientific literature, followed by a multiple-case study involving software-producing organisations. Both phases' results are compared and subsequently analysed to formulate the DoR adoption (DA) framework. This is validated through expert opinion interviews resulting in a final version.

Concluding, most teams applying criteria to determine item readiness do so explicitly and are aware of the DoR concept. The most common rationales for DoR usage are reduced unclear items and communication overhead. Although INVEST occurs, a plethora of alternative criteria is identified. Teams do not always adhere to all criteria in their DoR, preferring flexibility and degrees of readiness.

Glossary

ASD - Agile Software Development

ATA - Agile Transition and Adoption (framework)

ATP - Agile transformation/- transition process

DA - DoR Adoption (framework)

DEF - Data Extraction Form

DoD - Definition of Done

DoR - Definition of Ready

DPO - Data Protection Officer

MLR - Multivocal Literature Review

PO - Product Owner

QUS - Quality User Story (framework)

SAFe - Scaled Agile Framework

SLR - Structured Literature Review

SPO - Software Producing Organisation

Contents

Abstract	i
Glossary	ii
List of Tables	v
1 Introduction	1
1.1 Agile Software Development and Scrum	1
1.2 Problem Statement	2
1.3 Research Aims	3
1.4 Thesis Outline	4
2 Research Questions	5
3 Research Method	7
3.1 Multivocal Literature Review	8
3.2 Exploratory Case Study	9
3.3 Comparison	13
3.4 Expert Opinion	13
4 MLR Protocol	15
4.1 Initial Search	15
4.2 Source Selection	17
4.3 Snowballing	20
4.4 Data Extraction	21
5 Literature Review Results	23
5.1 Explicit or Implicit DoR Application (SQ1)	24
5.2 Rationale for Applying the DoR (SQ2)	26
5.3 Criteria Applied in the DoR (SQ3)	29
5.4 Adherence to the DoR (SQ4)	34
5.5 Summary	36
6 Case Study Results	37
6.1 Case Characteristics	37
6.2 Thematic Coding Steps	40
6.3 Results	43
6.4 Summary	61

7	Comparison	62
7.1	Comparative Analysis	62
7.2	Framework Construction	72
8	Expert Opinion Results	78
8.1	End Results	79
8.2	Framework Adjustment	81
9	Conclusion	84
10	Discussion	87
10.1	Implications	87
10.2	Threats to Validity	88
10.3	Additional Findings	90
10.4	Future Research	91
A	Ethics Quick Scan	97
B	Literature and IDs	105
C	Literature Quality Scores	110
D	Data Extraction Forms	113
E	Interview Question Protocol	122
F	Case Study Data	124
G	Expert Opinion Protocol	125
H	Expert Opinion Data	127

List of Tables

4.1	Search queries per sub-question and the number of hits.	17
4.2	The inclusion and exclusion criteria.	18
4.3	The quality criteria for scientific literature.	19
4.4	The quality criteria for grey literature.	19
4.5	The number of items after each filtering activity.	22
5.1	Aspects of DoR explicitness found in literature.	24
5.2	Types of rationale found in literature.	26
5.3	Criteria found in literature.	30
5.4	Types of criteria adherence found in literature.	34
6.1	The case characteristics.	38
6.2	The type of rationales found in the case study.	44
6.3	The construction aspects found in the case study.	47
6.4	The aspects of explicitness found in the case study.	49
6.5	The criteria found in the case study.	51
6.6	The importance of criteria found in the case study.	56
6.7	The adherence aspects found in the case study.	58
7.1	The comparison of rationales.	63
7.2	The comparison of explicitness aspects.	66
7.3	The comparison of criteria.	68
8.1	The expert characteristics.	78

Chapter 1

Introduction

1.1 Agile Software Development and Scrum

In 2001 the Agile Manifesto was conceptualised, synthesizing the authors' novel views on software development into four values and 12 principles (Beck et al., 2001). Software development methods based on the Agile Manifesto have since taken over from traditional methods to be the most popular worldwide (Hoda et al., 2018).

Scrum or Hybrid Scrum (Scrum and another method combined) is reported by the 16th annual State of Agile Report to be used by 87% of practitioners, forming the mainstream in contemporary Agile Software Development (ASD) (State of Agile, 2022). Thus when referring to ASD in this report the envisioned method is Scrum or a Scrum-derivative.

As a method Scrum has several key aspects which form its basis (Sutherland & Schwaber, 2020). The first is that a team of around 10 people is the core organisational form, consisting of a Product Owner (PO), a Scrum Master, and developers.

The PO represents stakeholders and their needs and is responsible for managing the product backlog. A Scrum Master oversees the implementation and improvement of Scrum in the team and helps them in understanding the method. The developers are tasked with planning for the sprint and developing the sprint result, usable and valuable increments of the final product.

A second aspect is the key artifacts of Scrum; product backlog, sprint backlog, items, and increments. Work is collected in a prioritised list called the product backlog, containing items describing requirements for the final product. User stories are a commonly used format for representing backlog items, containing only the essential requirement elements from the perspective of the end user (Lucassen et al., 2015).

The sprint backlog contains the sprint goal (why), along with the items chosen for inclusion in the sprint (what) and the increment delivery plan (how). An increment is the sum of backlog items converted to facets of the final product in the sprint, which are usable and thus create value.

The last aspect is the main events already mentioned; sprint and sprint planning. The sprint is a predefined period in which the goal is to develop an increment that delivers value to the stakeholders. In sprint planning, the sprint's goal and scope of development are determined, while developers consider how to convert backlog items into increments.

A focal point in the Scrum method is deciding what backlog items are ready to be worked on in a sprint and when the development of items is done so that they are in a deliverable increment. Although not in the Scrum guide, the Definition of Ready (DoR) is suggested by some sources to determine which product backlog items are ready for inclusion in a sprint backlog (Dalton, 2018; Rubin, 2012; Sutherland & Schwaber, 2020).

The DoR is a collection of criteria for product backlog items about their quality and completeness (Rubin, 2012). In general, it is used during backlog refinement and sprint planning, where backlog refinement ensures that items are ready (e.g. completely described) to be planned. Some benefits of the DoR experienced in practice are reduced waste, meaning less work is done which does not add value, and an improved workflow in development (Power, 2014).

A related concept is the Definition of Done (DoD) which acts as a closing gate to the sprint, applied to determine when an item is successfully developed into an increment, or in other words, done (Sutherland & Schwaber, 2020). Consequently, the DoR and DoD can be applied together to improve the increment value through monitoring sprint in- and output with agreed-upon criteria.

1.2 Problem Statement

Despite the suggestion to use a DoR in ASD and both claimed and demonstrated benefits, its adoption rate by Agile teams is low (Klotins et al., 2021; Wagenaar et al., 2018). This contrasts with a high adoption rate of the prerequisite concepts product backlog and backlog items and of the related activity backlog refinement.

In addition, limited research has been performed on how the DoR is applied in practice and what its benefits perceived by practitioners are. The case studies by Power (2014) and Diebold et al., (2018) are anomalies, examining the usage and subjective benefits of a DoR in practice, which only provides a limited frame of reference. Both studies were performed within one company meaning that results may not be generalisable to other Agile teams.

Several scientific works recommend requirement quality frameworks evaluated with practitioners or in case studies (Heck & Zaidman, 2014; Lucassen et al., 2017). Heck & Zaidman (2018) expand this with a review of scientific literature on quality criteria in search of confirmation for their framework. Additionally, Lucassen et al., (2016) investigate the use and perceived effectiveness of user stories in practice, finding that interviewees are not using quality frameworks due to unawareness.

However, requirements quality is only one part of backlog item readiness, a notion that also includes ASD process-oriented aspects. A comprehensive overview of what constitutes a ready backlog item to Agile teams thus remains unexplored in scientific literature.

Grey literature recommends among others the INVEST criteria conceptualised by Wake (2003), for determining item readiness: Independent, Negotiable, Valuable, Estimable, Sizeable and Testable (Aschauer et al., 2019; Barjis, n.d.; Dalton, 2018). However, INVEST is merely based on the creator's intuition, and criteria are open to interpretation. Reports on DoR practice in grey literature are likewise scarce, with coverage being mostly suggestions or anecdotal examples.

Together this shapes a theoretical problem concerning a shortage of scientific insight into the DoR and its usage in ASD teams: The problem manifests as a lack of knowledge about its application (*how*) and the occurring benefits (*why*):

It is unknown how and why ASD teams adopt and apply the DoR in practice.

The goal of this study is to close the gap between literature and practice, by exploring how the DoR is applied in ASD by practitioners. When this issue is addressed more focused research can be performed on the objective and subjective benefits of the DoR in ASD.

1.3 Research Aims

The first aspect to be covered in addressing the problem is why the adoption rate of the DoR is relatively low, considering that prerequisites for its application are already being applied by Agile teams. An assumption is that teams may be informally applying a DoR within backlog refinement, due to Agile requirement engineering's reliance on face-to-face communication (Inayat et al., 2015).

Additionally, it should become apparent how Agile teams define what constitutes a ready backlog item and what criteria are applied to determine this. Are teams applying their interpretation of the INVEST acronym, or are they using a broader spectrum of criteria such as those proposed in the scientific literature? For companies implicitly applying a DoR the quality criteria may be ad-hoc in nature, depending on the item at hand (Lucassen et al., 2016).

Lastly, it should become apparent if the currently available definitions of the DoR in literature match its usage in practice, or if an extension in the form of a framework is required for the benefit of future research. Examples of aspects that might be included are when the DoR is applied, by whom it is applied, and what range of criteria may be considered aside from INVEST.

1.4 Thesis Outline

The continuation of the report is as follows. First, the research question and the derived sub-questions are covered (Chapter 2) which explains how the research problem and aims are addressed. Secondly, the research method is discussed (Chapter 3), concerning how the questions are to be answered.

Following is the protocol for conducting a Multivocal Literature Review (Chapter 4). Consequently, the literature review results are presented as the conclusion to the first research phase (Chapter 5). The case characteristics and case study results from the next phase are discussed in Chapter 6.

In the third phase, the MLR and case study results are compared and analysed, leading to the construction of the DoR adoption framework (Chapter 7). This framework is validated in the last phase through expert opinion interviews, resulting in an adjusted model for its representation (Chapter 8).

The research is concluded in Chapter 9, providing the answers to the posed sub-questions, which accordingly addresses the main question of how the DoR is used in practice. In Chapter 10 the results' implications, validity threats, additional findings, and future work are discussed, finalising the report. Additional materials are attached in the subsequent appendices.

Chapter 2

Research Questions

Based on the problem statement and research aims the following main research question is posed:

RQ: How is the Definition of Ready applied in Agile software development teams?

To comprehensively answer this question it is divided into several sub-questions corresponding to the research aims. The first sub-question concerns the explicitness of Agile software development team's Definition of Ready:

SQ1: Is the Definition of Ready applied explicitly or implicitly by Agile software development teams?

It should be examined if Agile software development teams apply criteria to determine backlog item readiness without explicit definition and/or awareness of the DoR. This type of concept usage will from now on be referred to as an implicit DoR.

The implicit application by teams, most notably unawareness, can possibly explain the concept's relatively low adoption rate found in previous works. When it is known whether Agile teams apply the DoR explicitly or implicitly it can be examined what the rationale is for explicitly applying a DoR. This leads to the second sub-question:

SQ2: What is the rationale of Agile software development teams for applying the Definition of Ready?

From this, it can be derived if the rationale given by Agile software development teams for explicitly using the DoR adheres to its description in scientific- and grey literature. Diverging rationales can point to a gap between how the DoR is applied in practice and how it is suggested in theory. To further investigate the usage of the DoR a third sub-question is posed:

SQ3: What criteria are applied by Agile software development teams in their Definition of Ready?

It should become apparent by what criteria Agile software development teams determine backlog item readiness. As discussed several readiness or quality criteria frameworks are suggested in different types of literature and web resources, but the usage of criteria in practice remains a blind spot. When it is known what criteria Agile teams apply it can be investigated when a backlog item is considered to be ready, forming the fourth sub-question:

SQ4: When do Agile software development teams consider a backlog item sufficiently ready to move into a sprint?

The perception of the ready state attributed to backlog items shall be investigated. Or in other words, to what extent do ASD teams adhere to their DoR criteria before moving an item to the sprint?

Assumed influential aspects are the degree to which selected criteria are met before an item is marked ready and at what moment readiness is determined. The answers to the previous sub-questions may offer insights into the DoR's practitioner usage which are not included in the literature. Consequently, the fifth and last sub-question is as follows:

SQ5: What would a framework for the adoption of Definition of Ready encompass considering its usage in practice?

Aspects of DoR's practitioner usage which are currently not specified across existing definitions of the concept should be considered for the framework. In this, the goal is to propose a framework for DoR adoption based on practitioner usage when possible supported by literature. However, this research is not aimed at finding the most ideal usage of DoR. Thus the framework aims to clarify what practical aspects should be taken into account when researching or adopting the DoR.

Chapter 3

Research Method

The main goal of this research is to close the gap between literature and practice by exploring how the Definition of Ready is applied in ASD. In this chapter the research method is covered which contains four phases presented in Figure 3.1.

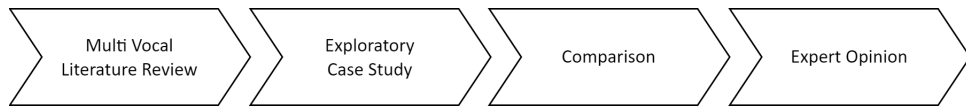


FIGURE 3.1: The research method phases.

In the first phase a Multivocal Literature Review (MLR) is undertaken, a form of Structured Literature Review (SLR), to demonstrate the importance of both the research question and case study (Yin, 2017).

After the analysis of the state-of-the-art in literature (MLR) and the state-of-the-practice (Case Study) is completed a comparison is made to identify and address gaps between them. The research is closed by an expert opinion phase wherein the proposed framework for DoR adoption is validated through interviews.

This four-phase methodological set-up provides a comprehensive overview of the literature and practical perspectives fitting to the research aims' exploratory nature. In the following the four phases are discussed further, covering their design and the rationale behind them.

The first three phases are inspired by other ASD research which combine a literature review with a case study, after which the results are compared (Drury-Grogan et al., 2017; Islam & Storer, 2020; Kopczyńska et al., 2022; Wagenaar et al., 2018). Expert opinion as the closing phase is derived from Wieringa's (2014) design science methodology, in which practical validation forms a key aspect.

Tom et al., (2013) leverage a similar ordering of phases in their exploratory research on technical debt in software development. They conduct an MLR in parallel with semi-structured interviews after which a framework is developed through comparative analysis. For practical reasons the MLR and Case Study are performed sequentially in this research.

Their constructed framework is consequently validated in informant reviews, which is comparable to expert opinion interview validation, with the main difference being the medium of conducting validation (textual comments vs. interview comments).

Following are phase descriptions that additionally address possible validity threats arising from the design and counter-measures to mitigate them. To safeguard the ethics in this research the Ethics and Privacy Quick Scan as required by the Research Institute of Information and Computing Sciences at Utrecht University was performed. No issues related to ethics or privacy were found, full results are presented in Appendix A.

3.1 Multivocal Literature Review

An MLR is performed to gain a comprehensive insight into the Definition of Ready state-of-the-art. Currently, no DoR-specific SLR exists which means there is no oversight on the available literature, making research into the concept more difficult.

Likewise, a considerable amount of the concept's coverage is found in handbooks, internet resources, and student theses. These sources are classified as grey literature of varying shades or tiers, determined by the explicitness and transparency in production, moderation, or editorialisation and the extent of determinable author expertise (Garousi et al., 2019).

MLR is conceptualised as an alternative to SLR, facilitating the inclusion of both scientific- and grey literature. The process attempts to mitigate the reliability and bias issues present in grey literature, but completely solving them is not possible due to the commonly occurring lack of scientific basis and methodological rigour.

Nevertheless, grey literature is included in this research to get a comprehensive overview of the state-of-the-art including practitioner views. This broadens the generalisability of the results mitigating the external validity threat occurring in SLR due to its focus on scientific literature.

The need for MLR is determined by the Garousi et al., (2019) checklist which considers necessity from the perspective of the research, the scientific community, and practitioners. Five of the seven points apply to this research suggesting that including grey literature in the review is warranted.

To conduct an MLR a three phase process is used which will be covered briefly. The first phase requires the preparation of the literature review, this begins with determining the goal and establishing the need for MLR, which was already discussed.

This phase is concluded by constructing a literature review protocol containing queries, in- and exclusion criteria, quality assessment criteria, and extraction and analysis procedures. In Chapter 4 the literature review protocol, validity threats, and their counter-measures are covered extensively.

The next phase is review execution wherein literature is found based on the posed queries and filtered based on criteria, which is likewise reported in Chapter 4. This phase also comprises data extraction from literature into data extraction forms (DEFs) and analysis of the recorded data. The final phase is to report the analysis results in Chapter 5, which address the MLR goal and provides the first insights for answering the research sub-questions.

3.2 Exploratory Case Study

To investigate how the DoR is applied in Agile teams the case study approach is chosen. This approach is an appropriate means for exploratory research into phenomena in their real-world context (Yin, 2017).

Contrary to the previous case studies of DoR covered in Section 1.2 this study will focus on multiple companies and takes a broader perspective on DoR usage. The desired result of the case study is an insight into the state-of-the-practice of DoR which similarly to the state-of-the-art is an aspect under-examined by scientific research.

Creating a case study protocol is a requirement for ensuring the successful execution of the study (Wohlin et al., 2012). Various templates exist for constructing a protocol containing case study design and procedures. In this research, the protocol template by Brereton et al., (2008) is applied.

Sections such as data collection are covered more extensively, while non-applicable parts like case study roles are omitted. Since the research background and sub-questions were already discussed the study design is presented here first.

3.2.1 Case Study Design

The planned case study is a multiple-case study in which each Agile team is studied holistically as a single case, the multiple individual cases are then compared to reach a common understanding of how the DoR is used in practice (Yin, 2017).

Since a singular DoR is generally applied throughout the complete Agile team the holistic study design is chosen in which the case is analysed by one unit of analysis. An embedded case study requires that data is collected from multiple roles forming multiple units of analysis.

Because the Scrum guide suggests that Agile teams consist of ten or fewer members, a single participant per case will be involved in the case study as a representative of their team (Sutherland & Schwaber, 2020).

3.2.2 Case Selection

To perform a case study one must know what is considered a case and when a case is to be included. As noted before Agile teams are considered a single case, but due to increased interest in Agile methodology from domains outside of software development a specification is needed (Ciric et al., 2019).

Software producing organisations (SPOs) in the Netherlands of differing sizes will be contacted to find practitioners from Agile teams to represent their case as a participant for data collection (Wagenaar et al., 2018).

If a company has multiple Agile teams with each their own implicit or explicit DoR they can be considered single cases which may be investigated separately. The aim is to find between five and ten organisations that wish to provide an employee to represent their case in the study.

Additionally, in contacting the SPOs a simplified explanation of the Definition of Ready is given as follows: *“The Definition of Ready involves setting criteria for an item on the Backlog before it can be worked on in a Sprint.”*.

To this is added that there is also the opportunity for an interview when there is no explicitly established DoR in the team, or when they are unfamiliar with the concept but nevertheless apply criteria to items before the sprint. These points are clarified to ensure that teams with a potentially implicit DoR are not repelled.

The type of sampling used in case selection corresponds to convenience sampling, which Wohlin et al., (2012, p. 93) describe as: *“The nearest and most convenient persons are selected as subjects.”*.

Convenience sampling is leveraged to ensure sufficient participants which adhere to the criteria are found within the limited time frame of the research project. A downside to this type of sampling is a lack of control over case characteristics which could in turn cause an uneven representation, this threat is discussed further in Section 3.2.6.

3.2.3 Data Collection

In this case study the main way of data collection is through semi-structured interviews with the participants, in which each participant is interviewed once. Semi-structured interviews give the interviewer freedom to probe and explore interviewee knowledge beyond what was intended in the interview protocol (Kajornboon, 2005).

Audio recordings of the interviews will be made with explicit informed consent from the participants. Anonymous transcripts are then made of the interviews which are used for data analysis.

After finalisation of the case study, the audio recordings shall be removed. All case study data is stored locally or on personal cloud storage provided by Utrecht University during the research. The transcripts are analysed using NVIVO¹ software for which the files are stored locally. NVIVO version 1.7.1 is used at the time of processing.

For archiving the NVIVO data set after the research the Open Science Framework platform is used², which is approved by Utrecht University. The data is stored in a private project on the OSF server in Frankfurt, Germany, which can only be accessed through the provided url.

3.2.4 Interview Protocol

An interview question protocol is constructed as suggested by Brereton et al., (2008) to enable data collection in the interviews. The questions posed are formulated after the completion of the MLR, which may provide additional insights into how the SQs can be answered empirically.

In the following the protocol’s structure and questions are elaborated upon. The full version of the question protocol is presented in Appendix E, which shows the precise mapping between the interview questions and the research’s sub-questions.

¹<https://lumivero.com/products/nvivo/>

²<https://osf.io/>

The structure of the interview question protocol is partially based on Roberts (2020) and Wohlin et al., (2012). Both recommend to start with an introduction to put the interviewee at ease and to explain the interview's objective.

Following are introductory questions about the interviewee's context which shape the case characteristics, such as their role and the type of software their team develops. This likewise serves as a way to express mutual interest in the participant's situation.

Consequently the main questions are as follows:

1. Do you use criteria for determining when a backlog item is ready for sprint planning, and if so do others in the team as well?
2. Is the DoR an agreement between people in the team and/or with external stakeholders?
3. Why do you use the DoR?
4. What criteria are used in the DoR?
5. To what degree is the DoR adhered to when an item is moved into the sprint?

Each main question is accompanied by a number of sub-questions which may be used as probes to ensure that all relevant aspects are covered. The main questions are ordered to intuitively form a story about the team's DoR which touches all research sub-questions.

First the presence and explicitness of the DoR is covered followed by its internal and potentially external communication. Then the rationale for applying a DoR is uncovered after which the criteria are discussed, which in turn could be influenced by the previous points. When the criteria are known the degree of adherence to them, and consequences of non-adherence are covered.

The interview is closed with concluding questions concerning whether there is additional information on the DoR or team which has not been discussed so far, and whether documentation regarding the DoR is available for sharing. Additionally the interviewee is asked whether they would like to receive the final thesis report, which contributes to the dissemination of findings to the practitioner audience.

3.2.5 Data Analysis

In the case study, thematic coding is applied as a realist data analysis method to report on meanings, experiences, and the reality of the participants involved (Robson, 2002). Thematic coding is a qualitative analysis method that relies on coding to determine the meaning of text segments. Once the text is coded the segments can be accumulated into themes which are aspects related to the sub-questions.

Codes and themes are determined inductively during analysis based on the data or in some cases pre-constructed based on theory. The inclusion of theory reinforces the analysis by highlighting data features that could be overlooked without this knowledge. After completing the MLR it shall be determined what aspects of literature apply to coding.

To perform thematic coding several steps are executed, some of which are in parallel, or are performed during data collection. Parallel execution in this case means that codes and themes can be discovered at any time in the process and that analysis is continuous. This continuity allows analysis insights to be incorporated into further data collection efforts. In the Section 6.2, each step will be reported upon to demonstrate the analytical quality.

3.2.6 Case Study Validity

The first type of validity to account for during the case study is construct validity. This validity concerns the generalisation of the case study to the concept or theory underlying it (Wohlin et al., 2012). To ensure construct validity multiple sources of evidence are used, resulting in the multiple-case study design. Also, a chain of evidence is to be constructed by transcribing and coding the interview contents, documented in Appendix F.

The second type of validity relevant to case studies is internal validity, or whether the causal relationship between intervention and outcomes can be proven. However, since this research is exploratory rather than explanatory internal validity is of limited relevance. This is emphasized by a lack of intervention and subsequent outcomes in the case study.

Another validity type to consider is external validity, the extent to which the case study results can be generalised. The external validity of the case study is threatened by focusing only on SPOs from the Netherlands. Another threat that could occur is an uneven representation of exact Agile methodology (i.e. Scrum vs. Hybrid Scrum), company sizes, roles from Agile development, and in-house or external development.

Due to time and resource constraints, it is difficult to counter these external validity threats within the scope of this research. In future research, this generalisability could be improved by performing a study on a larger and broader selection of cases representing the mentioned factors.

Lastly, the reliability as a validity type is to be taken into account. This pertains to the repeatability of research by other researchers leading to the same results. The current case study protocol is presented to provide the opportunity for repetition by highlighting the exact steps taken to come to the conclusions, which likewise leads to a chain of evidence.

3.3 Comparison

The research method's third phase encompasses a comparison of the MLR and case study results to draw conclusions and answer the main research question. This analysis builds upon the previous chains of evidence to enhance construct validity. In this comparison, the aim is to highlight, and where possible, close the gap between literature and practice.

For each sub-question, various aspects can be compared based on the overlap between MLR findings and coding themes. An example of an aspect to compare could be whether the different degrees of DoR explicitness (SQ1) assumed in literature (Section 5.1) are observed in practice and if so what practitioner views underlie this.

This comparative information can be represented in table format for clarification as leveraged by Kopczyńska et al., (2022). Table format representation likewise serves to improve the reliability of analysis by offering insight into how conclusions are drawn. The result of this phase is to construct the framework as posed in SQ5, with special attention to aspects from practice overlapping with literature, and relevant novel aspects from practice.

To construct the DoR adoption framework the framework by Gandomani and Nafchi (2015) is adapted. This framework aims to assist teams in the Agile transformation/-transition process (ATP) by iteratively, continuously and gradually introducing facets of ASD. In Section 7.2 this adaptation and the preliminary framework are discussed in more detail.

3.4 Expert Opinion

To validate the framework expert opinion is used from between two to four experts. A minimum of two is required ensuring multiple independent opinions are available to enhance construct validity, while four is chosen to bound the effort.

Expert opinion is suggested in Wieringa's (2014) design science methodology as a means to validate a designed artifact in interaction with its context, based on experts' expectations. Although a framework is not an artifact in the traditional sense, it describes how a DoR takes shape instead of being a concrete instance, expert opinion is applied to enhance the validity.

For data collection, the expert opinion sessions will be recorded after which anonymous transcripts are made. These transcripts are then analysed to extract the participants' opinions regarding the previously mentioned aims. Together this forms a chain of evidence which aids the expert opinion phase's construct validity. Depending on the results the framework is either adjusted or remains in the preceding phase's form with expert opinion results added as contextualisation.

3.4.1 Validation Aims

The first aim of this validation phase is to examine if the framework is clear to the experts and whether they expect it to be understandable to other practitioners. This ensures that the framework is suitable for an audience beyond the scientific community, and corresponds to the views regarding DoR in practice. Should the framework be incomprehensible to experts adjustment is required, which also goes for expected misunderstanding by practitioners.

Secondly is to validate whether the framework is sufficient to enable well-informed implementation of a DoR for a team, project, or organisation. Although the foremost aim of the research is to aid science in understanding practitioner usage of the DoR, the framework should be applicable in practice. If the framework passes this validation aspect it's grounding in practice is confirmed further.

Thus, the sessions are structured by the DoR aspects covered in the framework, negating the need for a strict question-based interview protocol. Nevertheless, a short protocol is constructed to aid in performing the expert opinion interviews and improve reliability as presented in Appendix G.

Additionally, to ensure the experts are familiar with the framework before its validation they are offered the DoR adoption framework a week in advance, without the accompanying description text. In the expert opinion sessions a global explanation of the framework is given akin to that in Section 7.2.2.

3.4.2 Expert Selection

To select experts for this phase selection criteria are required, which in turn define what an expert in ASD and DoR is in the case of this research. The selection again takes place based on convenience sampling due to time limitations and scarcity of willing participants. In this an external validity threat arises since the results may not be generalisable to practitioners which fall outside of this definition of an expert.

Firstly, the participants should be available for the expert opinion interview within the set time frame. Likewise, they should be willing to cooperate in the interviews which will be recorded and processed, for which informed consent is likewise required. The offered informed consent form is the same as that leveraged in the case study.

The participants should have relevant knowledge about ASD methodology and the DoR concept so that they can cooperate in validation and substantiate their line of thinking. This knowledge is verified in the selection of experts by inquiring about their familiarity with both topics and by checking credentials such as certifications.

Subsequently, the experience of participants in implementing ASD, teaching ASD, or enforcing ASD usage is important. Experienced participants are necessary since their experience can provide additional insights besides those arising from their knowledge of theory.

Experience is accounted for in selection by focusing on ASD trainers, and by aiming at more than five years experience in this role, preferably working with the DoR all this time. Trainers are expected to be familiar with a wide variety of ASD team circumstances, which provides additional value in judging the framework.

Chapter 4

MLR Protocol

As discussed in Chapter 3 an MLR is performed to provide the theoretical basis for the research. For this, the Garousi et al., (2019) process is followed, of which the justification for selection and main phases were discussed.

A main artifact based on this process is a literature review protocol which is covered in the following sections. Kitchenham and Charters (2007) advise constructing a predefined review protocol to reduce researcher bias and to report on the review steps in detail to improve reliability. This literature review protocol is therefore dual purpose, presenting both the predefined steps and methods and the intermediate results of each step.

For ease of discussion the unique items included in qualitative filtering are given IDs, which are defined in Appendix B. In this ID format an *S* signifies scientific literature, a *G* grey literature, and an *R* items removed in qualitative filtering or thereafter.

This chain of evidence likewise aids in improving construct validity, providing traceability from source to conclusions. The overarching protocol begins with the second phase, Conducting the MLR, since the first phase concerning the need for an MLR and its goal were already covered.

4.1 Initial Search

The initial literature search is performed within two resources; Google Scholar and Google Search. Initially, the sources are applied to searching respectively only scientific literature and grey literature.

However, Google Scholar indexes scientific and grey literature meaning it can be applied to searching both types, which is taken into account in further steps. Contrarily, Google Search provides a negligible amount of scientific literature on the DoR, as a consequence it is solely used for finding grey literature.

Thus, Google Scholar is chosen as the main resource for Scientific Literature since it indexes a wide array of relevant journals and conference proceedings in the Software Engineering domain. Grey literature found using Google Scholar is likewise taken up in the MLR results as to not miss potential novel hits.

Google Search suffices for finding grey literature being the de-facto standard for general web searches. To collect the found items the Zotero¹ reference management software package is used, which will also be used for tagging items as in- or excluded in further review steps.

For grey literature, various source types are categorised by Garousi et al., (2019). In this research, the initial focus is on books, blog posts, company or organisational web pages, government reports, student theses, and white papers. Source types such as tweets are omitted to bound the effort and due to an expected lack of relevance to this research.

4.1.1 Search Queries

To find literature in the resources ten structured queries are used as presented in Table 4.1, along with the sub-question they correspond to and the number of resulting hits. SL queries are exclusively used in Google Scholar and GL queries exclusively in Google Search. Notably, SQ5 does not directly correspond to any queries since it is addressed by analysing the answers to preceding questions.

The ten search queries are constructed according to a three-part pattern merged by the AND operator indicating that items should contain all main parts of the query. All queries but one start with ‘definition of ready’, the main concern of the research which should filter out any items not discussing the topic. Terms between quotation marks are required to be present verbatim in the resulting items.

The second query for SQ3 starts with (agile OR scrum OR “just-in-time” OR “just in time”) to find related literature on requirement quality that does not necessarily mention DoR. The usage of OR between brackets indicates that items should contain either X or Y or both. Each query ends with a sub-question-specific term providing more focused results. This term is refined based on trial-and-error of which the current version gives the most satisfying results.

Another consequence of trial-and-error is a deviation between the queries used for searching SL and GL. In the case of SQ2 this is due to a lack of novel results when applying both SL queries to GL in Google Search, with the second query providing the most fruitful result.

The deviation for SQ3 is similarly caused by a lack of novel and/or useful results when applying the second SL query in Google Search. An adjusted version of the first SL query is used to avoid an abundance of results covering INVEST as opposed to other criteria. Although the original query contained the ‘OR’ operator, including it resulted in a disproportionate coverage of INVEST compared to other criteria in the results.

Consequently, the search is stopped after the first ten pages containing ten items each, bounding the effort for each query at 100 items. Thus the Initial Search considers 1,000 items in total, which does include duplicate items found using multiple queries. This nevertheless high number is desired for a comprehensive overview of the available literature on DoR which has not been indexed before, and to decrease the internal validity threat of overlooking literature.

¹<https://www.zotero.org/>

SQ	Query	Hits
SQ1	<i>Scientific Literature</i>	
	“definition of ready” AND (agile OR scrum) AND (explicit OR implicit)	147
	<i>Grey Literature</i>	
	“definition of ready” AND (agile OR scrum) AND (explicit OR implicit)	18,600
SQ2	<i>Scientific Literature</i>	
	“definition of ready” AND (agile OR scrum) AND “rationale”	68
	“definition of ready” AND (agile OR scrum) AND “benefits”	244
	<i>Grey Literature</i>	
	“definition of ready” AND (agile OR scrum) AND “benefits”	40,800
SQ3	<i>Scientific Literature</i>	
	“definition of ready” AND (agile OR scrum) AND (criteria OR INVEST)	327
	(agile OR scrum OR “just-in-time” OR “just in time”) AND (“user story” or requirement) AND “quality”	7,460
	<i>Grey Literature</i>	
	“definition of ready” AND (agile OR scrum) AND “criteria”	48,400
SQ4	<i>Scientific Literature</i>	
	“definition of ready” AND (agile OR scrum) AND “is ready”	90
	<i>Grey Literature</i>	
	“definition of ready” AND (agile OR scrum) AND “is ready”	15,500

TABLE 4.1: Search queries per sub-question and the number of hits.

4.1.2 Initial Filtering

Additionally, some inclusion criteria are already applied in this step to filter irrelevant literature. This applies to the language (English or Dutch), a date after the DoR’s conceptualisation by Jakobsen and Sutherland (2009), and the topic being Software Development. Inaccessible literature and duplicate results for the same query are excluded immediately as well.

Lastly, grey literature items which merely mention DoR or provide a short definition (1 - 2 sentences) are excluded as well. These items are deemed irrelevant for in-depth research into DoR. After performing this step 644 items remain, of which 250 are scientific literature and 394 grey literature.

4.2 Source Selection

In the previous section, some preliminary in- and exclusion criteria were covered, which is extended in this step with sub-question-specific criteria. The additional criteria are presented in Table 4.2, ordered by sub-question, together with those already applied in the category all. By applying this list only literature expected to be relevant for both the DoR and sub-questions at hand remains.

After applying the criteria 243 items are left, of which 51 scientific literature and 192 grey literature. Of the grey literature, 136 items are either blog posts or web pages with the remaining 56 items being of types that Garousi et al., (2019) call 1st tier grey literature. For this literature tier, the authority and knowledge of the producer and the outlet control such as a reliable publishing organisation are largely determinable, which only goes partially for blog posts and web pages.

SQ	Criteria
All	Inclusion
	English or Dutch, year of publication >2009 (Jakobsen & Sutherland, 2009), literature is about Software Development.
	Exclusion
	Inaccessible literature, duplicate literature within query results, only mention or shortly describe DoR.
SQ1	Inclusion
	The literature mentions that teams have or should have a notion of backlog item readiness for inclusion in a sprint.
	Exclusion
	The literature uses the term ‘ready’ but means ‘done’.
SQ2	Inclusion
	The literature mentions the rationale behind- or consequences of having a DoR.
SQ3	Inclusion
	The literature mentions criteria for determining backlog item readiness. Or, the literature mentions criteria for user story quality in the context of Agile and/or Scrum, which can be applied for determining readiness.
SQ4	Inclusion
	The literature mentions an extent to which DoR criteria need to be met before a backlog item is considered ready.

TABLE 4.2: The inclusion and exclusion criteria.

4.2.1 Qualitative Filtering

For the remaining items, the quality is checked so no literature of inferior publishing or research quality enters the results. To determine quality separate lists for both scientific- and grey literature are constructed. The quality criteria list used for scientific literature is given in Table 4.3 and the list for grey literature in Table 4.4.

Both lists are adapted from existing quality criteria lists, the scientific literature list from the ‘checklist for qualitative studies’ by Kitchenham & Charters (2007), and the grey literature list from that by Garousi et al., (2019).

Adaptation of quality criteria lists based on appropriateness for the research question at hand is recommended by both sources. Garousi et al., (2019) also indicate that combining lists for scientific- and grey literature in a review enables the systematic collection of evidence from a wider array of sources by objectively assigning a rigour score.

The two lists rely on a scoring mechanism adapted from Fliefel (2013) wherein each criterion has a numerical score used to calculate a total score. For the scientific literature list, the answer options are adapted from Fliefel (2013) as well, while for grey literature the binary answer model from Garousi et al., (2019) is kept. The total score is normalised to one by dividing with the maximum score, for which the quality coefficients are: 0.8–1 Excellent, 0.5–0.8 Average, <0.5 Poor.

In the continuation of the literature review, all literature rated Poor is excluded. Likewise, grey literature scoring zero on criteria 5.1 and 5.2 are excluded when overall scoring is average or lower. The reason is that grey literature is overall of lower quality than scientific literature, meaning additional caution must be had.

Item	Description	Rating
1	How credible are the findings?	0: Clearly incredible 1: Possibly credible 2: Clearly credible
1.1	If credible, are the findings important?	0: Unimportant 1: Important
2	How has knowledge or understanding been extended by the research?	0: No extension 1: Extension
3	How defensible is the research design?	0: Clearly indefensible 1: Possibly defensible 2: Clearly defensible
4	How well defined are the sample design/target selection of cases/documents?	0: Clearly insufficient 1: Possibly sufficient 2: Clearly sufficient
5	How clear and coherent is the reporting?	0: Clearly insufficient 1: Possibly sufficient 2: Clearly sufficient
6	How adequately has the research process been reported?	0: Clearly inadequate 1: Possibly adequate 2: Clearly adequate

TABLE 4.3: The quality criteria for scientific literature.

Item	Description	Rating
1.	<i>Authority of the producer</i>	0 - 2
1.1	Is the individual author associated with a reputable organisation?	0: No 1: Yes
1.2	Does the author have expertise in the area? (E.g. job title)	0: No 1: Yes
2.	<i>Methodology</i>	0 - 1
2	Is the source supported by authoritative, contemporary references?	0: No 1: Yes
3.	<i>Objectivity</i>	0 - 2
3.1	Does the work seem to be balanced in presentation?	0: No 1: Yes
3.2	Is the statement in the sources as objective as possible, or is the statement a subjective opinion?	0: No 1: Yes
4.	<i>Date</i>	0 - 1
4	Does the item have a clearly stated date?	0: No 1: Yes
5.	<i>Novelty</i>	0-2
5.1	Does it enrich or add something unique to the research?	0: No 1: Yes
5.2	Does it strengthen or refute a current position?	0: No 1: Yes

TABLE 4.4: The quality criteria for grey literature.

The resulting quality scores of scientific- and grey literature are presented in Appendix C. After qualitative filtering 92 items remain, of which 44 scientific literature and 49 grey literature. When removing the duplicate items which are found in multiple queries, but are only checked for quality once, a total of 54 unique items is left. 28 Items are scientific literature and 26 items grey literature, these form the sets to be processed in the following steps.

Through applying the in/exclusion and quality criteria it became apparent that blog posts and web-pages do not include sufficient novel knowledge to warrant their further inclusion in the review. Combined with the overall low quality it is decided to exclude these source types, removing 136 items from the grey literature set.

4.3 Snowballing

An adjustment made to the MLR process is to trial backward- and forward snowballing after in/exclusion and quality criteria are applied to determine the merit for its full implementation. The reason is that the initial set of sources already captured 1,000 items of various types, for which reasonable effort would be exceeded by attempting snowballing at the second phase's first step as suggested by Garousi et al., (2019).

Thus, snowballing is trialed on the top three scientific and grey literature items, in terms of quality score and relevance (times found with queries). These items are expected to be the most viable for both types of snowballing. The included items are as follows:

Scientific Literature

- Heck, P., & Zaidman, A. (2018) : Quality score 1.0, found in 5 queries.
- Heck, P., & Zaidman, A. (2014) : Quality score 1.0, found in 4 queries.
- Wagenaar et al., (2018) : Quality score 1.0, found in 3 queries.

Grey Literature

- Aschauer et al., (2019) : Quality score 1.0, found in 2 queries.
- Kneuper, R. (2018) : Quality score 1.0, found in 2 queries.
- Reeder, L. de. (2019) : Quality score 0.88, found in 4 queries.

Backward- and forward snowballing and applying in/exclusion criteria resulted in four new items. Three of these four items were different editions of the Scrum guide by Sutherland & Schwaber (2020), which was already included in the introduction (Section 1.1). Thus snowballing is not used further in this literature review, which is bolstered by the promising number of 55 filtered unique items already found through the structured search.

4.4 Data Extraction

To capture the information relevant to the research's sub-questions from the remaining literature, and to improve review result traceability, two Data Extraction Forms are designed in Excel.

In these forms, each item has a unique ID, APA reference, and four columns corresponding to the sub-questions. Including the APA reference for keeping basic information is recommended by Kitchenham & Charters (2007) for ease of further processing.

Additionally, in the scientific literature Data Extraction Form the research type based on categorisation by Wieringa et al., (2006) is added. The research type provides context to the registered information, which is after synthesis discussed in Chapter 5. An exception are Structured Literature Reviews which are not included but are nevertheless used as a category in the form.

The categories by Wieringa et al., (2006) are as follows:

- **Evaluation Research:** A problem or the implementation of a Requirements Engineering technique is investigated in practice through for example case studies or empirical mathematical research.
- **Proposal of Solution:** A solution technique is proposed and argumentation for its relevance is given without full validation.
- **Validation Research:** The properties of a solution proposal that is not yet implemented in Requirements Engineering practice is investigated through for example experiments.
- **Philosophical Papers:** A new perspective for looking at things is given such a new conceptual framework.
- **Opinion Papers:** The author's opinion is given about the valuation of something, or how something should be performed.
- **Personal Experience Papers:** The author's personal experience concerning one or more projects is given, stressing what instead of why.

The resulting filled-in Data Extraction Forms are presented in Appendix D. Where possible the information from the items in the sub-question columns is included verbatim for ease of traceability.

When information is in verbatim it is enclosed in quotation marks, if not it concerns a summary of the segments discussing the respective sub-question. In the case of an empty cell, the item does not discuss any aspects of the sub-question, determined by the hand of the in/exclusion criteria used previously in the selection step.

Through filling in the Data Extraction Forms it is decided that two more scientific literature items (R4 and R5) are removed since another version by the same authors is already included, meaning no novel information could be extracted.

This brings the final number of items included in the review results to 53 unique items, of which 26 are scientific and 27 are grey literature. In Table 4.5 the intermediate number of items after each filtering activity is presented. The synthesis of extracted data and its results are discussed in the review results in Chapter 5.

	Initial Selection	Preliminary In/Exclusion	Full In/Exclusion	Quality Criteria	Unique Items	Final Count
Total	1,000	644	243	93	54	52
SL	-	250	51	44	28	26
GL	-	394	192	49	26	26

TABLE 4.5: The number of items after each filtering activity.

Chapter 5

Literature Review Results

This chapter discusses the results of the MLR structured by the first four sub-questions. A distinction is to be made concerning the main research question; in the MLR literature is found which suggests the DoR or makes assumptions about its usage, while other literature describes how the DoR is applied in practice.

The first type of literature is used for identifying choices that can be made in DoR application. The second type of literature forms a frame of reference to compare the results of the case study, which can in turn directly answer the research question.

Another differentiation is to be made between literature which directly mentions the DoR concept and literature which does not. The literature which does not mention the concept is included due to its assumed applicability to DoR, in some cases because INVEST is referred to which is often mentioned in DoR literature. This literature mainly covers the quality of requirements in the user story format which may potentially be included in a DoR.

Some examples are S13, S14, and S15. In these works the authors elaborate a framework with criteria for requirement quality, but not with the specific intent of inclusion in a DoR. Nevertheless, the framework is inspired by the Heck & Zaidman framework (2014) for which the authors do mention inclusion in a DoR. Additionally, the framework's criteria form an example of how user story quality can be incorporated in a DoR.

5.1 Explicit or Implicit DoR Application (SQ1)

In the literature three aspects deemed indicative for the explicitness of a DoR are covered; between whom the DoR is an agreement when described as such, how it is conceptualised and how it is enforced. Per aspect the different perspectives are categorised as codes in Table 5.1.

Aspect	Description	Code and IDs
Agreement between ...	Between whom the DoR is an agreement when it is described as such?	Team [S1, S5, S16, G2, G5, G14, G16]; Customer and Team [S3, G11]; PO and Team [S9, S17, S18, S19, G7, G8, G18]; Across Teams [S17, S26, G9, G15]; Specific Roles [G7]; No Agreement [G19].
Conceptualised as ...	Is the DoR conceptualised as a checklist of criteria or another form?	Grid [S3]; Checklist [S5, S6, S7, S17, G2, G5, G19, G26]; Collection of Criteria [S1, S4, S18, S26]; No Well-defined List [S15, G8].
Is enforced through ...	How are the DoR criteria enforced?	Mechanism [S3, S7, S9, S18, G1, G5, G7, G12]; Social Control [S15, G8, G14].

TABLE 5.1: Aspects of DoR explicitness found in literature.

5.1.1 Agreement

The first aspect concerns whom the DoR is an agreement between if it is described as such in the literature. Agreement and roles involved are assumed to be a factor in DoR explicitness since it determines who interacts with the concept and whether there is a mutual agreement about its contents.

In G19 it is noted that teams in the case study did not have a DoR, meaning there is no agreement on what constitutes a ready item. In the case of an agreement, it is most commonly between all members of the team. G5 remarks that this team-wide agreement helps in sprint planning sessions and the DoR effectivity should be reflected upon in a sprint retrospective session.

A related form of agreement is between PO and the team. In this, the PO has a certain responsibility towards defining the DoR as mentioned in G18, or to make items meet the DoR as in S18. There should be both agreement between a PO and their team and within the team on the contents of the DoR.

An equivalent to this is an agreement between the customer and the team, considering the PO represents the customer in external development. In this category, S3 proposes explicit discussion between the customer and software provider on what thresholds to use for DoR criteria.

In the case of larger projects, it is recommended to create an agreed DoR across teams. S17 reports that a business unit at Cisco has a cross-team DoR due to interdependencies in their development work. Having no DoR or an incompatible DoR with that of other teams causes issues in the product's overall development. Lastly, G7 proposes a process wherein the DoR is created by a PO and applied by a sub-team consisting of the PO, a quality member, and the tech lead.

5.1.2 Conceptualisation

A second aspect of explicitness is in what format the DoR is applied, for which a grid, checklists, collection of criteria, and no well-defined list are mentioned in the literature. These formats range from most explicit (a grid) to least explicit (no well-defined list).

A checklist is prescribed most commonly, for example, Dalton (2018, P. 164) suggests in G5: *“Develop a checklist to outline the criteria for the agile team’s definition of ready. . . . Post the definition of ready in the location visible to all team members.”*

In S3 a grid is proposed with predefined, weighted scores per criterion on which thresholds can be set. This is considered to be more explicit than a checklist, going beyond the binary options of a checklist by adding fully defined answer options.

For collections of criteria, it is noted that the DoR should have explicit criteria, but it is not always stated in what form they are to be handled. S26 adds that the collection of explicit DoR criteria should be kept in a shared (digital) place visible to all team members.

Support for the notion of an implicit DoR is found in studies reporting the usage of criteria in the form of no-well defined list. The Evaluation Research source S15 reports that some practitioners use self-defined quality criteria for requirements not explicitly documented, relying on the writer’s experience and peer review.

Unawareness is mentioned in S15 and G25 as a reason for not using a structured list of quality guidelines. More support for the implicit usage of DoR is given in the Evaluation Research paper S4 by Diebold et al., (2018, P. 6): *“Definition of Ready and Definition of Done did not replace existing practices. Some criteria already existed in the minds of team members.”*

5.1.3 Enforcement

The last aspect is how the DoR is enforced; through a specific predetermined mechanism or loosely by social control. A predetermined mechanism mentioned by S9 and S18 is that the PO is responsible for making sure that backlog items adhere to the DoR.

Contrary to this are sources that indicate teams use a form of social control to safeguard requirement quality and adherence to the DoR. In the case study in G8 the team has a DoR but enforcement is achieved through informal discussion and agreements between the PO and the team.

An approach between a mechanism and social control, but categorised as the latter, is described in G14. Each team member who works on an item is responsible for checking DoR adherence which is effectuated by social control in the team.

5.2 Rationale for Applying the DoR (SQ2)

Six types of DoR rationale are identified in the literature and presented in Table 5.2, along with the specific rationales they contain. In this case a rationale represents a reason for teams to apply the DoR, categorised as types according to where in the ASD process it manifests.

Type	Description	Rationale and IDs
Sprint Planning	The rationale manifests in the sprint planning activity.	Estimation [S3, S7, S17, G1, G2, G11]; Planning Efficiency [S23, G5, G15, G22].
Sprint Execution	The rationale manifests in the execution of the sprint.	Waste [S17, S23, G6, G13, G23]; Productivity [S15].
Communication	The rationale manifests in communication in the team or with the customer.	Communication Overhead [S4, S23, G3, G13]; Understanding [S5, S6, S8, S20, S22, G4, G12, G18].
Documentation	The rationale manifests in requirements documentation.	Documentation Quality [S4, S14]; Transparency [S5, S6, G2].
Final Product	The rationale manifests in aspects of the final deliverable.	Software Quality [S5, S6, S15, G2]; Customer Satisfaction [S10, G22, G24]; ISO Certification [S16]; Sprint Success [G4, G5, G20, G22].
Agility Rationale	The rationale is that the DoR comes with the adoption of Agile methodologies.	Agility Rationale [S24, G25].

TABLE 5.2: Types of rationale found in literature.

5.2.1 Sprint Planning

The first rationale type expresses itself in sprint planning and comprises improved item estimation and overall planning efficiency. Estimation is concerned with sizing items and adding estimates in terms of effort to items (Rubin, 2012; Sutherland & Schwaber, 2020).

Although estimation can take place during either or both backlog refinement and sprint planning, the benefit expresses itself in the ease of planning sprints. In S3 and S7 the INVEST DoR criteria are prescribed respectively to make more accurate estimates and help in sizing items.

Another planning rationale is efficiency, which the DoR aids in by helping to identify faulty items early as stated in G5. To this S23 adds that combining a DoR with backlog refinement can relieve sprint planning from lengthy discussions and un-ready user stories.

5.2.2 Sprint Execution

Closely related to sprint planning is sprint execution, a first rationale of this type is that a DoR reduces waste. Waste is work done that does not add value to the customer or stakeholder (Bufon & Leal, 2019). To prevent waste unclear and ambiguous user stories should be avoided through a DoR according to S23. S17 states the DoR helps the team to understand items so that the sprint scope is clear when they are combined after planning.

Second is the rationale that having a DoR improves sprint productivity in general. For this S15 finds that practitioners using quality guidelines for user stories, such as those potentially in a DoR, agreed more often that their usage increased productivity.

5.2.3 Communication

In executing a sprint communication within the team is vital, from which the third rationale type is derived. In the case study S4 it is found that the DoR reduced communication overhead meaning less discussion is required regarding items during the sprint. In this case, it was achieved through improved documentation, which is also covered as a rationale type itself later on.

Another rationale to improve communication is to have the DoR foster understanding of items by improving their overall quality, which might reduce communication overhead in turn. This is also related to the prevention of waste mentioned in the previous type. For this rationale Heck & Zaidman (2015, P. 1) state in S5: *“According to the practitioners, good quality agile requirements help the understanding within the team.”*

5.2.4 Documentation

As mentioned documentation is a rationale type for using a DoR, it's found rationales are twofold. Firstly a DoR is expected to improve item documentation quality, either on an overall level as experienced in S4 or on a user story level as found in S14. It should be noted that in S14 the number of quality defects in user stories was reduced by applying criteria, but participants did not perceive the improvement as such.

Another documentation rationale is improved transparency between the team and the surrounding organisation. S5 notes that practitioners think quality criteria for items aid in their traceability and accountability to the organisation.

5.2.5 Final Product

The development of items in a sprint should result in a final product, wherein the second to last rationale type manifests. One aspect of the sprint's final product is quality, which is expected to increase by having a DoR.

In S5 and S6 it is assumed that correctly specified Agile requirements increase the final product's quality, which is considered to be true for requirements in traditional development. Practitioners using requirements templates and guidelines were found to agree that these practices improve the final product's quality in S15.

Another aspect is the satisfaction of the recipient with the final product, or whether it is valuable to them, which is assumed to improve by DoR usage. An example is found in S10, where it is assumed that a DoR contributes to the Agile feature 'customer satisfaction', which corresponds to the Agile value working software over comprehensive documentation.

In the Sprint success rationale, the perspective is inverted, it concerns whether the team meets the sprint goal. To this Rubin (2012, P. 110) states in G20: "*A strong definition of ready will substantially improve the Scrum team's chance of successfully meeting its sprint goal.*". Presumably, this is effectuated through the planning, sprint execution, and communication rationale types. However, an underlying rationale is not explicitly linked in this literature.

A last rationale is uncovered in S16 which proposes using a DoR to help Scrum development adhere to ISO 9001 certification requirements. This certification relates to a quality management system for organisations developing products or providing services, which is audited by a third party.

5.2.6 Agility

Concluding is the Agility rationale uncovered in S24. In this Evaluation Research participants indicated that they use some artifacts because they come included with ASD methods such as Scrum. G25 confirms this finding in a case study between two projects, where it was likewise found that teams adopt certain artifacts such as a DoR because their usage is advised when applying ASD methods.

5.3 Criteria Applied in the DoR (SQ3)

In the literature 25 different DoR criteria are found which were either suggested or used in practice. The criteria are presented in Table 5.3, where they are grouped into five categories that structure their discussion. This categorisation is based on whether it is part of INVEST, what aspect of ASD it affects most, or how it assists in writing user stories.

The first category concerns criteria that are part of INVEST or correspond directly to the criteria's intent. For the second type the categories are derived from the two phases in ASD the DoR mainly pertains to (planning and development) and team cooperation which is integral to the methodology. The last category concerns the formulation of requirements in the user story format as commonly leveraged in ASD.

Concluding, all included criteria are mentioned in multiple sources, criteria only mentioned by a single source are excluded except the Context-Specific criteria such as Sustainability (S1) or Data Protection (S11) . Furthermore, three different criteria are broken down into variants; as an example, Responsible Person is broken down by the exact responsibility mentioned.

5.3.1 INVEST

The INVEST criteria were conceptualised to aid in securing user story quality in the Extreme Programming ASD method (Wake, 2003). In the widespread diffusion of Scrum, this mnemonic has been repeated often as an example for practitioners to shape their DoR.

S15 finds that 23,5% of practitioners surveyed use INVEST, compared to 33% using self-defined user story quality guidelines and 39.5% not using any guidelines. However, other aspects of the ASD process such as planning and prioritisation are not covered by INVEST, causing G2, S5, and S7 to suggest using additional criteria in a DoR.

Although INVEST is closely related to user stories there is no overlap of criteria between the two categories in this collection. The criteria in the latter are kept separate from INVEST since they can be used independently of the mnemonic, and pertain to different user story aspects. Likewise, INVEST criteria may be at odds with those in other categories such as Independent vs. Explicit Dependencies, indicating INVEST is not exhaustive.

Independent

The first INVEST criterion, independent, prescribes that backlog items on the user story level should not depend on each other so that they can be planned and implemented freely (Wake, 2003). This freedom is important for enabling the Agile Manifesto principles of customer satisfaction through early and continuous delivery of working software on a short timescale (Beck et al., 2001).

DoR Criterium	Description	Code and ID
<i>INVEST:</i>		
Independent	There should be no conceptual overlap between user stories, allowing them to be planned and implemented freely (Wake, 2003).	[S3, S5, S6, S7, S13, S15, G1, G2, G5, G16, G17, G19]
Negotiable	A user story should not be an explicit contract for a feature, it is open for negotiation between stakeholders (Wake, 2003).	[S3, S5, S6, S7, S15, G1, G2, G5, G13, G16, G19]
Valuable	A user story should be valuable to the customer or receiving stakeholder (Wake, 2003).	[S3, S5, S6, S7, S15, G1, G5, G16, G17, G19]
Estimable	A user story should be estimable, allowing it to be prioritised and planned (Wake, 2003).	[S3, S5, S6, S7, S13, S15, G1, G2, G5, G16, G19]
Small	A user story should be small so that the scope can be determined (Wake, 2003).	[S3, S5, S6, S7, S12, S13, S15, G1, G2, G3, G5, G13, G14, G16, G19, G20]
Testable	A user story should be able to be tested, which can be tried by writing a test for the story (Wake, 2003).	[S3, S5, S6, S7, S15, S17, S21, G1, G2, G5, G10, G14, G16, G19, G20, G23, G26]
<i>Sprint Planning:</i>		
Demo Known	The team knows what it means to demo the user story (Power, 2014a).	[S17, S21, G11, G16, G20, G23, G26]
DoD	The backlog item has a DoD.	[S4, G3, G11]
Estimated	The effort of the item has been estimated.	[S12, S17, S21, G2, G4, G10, G11, G13, G16, G17, G20, G23, G26]
Explicit Dependencies	Dependencies between backlog items are identified and linked.	[S5, S6, S13, S17, S21, G16, G20, G23, G26]
External Dependencies	External dependencies are identified and resolved.	[G2, G20, G23]
Prioritised	The backlog item has been prioritised in relation to others in the backlog.	[S5, S6, S7, G10, G11, G13]
<i>Team Cooperation:</i>		
Responsible Person	The person responsible for certain aspects of the item is determined.	Accepting Item [S17, S21, G26]; Developing Item [G16, G26]
Reviewed	The team or stakeholders have reviewed the item.	Team [S17, S21]; Stakeholders [S9]
Understandable	The backlog item is understood by everyone on the team.	[S7, G2, G3, G11, G16, G20]
<i>Development:</i>		
Additional Documentation	Where applicable and appropriate additional documentation is included.	[S5, S6, G10, G16]
Context Specific	DoR criteria which are dependent on the context of the software under development.	Sustainability [S1]; Data Protection [S11]; Assurance Impact [S25]; System Distribution [G21]
No Contradiction	Neither items nor comments within items should contradict each other.	[S5, S6, S13]
No Duplicates	Each backlog item should be unique.	[S5, S6, S13]
Non-functional Requirements	The backlog item is accompanied by non-functional requirements.	[S2, S7, S17, S21, G11, G20]
Traceable	The backlog item is traceable throughout sprint execution.	Unique Identifier [S7, G11]; Tool Usage [S5, S6, G2]; Forward Traceable [S7]
User Experience Artifacts	The user experience artifacts required are done and reviewed by the team.	[S17, S21]
<i>User Stories:</i>		
Full Sentences	User stories are described in linguistically correct and unambiguous sentences.	[S5, S6, S13]
Problem Oriented	The user stories focus on the problem and not the solution.	[S5, S6, S13]
Template	The user stories follow the <role>, <means>, <end>user story template (Cohn, 2004).	[S4, S5, S6, S13, S17, S21, G1, G2, G11, G13]

TABLE 5.3: Criteria found in literature.

Negotiable

Negotiable is a criterion concerned with allowing for flexibility of feature development through fostering discussion between the team and stakeholders. G13 states that flexibility in delivery can be achieved by coming to a sufficiently general description of an item with stakeholders so that incremental delivery is possible.

Valuable

Delivering value to the customer in sprints is a key aspect of ASD and is thus considered an INVEST criterion (Rubin, 2012). The importance of valuable items is also pressed by sources not mentioning INVEST. An example is G17 in which Nogués and Valladares note as a criterion (2017, P. 41): “*Is this item adding value by itself? Or it must be done with other items on the list?*”. In this the valuable criterion is combined with independent.

Estimable

Another criterion is to form estimable items so that they can be prioritised in the product backlog and planned. This should not be confused with the latter criterion estimated, signifying that the item’s estimable state has been acted upon. In the Quality User Story framework (QUS) from S13, atomic is a criterion that makes items more estimable by ensuring they only concern a single feature.

Small

A related INVEST criterion is small, which ensures that user stories are of the right granularity. Among others, S3 and S5 indicate sufficiently small user stories offer certainty that they can be completed within the sprint. To this S13 adds that a set of smaller stories is easier to estimate. Practitioners are found to perceive the same accuracy and ease of estimating smaller stories in S12.

Testable

The last INVEST criterion is that user stories should be testable, which can be tried by creating a test for the story. Although the original INVEST conceptualisation mentions tests, a variety of sources recommend creating acceptance criteria for verifying testability. S7 states that writing acceptance tests upfront goes against ASD’s intent because requirements might not be implemented or changed, causing the test cases to be rendered obsolete. Acceptance criteria stating when a user story should be accepted are proposed as an alternative.

5.3.2 Sprint Planning

DoR Criteria in this category concern activities and backlog item information which directly influence the planning of activities for the sprint. The first criterion is that it should be known what it means to demo the item, determining beforehand what work needs in the sprint to enable this proof of value (Power, 2014a).

Similar is the criterion of requiring the Definition of Done to be defined before developing the item in the sprint. In S4 an ASD method is implemented in which the DoD is required by the DoR, which should hold later in the planning and refinement process than the other DoR criteria. Only for prepared and verified requirements a DoD is created which includes work to be done for developers and testers, with a scope beyond the aforementioned demo criterion.

A criterion important for sprint planning is whether the backlog item has been estimated in terms of effort. Some sources such as G11 and S17 refer to this practice as sizing, where the effort is represented as a relative (scope) size to others in terms of story points. Expected Implementation Duration is proposed in S12 as a less abstract alternative to story points, quantifying user story scope as the expected number of days required to implement an item.

In some cases, a backlog item depends on other backlog items or external actors or teams for their completion. For the first case, it is mentioned that the dependencies between items should be made explicit through linking by tags or IDs, to which S13 adds that this especially holds when they are not obvious.

External dependencies should not only be identified but also resolved according to some sources. However, G23 claims that resolving should not always be the end goal, because external dependencies can be necessary for facilitating cross-team development.

The last criterion is again related to a key part of planning, namely that backlog items should be prioritised before they are planned. S7 mentions priority is important in ASD planning since it represents the item's stakeholder value at a given point in time, and the highest priority items are planned for development.

5.3.3 Team Cooperation

Team cooperation criteria relate to communication about a backlog item and the division of roles in the team for work related to it. The latter is captured by the criterion Responsible Person, for which sources have varying definitions. Sometimes the person responsible for accepting the item is to be specified, signifying who will test the acceptance criteria.

To this G26 adds that for their project the developer who will implement the item should be determined. In G16 this is described as ownership of the item, indicating the person responsible for getting it done.

Secondly is the criterion that either the team or stakeholders should have reviewed the item so that they are aware of the item and what it encompasses. S9 is the only source requiring stakeholder review in the DoR due to its focus on external software development for traditional industrial companies, which in the past caused communication issues.

In the third criterion team review is expanded, stating the item should also be understood by the team so the demand of the customer is clear and correct. G20 recommends that the backlog item is understood in particular by developers, to help in deciding whether they can develop it.

5.3.4 Development

Another category of criteria expresses in the development of the item during the sprint, they are for example used to help inform the development team about implementation aspects. Additional documentation can be required and should be of the appropriate type. Examples of this documentation given in G10 are mock-ups or wireframes.

Depending on the context, specific DoR criteria are mentioned in the literature, related to aspects such as assurance, data protection, and sustainability. Regarding sustainability aspects, S1 finds practitioners agree that knowing the sustainability impact of backlog items should be part of the DoR.

The sustainability dimensions used in the research are environmental, as well as economic, technical, social, and individual. Inclusion in the DoR ensures that the impact on these dimensions is both assessed before the sprint and that there is a commitment to this assessment.

Two related criteria with impact on development are that there should be no contradicting or duplicate items. No contradiction means that information like additional documentation within an item should not contradict each other, as well as no contradiction between items is allowed to exist.

For this criterion S13 notes contradiction conflicts can cause implementation errors and rework. The same sources mention not allowing duplicate items. S5 states duplicates can confuse when discussed separately, resulting in deviations.

In addition to documentation backlog items can be expanded with non-functional requirements, sometimes called quality requirements. S17 reports teams in Cisco need to identify architecture criteria related to among others performance and security before a backlog item is ready. Additionally, two sources are found in the MLR that recommend discussing non-functional requirements as early as possible in development (S2 & S7).

To track backlog items throughout development another recommendation is to include required traceability aspects in the DoR. Some sources accomplish this by attributing unique identifiers to user stories, others advise using a tool for tracking items from planning until delivery. S7 goes beyond this by suggesting requirements' relation to source code and test cases should be clear so that changes become easier to implement.

Concluding is another criterion for expanding backlog items. The Personal Experience report in S17 again reveals that the organisation requires backlog items to be extended with a non-functional part in the form of user experience artifacts.

5.3.5 User Stories

The last category contains three criteria directly related to the user story representation of backlog items. First is the criterion that user stories should be formed as linguistically correct and unambiguous full sentences. For this S5 and S6 specify correct language should be used, to which S13 adds that terms with multiple interpretations should be avoided.

Secondly, user stories should be problem-oriented instead of proposing a solution according to S13. S5 and S6 are more lenient and state that while the focus should be on the problem at hand, solutions can be included in some form. Since this framework applies to both feature requests and user stories, assumably the solution should be included as a comment in the case of user stories, and not in the story itself.

Lastly, an often-found prescription is to use the user story template as defined by Cohn (2004): $\langle role \rangle, \langle means \rangle, \langle end \rangle$. Adhering to this template aids in ensuring stories are uniform as required by S5 and S6, and minimal and well-formed as stated in S13.

5.4 Adherence to the DoR (SQ4)

Discussion of the degree of adherence to DoR criteria in literature is scarce compared to coverage of the other sub-questions. Three different perspectives are uncovered which are presented in Table 5.4; Full, Partial and Threshold.

Adherence to DoR criteria	Description	IDs
Full	All criteria should hold for a backlog item to be ready.	[S4, G2]
Partial	Not all criteria should necessarily hold for a backlog item to be ready.	[S5, S6, S17, G1, G5]
Threshold	Criteria have values by which readiness is determined based on thresholds.	[S3]

TABLE 5.4: Types of criteria adherence found in literature.

5.4.1 Full Adherence

The first perspective pertains to literature which describes that all criteria should hold before a backlog item is considered ready. S4 reports an Evaluation Research case study in which all criteria need to hold for the DoR. However, this definition consisted of only three criteria, of which the last concerns that items have a DoD. The first two criteria state that an item must have a stated role, and should follow the user story template.

In the grey literature, G2 recommends that items are either done or not done, and that in-between states should not exist. This signifies that either all selected criteria should be adhered to, or the item needs to be worked on until this is the case. However, the authors do note that some criteria might not be applicable to certain items allowing them to be skipped.

5.4.2 Partial Adherence

The second, most common perspective is that not all criteria in the DoR should hold for an item to be considered ready. An example is an item for which a criterion is not able to hold, but the rest of the criteria are achieved meaning the item is as ready as it can be.

In S5 it is mentioned that violations of criteria should be judged on importance by the author of the requirement. This viewpoint is likewise stated in G5 by Dalton (2018, p. 164): *“Avoid rules that require full compliance to DOR at all times, allowing for exceptions based on specific attributes of the user story.”*

S17 is a Personal Experience Paper that takes a different angle by stating that requirements do not need to be fully defined to be ready, but sufficiently so that there is confidence and understanding in the team.

G1 repeats this viewpoint, and recommends Pareto efficiency is reached in defining requirements, meaning that additional effort in defining the requirement would not result in more understanding. They add to this that the requirement should be defined sufficiently to allow an estimation of development effort.

A suggestion made by S6 is that criteria can have different times in an ASD process during which they should hold. Some criteria should hold when the requirement is created, while others may hold later, just in time for a step which can be specified in the DoR.

5.4.3 Threshold Adherence

Lastly is the perspective that criteria should have weighted scores, for which separate thresholds are set to determine when the requirement is ready. This perspective is differentiated from partial adherence since it features an explicit mechanism to quantify what degree of adherence is desired from which requirement.

S3 proposes this perspective as a scoring grid based on the INVEST criteria. In this grid, each criterion has a score ranging from poor (0) to excellent (3) accompanied by a description to determine the applicable score. The criteria thresholds are to be discussed with the customer in case of contracted software development, not achieved thresholds after evaluation form a direction for further requirement refinement.

This approach is somewhat similar to that used in the experiment in S5, where the quality criteria were converted to a checklist with score calculation per criterion aggregated into a total score. However, this approach is only used for experimentation and the authors suggest that in practice having a checklist to assist in reviewing requirements takes priority over explicit scoring mechanisms.

5.5 Summary

The MLR results discussed in this chapter cover the literature perspective for the first four sub-questions. First three aspects of DoR explicitness were uncovered, each containing several codes identified in literature. DoR agreement within the team is often mentioned in literature.

Additionally, some literature mention agreement across teams or with the customer on the DoR. The DoR is conceptualised in many cases as a checklist. Collection of criteria with an unspecified form, a criteria grid and no well-defined list were likewise covered in literature. Enforcement of the DoR is done either through a predefined mechanism or loosely by social control within the team.

Secondly six main rationale types for adopting a DoR were identified in literature, with each type containing specific rationales. Often mentioned were improved estimation as part of Sprint Planning, reduced waste in Sprint Execution, and more requirement understanding in Communication. In the Final Product type the greatest variety of rationales was found; software quality, customer satisfaction, ISO certification and sprint success were all covered in literature.

An important aspect of the DoR is the criteria it contains, which was covered in the results for the third sub-question. In total 25 criteria were discovered which are all mentioned multiple times in literature, with the exception of Context Specific criteria. The criteria were categorised based on whether it is part of INVEST, what aspect of ASD it affects the most, or how it assists in applying user stories.

Lastly three different forms of adherence to DoR criteria were found. The most common form in literature is partial adherence, wherein not every criterion needs to hold for a backlog item to be ready.

Alternatives are to require that all DoR criteria hold before an item is ready, except those not applicable the item, or to have a system with scores. In the latter each criterion has a score and thresholds can be set on the desired level for the team or project.

Chapter 6

Case Study Results

The results of the exploratory case study are reported in this chapter. First, the case characteristics are covered which provide the cases' context concerning the participant's role, the team, and the broader organisation they belong to. Secondly, the in-between results of the thematic coding analysis steps are reported to provide transparency.

Lastly, the final findings of the analysis are presented and discussed, forming the conclusion of the exploratory case study. This provides the first part of the answer to sub-questions one through four, which are substantiated further in the comparison (Chapter 7).

The full interview transcripts, case characteristics, codes, and themes referred to in this chapter are included in Appendix F bundled in an NVIVO file. In the software's coding tab themes and sub-themes are capitalised, while codes and sub-codes are in lowercase.

6.1 Case Characteristics

In Table 6.1 the characteristics of the eight cases are presented. In the continuation, each case is referred to by an identifier ranging from C1 to C8 for anonymisation. Following is the role of each participant representing the case, most of which are commonly found in ASD.

ID	Role	Method	Product	Product Focus	Customer	Cooperation	Team Age
C1	Product Owner, Developer	Scrum	Web Development (CMS)	Fixed Product (Customisation)	External	Independent Team	8 years
C2	Product Owner	Scrum, SAFe	Cloud Infrastructure	Product Portfolio	Internal	Agile Release Train	4 years
C3	Product Owner	Kanban, Scrum	E-commerce Partner Service Portal	Product Portfolio	External	Cross-Team Projects (Micro-services)	2.5 years
C4	Consultant, Developer	Scrum, SAFe	Digital Transformation of Business Processes (Low-Code)	Custom Software (Assignment)	External	Independent Team	3 years
C5	Tribe Lead	Scrum	Administrative Healthcare Software	Product Portfolio	External	Cross-Team Development	2.5 years
C6	Product Owner, Manager	Scrum	Customer Experience Portals	Custom Software (Assignment)	External	Independent Team	2 months
C7	Tech Lead	Scrum	Fund Management Application	Fixed Product	Internal	Independent Team	3.5 years
C8	Developer, Scrum Master	Scrum	Hotel Booking Tool	Custom Software (Assignment)	External	Independent Team (Micro-services)	2.5 years

TABLE 6.1: The case characteristics.

6.1.1 Participant Role

Two roles which may require further clarification are Tribe Lead (C5) and Tech Lead (C7). A tribe is an Agile organisational unit consisting of multiple teams, sometimes called squads, coordinated by a Tribe Lead (Kerr et al., 2018). The Tribe Lead set the tribe's goal and coordinates this with the involved teams' Product Owner. Kua (2014) defines a Tech Lead as the leader responsible for a team of developers, who themselves write code with the team.

6.1.2 ASD Method

For each case, the ASD methods applied in the team are recorded, with Scrum being the most common primary method. One case (C3) leverages Kanban as the primary ASD method but supplements this with elements from Scrum such as the sprint.

Kanban is derived from Lean manufacturing and is based on the visualisation of the workflow on a Kanban board, to reduce work in progress, focusing effort only on what's requested by the customer (Ahmad et al., 2013). This in turn should provide a constant delivery of value to the customer.

In two cases (C2 & C4) Scrum is applied in the Scaled Agile Framework (SAFe) which aims to aid organisations in managing multiple teams, programs, and portfolio development in an Agile way not accommodated by Scrum (Brenner & Wunder, 2015).

As an implication the teams can be part of an Agile Release Train (ART) such as in C2, meaning there is coordination beyond the team on what is to be developed. In C4 the customer provides the Product Owner for the team and applies the SAFe framework in their organisation.

6.1.3 Product Delivery

Next, the product developed by the case's team is described in general, which is recorded since besides providing context it could influence the DoR. A variety of domains is represented in the eight products, ranging from general systems such as a CMS to specific software like a hotel booking tool.

Additionally, the product focus is described, differentiating between three main types; custom software, a fixed product, or a portfolio of products. The product focus could again influence the DoR's usage and contents warranting its inclusion for analysis.

In the case of custom software, the team develops the product on an assignment basis for a customer outside of their organisation, or in other words an external customer. Sometimes the team is specifically formed for the assignment, as goes for C4. Typical for this type of development is greater involvement of the customer and focus on delivering value.

When the product is fixed this means the team works (mainly) on one product for either an internal customer in their organisation, such as another department, or an external customer. This can still involve customisation of the product for the customer, as in C1, but the long-term focus is on the delivering same core product for all customers.

Lastly is the portfolio of products meaning the team works on alternating products from a set collection. Products in the portfolio relate to each other, an example is C3 which develops a set of micro-services as part of a landscape. Again both internal and external customers are catered to by cases with this focus.

6.1.4 Team Cooperation

Besides this, cooperation with other teams in the organisation is considered, since this may affect cross-team agreement on the DoR. Most teams in this study are independent, meaning there is little to no cooperation with other teams in development. As an implication, these teams may have a significant degree of control over their DoR.

Contrary to this are teams who cooperate with others in cross-team development. This cooperation can take different shapes, C3 and C8 cooperate on a technological level by occasionally building micro-services also used by other teams. In both organisation there is likewise coordination on a project level meaning teams work towards the same goal or for the same customer. As noted in the MLR (Section 5.1) this might require agreement on the DoR.

6.1.5 Team Age

Concluding, the team's age is documented which could be an explanatory factor in how the DoR is applied and developed. Although a team's age does not guarantee it has had the same composition the whole time it can partially reveal its maturity. The team ages vary from young (2 months) to old (8 years), with the majority of the six teams being between 2.5 years and 4 years old.

6.2 Thematic Coding Steps

As described in Section 3.2.5 five thematic coding steps are performed to analyse the case study data as suggested by Robson (2002). In the following, the execution and in-between results of the first four steps are reported briefly. The final step's results are discussed in Section 6.3.

6.2.1 Data Familiarisation

Data familiarisation is achieved through transcribing the interview recordings and re-reading them as noted previously. In addition, the transcripts are loaded into NVIVO where the case characteristics originating from the introductory questions are extracted first. The case characteristics are kept separately from codes to avoid confusion.

An exception to this is team age which has been added in retrospect. Since not all team ages were apparent because it was not asked directly, four cases were contacted after the interview (C2, C4, C6, C7), to which the response is recorded at the end of the corresponding transcripts.

6.2.2 Initial Code Generation

The next step is to code segments of the transcripts based on their meaning as a means of data categorisation, to gain deeper insights beyond reading. Where possible codes are reused across transcripts providing an additional understanding of similarities across cases. Some codes have a close dependency on another code, which in turn forms sub-codes.

As Robson (2002) states codes can be derived inductively from the data or can be based on theory. In this step, the focus is on inductive codes to extract the practitioners' DoR usage without inferring from theory. However, codes based on theory cannot be avoided when reality corresponds closely to theory. An example is the Communication overhead rationale found both in theory (Table 5.2) and applied as a code.

Concluding, 103 codes are formulated from the interview transcripts, of which 12 are sub-codes. For infrequently occurring codes, such as those that only occur once, it has been attempted to merge them with other codes if this does not compromise the code's significance.

6.2.3 Theme Identification

Following is to group the loose collection of codes into initial themes. In this step, the inclusion of theory becomes inevitable since the coded segments are directly influenced by the interview questions, which are in turn derived from MLR findings. As a result, six main themes are identified forming the first grouping of codes:

- Adherence
- Construction
- Criteria
- Explicitness
- Rationale
- Others

The themes Adherence (Section 5.4), Criteria (Section 5.3), Explicitness, (Section 5.1) and Rationale (Section 5.2) are derived from the MLR results. Construction is derived inductively and concerns codes that cover how the team's DoR came into being. The last theme is Others, which is what Robson (2002) titles as a residual category capturing all codes that do not fit the other themes. In this theme the team age is recorded as a code.

6.2.4 Thematic Network Construction

Although the grouping of codes into initial themes assists in analysing the codes, further categorisation is required to achieve full comprehension of the data. Within the first five initial themes codes are categorised into sub-themes based on their influence on- or relationship to the main theme. Resulting are a total of 19 sub-themes spread out across the five main themes.

Together this organisation of themes forms a thematic network presented in Figure 6.1. At the core is the overarching theme of the case study; the application of the DoR in practice. The sub-themes are gathered in the boxes indicated by a dotted line adjacent to the corresponding main theme.

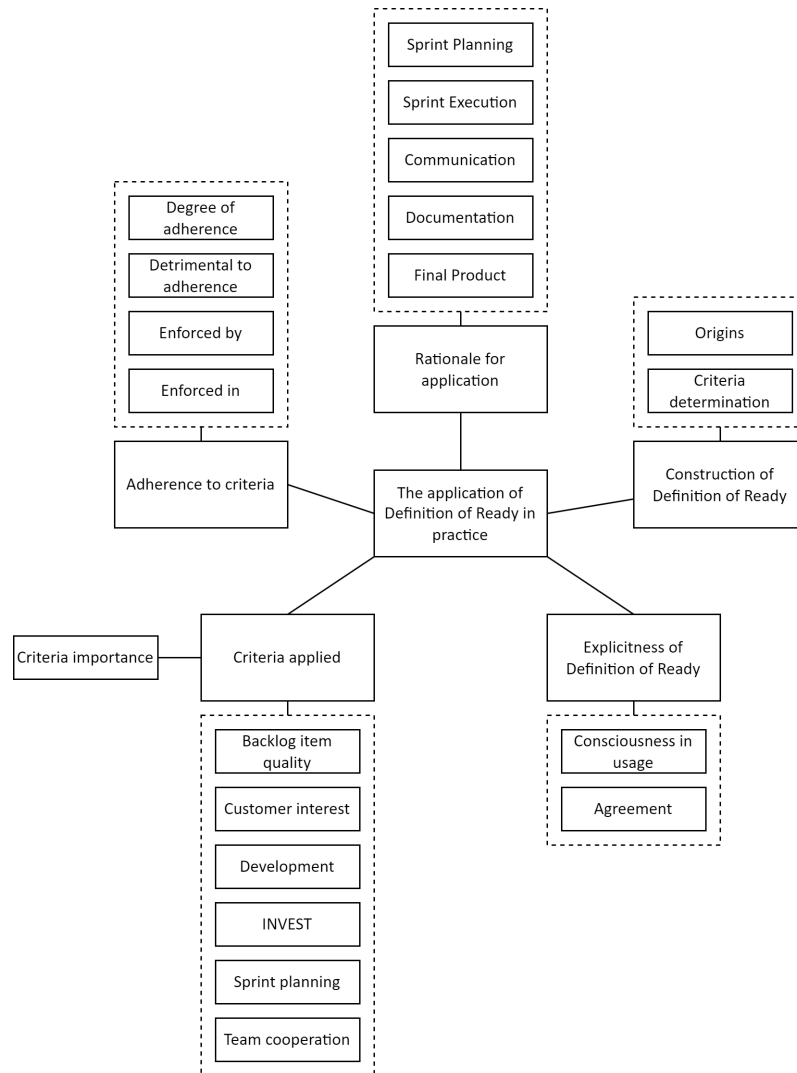


FIGURE 6.1: The full thematic network.

Two deviations in the figure are the omission of Other as a theme and the direct linkage between Criteria importance and Criteria applied. Other is omitted since it only contains the team age as a case characteristic and the daily development review during a sprint, which is only leveraged by one team (C8).

Criteria importance is a code with a broad scope containing all segments discussing what criteria are important to a team, this proves difficult to narrow down further into separate codes and directly concerns the mentioned criteria sub-themes. Thus this code is kept separate within the main criteria applied theme.

6.3 Results

As the final step of thematic coding analysis, the categorised data from the previous steps in the thematic network (Figure 6.1) is integrated and interpreted. The following discussion is structured by the main themes, starting with Rationale for application and proceeding in a clockwise direction.

This ordering is based on the expected occurrence in case a DoR is adopted for the first time. First a reason or possible consequences of adopting a DoR are considered after which the decision could be made to construct a DoR. Following some form of agreement and method of application are adopted which determines the explicitness. Afterward, the DoR criteria are applied in item refinement and/or sprint planning leading to adherence which is enforced in some way.

It is possible that this ordering occurs in a less-structured manner in practice, with the various decisions about the noted sub-themes performed in parallel. This ordering is discussed further in constructing the framework, where possible backed up by cases and literature (Section 7.2).

6.3.1 Rationale for Application (SQ2)

The sub-themes for rationales are derived from the MLR categorisation which is based on where in the ASD process the rationale manifests (Table 5.2). Consequently, the same ordering is taken up for ease of comparison in the next research phase. A total of 15 rationales is uncovered in the case study presented in Table 6.2.

Rationale	Description	IDs
<i>Sprint Planning</i>		
Constructing DoD	Knowing test scenarios helps construct the DoD.	[C4]
Uniform way of working	Making the process of item refinement predictable.	[C2, C4, C6, C7, C8]
Used as blockage	Can be used negatively to block work on items.	[C1]
Ease of checking item readiness	It is clearer when an item is ready.	[C1, C4, C5, C6]
No need in refinement	The need for a DoR was not experienced.	[C8]
Improved item estimation	Clearer items are easier to estimate.	[C4, C7]
Remembering work for readying item	Not forgetting aspects such as performing a check.	[C1, C7]
<i>Sprint Execution</i>		
Reduced waste	Less work is done which does not add value.	[C3, C7]
Reduced scope creep	The item's scope changes less during development.	[C2, C4, C7]
<i>Communication</i>		
Fostering discussion about item	Eliciting discussion about what is ready within and outside the team.	[C2, C3, C4]
Reduced communication overhead	Less discussion is required about items during the sprint.	[C1, C3, C4, C5, C6, C7, C8]
Unclear items	Unclear items enter the sprint.	[C1, C2, C3, C4, C5, C7, C8]
<i>Documentation</i>		
Accountability towards customer	It is clearer for the customer what to expect.	[C5, C6]
Improved documentation	It is clearer afterwards what has been implemented.	[C2, C5]
<i>Final Product</i>		
Improved quality	The delivered item's quality is higher.	[C5]
Improved value	Building the item as intended improves the value.	[C2, C5]

TABLE 6.2: The type of rationales found in the case study.

Sprint Planning

Mentioned most often by cases is the rationale that having a DoR makes the process of refining items more predictable by offering a uniform way of working in the team. One case (C2) notes that the team size of 13 members reinforced the need for a uniform way of working because it would require too much effort from the PO to check everyone's work.

Related is the rationale that a DoR makes it easier to check item readiness because the set of criteria makes expectations for a ready item clear to the team. C6 adds that knowing these expectations may increase the feeling of responsibility of team members for readying items.

To this, it can be added that having a clear list of requirements for items can help to remember the work required to get an item ready. C7 gives the example that forgetting to resolve an external dependency can cause the item to be delayed while waiting for the dependency to be resolved, which can be significant.

Another rationale is that having a DoR can make items clearer which then aids in their estimation. C7 states that they check whether items are clear before estimation, because unclear items may not be estimated correctly in terms of points. It is suggested by C4 that although the DoR helps in writing clearer items, refinement is needed for further clarification to improve estimation and planning.

Two counterarguments against the DoR in this sub-theme are that it can form too much of a blockage for including items in a sprint and that the need for a DoR was not experienced during refinement and planning.

For the first C1 states that according to them the DoR may be used to bar items from the sprint because they are unclear, while they sometimes only become clearer during development. The latter is given by C8, a case with an implicit DoR. They did not consciously experience any issues with unready items in refinement or planning which could be solved by a DoR from their perspective.

Lastly, C4 suggests that formulating a DoR can help teams in constructing their DoD. A DoR can require an item's acceptance criteria or test scenarios to be known before development, on which the DoD can be based and expanded.

Sprint Execution

In sprint execution two rationales are identified; reduced waste and reduced scope creep. As noted in the MLR results (Section 5.2), waste is related to delays in item development occurring during the sprint.

Regarding waste reduction due to the DoR C3 says that unclear stories can cause the effort to be wasted, work to be reversed or mistakes to be made. To this C7 adds that confusion can cause what is developed to be 'thrown away' requiring a fresh start, which delays development severely.

Scope creep is mentioned explicitly as such by C2, stating that adhering to a DoR reduces the risk of stories increasing in size during a sprint. Aside from a size increase C4 mentions the size can decrease as well because unclear items can cause a gray area to exist between items, in turn leading to ambiguity about scopes.

Communication

Unclear items are mentioned several times already and are stated by the majority of cases as a reason to adopt a DoR, or the reduction thereof as a consequence of its application. Most cases mention delays as a final consequence of ambiguous items.

According to C5 unclear items are handled in two ways, they are either planned into the sprint with ambiguities remaining or not planned until it is clarified further. Both cases cause delays meaning that unclear items should be prevented from reaching the planning activity by adhering to a DoR during their preparation.

Related to the delays caused by ambiguous items is the rationale that a DoR may reduce communication overhead as well. Although this may seem contradictory to the fostering of discussion about an item's rationale, their co-occurrence is explained by several cases. Likewise, it ties in with the previous rationale of reducing unclear items.

C3 states including the DoR in creating items and refinement discussions might take more time beforehand, but will consequently reduce the discussion about an item during the sprint, because it is now clear in itself and to the team.

This is confirmed by C4 which takes the customer's involvement in refinement as an example. They describe that a DoR is used to agree on what ready means to both parties complemented by discussion in refinement sessions, which leads to fewer meetings during the sprint about item implementation aspects.

Documentation

Two documentation-related rationales are found in the case study; improved accountability towards the customer and improved overall documentation of the item. The latter rationale takes the team's internal perspective, where item documentation may be used as a reference for future product development.

The first rationale is related to managing the customer's expectations about an item and developing the item such that accountability can be attributed to the customer. C5 asserts that the DoR makes it easier to check with the customer whether the stories correctly represent their expectations before development.

To this C6 adds that when an item is agreed with the customer the accountability partially shifts towards them when it is built according to specification. This accountability likewise aids in operating in a way billable to the customer with approved item documentation forming evidence.

About the DoR improving documentation C5 notes that the more complete items due to the DoR can be used down the line to enquire about previous feature implementations. They state that the documentation can still prove useful years onward in finding about design choices, implementation choices, and value offered to the end-user.

Final Product

The last rationale sub-theme concerns the final product for which the rationales addressed directly by cases are scarce. C2 is the only case that discloses they have perceived improvements in the final item's implementation due to a DoR: *"I think scope creep can be very detrimental, wanting to do a lot of work at once and thus reducing the quality."*

An improvement in value is noted by C2 as well in the same breath with improved implementation quality, suggesting that in their view the value for the end-user is tied to the implementation quality. C5 links the improved value to the previously noted accountability towards the customer, stating that the clearer stories due to a DoR increases the chance the item is built correctly for the customer.

6.3.2 Construction of the DoR (SQ5)

For the construction of the DoR, a distinction is made between two sub-themes as presented in Table 6.3. The first concerns how the teams determine what criteria are part of their DoR at a certain point in time. Secondly, the origins of the DoR are discussed; from what sources do teams draw inspiration in construction?

Aspect	Description	IDs
<i>Criteria Determination</i>		
Same DoR for all projects	The DoR remains the same for all projects.	[C6]
Standardised backlog items	Reoccurring backlog items are used as a template.	[C6]
Type of work decisive	The type of work influences the relevant criteria.	[C4, C7, C8]
<i>Origins</i>		
Organisational DoR	Adapted from a DoR suggested by the organisation.	[C1, C4, C7]
Team member experience	Arisen from experience outside of the team.	[C2, C5, C6, C7]
Within team experience	Arisen or developed further from experiences inside of the team.	[C3, C4, C5, C6, C7]
Other sources	Inspired by other sources such as literature.	[C3, C5, C6]

TABLE 6.3: The construction aspects found in the case study.

Criteria Determination

The first aspect of criteria determination is that criteria may differ due to the type of work performed by the team. Three cases state that the work may differ based on whether it concerns mainly back-end development, front-end development, or both.

As an example, C7 mentions the difference between a front-end development team and a back-end team that works on an API. For the front-end team, a UI design is always required in the DoR, otherwise, it is unclear what to develop. While for an API a UI design is never required because the end-user does not directly interact with the feature.

C4 goes beyond this by stating that their DoR might differ per item depending on what type of development is required. This somewhat corresponds to the non-applicable item aspect of adherence in Table 6.7, wherein not always every criterion of the DoR is necessarily used.

The other aspects of criteria determination are only discussed in C6. Their team's DoR firstly remains the same across projects although they do develop assignment-based custom software for external customers. The reason is that in their way of work, the customer does not have an influence on the DoR criteria and item estimation.

Additionally, a new development in their team is the construction of standardised backlog items based on reoccurring work in assignments. This is due to the perception that the setup of a software platform for a customer always involves a few of the same facets. These items are supplemented with acceptance criteria for future usage as templates to standardise part of an assignment's setup.

Origins

Four types of origins are found in the case study with within-team experience being the most common. Drawing inspiration from them when adopting a DoR could influence both how the DoR is used and what criteria it contains.

By within-team experience, it is meant that through shared experience from working together in the team, a DoR is constructed. All noted cases also state that their DoR has developed further after construction based on experiences in the team. About this C3 says: “.. *I think it's also a bit of a living thing (the DoR), which you can supplement with elements you run into.*”.

Secondly, experiences from outside the team by team members are used in constructing a DoR. This mostly applies to people having worked in scrum teams before, either in the same or in a different organisation. C2 specifically mentions previous experiences from team members who have been Scrum Masters, which might prove valuable since this role is aimed at implementing ASD aspects.

The organisational DoR can in some cases offer additional inspiration on which to base a DoR. An organisational DoR is a standard within the organisation that suggests how to construct a DoR. All three cases confirm that although the DoR was based on this standard, there was no strict enforcement from within the organisation about the exact criteria.

Concluding, some teams take their inspiration for constructing a DoR from other sources such as literature. Both C3 and C5 indicate that the search was not performed in a very structured manner and that Google was used to find some examples from literature or websites. About this C6 says that the examples on the internet are very diverse and thus have not constituted the main inspiration for them.

6.3.3 Explicitness of the DoR (SQ1)

The explicitness of the DoR is divided into two sub-themes as provided in Table 6.4. Firstly, it is covered between whom the DoR is an agreement if the cases mention an agreement. Secondly, it is considered how conscious the teams are about their application of the DoR concept.

Aspect	Description	IDs
<i>Agreement</i>		
With external party	Agreement with external party such as a customer.	[C1, C4]
Within team	Agreement between the team members.	[C1, C2, C3, C4, C5, C6, C7, C8]
With other departments	Agreement with other related departments in the organisation.	[C5]
No customer interference	The customer does not influence the DoR.	[C6, C8]
Organisation checks	The organisation checks DoR as part of the way of working.	[C1, C4]
<i>Consciousness in Usage</i>		
DoR called as such	The DoR is known by this name in the team.	[C1, C2, C3, C4, C5, C7]
Documented as	As backlog software template or a checklist.	[C1, C2, C3, C4, C5, C7]
DoR not called as such	The DoR is not known by this name in the team.	[C6, C8]
Unconscious usage	The DoR is unconsciously applied in creating and refining items.	[C1, C3, C5]

TABLE 6.4: The aspects of explicitness found in the case study.

Agreement

Agreement between members of the team is mentioned in all interviewed cases, and can thus be considered a prerequisite for the application of the DoR in practice. About this C4 states that when forming a team in their organisation the DoR is agreed upon in the first sprint, or as they sometimes call it ‘sprint zero’.

Nevertheless, other forms of agreement are also mentioned in cases. Two cases (C1 & C4) state that an external party is (sometimes) directly involved in setting DoR criteria. For C1 this has occurred once during development with a third-party developer for the same customer, in this case the DoR was agreed upon with this other team.

In C4 the agreement with an external party is more common which can be explained through the organisation’s way of working. This organisation develops custom software on an assignment basis for external customers, who in turn provide the Product Owner for the team.

Contrary to this are teams who state that the external customer has no interference at all with the DoR. As noted before this is the case for C6 where the DoR remains the same across assignments for differing customers. C8 adds that the customer does influence the Product Owner in discussions, but that this does not concern the team’s DoR.

Another form of agreement is across the organisation, when there is an organisational check on whether the DoR adheres to the way of working. C1 states that forming a DoR is not completely without obligation because once a year the overarching organisation checks the adherence of their way of work to the organisational values.

Consciousness in Usage

The first aspect of consciousness is whether the teams call the DoR as such, or in other words whether the concept is known. The majority of teams are familiar with the DoR concept, with some cases stating the term without first enquiring about their familiarity in the interview (C1, C3, C7).

Contrary are two teams that do not call the DoR as such. C6 is unfamiliar with the term but recognises it is very comparable to what they call writing ‘actionable’ items. The contrary is C8 which is both unfamiliar with the concept and only uses facets of item readiness akin to an implicit DoR.

It must be noted that there might be selection bias under the assumption that teams with knowledge about the DoR are more likely to participate. Nevertheless, some cases unfamiliar with the concept were reached which is according to the intent of the generalised DoR explanation given in contacting them.

An additional facet of consciousness is the way the team interacts with their DoR. Firstly is the way the DoR is documented for which two types are identified in practice. Three teams (C2, C4, C5) are using, or have used, a DoR in the template for writing items in backlog management software such as Jira¹. This template aims to ensure that only complete items end up on the backlog before discussion in refinement.

Four other cases (C1, C3, C5, C7) document the DoR as a checklist that can be used as a reference in refining and discussing items. C5 previously applied a backlog software template but has switched to a documented checklist, because they noticed an abundance of checks when submitting an item caused them to be ignored.

Lastly is a residual aspect concerning unconscious usage of the DoR in teams who are nevertheless aware of the concept and its existence in the team. These teams have a DoR which is remembered by members and used without explicit realisation in writing stories and refinement. As C5 notes the list of DoR criteria can be used by the PO or Scrum Master to refer to when an item is unready, but is otherwise not explicitly checked.

6.3.4 Criteria Applied in the DoR (SQ3)

In Table 6.5 the DoR criteria found in the case study are presented. A total of 37 criteria is identified and divided over 6 sub-themes. The criteria range from context-specific and being merely used by a singular team to applicable to ASD in general and used by a majority of teams. The following discussion evidently pays additional attention to the latter, widely applicable criteria.

¹<https://www.atlassian.com/software/jira>

Criterion	Description	IDs
<i>Backlog Item Quality</i>		
Formulated in template	For example the <role>, <means>, <end>user story template.	[C1, C2, C4, C5, C6]
Has a title	The item has a title for clarity.	[C2]
Has complete description	The item is described completely.	[C7]
Has roles	It is clear what roles have what permissions.	[C4]
Has scope	The backlog item clearly describes its scope.	[C2]
Is problem oriented	The item focuses on the problem and not the solution.	[C3, C4]
Is simple	The item is simple for understandability.	[C3]
<i>Customer Interest</i>		
Agreed with customer	The customer has to check the item and agree with it.	[C6, C7]
Has business acceptance	It is known how and who in the organisation will check acceptance.	[C7]
Has value estimate	The item has an estimated value in currency it will deliver.	[C4]
Is value oriented	The value for the business needs to be clear in the story.	[C5]
Stakeholders involved	The stakeholders are involved in shaping and developing the item.	[C1, C7]
<i>Development</i>		
Checks performed	Checks such as for privacy and personal data risk are performed.	[C1]
Has breakpoints	Breakpoints and changes to the platform need to be clear.	[C8]
Has process description	It is clear where in the customer's process the item is to be applied.	[C4]
Has reproduction scenario	If the item is a bug it requires a reproduction scenario.	[C5]
Has scenarios	The item is accompanied by scenarios with (un)happy flows.	[C3, C4]
Has success measurement	It is known how the item's success in development is measured.	[C3]
Has UI design	The item is accompanied by the required UI design.	[C3, C4, C6, C7]
Is aligned with architecture	The expected implementation should fit the software architecture.	[C7]
Takes mobile into account	The item takes mobile design into account.	[C1, C8]
Takes privacy into account	The item has privacy by design and privacy by default.	[C1]
<i>INVEST</i>		
Has acceptance criteria	The item has acceptance criteria which are to be satisfied at delivery.	[C2, C3, C5, C6, C7]
Has test plan	It is clear how the item will be tested.	[C7]
Is as small as possible	The item is as small as possible to support incremental development.	[C5, C6]
<i>Sprint Planning</i>		
Definition of Done	The DoD needs to be known as a criterion in the DoR.	[C2, C8]
Dependencies to others	The dependency to other stories needs to be known and resolved.	[C1, C3, C7]
External dependencies	The external dependencies are identified and resolved.	[C1, C3, C6]
Has clear way to demo	It is clear how the item will be demonstrated to the stakeholders.	[C7]
Has deadlines	The deadlines of the item are known.	[C3]
Is estimated	The item is estimated before entering the sprint.	[C2, C4, C5, C6, C7, C8]
Is prioritised	The item has received a priority on the backlog.	[C7]
Overarching collection	Item is part of a collection such as an epic.	[C2, C4, C6]
Release calendar checked	The item takes into account the calendar with downtime.	[C7]
Tasks identified	The tasks required for implementing the item are known.	[C1, C5]
<i>Team Cooperation</i>		
Refined before planning	The item is refined within the team or with others before planning.	[C5, C8]
Understood by everyone	Everyone has to understand what needs to be built.	[C7]

TABLE 6.5: The criteria found in the case study.

Backlog Item Quality

The first category of criteria concerns safeguarding the quality of the backlog item's description. Noted most often by teams is that backlog items have to be formulated according to a template, for example in the user story form. C6 States that they are currently using user stories but do not use a standardised formulation yet, which they intend to implement sometime soon.

For this C3 indicates that they used to strictly adhere to the user story template, but do not follow this anymore. They give an example as the reason, wherein the item concerned a broken web page and the user story ended up as: “*As a user, I want to see a working page, so that I can do my job.*”.

In this case, the PO prefers flexibility and would suggest just including a screenshot of the error together with ‘fix this’ as text. As a rule of thumb, the team requires that the context should be clear enough from the item instead.

To their usage of a template for item formulation C2 adds explicitly that the description needs to convey the item’s scope. According to them the scope helps in achieving consensus in the team about what the item is and what it is not and improves the accuracy of estimation.

Another criterion mentioned twice is that the item needs to be problem-oriented. Both cases (C3 & C4) state that the item description needs to stay as far away as possible from the solution. C3 notes as an exception a relatively simple item that has one clear solution beforehand.

An addition to the role part of the user story template is made by C4. They go beyond simply stating a simple end-user role and require it to be known who in the business will use the delivered item at what place. This leads to a registration of which roles have certain permissions in the application.

Although C3 and C7 do not require item formulation through a template they do have quality criteria. For C3 the items should be simple, providing some context and being understandable to team members. They want to avoid items with a lot of text or an abundance of scenarios and acceptance criteria.

For C7 the description should be complete which is enforced through a subjective ‘sanity check’ of each item. There is no clear definition of what complete means, which may rather be an informal agreement in the team.

Lastly, C2 requires items to have a title. While not specifically noted by the case it can be assumed this is due to their usage of Jira backlog management software, which asks for a title when creating an item. This title is expected to aid in distinguishing backlog items.

Customer Interest

The next sub-theme captures five criteria that focus on the customer’s interest. They mainly concern agreement with the customer on the work to be delivered, or the maximisation of value for them.

Firstly, some teams require that the items are agreed upon by the customer before planning them into the sprint. In C6 the external customer needs to check and agree with items before a deadline, otherwise, the team assumes that items are correct. For C7 multiple branches of the organisation sometimes have to agree with the item, while other times it concerns a single internal customer.

Another criterion is that the stakeholders are identified and consequently involved in shaping and developing the item. As stated by C7 this goes beyond checking and agreeing, the end-users as stakeholders need to be involved beforehand in discussing what is to be built.

Closely related to these two criteria is that clarity about the business acceptance is required by C7 before the sprint. This means it needs to be known which internal customer needs to accept the feature before releasing it, and in which application environment this will be done.

The last two criteria involve the value which will be delivered to the customer by developing the item. C4 indicates that the value the item will deliver in currency is asked by the customer as part of the DoR. Although difficult to estimate an attempt is always made, which aids the PO in estimating the item's priority.

Secondly, C5 requires aside from the formulation according to a template that items clearly indicate the business value; what does the customer want to use this for, why are we building this and why do they want this story? This criterion could originate from their focus on product portfolio development for external customers, where there is not always a singular customer for whom the product is developed specifically.

Development

Some criteria in a DoR are aimed at aiding in development during the sprint, of which the following are examples. Mentioned most often by cases is that an item should be accompanied by a UI design so it is clear how the item's visual representation should be created in development.

In the most simple form, this can be the text that needs to be placed somewhere in the interface, as stated by C3. The other cases specify that the item's UI design needs to be clear to a greater extent, such as where a button goes and what it will look like, for which C6 suggests a wire frame.

Somewhat similar to knowing the UI design is that C8 requires the applied breakpoints and changes to the platform to be documented. A breakpoint signifies a point of transition where an adaptive UI design adapts to the device's screen size. (Sinha & Karim, 2015).

Likewise related is the criterion that the item should take mobile design into account before development. For C1 this means thinking 'mobile first', which originates from their focus on website development. This means the UI should in the first instance work well on mobile devices and likewise work on larger screens.

A criterion also mentioned multiple times is that the item should be accompanied by scenarios for the benefit of development (C3 & C4). These scenarios relate to the happy and unhappy flows in the process the item supports, or in other words what should occur when things go right and when things go wrong. As mentioned C3 notes that an abundance of scenarios should be avoided and that in case of a high number, a flowchart is preferred.

The remaining DoR criteria in this sub-theme are context-dependent and thus only occur in a single case each. Firstly, C1 requires that items take privacy by design and privacy by default into account. Privacy by design encompasses seven principles for embedding privacy into software, wherein privacy by default signifies that users do not have to act to protect their personal data (Cavoukian, 2010).

Additionally, C1 requires that in some cases assessments for privacy are filled in and checked with the Data Protection Officer (DPO) before the sprint. Not performing this check in time can mean the DPO is not available causing delays.

C7 applies a different type of check on items before the sprint, namely whether it fits the overall software architecture. This architecture is represented as an architecture board, for which the check concerns if the envisioned item fits the organisational way of working in terms of architecture, and whether everyone on the team agrees on this.

Two criteria concern the expansion of items with contextual information for the benefit of development. C4 wants items to be accompanied by a process description that expresses where in the customer's process the functionality shall be used, and what process steps it involves.

For items that concern bugs C5 asks that they are accompanied by a reproduction scenario so that developers can emulate it. This reproduction scenario is enforced through Jira where it is required when making an item of the 'bug' type. In this case, the support and account management department is responsible for delivering this reproduction scenario to the team.

Concluding, C3 notes as a distinct point in their DoR that it needs to be known how to measure an item's success. It is expected this concerns the quantification through metrics of how well an item has been implemented, as opposed to simply measuring its acceptance.

INVEST

Three criteria from the cases are directly related to the INVEST principles for writing backlog items. Firstly, teams require that the acceptance criteria to be satisfied at the item's delivery are known before the sprint. This relates to the testable part of INVEST, for which in the literature some sources recommend writing acceptance criteria as a means of verifying an item's testability (Table 5.3).

To this C2 adds that for them this partially sets the item's scope because acceptance criteria may define what the story includes on delivery. For them writing the acceptance criteria may also partially reveal what the technical implementation of the item encompasses.

An addition to the testability from INVEST is given in C7, which demands that the test plan for items is known. This differs from acceptance criteria since a test plan considers aspects of tests such as their scope, objective, environment, methods, techniques, and tools, which goes beyond when an item is to be accepted (Quadri & Farooq, 2010).

The last INVEST criterion found concerns that items are kept as small as possible, corresponding to the small part of INVEST. C5 achieves this by splitting items into as small as possible sub-tasks before planning. In C6 the team attempts to keep items as small as possible, and an item should be split when reaching 13 points during estimation.

Sprint Planning

For sprint planning a wide variety of DoR criteria is identified in the case study. Noted most commonly is that an item should be estimated in terms of effort before it is planned into the sprint. Four teams mention that they use story points to represent this effort estimation (C2, C6, C7, C7).

In two cases the teams specifically mention the term planning poker as the method to agree on the item estimates (C2 & C8). In a discussion session each team member individually attributes story points to an item and writes it on a card, after which the cards are revealed at the same time (Mahnič & Hovelja, 2012). If the results differ too much the item is discussed further after which a re-vote is cast, continuing until there is consensus.

Aside from knowing the estimated effort of an item before the sprint, some teams require dependencies of an item to be known. Firstly, teams want to know what other items on the backlog an item depends on for its delivery (C1, C3, C7). C7 notes that not taking these item dependencies into account may cause an item to be developed prematurely, leaving it unusable for some time.

Another form of dependency is those external to the team, either relying on another department or organisation, or on another software service. Both C3 and C6 state that is important to know item dependencies to other services so that it is known before the sprint what external software is part of the scope. C3 notes as well that if the work depends on another team in the organisation it should only be marked ready when they are.

Related is the requirement that an item is (sometimes) part of an overarching collection such as an epic. C6 indicates that not all stories are connected to an epic, but those that do aid the management in achieving insight into the sprint planning. In C4's team, it is required that items are part of a feature (the level below an epic), for the benefit of longer-term planning and estimation.

To aid in planning two teams require that the tasks are known (C1, C5). Although not mentioned by these cases it may be assumed that this division aims to make it clear what work is required per item to subsequently divide this among team members.

The work on the item during the sprint is done when the Definition of Done is met. Some teams require the item's DoD to be known as a criterion of the DoR. C2 states that before the sprint it is decided what parts of the DoD apply to an item; the required documentation, the way of testing, and if a demo is required. C7 also notes, as a separate criterion, that a story should have a clear way to demonstrate it to the stakeholders.

When the item has to be finalised is determined through deadlines or milestones, as noted by C3 in their DoR. Additionally, C7 requires that the release calendar is checked because some items require deployment during application downtime. To determine in which sprint the item is to be developed it is prioritised. Although this may seem arbitrary, C7 do note this in their DoR so that the prioritisation of items is not forgotten.

Team Cooperation

The last two criteria relate to team cooperation, of which the refinement before planning is most commonly mentioned explicitly as a criterion by cases (C5 & C8). For C5 this refinement can involve a User Experience department, a Technical or Functional department, and the team itself. In C8 the refinement only concerns the team itself.

Concluding is the criterion that an item should be understood by everyone. Although this may be implied by other teams through discussion in refinement, C7 is the only team that explicitly notes this in their DoR. In this team, every team member needs to understand the assignment formed by the item before it is considered ready.

6.3.5 Criteria Importance (SQ3)

Aside from the contents of their DoR cases are enquired what criteria they find the most important, as presented in Table 6.6. Notably, this importance is from the perspective of the participant's ASD role unless stated otherwise. C8 is not represented since their DoR is implicit, which means the participant did not have a conscious grasp on what DoR criteria are important in their ASD process.

Case	Importance
C1	External dependencies
C2	Has scope, is estimated
C3	Is simple, has deadlines, describes the context
C4	Is estimated, formulated in template, has process description, has scenarios (happy flow)
C5	Is value oriented, has reproduction scenario (for bugs)
C6	Acceptance criteria, external dependencies, has UI design
C7	Has clear way to demo, acceptance criteria, understood by everyone

TABLE 6.6: The importance of criteria found in the case study.

The first case finds that knowing and resolving external dependencies is the most important. An example is given wherein a set data interchange format version needs to be adhered to by the team and the external party so that this remains stable to allow for further development. If this does not remain stable it could cause issues during development due to the external party making changes.

For the second case, two criteria are most important. Firstly the scope of items should be clear from their description. Likewise, the estimation is important because this aids in committing to a set amount of work in the duration of the sprint.

In C3 keeping items simple and clear is of the utmost importance, allowing team members to quickly grasp items when checking them during the sprint. Deadlines and milestones are also indicated to be important, which can be assumed to support the PO in planning. As noted C3 does not require a template for formulating items but finds it important that the context is clear enough.

Within C4's team, a story is never picked up without being estimated, forming an integral part of the DoR. Additionally following the template and having a process description is important. For the scenarios, happy flows are found to be more important than unhappy flows.

Following is C5, which indicates that the value orientation of items is the most important: “*Why are we building this (for the customer) in the first place?*”. For bugs, it is also important that the reproduction scenario is present so that it can be analysed what exactly goes wrong. Without these criteria, the team indicates they can not work on an item.

C6 has a somewhat different perspective and states that formulating the item as a user story is perhaps less important than acceptance criteria. In their case, the acceptance criteria cover the desired behavior in the ‘so that’ part of the user story format. For the development team, the external dependencies and the UI design are important as well, with most other DoR criteria in essence serving the PO and project lead.

Lastly, C7 finds demonstrability, testability, acceptance criteria, and item understanding in the team to be important. The first two are important to them since without them business acceptance can not be achieved, and the delivered work is unable to be verified with organisational stakeholders. Acceptance criteria are mentioned because they clarify for their team what exactly should be developed, for which the item should likewise be clear to everyone.

6.3.6 Adherence to the DoR (SQ4)

The last theme concerns the adherence to the DoR, which is presented in Table 6.7. Four sub-themes are used to categorise the 20 aspects to adherence identified in the case study.

Aspect	Description	IDs
<i>Degree of Adherence</i>		
Changes to item in sprint	When a criterion changes in the sprint this is discussed.	[C6, C8]
Effort besides DoR	Adhering to a DoR compromises item preparation only partially.	[C1]
Flexible adherence	Not all criteria are adhered to but an attempt is made to go as far as possible.	[C1, C2, C3, C4, C5, C7]
Full adherence	All criteria are adhered to before inclusion in a sprint.	[C6]
Item may become clearer	Some aspects of an item may become clearer when developing it.	[C1, C3, C5]
<i>Detrimental to Adherence</i>		
Ad-hoc items	Smaller items are taken up during the sprint without refinement.	[C2, C3, C5]
Carelessness	Carelessness may cause criteria to not be adhered to.	[C1]
Hurrying	Items are taken up in a hurry without considered all criteria.	[C5, C7]
Non-applicable criteria	Some criteria may be irrelevant for an item.	[C1, C2, C5, C7]
<i>Enforced By</i>		
Experience based	Items readiness is assessed through experience.	[C8]
Incomplete items returned	Incomplete items are returned to their creator.	[C5]
Product owner checks	The Product Owner pays attention to DoR adherence.	[C1, C2, C3, C4, C5, C6]
Product owner clarifies	The Product Owner clarifies items with unclear specification by the customer.	[C8]
Scrum master checks	The Scrum Master pays attention to DoR adherence.	[C5]
Shared responsibility	The responsibility for checking adherence is shared across the team.	[C2, C3, C4, C6, C7, C8]
Software architect checks	The Software Architect pays attention to DoR adherence.	[C1]
<i>Enforced In</i>		
Before estimation	The item is only considered for estimation when it is deemed ready.	[C1, C4]
Refinement	The DoR is enforced in the refinement sessions.	[C1, C3, C4, C6, C7]
Technical refinement	The DoR is enforced in technical refinement sessions.	[C1, C6]

TABLE 6.7: The adherence aspects found in the case study.

Degree of Adherence

Firstly is the sub-theme about the degree to which teams adhere to their DoR. Four teams indicate that they do not always adhere to all criteria, but an attempt is made to adhere to the largest extent possible (C1, C2, C5, C7). C5 states that their goal is to adhere to 100% of DoR criteria, and if criteria cannot be adhered to before the sprint, the item needs to be supplemented during development.

C3 and C4 indicate that they take a pragmatic approach in deciding to mark items ready that do not adhere to all criteria, without mentioning their goal is full adherence. This is described by C4 as ‘cherry picking’, focusing on adhering to the criteria deemed important.

Closely tied to this is the perception that sometimes items may become clearer during the sprint. About this C5 expresses that sometimes aspects of an item are left open in refinement, mostly those on the solution, since it may only be decided during coding.

Only one team indicates that they require all DoR criteria to hold before an item is moved into the sprint. C6 states that items that are not ready are marked with a red flag in Jira which is not removed until all criteria are ticked off. Consequently, items marked with a red flag are never moved into the sprint.

During the sprint changes can be made to the criteria an item has already adhered to in some cases. C6 gives an example wherein acceptance criteria might change during the sprint, which is discussed with the customer. For C8 deviation from what is discussed in refinement about the item is done in agreement with the PO.

Lastly, C1 notes that working on an item to adhere to a DoR is only part of its preparation. In their experience, other work is required to further clarify a story to the team, although the item may already be marked as ready.

Detrimental to Adherence

Some aspects are assumed to be detrimental to the degree a team adheres to its DoR. Mentioned most often is that for some items certain criteria might be irrelevant, causing them to not be adhered to (C1, C2, C5, C7). To this C1 adds that this non-applicability is sometimes wrongly assumed, leading to additional preparatory work in the sprint being required.

Secondly, multiple teams make use of what C2 dubs ‘ad-hoc’ items. These are smaller items, such as fixing issues that occur during development, which are taken up in the sprint without refining and estimating them as one would perform usually. Consequently, these items often do not adhere to the DoR (in full), leading to lower usage of- or adherence to the DoR.

Another aspect mentioned twice (C5, C7) is when teams are in a rush during sprint planning, hurrying may cause less strict adherence to the DoR. C5 notes that because of hurrying, including a UI design or identifying external dependencies may be skipped in their team.

Concluding, carelessness can mean fewer DoR criteria are adhered to according to C1. The participant stated that a lack of adherence is caused rather by careless interaction with the DoR than the person being in a rush.

Enforced By

Through enforcement, teams ensure that the DoR is applied in creating and refining items, and unready items do not enter the sprint. For enforcing the DoR a variety of roles can be held responsible, of which a shared responsibility across the team is mentioned most often.

About shared responsibility, C3 notes that a form of ‘social control’ exists in the team which expresses in refinement sessions where every item is discussed in the team. Should an item be unclear in refinement an owner is appointed who has to clarify the owner further, after which it is taken up in a later refinement session.

C4 states that because there is not a set role in their team that enforces the DoR, it is sometimes unclear who has the final responsibility. Multiple teams solve this issue by giving the Product Owner final responsibility over DoR adherence, which C4 has experienced in other teams before.

An example is found in C5, in this team, the Product Owner and Scrum Master check the list of new backlog items every two weeks. Items that are not sufficiently ready for refinement are returned to their creator to be supplemented. In C2 the PO is also held responsible by the team when items that are insufficiently ready for refinement enter the session: “..I would be rebuffed as product owner if I drag something into the sprint that does not meet certain conditions.”.

In C8 the responsibility to supplement items unclear within refinement lies with the PO, which is tasked with discussing the design and demands with the customer so that it becomes sufficiently clear for the next refinement session.

C8’s implicit DoR is enforced based on experience instead of explicit criteria. According to them, the more experienced team members carry a larger degree of responsibility in preparing and refining items, which also expresses itself in higher accuracy during planning poker.

Question about items during the sprint, resulting from this lack of an explicit DoR, are handled by having a daily development review (as coded in the Others theme). In this daily session developers in the team (shortly) review each others work so that novice team members’ misconceptions are noted early.

A final responsible role for enforcement is encountered in C1, where besides the PO a Software Architect occasionally checks adherence to the DoR in the ‘technical’ refinement with the development team, an event which is elaborated in the next sub-section.

Enforced In

Three aspects are identified in the case study which signify when the DoR is enforced. Firstly, teams in general enforce the DoR during a common refinement session attended by the complete team. Most teams indicate that items are discussed one by one, to which C3 adds that this is eminently the moment to discuss any remaining ambiguities.

As noted items that are not ready after refinement are held back in some manner. Both C3 and C7 state that these items reoccur in a later refinement session in-between which they are elaborated further based on DoR criteria.

Additionally are teams who have a ‘technical’ refinement with the development team. C6 described that in their team the technical refinement does not include the PO and covers all aspects concerning the item’s technical design. Since the PO is not present it is a shared responsibility for the team to adhere to the DoR in this discussion.

Lastly, two teams (C1 & C4) add that the DoR needs to hold before estimation. This forms an additional locus of DoR enforcement since both teams already try to adhere to the DoR in the refinement sessions. About this C4 says that items only reach the sprint planning phase when they are clear to the team and subsequently estimated.

6.4 Summary

The case study results discussed in this chapter cover the practitioner perspective for the first four sub-questions. In discussing the practitioner rationale for DoR usage (Section 6.3.1) SQ2 is answered.

The found rationales most commonly manifest in sprint planning, for which making the refinement process predictable is mentioned most often. Mentioned repeatedly as well are that the DoR reduces communication overhead and that it prevents unclear items from entering the sprint.

Following was the discussion of construction aspects, which do not correspond directly to sub-question one through four. Nevertheless, it may aid in constructing the framework as part of answering SQ5. Most notably team member experience from outside and within the team is cited as an important factor in DoR construction.

Subsequently, aspects related to explicitness were gathered to answer SQ1. Firstly, some evidence for an implicit DoR was found in one case (C8) where the criteria are not documented and the DoR is not called as such. Secondly, agreement within the team is most frequently noted by cases, as well as conscious usage through knowing the concept and documenting the DoR.

The criteria in the DoR are a core part of its application. A total of 37 criteria were discovered, forming the answer to SQ3. Some frequently mentioned criteria are to formulate items according to a template, to include a UI design, for it to have acceptance criteria, and to estimate it before planning. Many criteria found were mentioned by a singular team, and are thus likely to be context-dependent.

Additionally, the importance of criteria to the teams was identified to contextualize the answer to SQ3. In this multiple teams indicated acceptance criteria, the item being estimated, having a UI design, and knowing the external dependencies are most important to them.

Lastly, adherence to the DoR was covered to provide an answer to SQ4. A foremost aspect is that not all criteria are always adhered to, but an attempt is made to adhere to the highest possible extent. Teams note frequently as well that the PO checks whether items adhere to the DoR, but that there is also a shared responsibility for ensuring adherence. Concluding, the DoR's adherence is often checked in refinement sessions.

Chapter 7

Comparison

This chapter covers the third research phase's results concerning the comparison of the MLR and case study outcomes. In this comparison, the research problem is addressed by highlighting and closing the gap between the DoR's state-of-the-art and state-of-the-practice. Likewise, the answers to sub-question one through four are substantiated further in this comparison.

Additionally, the resulting framework is presented which addresses practical aspects to take into account when researching or adopting the DoR. This is the first step towards answering the fifth sub-question. In Chapter 8 the framework is validated with experts and where required adjusted, forming the final answer to SQ5.

7.1 Comparative Analysis

For the comparison the perspective of the sub-themes from the case study is taken, since the research aims' main focus is on DoR usage in practice. It is primarily compared what observed practical aspects occur in the literature and what practical aspects the literature misses. Where viable a direct table-based comparison between aspects from the sources is made, in other cases, the description is textual.

The ordering of the themes from the thematic network (Figure 6.1) is followed, which is notably based on the expected order of occurrence in first-time DoR adoption. Consequently, the construction of the DoR is covered in the case results but does not explicitly occur as a theme in the MLR results. This is detailed further in Section 7.1.2.

Likewise, the importance of specific criteria to teams is not covered in the literature, which did become apparent in the case study. Therefore, this sub-theme is not included in the comparison but is nevertheless considered in framework construction in Section 7.2.

7.1.1 DoR Rationales (SQ2)

In Table 7.1 the comparison between rationales found in the case study and MLR is presented. MLR Rationales and IDs marked as ‘?’ indicate that no comparative aspect is found in the literature results, which is elaborated upon in the discussion.

Case Rationale	IDs	MLR Rationale	Literature IDs
<i>Sprint Planning</i>			
Constructing DoD	[C4]	?	?
Uniform way of working	[C2, C4, C6, C7, C8]	Planning Efficiency	[S23, G5, G15, G22].
Used as blockage	[C1]	?	?
Ease of checking item readiness	[C1, C4, C5, C6]	Planning Efficiency	[S23, G5, G15, G22].
No need in refinement	[C8]	?	?
Improved item estimation	[C4, C6]	Estimation	[S3, S7, S17, G1, G2, G11]
Remembering work for readying item	[C1, C7]	Planning Efficiency	[S23, G5, G15, G22].
<i>Sprint Execution</i>			
Reduced waste	[C3, C7]	Waste	[S17, S23, G6, G13, G23]
Reduced scope creep	[C2, C4, C7]	Productivity	[S15]
<i>Communication</i>			
Fostering discussion about item	[C2, C3, C4]	?	?
Reduced communication overhead	[C1, C3, C4, C5, C6, C7, C8]	Communication Overhead	[S4, S23, G3, G13]
Unclear items	[C1, C2, C3, C4, C5, C7, C8]	Understanding	[S5, S6, S8, S20, S22, G4, G12, G18]
<i>Documentation</i>			
Accountability towards customer	[C5, C6]	Transparency	[S5, S6, G2]
Improved documentation	[C2, C5]	Documentation Quality	[S4, S14]
<i>Final Product</i>			
Improved quality	[C5]	Software Quality	[S5, S6, S15, G2]
Improved value	[C2, C5]	Customer Satisfaction	[S10, G22, G24]

TABLE 7.1: The comparison of rationales.

Sprint Planning

In the sprint planning sub-theme, most correspondence is found with the MLR rationale Planning Efficiency. Firstly, creating a uniform way of working by implementing a DoR relates to Planning Efficiency. According to C4, C6, and C7, adopting a DoR provides predictability and structure in planning, which is expected to improve its efficiency.

The ease of checking item readiness also aids in Planning Efficiency, since a DoR makes it clearer when an item is ready, both when preparing for- and during refinement sessions. As C6 states, preparing questions about items based on the DoR can cause the refinement sessions to be more focused, which could increase their efficiency.

Furthermore, having a DoR can help in remembering what an item needs to adhere to before the sprint, meaning required work is not forgotten which otherwise may reduce efficiency. Both C1 and C7 noted that forgetting criteria can cause delays in including an item in the sprint, or during development.

Improved item estimation corresponds to the Estimation rationale from the MLR. This relation is signified by the fact that a DoR may according to both improve estimation accuracy by providing more complete or higher quality items.

Lastly, three rationales from practice were not identified in the literature. A possible explanation for not finding coverage on the misuse of DoR as a blockage and not perceiving the need for a DoR in refinement is that they are somewhat opinionated statements based on personal experience.

Sprint Execution

The reduction of waste during development in the sprint directly corresponds to the similarly named Waste MLR rationale. Both concern the fact that a DoR can prevent non-value-added development on items by requiring clear and complete specifications, and incorporating customer value.

Although a reduction in scope creep is mentioned in multiple cases it was not encountered directly as a rationale in the literature. The closest related MLR rationale is Productivity since scope creep can cause productivity issues. S15 states that participants using user stories and quality guidelines more often agree than non-users that they improve productivity.

Communication

A reduction in communication overhead is mentioned in most cases as well as four sources in the MLR. The DoR can reduce the discussion required about items during the sprint by ensuring that they are discussed and agreed upon as ready beforehand.

Closely related is the rationale that a DoR fosters discussion about the item during refinement. The cases indicate that a DoR offers a set of points to discuss the item, which could improve the subsequent intensity of the discussion. However, the rationale in this form was not identified in the literature, which mainly focuses on the aforementioned reduction in communication overhead during the sprint.

Likewise, the adoption of the DoR leading to fewer unclear items is mentioned frequently in both cases and literature. In the literature this is often worded as the DoR resolving a lack of understanding, such as the issue described in S20 by Rizkiyah et al., (2020, P. 270): *“Lack of understanding of the requirements at the planning meeting, which results in changes in the ongoing sprint.”*

Documentation

Firstly, in the case study, it became apparent that two cases find having a DoR to improve accountability towards their customer (C5 & C6). In the MLR the closest corresponding rationale concerns transparency and traceability of items towards others in the organisation by having quality criteria.

For ease of comparison, both other entities in the organisation and a customer are considered external parties to the team. Improved transparency towards an external party through higher quality requirements can be the first part of achieving this accountability. When items are clearly specified by leveraging a DoR, and subsequently agreed upon with the customer, the accountability partially shifts according to C6.

Additionally, improved item documentation quality is given as a rationale for the DoR in both cases and literature. For C5 the improved documentation could prove useful in the future as a reference to the items' implementation. S14 found that quality criteria, which could be part of a DoR, reduced the quality defects in requirements written in the user story format. Although the perspectives are different, both attribute benefits to (DoR) criteria in ensuring quality documentation.

Final Product

The last rationale category concerns the final product. Although mentioned only in one case (C5), an improved software quality (indirectly) caused by a DoR is mentioned multiple times in the literature. This gap could be explained by the case study's focus on Product Owners, who despite their grasp of the ASD process and planning may be less familiar with the final product's technical implementation quality than developers or Tech Leads.

An improved value as mentioned in cases C2 and C5 can be linked to the Customer Satisfaction MLR rationale. Both cases link the improved final product's value to the customer's satisfaction with it since a DoR causes requirements to be specified correctly. In the literature, G22 notes that practitioners perceive the DoR as important for achieving the Agile principle of customer satisfaction, which corresponds to these findings.

7.1.2 DoR Construction (SQ5)

Most aspects found in the case study related to constructing a DoR are not covered by the literature. These aspects are nevertheless included in the framework since they provide novel insight into how a DoR typically takes shape. Considered are that DoR criteria can be influenced by the type of development a team typically undertakes, and what inspiration sources are used.

Nevertheless, the fact some cases have an organisational DoR is somewhat similar to the notion from the literature that a DoR needs to be an agreement between teams, for instance in cross-team projects. Although an organisational DoR is rather aimed at creating a way of working and providing an example, as opposed to creating an agreed-upon and uniform DoR for projects.

7.1.3 DoR Explicitness (SQ1)

Table 7.2 presents the comparison between explicitness aspects found in cases and the MLR. Notably, the literature has greater coverage of how the DoR is an agreement than on how conscious teams are in applying the concept.

Case Aspect	Case IDs	MLR Aspect	MLR IDs
<i>Agreement</i>			
With external party	[C1, C4]	Between customer and team	[S3, G11]
Within team	[C1, C2, C3, C4, C5, C6, C7, C8]	Between team; PO and team	[S1, S5, S16, G2, G5, G14, G16]; [S9, S17, S18, S19, G7, G8, G18]
With other departments	[C5]	Across teams	[S17, S26, G9, G15]
No customer interference	[C6, C8]	?	?
Organisation checks	[C1, C4]	Across teams	[S17, S26, G9, G15]
<i>Consciousness in Usage</i>			
DoR called as such	[C1, C2, C3, C4, C5, C7]	?	?
Documented as	[C1, C2, C3, C4, C5, C7]	Checklist	[S5, S6, S7, S17, G2, G5, G19, G26]
DoR not called as such	[C6, C8]	No Well-defined List	[S15, G8]
Unconscious usage	[C1, C3, C5]	Collection of criteria	[S1, S4, S18, S26]

TABLE 7.2: The comparison of explicitness aspects.

Agreement

Four out of five DoR explicitness aspects related to agreement are found to correspond to literature. A novel aspect is that for some cases there is explicitly no customer interference in forming the DoR. The lack of literature coverage could be explained because, to authors or practitioners, this aspect might seem evident.

In contrast, the agreement with an external partner such as a customer is found in cases and literature. C1 gives an example in which DoR agreement was sought between their team, the customer, and an involved third-party SPO. In C4's organisation agreement on the DoR with the external customer as part of contracting is commonplace, which corresponds to G11.

In this Personal Experience Paper Hooles (2017) reports that the DoR can play an important role in contracting for external customer ASD development. The DoR needs to be discussed between the supplier (ASD team) and customer so that for ready items the supplier can ensure delivery and a fixed price can be negotiated.

As noted in Section 6.3.4, the agreement between team members seems a prerequisite for DoR usage in practice, since it is mentioned in all cases. In literature, this is found to directly correspond to between team agreement and PO and team agreement. The latter is likewise internal to the team, with the added notion that the Product Owner has the final responsibility for ensuring items are ready.

Only one case noted that the DoR is agreed with other departments (C5). In this case, DoR criteria have been agreed with the account management and support departments, since they are responsible for providing tickets, which results in items to be developed by the team.

This relates closest to agreement across teams, which is nevertheless somewhat different. In literature, this agreement is advised when teams are working on a project together, such as the cross-team development in Cisco described by S17. Albeit departments are not teams, the ASD team's dependency on them still requires agreement on the DoR for successful cooperation.

The last form of agreement is one wherein the organisation checks whether the implemented DoR fits the organisational way of working. Again the closest relationship is to agreement across teams. Similar to S17 teams are advised to adopt a DoR by the organisation, with a baseline set of criteria serving as an example to base the team's DoR on.

Consciousness in Usage

In the literature, an aspect regarding teams' consciousness in applying the DoR is how the DoR is documented. For the teams in the case study, this is most commonly in the form of a checklist, or a backlog management software template, with the latter corresponding to Tool Usage from the literature (S5, S6, G2).

DoR documentation in the form of a checklist is by contrast frequently identified in the literature. Although four cases indicated their DoR is documented in this form (C1, C3, C5, C7), its actual usage in three cases (C1, C3, C5) is unconscious, with team members remembering the criteria. This unconscious usage corresponds closer to the collection of criteria format that signifies it is unspecified how the list is to be handled.

Contrary to the majority of cases familiar with the DoR concept, the two which did not call the DoR as such (C6, C8) did not have a documented and well-defined list of criteria to determine item readiness. Both cases indicate that the agreement on when an item is ready is (for C6; likely) informal or verbal.

In Chapter 2 an implicit DoR is defined in this research as: “..*Agile software development teams apply criteria to determine backlog item readiness without explicit definition and/or awareness of the DoR.*”. Following this definition both C6 and C8 make use of an implicit DoR, since arguably their definition of a DoR is implicit, and there is no knowledge of the concept.

7.1.4 DoR Criteria (SQ3)

In Table 7.3 the criteria from the case study are compared with those from literature. All **INVEST** criteria except Estimable and Independent are identified in this comparison.

Case Criterion	Case IDs	MLR Criterion	MLR IDs
<i>Backlog Item Quality</i>			
Formulated in template	[C1, C2, C4, C5, C6]	Template	[S4, S5, S6, S13, S17, S21, G1, G2, G11, G13]
Has a title	[C2]	Completeness*	[S5, S6]
Has complete description	[C7]	Completeness*	[S5, S6]
Has roles	[C4]	?	?
Has scope	[C2]	?	?
Is problem oriented	[C3, C4]	Problem Oriented	[S5, S6, S13]
Is simple	[C3]	?	?
<i>Customer Interest</i>			
Agreed with customer	[C6, C7]	Stakeholders Reviewed	[S9]
Has business acceptance	[C7]	Responsible Person for Accepting Item	[S17, S21, G26]
Has value estimate	[C4]	Valuable	[S3, S5, S6, S7, S15, G1, G5, G16, G17, G19]
Is value oriented	[C5]	Valuable	[S3, S5, S6, S7, S15, G1, G5, G16, G17, G19]
Stakeholders involved	[C1, C7]	Negotiable	[S3, S5, S6, S7, S15, G1, G2, G5, G13, G16, G19]
<i>Development</i>			
Checks performed	[C1]	?	?
Has breakpoints	[C8]	?	?
Has process description	[C4]	?	?
Has reproduction scenario	[C5]	Additional Documentation	[S5, S6, G10, G16]
Has scenarios	[C3, C4]	Additional Documentation	[S5, S6, G10, G16]
Has success measurement	[C3]	?	?
Has UI design	[C3, C4, C6, C7]	User Experience Artifacts	[S17, S21]
Is aligned with architecture	[C7]	?	?
Takes mobile into account	[C1, C8]	?	?
Takes privacy into account	[C1]	Data Protection	[S11]
<i>INVEST</i>			
Has acceptance criteria	[C2, C3, C5, C6, C7]	Testable	[S3, S5, S6, S7, S15, S17, S21, G1, G2, G5, G10, G14, G16, G19, G20, G23, G26]
Has test plan	[C7]	Testable	[S3, S5, S6, S7, S15, S17, S21, G1, G2, G5, G10, G14, G16, G19, G20, G23, G26]
Is as small as possible	[C5, C6]	Small	[S3, S5, S6, S7, S12, S13, S15, G1, G2, G3, G5, G13, G14, G16, G19, G20]
<i>Sprint Planning</i>			
Definition of Done	[C2, C8]	DoD	[S4, G3, G11]
Dependencies to others	[C1, C3, C7]	Explicit Dependencies	[S5, S6, S13, S17, S21, G16, G20, G23, G26]
External dependencies	[C1, C3, C6]	External Dependencies	[G2, G20, G23]
Has clear way to demo	[C7]	Demo Known	[S17, S21, G11, G16, G20, G23, G26]
Has deadlines	[C3]	?	?
Is estimated	[C2, C4, C5, C6, C7, C8]	Estimated	[S12, S17, S21, G2, G4, G10, G11, G13, G16, G17, G20, G23, G26]
Is prioritised	[C7]	Prioritised	[S5, S6, S7, G10, G11, G13]
Overarching collection	[C2, C4, C6]	?	?
Release calendar checked	[C7]	?	?
Tasks identified	[C1, C5]	Small	[S3, S5, S6, S7, S12, S13, S15, G1, G2, G3, G5, G13, G14, G16, G19, G20]
<i>Team Cooperation</i>			
Refined before planning	[C5, C8]	Team Reviewed	[S17, S21]
Understood by everyone	[C7]	Understandable	[S7, G2, G3, G11, G16, G20]

TABLE 7.3: The comparison of criteria.

Backlog Item Quality

For backlog item quality the criterion about formulation in a template is identified frequently in both sources. In all cases, the user story formulation is used, which is likewise the case for literature. Nevertheless, this aspect is worded as ‘formulated in a template’ should teams want to follow a different template according to their preference.

Both criteria Items having a title and a complete description relate to Completeness from MLR. Completeness is marked with a ‘*’, since only S5 and S6 from the same authors specifically note adding titles and descriptions within this criterion. Other sources mention a general notion of backlog item completeness.

Lastly, the problem-oriented criterion from the literature is directly identified as such in cases C3 and C4. Notably, C3 does not strictly apply a template for formulating items but still requires them to be problem-oriented, except for simple items with an evident solution.

Customer Interest

Customer interest is a novel criteria category emerging as a sub-theme in the case study. In this category, all case criteria correspond to literature. Firstly, agreement with the customer on items is found in literature as the Stakeholders Reviewed criterion from S9.

Although this Personal Experience Paper focuses on external customers, C7 leverages this criterion for internal customers in their financial organisation. In this organisation the mentioned communication issues are likewise to be tackled. Additionally, they require that it is known who in the organisation is responsible for accepting the item, which directly corresponds to the literature as well.

Two cases (C4, C5) use criteria that correspond to Valuable from INVEST. C4 focuses on knowing the estimated value the item will provide, in currency. C5 by contrast requires that the items clearly describe the value it will deliver to the customer, which is likewise indicated to be an important criterion (Table 6.6).

The criterion stakeholders involved is lastly found to correspond to Negotiable from INVEST. Negotiable signifies that items are not an explicit contract, and should be open for negotiation with stakeholders (Wake, 2003).

Development

Mentioned by three cases (C3, C4, C5) is the MLR criterion that items should be accompanied by (relevant) additional documentation. The UI design as a more elaborate form of additional documentation is required by four cases (C3, C4, C6, C7), relating to the User Experience Artifacts criterion from literature.

The only other case criterion found in the literature is C1 requiring that items take privacy into account, corresponding to the data protection criterion from S11. Similar to this criterion, the lack of comparison in literature with other Development criteria could be explained by their context dependency.

INVEST

In the category concerning criteria that relate closest to INVEST, two case criteria correspond to Testable. Noted most often by cases is that items should have acceptance criteria, which are in turn a way of verifying an item's testability according to S7. C7 is the only case that follows Wake's (2003) suggestion to validate testability by writing tests in addition to acceptance criteria.

C5 and C6 indicate that according to their DoR items should be as small as possible, which relates directly to the Small criterion from INVEST. Both aim to facilitate the incremental development of software as the core of ASD through certainty about completion and improved estimation.

Sprint Planning

Firstly, both cases and literature note that a DoR can require the DoD to be known. This is to ensure that the item-specific DoD is available at the time of the finalisation of item development.

Also, both types of dependencies (between items and external) are represented in cases and literature. External dependencies are considered to be more important than those between items in practice, with multiple cases noting the former as one of their most important DoR criteria.

C7 is the only case requiring a clear way to demo an item as a separate criterion. In literature, this is included more often explicitly in the form of the Demo Known criterion. This forms a distinct difference with requiring acceptance criteria, which is widely included in literature and practice.

A criterion that does occur frequently across the sources is that items should be estimated before they are planned. As stated in Section 5.3 this is considered to be separate from the Estimable criterion from INVEST, which concerns the item's quality of being estimable. It can be assumed that by including estimated as a criterion teams expect items to be written so that they are estimable, however, no case included the INVEST criterion as such.

Related to estimated is prioritisation, which is reported frequently in literature as a possible criterion for a DoR. In practice, only C7 included this as part of their DoR. While other teams may practice prioritisation as part of ASD they do not include it in their DoR as a criterion.

Multiple cases note that their items should be part of an overarching collection, such as a user story epic, when applicable. This criterion is not found explicitly in literary DoR examples. In practice, this serves as a means of keeping track of items, such as in C6 in which the project leader uses epics for keeping an overview of teams' sprint planning.

The last criterion, tasks identified, corresponds to Small from INVEST. C5 notes that the items are divided into as small as possible sub-tasks, which is assumed to aid in estimating whether an item can be completed during the sprint.

Team Cooperation

For team cooperation in practice, the two criteria correspond to literature in varying degrees. The first, refined before planning, relates closest to two literary sources that state an item needs to be reviewed by the team. In refinement, the items are discussed by the team and ambiguities are resolved as stated by C5.

Lastly, the criterion that items should be understood by everyone (C7) corresponds to understandable from literature. Both aim to ensure that everyone in the team understands an item before inclusion in the sprint, preventing possible ambiguities or differing interpretations.

7.1.5 DoR Adherence (SQ4)

For the aspects related to DoR adherence a low degree of overlap was found between cases and literature. The foremost reason is that in the literature only the three aspects in the form of adherence types are identified; full, partial, and threshold adherence. Of these, threshold adherence is not identified in practice.

Partial adherence is identified most often in both cases and literature, which is in line with the flexible nature of ASD methods. In most cases mentioning the aspect it is noted that although adherence is not always complete, an attempt is made to adhere to the largest possible extent (C1, C2, C5, C7). This reality corresponds to the Pareto efficiency noted by S17, wherein items are sufficiently ready so that there is understanding and confidence in the team.

It is mentioned by S5 and G5 in the literature that exceptions causing criteria not to be adhered to can be made based on experience or item attributes. This is likewise found in cases, with four of them mentioning criteria are sometimes non-applicable reducing adherence (C1, C2, C5, C7). C3 and C4 mention ‘cherry picking’ criteria based on what they find important, without striving for full adherence.

Additionally, C8 specifically mentions that for their implicit DoR, experience is used as the way to enforce the criteria, which could result in experience-based partial adherence. In this case, there is a shared responsibility for ensuring that the implicit criteria are adhered to, similar to the within-team agreement on the DoR from the literature.

Full adherence to a DoR is as rare in practice as it is in literature. In the case study, only C6 indicated that all criteria are adhered to before including an item in the sprint. This is enforced through flagging items in Jira, which prevents un-flagged items from entering the sprint at all.

In the literature G2 concurs by stating that only a ‘ready’ and ‘unready’ state should exist, only allowing non-applicable items to be skipped. This contradicts the aspect that items may become clearer in the sprint, leading to higher adherence, as stated by flexible adherence cases C1, C3, and C5.

Concluding, there is a relationship between some remaining case aspects concerning who enforces the DoR and the types of agreement found in the literature (Section 5.1). The granting of responsibility to the Product Owner corresponds to an agreement between the PO and Team. In some cases, responsibility is given to the Scrum Master (C5) or Software Architect (C1), which is similar to the specific roles agreement from G7.

7.2 Framework Construction

As indicated in Section 3.3, the Agile Transition and Adoption (ATA) framework by Gandomani and Nafchi (2015) is adapted to be compatible with the DoR. The reference framework is presented in Figure 7.1.

7.2.1 Agile Transition and Adoption Framework

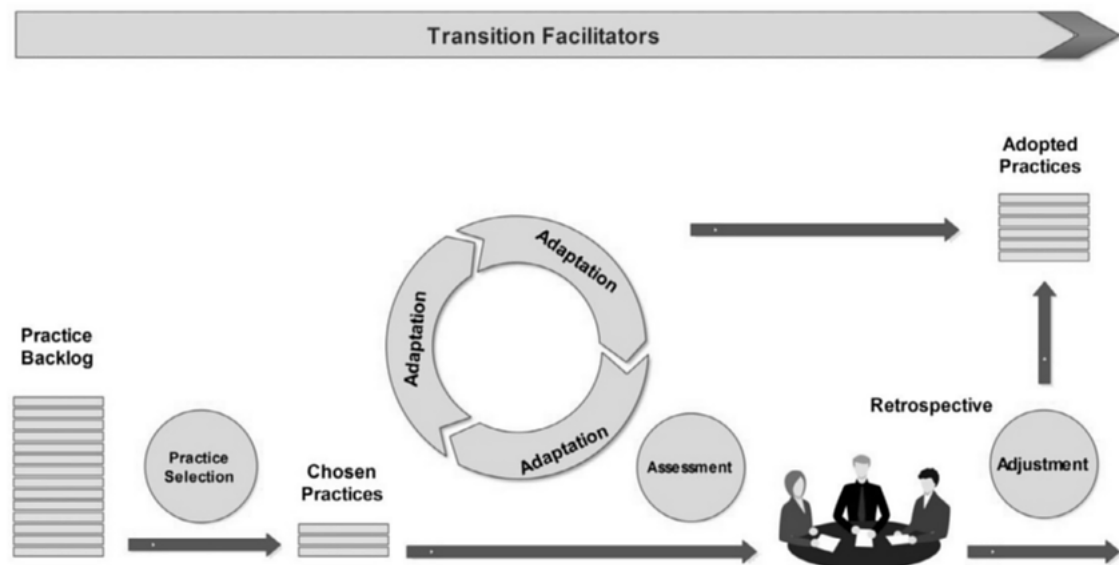


FIGURE 7.1: Practical Agile transition and adoption framework. Reprinted from “An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach”, by T. J. Gandomani and M. Z. Nafchi, 2015, *Journal of Systems and Software*, 107, p. 204-219.

The ATA framework is chosen due to its number of citations (136), grounding in practice and theory, and fit with DoR due to its Scrum-based structure. It is likewise compared to other Agile transformation frameworks by Jovanović et al., (2017) for the construction of their framework.

Gandomani and Nafchi (2015) conceptualise the framework based on a grounded theory study involving 49 practitioners and the widely applied Plan-Do-Check-Action (PDCA) approach. It aims to assist organisations and teams in adopting Agile practices and adapting them to their needs, which is firstly accomplished through its structural characteristics.

The framework focuses on the business value that can be derived from a transition to ASD and takes into account the Agile values. Secondly, the framework supports iterative transition, which incorporates the PDCA approach. In each iteration, only a part of the Agile practices is to be planned, adopted, evaluated, and adjusted.

Additionally, the process is continuous, meaning iterations follow each other up constantly causing an ever-growing adoption of Agile practices. This iterative continuity allows a cycle to be started whenever the need for a practice arises. Lastly is to enable a gradual transition, signified by the earlier noted selection of Agile practices, since changing organisational culture and mindsets requires time.

A second means of support offered by the framework are the key activities involved in adopting practices and transitioning to ASD. The initial activity is to select Agile practices from an Agile practice backlog, which is prioritised based on the business value and benefits the practices can offer. This is derived from the Scrum method which provides familiarity to ASD teams applying the framework.

The selection itself should be based on the needs and priorities of the organisation or team, wherein the noted focus on business value lies. For this, it is recommended to select only some practices for adaptation to enable an iterative, continuous, and gradual transition toward Agile.

Within selection, an assessment may be performed to determine what practices are applicable based on needs and priorities, capabilities, customer involvement, and project work. In the case of this framework roles such as project managers or Agile coaches are held responsible for assessment and selecting practices.

Following is a collection of chosen practices that are taken up for the next activity. The adaptation of- and to practices is an ongoing effort that should not be time-boxed, since changing organisational behaviour and cultures costs time. In all cases, the adoption of the practice should be the end goal. An additional goal can be the institutionalisation of practices, meaning the practice becomes fully ingrained in the organisation or team's way of working.

To monitor how the transition towards adopting the practices is proceeding the assessment activity is recommended. In this, the challenges encountered by the ASD team in adapting practices may become apparent through measuring metrics such as collaboration, development pace, and the realisation of benefits. This activity can be performed by the ASD team themselves or by external parties such as coaches or managers.

In the retrospective, the results of the assessment are reviewed and discussed by all team members. From this, the challenge areas, strengths, and weaknesses within the team are to be uncovered. The final concern of this meeting is what to adjust to improve upon the current transition conditions, leading to a plan for the future.

The last activity is an adjustment of the practices based on the plan, leading to further adaptation. It is suggested that teams focus on those practices that need adjustment first before continuing the adoption of new practices, to ensure that adaptation is effective.

A final element of the ATA framework is the transition facilitators noted in the top arrow. These facilitators are elaborated upon in a different work by Gandomani et al., (2014). Practitioners find that these facilitators make Agile transition and adoption easier, which includes among others; training, coaching, the buy-in of management and the team, and continuous meetings and negotiations.

7.2.2 DoR Adoption Framework

In Figure 7.2 the DoR adoption (DA) framework, adapted from the ATA framework, is presented. The main differences are the focus on a singular ASD team, the omission of transition facilitators, the transition from a backlog with Practice Selection towards rationalisation for DoR adoption, and the addition of aspects in the lower half.

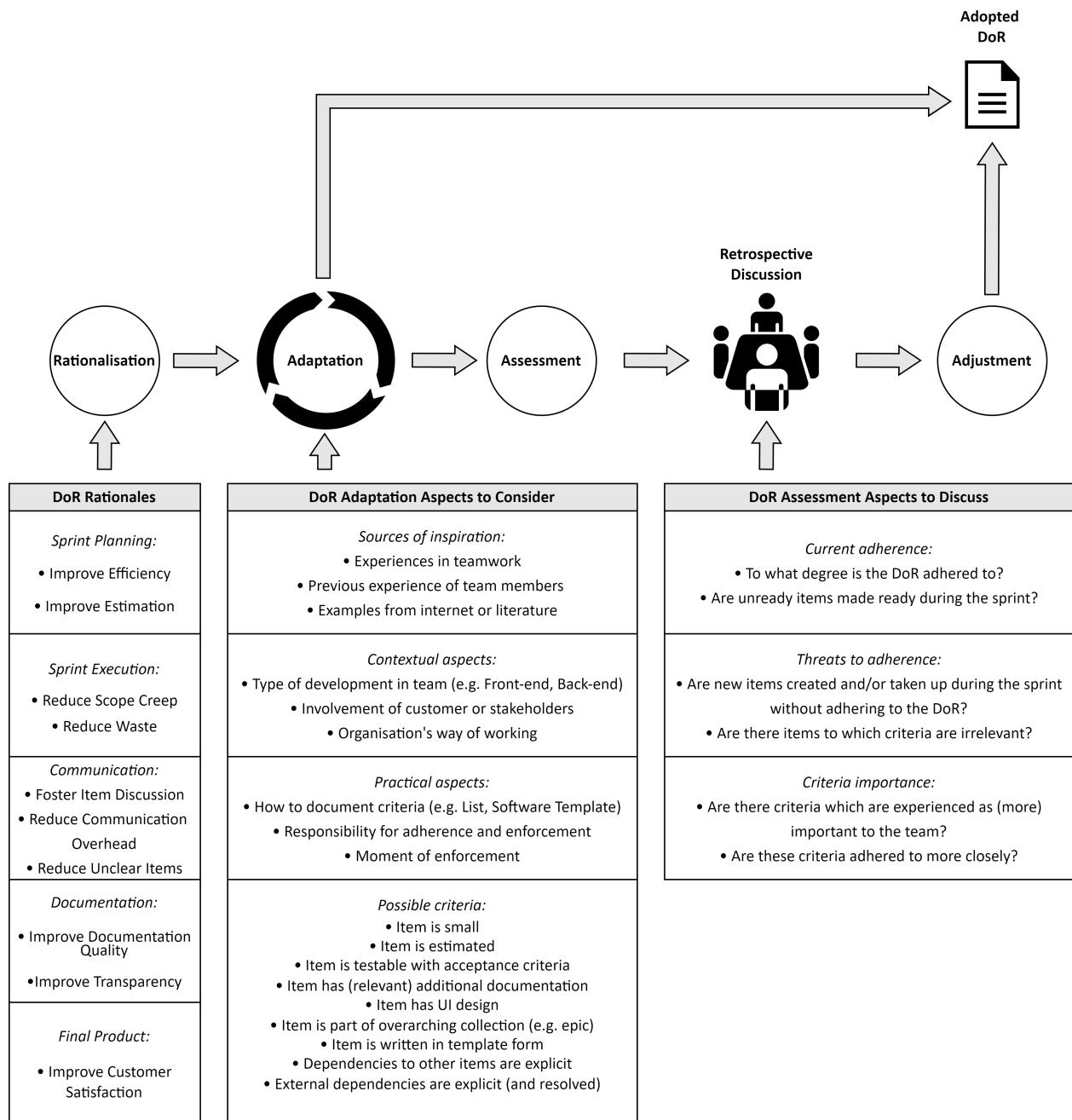


FIGURE 7.2: The DoR adoption framework.

Compared to the ATA framework, the DA framework focuses on individual teams instead of organisations and their teams. In practice, most teams in the case study were responsible for adapting and adopting their own DoR. Although sometimes an organisational DoR is present, or adoption of the concept is recommended, teams are given freedom in its adaptation.

The transition facilitators are omitted under the assumption that the teams leveraging the framework are already in ASD transition or have transitioned, diminishing their need for the adoption of the DoR as a single practice. In the following the DA framework's sequencing, activities, and aspects are elaborated upon.

Rationalisation

Since the ATA framework focuses on selecting multiple practices, and the DA framework merely on adopting the DoR, the new initial activity is rationalisation. In this activity it is to be assessed what DoR rationales apply to the ASD team at hand, and whether this warrants DoR adoption. The focus in this is on the (Agile) business value that a DoR can effectuate.

In practice and literature a variety of possible rationales for the adoption of a DoR are given, categorised in the framework according to where in ASD they manifest. Most rationales are backed up by both multiple cases and multiple literary sources to ensure their relevance. Two exceptions are reduced scope creep and fostering discussion, which were noted by multiple practitioners but did not occur in the literature.

Adaptation

In the next activity, the DoR is adapted by the team, and the team adapts to the DoR, with the end goal being adoption. Again, an additional goal is institutionalisation in the team. For both goals, agreement between all team members team on a DoR is required, as noted in Section 6.3.4.

Similar to the ATA framework, the adaptation activity is not time-boxed but should be a continuous effort that may span multiple ASD sprints. DoR adaptation involves several aspects which can be considered by teams, these are based on the previous literature and case study research. In the continuation, the different aspects are discussed briefly with the rationale for their inclusion.

Three sources of inspiration for adapting a DoR are noted in the framework, all mentioned in at least three cases. These sources mostly inspire the criteria, but other aspects can likewise be derived from them. The organisational DoR is moved to the contextual aspects since it is assumed to be part of the team's context and not every organisation offers this.

Furthermore, contextual aspects can be considered when adapting a DoR in a team. The type of development a team performs is considered a contextual aspect that may require DoR adaptation; it is an influential aspect in determining DoR criteria according to three cases.

Additionally, two cases indicate the DoR is agreed with the customer, or has been agreed with them before, thus it is advised that teams check whether agreeing criteria with a (longer term and external) customer is required.

Lastly, three cases note their DoR is adapted from an organisational one, of which two indicate that the organisation expects teams to have a DoR to fit the organisation's way of working. In the case of an organisational DoR, it is advised to use it as a starting point for adaptation.

Suggested next are practical aspects that can be considered which pertain to the way of applying the DoR. A majority of cases noted that their DoR is documented. Some cases initially use a checklist, later transitioning towards unconscious usage of the criteria. Others make (or made) use of a checklist integrated with backlog management software such as Jira, presented when submitting an item.

The responsibility for adherence and enforcement can be discussed when adapting the DoR. For adherence, it is common for team members to have a shared responsibility for ensuring items are ready after the refinement session. The Product Owner is frequently tasked by teams with checking and enforcing adherence to the agreed DoR criteria.

Another practical aspect is the moment at which the DoR is enforced. Some teams supplement the adherence after refinement sessions with an additional verification moment. Two cases note that items not agreed to be ready are not taken up during estimation. Others note that a technical refinement furthermore takes place within the development team, during which the DoR is likewise applied.

The last, but core aspect of adapting a DoR is the criteria it contains. The most commonly noted criteria by teams, which also occur in multiple literature sources, are included in the framework. From INVEST only Small and Testable are mentioned in more than two cases and literature. To Testable, Acceptance Criteria are added as a main way to check item quality.

Assessment, Retrospective Discussion and Adjustment

Similar to the ATA framework, an assessment of the transition toward adopting the DoR is recommended. In this assessment general metrics such degree of DoR adherence can be measured, or rationale-specific metrics such as time spent on planning sprints, estimation accuracy, or documentation quality.

After an assessment, a retrospective meeting is held, preferably with all team members, during which the results of the assessment are discussed. In this meeting the challenge areas of transitioning towards adopting the DoR are addressed, for which aspects to consider are noted. Following this, the decision can be made to adjust the DoR based on the findings in the retrospective discussion, which leads to further adaptation of the adopted DoR.

In the retrospective, it may first be considered to reflect on the degree of adherence, because a lack of adherence could potentially cause the rationales to be left unrealised. Flexible adherence is most common, meaning not all criteria are always adhered to, with most teams nevertheless attempting to adhere to the largest possible extent.

Likewise, some teams indicate that unclear (I.e. unready) items are sometimes planned under the assumption they become clearer during the sprint. It can be checked in assessment and subsequent retrospective discussion if this occurs, or whether these items remain unclear causing issues.

In some cases, smaller ('ad-hoc') items in terms of effort are created and taken up during the sprint. Some teams indicate that these items often do not adhere to the fullest extent of the DoR, or the DoR is not being applied to these items at all. This can be a threat to the usage of the DoR which could again result in unrealised rationales.

In turn, criteria may be irrelevant to certain items, which can cause them to not be adhered to. Examples of these items and the subsequent irrelevant criteria can be discussed to determine agreement on this irrelevance and further steps to take.

When enquired all teams with an explicit DoR could indicate that there are criteria with higher importance than others. In this context, it may be discussed in the retrospective if the importance of items is the same for everyone in the team, or whether agreement is to be sought. Another consideration is whether adherence to these items is more strict, or if additional effort is required.

Adopted DoR

After performing an initial iteration of these steps the DoR has been adopted by the team, which does not mean it remains static throughout its application. As noted earlier, the adaptation of the DoR can be a continuous process if required. Later in the DoR's life, further adjustments can take place based on emerging rationales or team experience. For this another round of assessment is suggested after which a retrospective can be used to discuss possible adjustments.

Chapter 8

Expert Opinion Results

To validate the DA framework three participants are involved in expert opinion interviews. In Section 3.4 the expert selection criteria are set, defining what constitutes an expert regarding ASD and DoR in the context of this research. To be an expert the participants should have both experience and knowledge on these topics, which is covered in Table 8.1.

ID	ASD Experience	DoR Experience	Certifications
E1	11 years	5 years	PSPO1, PSM1
E2	22 years	14 years	PSPO1, PSM1, PST, PAL
E3	9 years	5 years	PSPO1, PSM1, CAT

TABLE 8.1: The expert characteristics.

All experts have at least 5 years of experience with ASD as a coach or trainer, E2 and E3 likewise have experience in other roles such as developer or Scrum Master. Additionally, all experts have 5 years of experience, or more, in working with the DoR concept.

A variety of certifications is identified in the group of experts. All possess the Professional Scrum Product Owner I¹ and Professional Scrum Master I² certifications from Scrum.org. E2 also possesses the Professional Scrum Trainer³ and Professional Agile Leadership⁴ certifications from Scrum.org. Lastly, E3 has the Coaching Agile Transformations⁵ from ICAgile as well.

For transparency, the expert opinion interview transcripts are included in Appendix H. The results following from this are discussed (Section 8.1) and adjustments to the DA framework are made (Section 8.2). The discussion of the results is structured by the main questions from the expert opinion Protocol in Appendix G.

¹<https://www.scrum.org/assessments/professional-scrum-product-owner-i-certification>

²<https://www.scrum.org/assessments/professional-scrum-master-i-certification>

³<https://www.scrum.org/become-professional-scrum-trainer>

⁴<https://www.scrum.org/assessments/professional-agile-leadership-certification>

⁵<https://www.icagile.com/certification/coaching-agile-transformations>

8.1 End Results

8.1.1 Expert Understanding

The first question to participants concerns whether the framework is understandable from their perspective as an expert. E1 largely understands the framework but states that without the explanation some of the aspects in lower half are described too brief. From their perspective, some aspects are obvious and may have been omitted, but do not detract from the comprehensibility.

Something which could be clearer for E1 is the feedback loop between adjustment, the adopted DoR, and adaptation. In the explanation, this became clear to E1, with special attention to the continuity of adaptation signified by the cycle symbol. Their suggestion is to include an arrow in the framework model for clarification.

For E2 the framework is clear without the explanation, but the given explanation does add to their understanding. In their view, the rationales are a facet of the framework which is especially well thought out.

They likewise note that the inclusion of both fostering item discussion and reducing communication overhead is remarkable since at first glance they contradict each other. However, the expert agrees with their co-occurrence and states that this could be highlighted in the description.

Lastly, E3 does not have any questions about the framework after its explanation. Although it is attempted to describe the aspects briefly in the framework they do indicate it could be made more compact. In their view, there are also too many aspects to consider making the process of adopting a DoR too linear.

8.1.2 Expert Familiarity

Beyond their understanding, it is enquired whether the experts are familiar with the activities, their sequencing, and the aspects when aiding teams in adopting a DoR in practice. E1 recognises the importance of the DoR in practice and highlights that it is ideally used as a risk estimate for items, meaning unready items can still be taken up.

They indicate furthermore that the adaptation, retrospective, and adjustment activities occur in practice in some form. However, rationalisation and assessment in this form are novel to them. The re-occurrence of these rationales in the assessment provides ASD teams with a useful tool enabling them to work with continuous feedback: *“Why do I use the DoR, how do I use the DoR, and am I doing this well?”*.

E1 also recognises the criteria regarding dependencies and acceptance criteria in practice and confirms their importance. They miss a criterion that requires it to be known who is the stakeholder for the item or who is responsible for accepting it.

Similarly, E2 holds the opinion that the DoR is both important and to be used flexibly for item risk estimation, warning for the inflexible usage as a phase gate. They add that they recognise the risk of not using the DoR for items created during the sprint, for which the PO should be involved in estimating their risk.

The expert also recognises the rationalisation and adaptation in practice when adopting a DoR. The assessment, retrospective, and adjustment occur little to none and require additional attention from trainers. Mentioned as well is that efficiency is not a main goal of Scrum, it is rather effectiveness balanced with efficiency. They would like to see this balancing mentioned in the framework.

E3 recognises the activities and their sequence in their way of introducing a DoR. They first let ambiguities arise and ask whether teams would like to resolve this to rationalise the DoR. In adaptation they keep the initial DoR simple, with a few agreements and criteria, fearing for over-documentation.

A main difference is E3 prefers teams to start with a small DoR because otherwise, the process for readying items may become too cumbersome. While they do propose some criteria they let the team decide what is important to them. Likewise, the DoR reoccurs in retrospectives when relevant, and can be adjusted based on the discussion. They also acknowledge the risk of introducing items during the sprint in practice.

Concluding, E3 agrees that the acceptance criteria and knowing dependencies are common criteria in practice, and states that knowing the stakeholder is also part of the DoR they advise to teams.

8.1.3 Expected Practitioner Understanding

Subsequently, the experts are enquired if they expect other practitioners (I.e. ASD team members) to understand the framework. E1 is positive about the comprehensibility to practitioners but states the best fit is for Scrum Masters instead of ASD teams as a whole.

Scrum Masters could teach themselves the process of DoR adoption through the framework and apply it, while the team members merely apply the DoR and are facilitated in the adoption by the Scrum Master. According to them, it could be a highly useful tool for Scrum Masters to apply when the DoR is not working well or requires adjustment.

E2 notes that in their experience models are often misunderstood in practice, but adding the description would make it comprehensible to practitioners. The description should pay additional attention to the continuity of adaptation and that assessment, discussion, and adjustment are often overlooked in the DoR context.

Lastly, E3 expects the process steps to be understandable to practitioners, but the aspects below offer many exceptions and examples which is at odds with the lightweight Scrum framework with few prescriptions.

8.1.4 Expected Practitioner Usability

To conclude it is enquired whether the experts expect the framework to be usable for practitioners to successfully adopt a DoR. E1 is positive that the framework is usable, likewise for ASD teams that already have a DoR, since in their view all teams have a DoR in some form albeit not formalised.

Additionally, they would like to see an example DoR added, preferably from practice, so that teams can mirror this in their adoption. For the Scrum Master using the model, a list of criteria can be useful, for which the collection from Section 7.1.4 could be used.

E2 expects the framework to be usable by practitioners as well. For them a possibly useful addition may be a negative and a positive example of a DoR from ASD, so that teams know how to correctly adopt their own version.

Lastly, for E3 to use the framework as an ASD team member they would like to see fewer exceptions and examples, offering more freedom to fill in these aspects themselves. The aspects and the list of possible criteria might be too guiding, since they prefer to let ASD teams start small with their DoR focusing on their needs, preventing a large and inflexible DoR to arise due to adopting too soon.

As an alternative, they agree that one can differentiate between the aspects required to complete the steps and those which are optional, and to formulate the adaptation aspects as general questions to consider. Also, according to them a definition of the DoR could be good to include, since the DoR is often mistaken for the DoD by practitioners.

8.2 Framework Adjustment

The DA framework is adjusted based on the expert opinion interview results with the primary focus being the model, presented in Figure 8.1. Suggested adjustments to the description are not carried out further since they were already described in detail in the previous section, or were already present. An example of the latter is to emphasise the continuity of adaptation, which was covered in Section 7.2.

A first point of adjustment is the inclusion of an arrowhead from Adopted DoR to adaptation, highlighting the continuity of adaptation and the feedback loop as suggested by E1. Further adjustments to the process in the model's top half were not suggested by experts.

In the model's lower half the first adjustment is the inclusion of the criterion Stakeholder is known, noted by E1 and E3. Although this criterion was not explicitly identified in the case study, it is implied in criteria used in multiple cases such as Stakeholders involved and Agreed with customer.

Another adjustment to benefit the DA framework's usability to practitioners is to include an example DoR, as noted by E1 and E2. In this example both application rules and criteria recognised by experts in practice are incorporated. The second rule, 'Unready items may enter the sprint but are considered risky', refers to the flexible adherence found in practice.

Lastly, a set of suggestions by E3 for improving the practitioner usability is given. Although E3 suggested to include a definition of DoR, this is deemed superfluous when including an example. In contrast, the feedback that the model's aspects are too elaborate and restrictive is implemented.

To reduce the restrictiveness the phrasing of 'Possible criteria' is changed to 'Examples of criteria to include in the DoR'. For this, E3 noted that the list of criteria might be too guiding, and could be interpreted as the only options for DoR criteria by practitioners.

Where possible the number of aspects is reduced and henceforth formulated as more general questions an ASD team might ask during the DoR adoption process. In this transition, the intent is to leave the aspects which are deemed required to conduct the activities, and to omit exceptions.

The exceptions omitted are whether unready items are made ready during the sprint, whether there are items to which criteria are irrelevant and criteria importance. Although these aspects occur in practice they are not considered essential to discuss in a retrospective for the successful adoption of a DoR.

Concluding is an adjusted version of the DA framework model catered towards improved usability for practitioners, achieved through as few changes as possible to safeguard the correspondence to the framework's contents and intent. Consequently, the previous version in Figure 7.2 offers a more comprehensive overview of the DoR in practice suited for scientific usage.

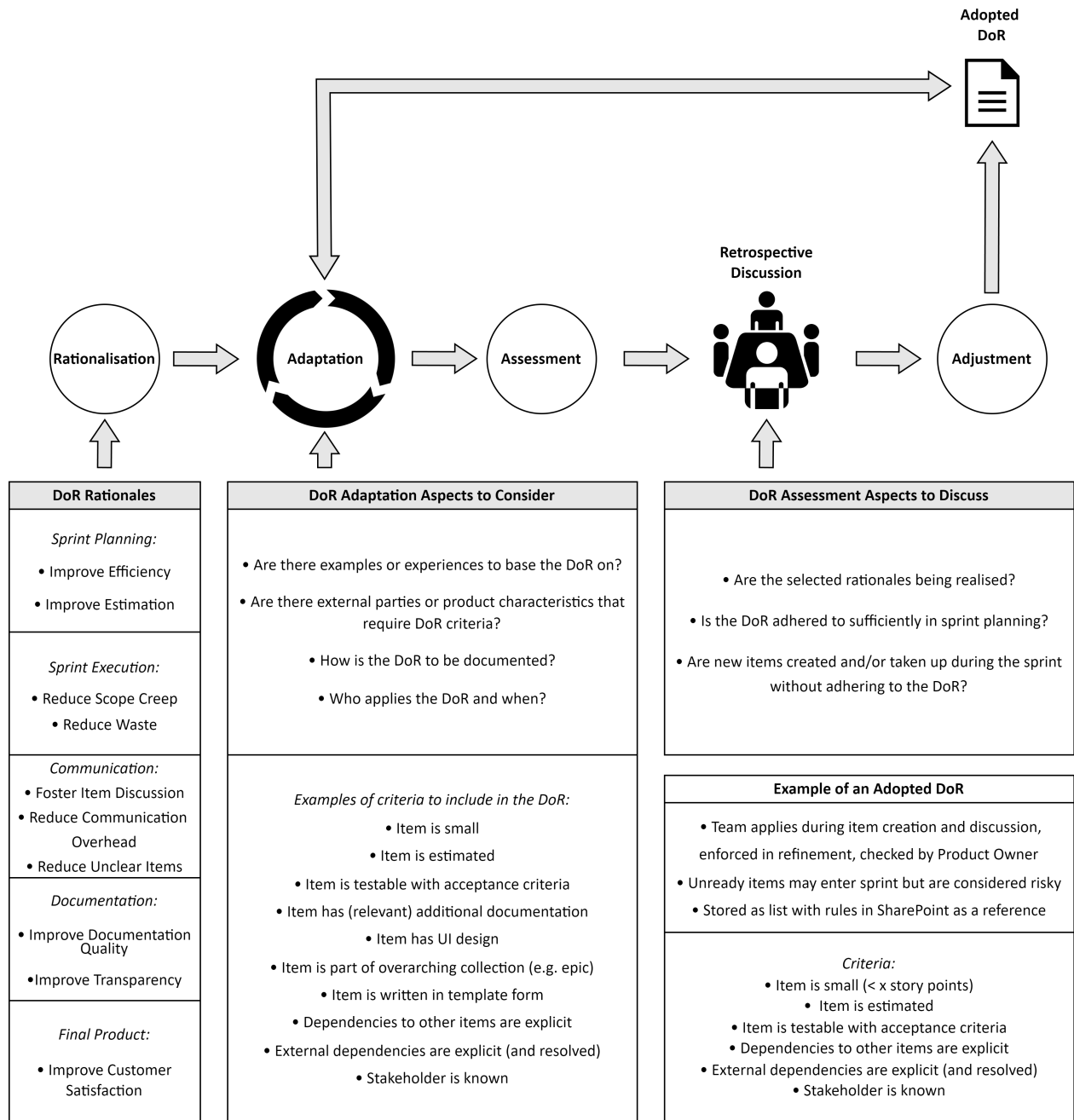


FIGURE 8.1: The adjusted DoR adoption framework.

Chapter 9

Conclusion

In this chapter, the final answers to the sub-questions posed in Chapter 2 are given. By answering the sub-question the answer to the main question is to become apparent:

How is the Definition of Ready applied in Agile software development teams?

The concluding answers focus on the practitioner perspective from the case study and expert opinion phases, where required supplemented with the literature from the MLR which was used in the comparative analysis. This corresponds to the overall research aim to close the gap between literature and practice.

SQ1: Is the Definition of Ready applied explicitly or implicitly by Agile software development teams?

The first sub-question concerns whether there are teams that apply a DoR without explicitly defining it in the team and/or are unaware of the DoR concept. In this, the aim is to find a possible explanation for the relatively low adoption rate of DoR compared to other ASD practices encountered in previous works.

To characterise the DoR's explicitness two aspects are uncovered in practice; the type of agreement and the consciousness in usage. The majority of teams in the case study agree on the DoR within the team, with some teams additionally agreeing with an external party. This also includes the provision of an organisational DoR which teams may adapt.

Additionally, most teams who use criteria for determining item readiness are familiar with the DoR concept and use it in the form of a checklist or backlog management software template, with some teams later transitioning towards unconscious usage. In this unconscious usage, the DoR criteria are remembered by team members during sprint planning and refinement sessions.

Concluding, the explicit application of the DoR as performed by six teams is the most common, two teams contrarily make use of an implicit DoR. The latter teams likewise indicate that the customer does not interfere in how they determine items to be ready for the sprint.

SQ2: What is the rationale of Agile software development teams for applying the Definition of Ready?

To find out why ASD teams apply the DoR and what consequences are perceived it is investigated in the case study what rationales occur in practice. Most of the 15 practitioner rationales can be linked to those from literature. Two exceptions are that a DoR may aid in constructing a DoD and that it may foster preemptive discussion about items, instead of reducing it.

The most common rationales in practice are the reduction of unclear items that enter the sprint and reduced communication overhead. These rationales are related, the DoR ensures items are clear and agreed upon before the sprint, which causes a reduction of item discussion due to ambiguities during the sprint.

Another frequently identified rationale is that the DoR offers a uniform, or rather predictable way of working in sprint planning. This is related to the Planning Efficiency rationale from the literature in the comparison since an improvement in sprint planning efficiency is expected as a result. Another improvement to planning efficiency is that the DoR makes it easier to check item readiness by having a clear and agreed-upon collection of criteria.

SQ3: What criteria are applied by Agile software development teams in their Definition of Ready?

A broad spectrum of 37 DoR criteria is identified in practice, which have been divided into six categories spanning aspects of ASD. The categories concerning the item's development and sprint planning each contained the most criteria.

The criteria occur in varying degrees, ranging from context-specific and being applied by a singular team, to general and leveraged by multiple teams. Many criteria encountered in practice are of the first type, which is expected since a DoR is adapted to the specificities of the team's context.

In literature, the INVEST mnemonic for DoR criteria is mentioned frequently. Contrarily, in practice the mnemonic does not occur in full as DoR criteria. Negotiable, Valuable, Small, and Testable all occur explicitly in multiple DoRs from practice, while Estimable is only implied in the Estimated criterion, and Independent does not occur in any form.

Thus, INVEST may provide ASD teams a starting point for selecting DoR criteria, but a plethora of alternatives exist that can be offered as an example. The most common criterion beyond the mnemonic and items being estimated is the formulation of items according to a template for which the user story format is preferred by ASD teams.

Other criteria used by multiple teams are; the inclusion of a UI design for items that pertain to the product's front-end, dependencies between items being explicit, external dependencies being known, and items being part of an overarching collection. In the expert opinion phase, two independent experts confirmed the importance of knowing internal and external dependencies to prevent waiting for them to be resolved.

SQ4: When do Agile software development teams consider a backlog item sufficiently ready to move into a sprint?

The most common adherence to the DoR in practice can be characterised as Partial, from Section 5.4. This corresponds with the commonality of partial adherence found in literature. In this DoR usage teams do not always adhere to all DoR criteria before moving an item into a sprint, which is in line with the overall flexible nature of ASD methodology.

Most teams leveraging flexible adherence attempt to adhere to the greatest extent possible. For unready items moved into the sprint multiple teams note that they expect them to be clarified further during development. Frequently given explanations for non-adherence are non-applicable criteria and ‘ad-hoc’ items created during the sprint for which the DoR is not adhered to.

In the expert opinion phase two experts recognised this form of adherence in coached and/or trained ASD teams. They state that in their view the DoR is can be used as a risk estimate for taking up items in the sprint, in which less ready items should be considered a higher risk.

Lastly, it is uncovered who is responsible for enforcing the DoR and when this occurs. For most teams there is a shared responsibility for checking items’ adherence to the DoR, to which the PO pays additional attention. Refinement sessions are noted in general as the moment of checking enforcement.

SQ5: What aspects should a framework for the adoption of Definition of Ready incorporate considering its usage in practice?

Resulting is the DA framework (Figure 7.2) adapted from the ATA framework by Gandomani and Nafchi (2015). The framework starts with rationalisation for DoR adoption since it only concerns this single practice. In this the team assesses which rationales apply to their situation, and if this warrants the adoption of a DoR. The focus is on potential business value, for which 10 rationales from practice are given as an example.

Following is the adaptation of- and to a DoR by the team, a continuous process striving towards adoption. In this several aspects from practice are suggested to consider; sources of inspiration, team contextual aspects, and practical aspects. Also, a list of example criteria frequently identified in practice is offered, where possible supported by the literature.

The adoption process is assessed by measuring general or rationale-specific DoR metrics corresponding to those chosen earlier. In a retrospective the results are discussed for which several guiding aspects are noted; the current adherence, threats to adherence, and the importance of criteria. The closing activity is to adjust the DoR where required and to continue adaptation.

In expert opinion validation the framework with explanation received overall positive feedback on expert understanding and familiarity, and expected practitioner understanding. Adjustments are primarily made to the framework’s model (Figure 8.1) to improve practitioner understanding while conserving its contents and intent. Examples are the addition of an example DoR and the formulation of aspects as general questions.

Chapter 10

Discussion

Concluding is the discussion of the results focusing on a variety of factors; the practical and scientific implications, threats to validity, additional findings, and possible future research to conduct.

10.1 Implications

The practical implication of the research is that ASD teams now possess a validated framework for adopting the DoR in case they do not have a (formalised) DoR, or to improve their existing DoR. Before such a framework did not exist, with teams having to rely on anecdotal handbooks for autonomous adoption, or required trainers to assist.

This DA framework provides practitioners with a comprehensive overview of what aspects other ASD teams consider, where possible supported by the literature. Additionally, it offers support through activities sequenced in a feedback loop for the continuous adaptation and improvement of the DoR.

A scientific implication is that the body of knowledge in the literature regarding the DoR's application in practice is extended. This includes DoR aspects that were under-reported, such as practitioner adherence and sources of inspiration. A remaining blind spot is what underlies the DoR's relatively low adoption rate compared to other ASD practices, which could not be explained.

Previously a knowledge gap existed about how practitioners apply the DoR, with conducted case studies only considering singular cases. This gap is closed in terms of methodology (multiple-case study and expert opinion) and results. Most examples from grey literature were likewise anecdotal, which is now extended with eight DoRs from practice.

Closing the gap paves the way for further research (Section 10.4) by offering clarity on why ASD teams adopt the DoR, how they apply the DoR using what criteria, and their ways of enforcing it. In this, the DA framework can also prove useful for conducting empirical investigation into DoR adoption.

10.2 Threats to Validity

For case study research Wohlin et al., (2012) describe four types of validity based on a classification by Yin (2017). The same types are noted for experiments based on Cook and Campbell (1979), except for conclusion validity which corresponds to reliability for case study research. Although not an experiment or case study, the types can be used to discuss MLR validity as well (Kolukisa Tarhan et al., 2020; Peltonen et al., 2021). In the following, the types are covered to reflect upon all four research phases.

10.2.1 Construct Validity

Construct validity concerns whether the studied operational measures correctly represent the researcher's intent and the aim of the research questions. A first threat according to Wohlin et al., (2012) is inadequately defining what is studied, which is countered by first exploring the DoR concept in depth in the MLR.

Additionally, case study participants were enquired whether they are familiar with the DoR, and if they can explain it to ensure a similar interpretation. Although Yin (2017) recommends letting participants review the resulting report to counter diverging interpretations of questions or answers, this was not performed for the case study and expert opinion phases.

Nevertheless, two other tactics suggested for increasing construct validity were applied. Both the MLR and case study have a full chain of evidence, providing traceability both ways from sources to findings. For the expert opinion phase, a reduced chain is present in the form of expert IDs and transcripts, since thematic coding was not applied.

A last tactic is to use multiple sources of evidence which goes for all four phases. In the MLR both grey- and scientific literature from Google Scholar and Google Search were indexed. For the case study, eight participants from independent SPOs were consulted. Subsequently, the variety of sources was triangulated in analysis, after which the framework was validated by three independent experts.

10.2.2 Internal Validity

Internal validity is of importance when causal relationships are investigated in research, to which influences without the researcher knowing are a threat. Although this validity type is of limited relevance to exploratory research it is still addressed for completeness.

A threat to internal validity for the MLR lies in source selection wherein sources may be overlooked, caused by the search queries' focus and applying in/exclusion criteria with bias. For mitigation broad structured queries are constructed based on preliminary findings from literature, which are improved based on trial-and-error.

This methodology is likewise leveraged in in/exclusion criteria construction, which attempt to filter as many possible irrelevant sources without overlooking relevant knowledge. In both cases a point of weakness is that no other researcher is consulted to review source selection which might prevent further bias.

Yin (2017) notes the internal validity threats in case studies mainly arise due to making inferences, and are likewise difficult to counter. Thus, the threats are mitigated in case study data analysis for which five techniques are suggested.

The main technique leveraged in this research is pattern matching, wherein the empirical patterns based on case study findings are compared with predicted patterns. When similarity is observed between the patterns this strengthens the internal validity, since this demonstrates that the correct influences have been considered.

In this research, the predicted, rivalling patterns are formed by the MLR results, pattern matched in the comparative analysis phase with case study results to construct the DA framework. Since there was overall a large degree of overlap the internal validity is safeguarded.

10.2.3 External Validity

For the MLR external validity covers whether the results can be generalised. Due to the broad scope and inclusion of grey literature a significant amount of the literature has been covered aiding the generalisability, nevertheless some threats exist.

Firstly, excluding blog posts and web pages on the basis that they provide little novel knowledge harms the generalisability of the research to these source types. Also, inaccessible sources or those in languages other than Dutch or English are excluded, which might harm the MLR's overall generalisability somewhat.

External validity in case studies pertains to the generalisability of results outside of the investigated cases. As noted in Section 3.2.6 the research only focused on SPOs from the Netherlands which harms the generalisability to those in other countries. Due to time and resource constraints, this threat was difficult to counter, and should thus be addressed in future research.

Another possible threat was the uneven representation of case characteristics in the case study. This has however only manifested partially in the team's ASD method (Scrum) and customer type (External). For Scrum, this could be explained by the current popularity of the method, with 87% of practitioners surveyed by the annual State of Agile report using it (State of Agile, 2022).

Thus, the results are expected to be generalisable to other Dutch SPOs applying Scrum in ASD for an external customer, independent of their age or product. This is further confirmed in expert opinion, with trainers recognising the perceived aspects in practice along with the posed framework steps.

An external validity threat to this validation phase again arises in participant selection, which focused only on certified experts in the role of trainer. In future work, this could be extended to other practitioners to improve the generalisability and test their understanding of the DA framework.

10.2.4 Reliability

The reliability of research concerns whether the research is repeatable by other researchers, such that the results remain the same. In this, the goal is to reduce biases and errors in the research that could be revealed in repetition. Wohlin et al., (2012) note as threats to reliability a lack of insight in interview protocols and coding.

To this, Yin (2017) adds transparency is the foremost method of mitigating reliability issues in research, achieved through making procedures explicit. This approach is followed throughout all research phases with pre-defined MLR, case study, and expert opinion protocols, and inclusion of in-between steps, case characteristics, and (coded) transcripts.

Furthermore, for the MLR and comparative analysis, the complete results in the form of tables are included, with the intermediate results of the MLR likewise reported in full (Chapter 4). As mentioned earlier, a chain of evidence is constructed for these results by applying IDs for all sources, which also holds true for expert opinion.

A reliability threat specific to the MLR is that it is performed by a singular researcher. This may firstly introduce bias in constructing the protocol, specifically the search queries and in/exclusion criteria, since no secondary party has validated them. Secondly, in data extraction a secondary researcher was not involved to review the process which could likewise result in bias.

10.3 Additional Findings

A first additional finding is the sources of inspiration ASD teams use for constructing a DoR. Although this was not explicitly part of the sub-questions it was enquired in the case study protocol for contextualisation. Notably, most teams indicate that they are inspired by team member experience within- and outside the team.

Similarly, teams were enquired whether there are DoR criteria that they find to be more important than others. All participants except one could indicate what criteria are deemed important above others. A wide variety of criteria was reported, with some criteria being deemed important by multiple teams, such as knowing the external dependencies.

Lastly, in expert opinion, the experts voiced their opinion on the DoR as a concept without explicitly being enquired about it. For two experts the DoR is to be applied flexibly as a risk estimate, meaning unready items can be included in a sprint but are to be considered riskier. The third expert mainly fears that teams over-document their DoR making the process of readying items cumbersome.

10.4 Future Research

Based on the overall research and the previous discussion some future work is considered. Firstly, the case study can be extended in terms of the number of participants to make the results more robust, which might lead to new insights for inclusion in the DA framework. The case study can also be extended by including SPOs from other countries since organisational ASD cultures could differ across the world.

Furthermore, the number of participants in the case study and expert opinion may require extension to test whether saturation is reached. As it stands it is uncertain whether no more novel information or viewpoints can be gathered from additional participants, for the reason the current numbers are the result of bounding effort (Wohlin et al., 2012).

Additionally, the influence of case characteristics noted in this research on the DoR could be investigated further. Although this could not be proven within the scope of the current research, it would be interesting to observe what impacts how a team applies the DoR. For this, the existing noted case characteristics could be used, or an extension can be undertaken where viable.

For the analysis activities there are some limitations which may be tackled in the future. Firstly, in analysing the case study criteria (Table 6.5) the interrelationships between DoR criteria are not considered in full. In this case further analysis may result in novel insights based on the case study findings. Likewise, the comparison phase focuses mainly on the perspective of the case study, while the opposing perspective (I.e. from the literature) may provide additional insights.

Another opportunity for future work lies in the further validation of the DA framework. Empirical research could be undertaken wherein practitioners other than trainers apply the framework in practice to validate whether it is comprehensible and usable for them. As indicated, this extension would also serve to increase the generalisability of the validation results.

A last point to address in future work is to research what the measurable effects of the DoR are, to objectively prove its benefits and drawbacks. To achieve this it should first be uncovered what metrics can be used to quantify the DoR's effects, for which the DA framework after empirical validation could form a starting point. Following, empirical research can be undertaken in which a DoR is introduced in multiple teams and the effects are measured over time.

In this line of research the relationship between the DoR and DoD could be investigated as well. As it stands the DoR has been researched in isolation, but the interaction with the DoD may be considered to comprehensively investigate its effects. For this the study by Kopczyńska et al., (2022) on the DoD's benefits may provide a starting point.

References

- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. In *2013 39th euromicro conference on software engineering and advanced applications* (pp. 9–16).
- Aschauer, B., Hruschka, P., Lauenroth, K., Meuten, M., & Rogers, G. (2019). *Handbook of RE@ agile according to the IREB standard*. International Requirements Engineering Board. Retrieved from http://www.compliance-technologies.com/DS/handbook_cpre_al_re@agile_v1.0.pdf
- Bambazek, P., Groher, I., & Seyff, N. (2022, June). Sustainability in agile software development: A survey study among practitioners. In *2022 international conference on ICT for sustainability (ICT4S)*. IEEE. doi: 10.1109/ICT4S55073.2022.00013
- Barjis, J. (n.d.). *Definition of ready definition of done*. <https://www.instituteengmgt.com/wp-content/uploads/DoRDoD.pdf>.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). *The agile manifesto*. Retrieved from <http://agilemanifesto.org>
- Brenner, R., & Wunder, S. (2015). Scaled agile framework: Presentation and real world example. In *2015 ieee eighth international conference on software testing, verification and validation workshops (icstw)* (pp. 1–2).
- Brereton, P., Kitchenham, B., Budgen, D., & Li, Z. (2008). Using a protocol template for case study planning. In *12th international conference on evaluation and assessment in software engineering (ease) 12* (pp. 1–8). doi: 10.14236/ewic/EASE2008.5
- Bufon, M. T., & Leal, A. G. (2019). Method for identification of waste in the process of software development in agile teams using lean and scrum. In *Knowledge management in organizations: 14th international conference, kmo 2019, zamora, spain, july 15–18, 2019, proceedings 14* (pp. 466–476). doi: 10.1007/978-3-030-21451-7_4
- Buglione, L., & Abran, A. (2013). Improving the user story agile technique using the invest criteria. In *2013 joint conference of the 23rd international workshop on software measurement and the 8th international conference on software process and product measurement* (pp. 49–53). doi: 10.1109/IWSM-Mensura.2013.18
- Cavoukian, A. (2010). *Privacy by design*. Information and Privacy Commissioner of Ontario. Retrieved from <https://privacysecurityacademy.com/wp-content/uploads/2020/08/PbD-Principles-and-Mapping.pdf>
- Ciric, D., Lalic, B., Gracanin, D., Tasic, N., Delic, M., & Medic, N. (2019). Agile vs. traditional approach in project management: Strategies, challenges and reasons to introduce agile. *Procedia Manufacturing*, 39, 1407–1414. doi: 10.1016/j.promfg.2020.01.314
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.
- Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings*. Houghton Mifflin.

- Dalton, J. (2018). *Great big agile: An os for agile leaders*. Apress.
- De Reeder, L. (2019). *Defining maturity of agile requirements engineering practices* (Master's thesis, Utrecht University). Retrieved from <https://studenttheses.uu.nl/handle/20.500.12932/34926>
- Diebold, P., Theobald, S., Wahl, J., & Rausch, Y. (2018). An agile transition starting with user stories, dod & dor. In *Proceedings of the 2018 international conference on software and system process* (pp. 147–156). doi: 10.1145/3202710.3203145
- Drury-Grogan, M. L., Conboy, K., & Acton, T. (2017). Examining decision characteristics & challenges for agile software development. *Journal of Systems and Software, 131*, 248–265. doi: 10.1016/j.jss.2017.06.003
- Fliefel, R. (2016). *Novel strategies in the management of exposed necrotic bone and bone defects in oral and maxillofacial surgery : management of osteomyelitis and medication-related osteonecrosis of the jaw "mronj"* (Unpublished doctoral dissertation). Ludwig-Maximilians-Universität.
- Forowicz, P. (2020). Approaches to business analysis in scrum at stepstone—case study. *Data-Centric Business and Applications: Towards Software Development (Volume 4)*, 211–223. doi: 10.1007/978-3-030-34706-2_11
- Gandomani, T. J., & Nafchi, M. Z. (2015). An empirically-developed framework for agile transition and adoption: A grounded theory approach. *Journal of Systems and Software, 107*, 204–219.
- Gandomani, T. J., Zulzalil, H., Abdul Ghani, A. A., Sultan, A. B., & Sharif, K. Y. (2014, 07). Exploring facilitators of transition and adoption to agile methods: A grounded theory study. *Journal of Software, 9*, 1166–1178. doi: 10.4304/jsw.9.7.1666-1678
- Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology, 106*, 101–121. doi: 10.1016/j.infsof.2018.09.006
- Goldstein, I. (2013). *Scrum shortcuts without cutting corners: agile tactics, tools, & tips*. Addison-Wesley.
- Heck, P., & Zaidman, A. (2014). A quality framework for agile requirements: a practitioner's perspective. *arXiv preprint arXiv:1406.4692*. doi: 10.48550/arXiv.1406.4692
- Heck, P., & Zaidman, A. (2015). Quality criteria for just-in-time requirements: just enough, just-in-time? In *2015 ieee workshop on just-in-time requirements engineering (jitre)* (pp. 1–4). doi: 10.1109/JITRE.2015.7330170
- Heck, P., & Zaidman, A. (2017). A framework for quality assessment of just-in-time requirements: the case of open source feature requests. *Requirements Engineering, 22*, 453–473. doi: 10.1007/s00766-016-0247-5
- Heck, P., & Zaidman, A. (2018). A systematic literature review on quality criteria for agile requirements specifications. *Software Quality Journal, 26*, 127–160. doi: 10.1007/s11219-016-9336-4
- Hoda, R., Salleh, N., & Grundy, J. (2018). The rise and evolution of agile software development. *IEEE software, 35*(5), 58–63. doi: 10.1109/MS.2018.290111318
- Hooles, A. (2017). How to contract successfully for agile software development. *Int'l. In-House Counsel J., 11*, 1.
- Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior, 51*, 915–929. doi: 10.1016/j.chb.2014.10.046
- Islam, G., & Storer, T. (2020). A case study of agile software development for safety-critical systems projects. *Reliability Engineering & System Safety, 200*, 106954.

- doi: 10.1016/j.res.2020.106954
- Jakobsen, C. R., & Sutherland, J. (2009). Scrum and cmmi : going from good to great. In *2009 agile conference* (pp. 333–337). doi: 10.1109/AGILE.2009.31
- Jørgensen, J. B., Christensen, H. L., Hansen, S. T., & Nyeng, B. B. (2022). Effective communication about software in a traditional industrial company: an experience report on development of a new measurement instrument. In *2022 ieee/acm international workshop on software-intensive business (iwsib)* (pp. 39–42).
- Jovanović, M., Mesquida, A.-L., Mas, A., & Lalić, B. (2017). Towards the development of a sequential framework for agile adoption. In *Software process improvement and capability determination: 17th international conference, spice 2017, palma de mallorca, spain, october 4–5, 2017, proceedings* (pp. 30–42).
- Jussilainen, N. (2013). *Metrics for agile requirements definition and management* (Master's thesis, HAAGA-HELIA). Retrieved from <https://www.theseus.fi/handle/10024/121608>
- Kajornboon, A. B. (2005). Using interviews as research instruments. *E-journal for Research Teachers*, 2(1), 1–9.
- Kerr, W. R., Gabrieli, F., & Moloney, E. (2018). Transformation at ing (a): Agile. *Harvard Business Review*, 1–19.
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Tech. Rep.). Keele University.
- Kiv, S., Heng, S., Wautelet, Y., & Kolp, M. (2018). Towards a goal-oriented framework for partial agile adoption. In *Software technologies: 12th international joint conference, icsoft 2017, madrid, spain, july 24–26, 2017, revised selected papers 12* (pp. 69–90). doi: 10.1007/978-3-319-93641-3_4
- Klotins, E., Unterkalmsteiner, M., Chatzipetrou, P., Gorschek, T., Prikladnicki, R., Tripathi, N., & Pompermaier, L. B. (2021). Use of agile practices in start-up companies. *e-Informatica Software Engineering Journal*, 15(1). doi: 10.37190/e-Inf210103
- Kneuper, R. (2018). *Software processes and life cycle models an introduction to modelling, using and managing agile, plan-driven and hybrid processes*. Springer.
- Kolukisa Tarhan, A., Garousi, V., Turetken, O., Söylemez, M., & Garossi, S. (2020). Maturity assessment and maturity models in health care: A multivocal literature review. *Digital health*, 6, 2055207620914772.
- Kopczyńska, S., Ochodek, M., Piechowiak, J., & Nawrocki, J. (2022). On the benefits and problems related to using definition of done—a survey study. *Journal of Systems and Software*, 193. doi: 10.1016/j.jss.2022.111479
- Kua, P. (2014). *Talking with tech leads*. Leanpub.
- Liskin, O., Pham, R., Kiesling, S., & Schneider, K. (2014). Why we need a granularity concept for user stories. In *Agile processes in software engineering and extreme programming: 15th international conference, xp 2014, rome, italy, may 26-30, 2014. proceedings 15* (pp. 110–125). doi: 10.1007/978-3-319-06862-6_8
- Lucassen, G., Dalpiaz, F., Van Der Werf, J. M. E., & Brinkkemper, S. (2015). Forging high-quality user stories: towards a discipline for agile requirements. In *2015 ieee 23rd international requirements engineering conference (re)* (pp. 126–135). doi: 10.1109/RE.2015.7320415
- Lucassen, G., Dalpiaz, F., Van der Werf, J. M. E., & Brinkkemper, S. (2016). The use and effectiveness of user stories in practice. In *Requirements engineering: Foundation for software quality: 22nd international working conference, refsq 2016, gothenburg, sweden, march 14-17, 2016, proceedings 22* (pp. 205–222). doi: 10.1007/978-3-319-30282-9_14

- Lucassen, G., Dalpiaz, F., Van der Werf, J. M. E., & Brinkkemper, S. (2017). Improving user story practice with the grimm method: A multiple case study in the software industry. In *Requirements engineering: Foundation for software quality: 23rd international working conference, refsq 2017, essen, germany, february 27–march 2, 2017, proceedings 23* (pp. 235–252). doi: 10.1007/978-3-319-54045-0_18
- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086–2095.
- McKenna, D. (2016). *The art of scrumhow scrum masters bind dev teams and unleash agility*. Springer. doi: 10.1007/978-1-4842-2277-5
- Nogués, A., Valladares, J., Nogués, A., & Valladares, J. (2017). Agile methodologies for bi projects. *Business Intelligence Tools for Small Companies: A Guide to Free and Low-Cost Solutions*, 29–56. doi: 10.1007/978-1-4842-2568-4_2
- Peltonen, S., Mezzalana, L., & Taibi, D. (2021). Motivations, benefits, and issues for adopting micro-frontends: a multivocal literature review. *Information and Software Technology*, 136, 106571.
- Pinto Lopez, R. (2020). *Study of user experience design of digital financial services* (Master's thesis, Aalto University). Retrieved from <https://aaltodoc.aalto.fi:443/handle/123456789/46168>
- Popović, T. (2015). Getting iso 9001 certified for software development using scrum and open source: A case study. *Tehnički vjesnik*, 22(6), 1633–1640.
- Power, K. (2014a). Definition of ready: An experience report from teams at cisco. In *Agile processes in software engineering and extreme programming: 15th international conference, xp 2014, rome, italy, may 26-30, 2014. proceedings 15* (pp. 312–319). doi: 10.1007/978-3-319-06862-6_25
- Power, K. (2014b). Social contracts, simple rules and self-organization: a perspective on agile development. In *Agile processes in software engineering and extreme programming: 15th international conference, xp 2014, rome, italy, may 26-30, 2014. proceedings 15* (pp. 277–284). doi: 10.1007/978-3-319-06862-6_21
- Quadri, S. M. K., & Farooq, S. U. (2010). Software testing—goals, principles, and limitations. *International Journal of Computer Applications*, 6(9), 1.
- Roberts, R. E. (2020). Qualitative interview questions: Guidance for novice researchers. *Qualitative Report*, 25(9).
- Robson, C. (2002). *Real world research: A resource for social scientists and practitioner-researchers*. Wiley-Blackwell.
- Rubin, K. S. (2012). *Essential scrum: A practical guide to the most popular agile process*. Addison-Wesley.
- Sinha, N., & Karim, R. (2015). Responsive designs in a snap. In *Proceedings of the 2015 10th joint meeting on foundations of software engineering* (pp. 544–554).
- State of Agile. (2022). *The 16th annual state of agile report* (Tech. Rep.). State of Agile.
- Sundermann, B. (2022). The agile swab: Testing for health and maturity.
- Sutherland, J., & Schwaber, K. (2020). *The scrum guide — the definitive guide to scrum: The rules of the game*. Retrieved from <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt. *Journal of Systems and Software*, 86(6), 1498–1516.
- Van Den Broek, R., Bonsangue, M. M., Chaudron, M., & Van Merode, H. (2014). Integrating testing into agile software development processes. In *2014 2nd international conference on model-driven engineering and software development (modelsward)* (pp. 561–574).

- Wadee, A. (2022). *The diffc model: Addressing documentation challenges in large-scale agile requirements engineering* (Master's thesis, Utrecht University). Retrieved from <https://studenttheses.uu.nl/handle/20.500.12932/42887>
- Wagenaar, G., Overbeek, S., Lucassen, G., Brinkkemper, S., & Schneider, K. (2018). Working software over comprehensive documentation—rationales of agile teams for artefacts usage. *Journal of Software Engineering Research and Development*, 6(1), 1–23. doi: 10.1186/s40411-018-0051-7
- Wake, B. (2003, Aug). *Invest in good stories, and smart tasks*. Retrieved from <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- Wakhisi, K., Roell, M., Okunieff, P., Smusz-Mengelkoch, N., Schneeberger, J., Commission, A. R., et al. (2022). *Phase 1 systems engineering management plan (semp)—arc its4us deployment project* (Governmental report). United States. Department of Transportation. Federal Highway Administration.
- Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements engineering*, 11, 102–107. doi: 10.1007/s00766-005-0021-6
- Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.
- Wirfs-Brock, R., & hvatum, L. B. (2018). Even more patterns for the magic backlog. In *Proceedings of the 25th conference on pattern languages of programs* (pp. 1–17).
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
- Yin, R. K. (2017). *Case study research and applications*. Sage.

Appendix A

Ethics Quick Scan

Response Summary:

Section 1. Research projects involving human participants

P1. Does your project involve human participants? This includes for example use of observation, (online) surveys, interviews, tests, focus groups, and workshops where human participants provide information or data to inform the research. If you are only using existing data sets or publicly available data (e.g. from Twitter, Reddit) without directly recruiting participants, please answer no.

- Yes

Recruitment

P2. Does your project involve participants younger than 18 years of age?

- No

P3. Does your project involve participants with learning or communication difficulties of a severity that may impact their ability to provide informed consent?

- No

P4. Is your project likely to involve participants engaging in illegal activities?

- No

P5. Does your project involve patients?

- No

P6. Does your project involve participants belonging to a vulnerable group, other than those listed above?

- No

P8. Does your project involve participants with whom you have, or are likely to have, a working or professional relationship: for instance, staff or students of the university, professional colleagues, or clients?

- Yes

P9. Is it made clear to potential participants that not participating will in no way impact them (e.g. it will not directly impact their grade in a class)?

- Yes

Informed consent

PC1. Do you have set procedures that you will use for obtaining informed consent from all participants, including (where appropriate) parental consent for children or consent from legally authorized representatives? (See suggestions for information sheets and consent forms on the website.)

- Yes

PC2. Will you tell participants that their participation is voluntary?

- Yes

PC3. Will you obtain explicit consent for participation?

- Yes

PC4. Will you obtain explicit consent for any sensor readings, eye tracking, photos, audio, and/or video recordings?

- Yes

PC5. Will you tell participants that they may withdraw from the research at any time and for any reason?

- Yes

PC6. Will you give potential participants time to consider participation?

- Yes

PC7. Will you provide participants with an opportunity to ask questions about the research before consenting to take part (e.g. by providing your contact details)?

- Yes

PC8. Does your project involve concealment or deliberate misleading of participants?

- No

Section 2. Data protection, handling, and storage The General Data Protection Regulation imposes several obligations for the use of personal data (defined as any information relating to an identified or identifiable living person) or including the use of personal data in research.

D1. Are you gathering or using personal data (defined as any information relating to an identified or identifiable living person)?

- Yes

High-risk data

DR1. Will you process personal data that would jeopardize the physical health or safety of individuals in the event of a personal data breach?

- No

DR2. Will you combine, compare, or match personal data obtained from multiple sources, in a way that exceeds the reasonable expectations of the people whose data it is?

- No

DR3. Will you use any personal data of children or vulnerable individuals for marketing, profiling, automated decision-making, or to offer online services to them?

- No

DR4. Will you profile individuals on a large scale?

- No

DR5. Will you systematically monitor individuals in a publicly accessible area on a large scale (or use the data of such monitoring)?

- No

DR6. Will you use special category personal data, criminal offense personal data, or other sensitive personal data on a large scale?

- No

DR7. Will you determine an individual's access to a product, service, opportunity, or benefit based on an automated decision or special category personal data?

- No

DR8. Will you systematically and extensively monitor or profile individuals, with significant effects on them?

- No

DR9. Will you use innovative technology to process sensitive personal data?

- No

Data minimization

DM1. Will you collect only personal data that is strictly necessary for the research?

- Yes

DM4. Will you anonymize the data wherever possible?

- Yes

DM5. Will you pseudonymize the data if you are not able to anonymize it, replacing personal details with an identifier, and keeping the key separate from the data set?

- Yes

Using collaborators or contractors that process personal data securely

DC1. Will any organization external to Utrecht University be involved in processing personal data (e.g. for transcription, data analysis, data storage)?

- Yes

DC2. Will this involve data that is not anonymized?

- No

International personal data transfers

DI1. Will any personal data be transferred to another country (including to research collaborators in a joint project)?

- Yes

DI2. Do all countries involved in this have an adequate data protection regime?

- Yes

Fair use of personal data to recruit participants

DF1. Is personal data used to recruit participants?

- Yes

DF2. Have potential participants provided this personal data voluntarily to be contacted about the research?

- Yes

DF3. If contact details have been provided by a third party, would participants expect their details to be passed on to the university and to be used in this way?

- Yes

DF4. If contact details have been gathered for a purpose other than research, would participants expect their details to be used in this way?

- Yes

Participants' data rights and privacy information

DP1. Will participants be provided with privacy information? (Recommended is to use as part of the information sheet: For details of our legal basis for using personal data and the rights you have over your data please see the University's privacy information at www.uu.nl/en/organisation/privacy.)

- Yes

DP2. Will participants be aware of what their data is being used for?

- Yes

DP3. Can participants request that their personal data be deleted?

- Yes

DP4. Can participants request that their personal data be rectified (in case it is incorrect)?

- Yes

DP5. Can participants request access to their personal data?

- Yes

DP6. Can participants request that personal data processing is restricted?

- Yes

DP7. Will participants be subjected to automated decision-making based on their personal data with an impact on them beyond the research study to which they consented?

- No

DP8. Will participants be aware of how long their data is being kept for, who it is being shared with, and any safeguards that apply in case of international sharing?

- Yes

DP9. If data is provided by a third party, are people whose data is in the data set provided with (1) the privacy information and (2) what categories of data you will use?

- Yes

Using data that you have not gathered directly from participants

DE1. Will you use any personal data that you have not gathered directly from participants (such as data from an existing data set, data gathered for you by a third party, data scraped from the internet)?

- No

Secure data storage

DS1. Will any data be stored (temporarily or permanently) anywhere other than on password-protected University authorized computers or servers?

- No

DS4. Excluding (1) any international data transfers mentioned above and (2) any sharing of data with collaborators and contractors, will any personal data be stored, collected, or accessed from outside the EU?

- No

Section 3. Research that may cause harm Research may cause harm to participants, researchers, the university, or society. This includes when technology has dual-use, and you investigate an innocent use, but your results could be used by others in a harmful way. If you are unsure regarding possible harm to the university or society, please discuss your concerns with the Research Support Office.

H1. Does your project give rise to a realistic risk to the national security of any country?

- No

H2. Does your project give rise to a realistic risk of aiding human rights abuses in any country?

- No

H3. Does your project (and its data) give rise to a realistic risk of damaging the University's reputation? (E.g., bad press coverage, public protest.)

- No

H4. Does your project (and in particular its data) give rise to an increased risk of attack (cyber- or otherwise) against the University? (E.g., from pressure groups.)

- No

H5. Is the data likely to contain material that is indecent, offensive, defamatory, threatening, discriminatory, or extremist?

- No

H6. Does your project give rise to a realistic risk of harm to the researchers?

- No

H7. Is there a realistic risk of any participant experiencing physical or psychological harm or discomfort?

- No

H8. Is there a realistic risk of any participant experiencing a detriment to their interests as a result of participation?

- No

H9. Is there a realistic risk of other types of negative externalities?

- No

Section 4. Conflicts of interest

C1. Is there any potential conflict of interest (e.g. between research funder and researchers or participants and researchers) that may potentially affect the research outcome or the dissemination of research findings?

- No

C2. Is there a direct hierarchical relationship between researchers and participants?

- No

Section 5. Your information.

This last section collects data about you and your project so that we can register that you completed the Ethics and Privacy Quick Scan, sent you (and your supervisor/course

coordinator) a summary of what you filled out, and follow up where a fuller ethics review and/or privacy assessment is needed. For details of our legal basis for using personal data and the rights you have over your data please see the University's privacy information. Please see the guidance on the ICS Ethics and Privacy website on what happens on submission.

Z0. Which is your main department? - Information and Computing Science

Z1. Your full name:

Tommy Versteeg

Z2. Your email address:

anonymized

Z3. In what context will you conduct this research? - As a student for my master thesis, supervised by:

Gerard Wagenaar

Z5. Master programme for which you are doing the thesis

- Business Informatics

Z6. Email of the course coordinator or supervisor (so that we can inform them that you filled this out and provide them with a summary):

Anonymized

Z7. Email of the moderator (as provided by the coordinator of your thesis project):

Anonymized

Z8. Title of the research project/study for which you filled out this Quick Scan:

What is Ready in a Definition of Ready

Z9. Summary of what you intend to investigate and how you will investigate this (200 words max):

The main problem the project will attend to is the lack of scientific substantiation for criteria used in the 'Definition of Ready' (DoR) concept from the Agile project management philosophy. Focused scientific attempts for defining and describing the DoR concept are scarce or non-existent, which is the root source of the perceived problem. There are scientific works about the DoR, Agile, Scrum or related concepts which include a definition of the DoR concept. However, the definitions are not comprehensive and do not cover the criteria for determining requirement 'readiness'.

The project's aim is to provide a scientific substantiation for possible criteria to be used in a Definition of Ready (DoR), based on scientific and grey literature through literature review, and industry knowledge from interviews. This substantiation should be accompanied by a definition of the DoR, and what demarcates the DoR from other Agile/Scrum concepts. Eventually this result can be used in science as a common definition which reduces the effort for starting research about DoR, and to avoid communicative conflicts due to diverging definitions being used. For the industries using DoR in Agile and/or

Scrum the substantiation may serve as a template on how to use and construct a DoR in projects, offering a scientific supplement to the grey literature about the philosophy and method

Z10. In case you encountered warnings in the survey, does supervisor already have ethical approval for a research line that fully covers your project?

- No

Scoring - Privacy: 0 - Ethics: 0

Appendix B

Literature and IDs

ID	Reference
S	<i>Scientific Literature</i>
S1	Bambazek, P., Groher, I., & Seyff, N. (2022). Sustainability in Agile Software Development: A Survey Study among Practitioners. 2022 International Conference on ICT for Sustainability (ICT4S), 13–23. https://doi.org/10.1109/ICT4S55073.2022.00013
S2	Behutiye, W., Karhapää, P., López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Rodríguez, P., Franch, X., & Oivo, M. (2020). Management of quality requirements in agile and rapid software development: A systematic mapping study. <i>Information and Software Technology</i> , 123, 106225. https://doi.org/10.1016/j.infsof.2019.106225
S3	Buglione, L., & Abran, A. (2013). Improving the User Story Agile Technique Using the INVEST Criteria. 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, 49–53. https://doi.org/10.1109/IWSM-Mensura.2013.18
S4	Diebold, P., Theobald, S., Wahl, J., & Rausch, Y. (2018). An Agile transition starting with user stories, DoD & DoR. <i>Proceedings of the 2018 International Conference on Software and System Process</i> , 147–156. https://doi.org/10.1145/3202710.3203145
S5	Heck, P., & Zaidman, A. (2014). A Quality Framework for Agile Requirements: A Practitioner’s Perspective (arXiv:1406.4692). arXiv. https://doi.org/10.48550/arXiv.1406.4692
S6	Heck, P., & Zaidman, A. (2015, August). Quality criteria for just-in-time requirements: just enough, just-in-time. In <i>2015 IEEE Workshop on Just-In-Time Requirements Engineering (JITRE)</i> (pp. 1-4). IEEE. https://doi.org/10.1109/JITRE.2015.7330170
S7	Heck, P., & Zaidman, A. (2018). A systematic literature review on quality criteria for agile requirements specifications. <i>Software Quality Journal</i> , 26(1), 127–160. https://doi.org/10.1007/s11219-016-9336-4
S8	Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. <i>Information and Software Technology</i> , 53(5), 521–534. https://doi.org/10.1016/j.infsof.2010.10.009
S9	Jørgensen, J. B., Christensen, H. L., Hansen, S. T., & Nyeng, B. B. (2022). Effective Communication About Software in a Traditional Industrial Company: An Experience Report on Development of a new Measurement Instrument. 2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB), 39–42.

S10	Kiv, S., Heng, S., Wautelet, Y., & Kolp, M. (2018). Towards a Goal-Oriented Framework for Partial Agile Adoption. In E. Cabello, J. Cardoso, L. A. Maciaszek, & M. van Sinderen (Eds.), <i>Software Technologies</i> (pp. 69–90). Springer International Publishing. https://doi.org/10.1007/978-3-319-93641-3_4
S11	Kneuper, R. (2019). Integrating Data Protection into the Software Life Cycle. In X. Franch, T. Männistö, & S. Martínez-Fernández (Eds.), <i>Product-Focused Software Process Improvement</i> (pp. 417–432). Springer International Publishing. https://doi.org/10.1007/978-3-030-35333-9_30
S12	Liskin, O., Pham, R., Kiesling, S., & Schneider, K. (2014). Why We Need a Granularity Concept for User Stories. In G. Cantone & M. Marchesi (Eds.), <i>Agile Processes in Software Engineering and Extreme Programming</i> (pp. 110–125). Springer International Publishing. https://doi.org/10.1007/978-3-319-06862-6_8
S13	Lucassen, G., Dalpiaz, F., Van der Werf, J. M., & Brinkkemper, S. (2015). Forging High-Quality User Stories: Towards a Discipline for Agile Requirements. 2015 IEEE 23rd international requirements engineering conference (RE) (pp. 126-135). https://doi.org/10.1109/RE.2015.7320415
S14	Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2017). Improving User Story Practice with the Grimm Method: A Multiple Case Study in the Software Industry. In P. Grünbacher & A. Perini (Eds.), <i>Requirements Engineering: Foundation for Software Quality</i> (Vol. 10153, pp. 235–252). Springer International Publishing. https://doi.org/10.1007/978-3-319-54045-0_18
S15	Lucassen, G., Dalpiaz, F., Werf, J. M. E. M. van der, & Brinkkemper, S. (2016). The Use and Effectiveness of User Stories in Practice. In M. Daneva & O. Pastor (Eds.), <i>Requirements Engineering: Foundation for Software Quality</i> (Vol. 9619, pp. 205–222). Springer International Publishing. https://doi.org/10.1007/978-3-319-30282-9_14
S16	Popović, T. (2015). GETTING ISO 9001 CERTIFIED FOR SOFTWARE DEVELOPMENT USING SCRUM AND OPEN SOURCE TOOLS: A CASE STUDY. <i>Tehnicki Vjesnik/Technical Gazette</i> , 22(6).
S17	Power, K. (2014a). Definition of Ready: An Experience Report from Teams at Cisco. In G. Cantone & M. Marchesi (Eds.), <i>Agile Processes in Software Engineering and Extreme Programming</i> (pp. 312–319). Springer International Publishing. https://doi.org/10.1007/978-3-319-06862-6_25
S18	Power, K. (2014b). Social Contracts, Simple Rules and Self-organization: A Perspective on Agile Development. In G. Cantone & M. Marchesi (Eds.), <i>Agile Processes in Software Engineering and Extreme Programming</i> (pp. 277–284). Springer International Publishing. https://doi.org/10.1007/978-3-319-06862-6_21
S19	Prokhorenko, S. (2012). Skiing and Boxing: Coaching Product and Enterprise Teams. 2012 Agile Conference, 191–196. https://doi.org/10.1109/Agile.2012.30
S20	Rizkiyah, K., Nisyak, A. K., & Raharjo, T. (2020). Agile-Based Requirement Challenges of Government Outsourcing Project: A Case Study. 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE) (pp. 267-273). IEEE.
S21	Santos, N. A. de L. F. F. dos. (2020). An agile process for modeling logical architectures: Demonstration cases from large-scale software projects [Doctoral Dissertation, Universidade do Minho]. RepositoriUM. http://repositorium.sdum.uminho.pt/
S22	Theobald, S., & Diebold, P. (2017). Beneficial and Harmful Agile Practices for Product Quality. In M. Felderer, D. Méndez Fernández, B. Turhan, M. Kalinowski, F. Sarro, & D. Winkler (Eds.), <i>Product-Focused Software Process Improvement</i> (Vol. 10611, pp. 586–593). Springer International Publishing. https://doi.org/10.1007/978-3-319-69926-4_48
S23	van den Broek, R., Bonsangue, M. M., Chaudron, M., & van Merode, H. (2014). Integrating testing into Agile software development processes. 2014 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 561–574.

S24	Wagenaar, G., Overbeek, S., Lucassen, G., Brinkkemper, S., & Schneider, K. (2018). Working software over comprehensive documentation – Rationales of agile teams for artefacts usage. <i>Journal of Software Engineering Research and Development</i> , 6(1), 7. https://doi.org/10.1186/s40411-018-0051-7
S25	Warg, F., Blom, H., Borg, J., & Johansson, R. (2019). Continuous Deployment for Dependable Systems with Continuous Assurance Cases. 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 318–325. https://doi.org/10.1109/ISSREW.2019.00091
S26	Wirfs-Brock, R., & Hvatum, L. B. (2020). Even more patterns for the magic backlog. <i>Proceedings of the 25th Conference on Pattern Languages of Programs</i> , 1–17.
G	<i>Grey Literature</i>
G1	Aschauer, B., Hruschka, P., Lauenroth, K., Meuten, M., & Rogers, G. (2019). Handbook of RE@ Agile According to the IREB Standard. International Requirements Engineering Board.
G2	Barjis, J. (n.d.). Definition of Ready & Definition of Done. https://www.instituteengmgt.com/wp-content/uploads/DoRDoD.pdf
G3	Chandrasekara, C., & Yapa, S. (2018). Effective Team Management with VSTS and TFS. Apress. https://doi.org/10.1007/978-1-4842-3558-4
G4	da Silva, A. A. C. (2016). Agile Forward: A simple toolkit for process improvement [Master's Thesis, Universidade do Porto]. Repositório Aberto. https://repositorio-aberto.up.pt/handle/10216/112244
G5	Dalton, J. (2018). Great Big Agile: An OS for Agile Leaders. Apress.
G6	Dimovska, L. (2018). Codification of a Novel Agile method: Fast Forward [Master's Thesis, Universiteit Utrecht]. Student Theses Repository. https://studenttheses.uu.nl/handle/20.500.12932/30660
G7	dos Santos Martin, J. M. A. (2020). Developing software as a medical device with an Agile methodology [Master's Thesis, Universidade do Porto]. Repositório Aberto. https://repositorio-aberto.up.pt/handle/10216/132653
G8	Forowicz, P. (2020). Approaches to Business Analysis in Scrum at StepStone—Case Study. In A. Poniszewska-Marañda, N. Kryvinska, S. Jarzabek, & L. Madeyski (Eds.), <i>Data-Centric Business and Applications: Towards Software Development (Volume 4)</i> (pp. 211–223). Springer International Publishing. https://doi.org/10.1007/978-3-030-34706-2_11
G9	Gillo Nilsson, C., & Karlsson, D. (2015). Implementing Agile project methods in globally distributed teams [Master's Thesis, Karlstad Business School]. DiVa. http://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-35321
G10	Goldstein, I. (2013). Scrum Shortcuts without Cutting Corners: Agile Tactics, Tools, & Tips. Addison-Wesley.
G11	Hooles, A. (2017). How to Contract Successfully for Agile Software Development. <i>International In-House Counsel Journal</i> , 11(45), 1–8.
G12	Hutten, E. (2015). Organizational design and agile software development: The effects of modularity on effective communication and collaboration [Master's Thesis, University of Twente]. UT Student Theses. http://essay.utwente.nl/66960/
G13	Jussilainen, N. (2013). Metrics for agile requirements definition and management [Master's Thesis, HAAGA-HELIA]. Theseus. https://www.theseus.fi/handle/10024/121608
G14	Kneuper, R. (2018). Software processes and life cycle models: An introduction to modelling, using and managing Agile, plan-driven and hybrid processes. Springer.
G15	Leeuwen, M. M. van. (2015). Agile Scaling @ Topicus: Scaling scrum with help of Agile Scaling frameworks at Topicus Finance [Master's Thesis, University of Twente]. UT Student Theses. http://essay.utwente.nl/67970/

G16	McKenna, D. (2016). <i>The Art of Scrum</i> . Apress. https://doi.org/10.1007/978-1-4842-2277-5
G17	Nogués, A., & Valladares, J. (2017). <i>Business Intelligence Tools for Small Companies: A Guide to Free and Low-Cost Solutions</i> . Apress. https://doi.org/10.1007/978-1-4842-2568-4_2
G18	Pinto Lopez, R. (2020). <i>Study of User Experience Design of Digital Financial Services</i> [Master's Thesis, Aalto University]. Aaltodoc. https://aaltodoc.aalto.fi:443/handle/123456789/46168
G19	Reeder, L. de. (2019). <i>Defining maturity of agile requirements engineering practices</i> [Master's Thesis, Universiteit Utrecht]. Student Theses Repository. https://studenttheses.uu.nl/handle/20.500.12932/34926
G20	Rubin, K. S. (2012). <i>Essential Scrum: A Practical Guide to the Most Popular Agile Process</i> . Addison-Wesley.
G21	Santos, R. A. S. dos. (2016). <i>Towards an SDLC for software development projects involving distributed systems</i> [Master's Thesis, PUCRS]. TEDE. https://tede2.pucrs.br/tede2/handle/tede/6830
G22	Shafagatova, A. (2015). <i>The importance and presence of agile principles in agile software development</i> [Master's Thesis, Universiteit Gent]. Scriptiebank. https://www.scriptiebank.be/scriptie/2015/importance-and-presence-agile-principles-agile-software-development
G23	Sundermann, B. (2022). <i>The Agile Swab: Testing for Health and Maturity</i> . https://pmworldlibrary.net/wp-content/uploads/2022/10/pmwj122-Oct2022-Sundermann-the-agile-swab.pdf
G24	Vanhanen, A. (2019). <i>Documentation of the product during a software development process: The scaled agile framework</i> [Master's Thesis, LUT]. Lutpub. https://lutpub.lut.fi/handle/10024/159594
G25	Wadee, A. (2022). <i>The DIFFC model: Addressing documentation challenges in Large-Scale Agile Requirements Engineering</i> [Master's Thesis, Universiteit Utrecht]. https://studenttheses.uu.nl/handle/20.500.12932/42887
G26	Wakhisi, K., Roell, M., Okunieff, P., Smusz-Mengelkoch, N., Schneeberger, J. D., Atlanta Regional Commission, Go Systems and Solutions, & Kimley-Horn & Associates. (2022). <i>Phase 1 Systems Engineering Management Plan (SEMP)—ARC ITS4US Deployment Project (FHWA-JPO-21-915)</i> . https://rosap.ntl.bts.gov/view/dot/62483
R	<i>Removed</i>
R1	Abdou, T., Kamthan, P., & Shahmir, N. (2014). <i>User Stories for Agile Business: INVEST, Carefully!</i> . <i>Social Media and Publicity</i> , 141.
R2	ESTAYNO, M., & MELES, J. (n.d.). <i>The Challenge of Being a Product Owner POs Responsibilities in Agile Projects</i> .
R3	Khanh, N. T., Daengdej, J., & Arifin, H. H. (2017). <i>Human stories: A new written technique in agile software requirements</i> . <i>Proceedings of the 6th International Conference on Software and Computer Applications - ICSCA '17</i> , 15–22. https://doi.org/10.1145/3056662.3056680
R4	Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2016). <i>Improving agile requirements: The Quality User Story framework and tool</i> . <i>Requirements Engineering</i> , 21(3), 383–403. https://doi.org/10.1007/s00766-016-0250-x

R5	Lucassen, G.G., Brinkkemper, Sjaak, Dalpiaz, Fabiano, University Utrecht, Sub Software Production, & Software Production. (2017). Understanding User Stories: Computational Linguistics in Agile Requirements Engineering [Utrecht University]. https://dspace.library.uu.nl/handle/1874/356784
R6	Patel, C., & Ramachandran, M. (2009). Story card based agile software development. <i>International Journal of Hybrid Information Technology</i> , 2(2), 125-140.
R7	Power, K. (2014). Metrics for understanding flow. In <i>Agile Software Development Conference (Agile 2014)</i> .
R8	Turek, M., & Werewka, J. (2016). Multi-project Scrum methodology for projects using software product lines. In <i>Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology–ISAT 2015–Part III</i> (pp. 189-199). Springer International Publishing.
R9	Araújo, M. J. B. (n.d.). Improvement of performance indicators after Scrum re-implementation in software development teams [Master’s Thesis, Universidade do Minho]. RepositoriUM. https://repositorium.sdum.uminho.pt/handle/1822/64918
R10	Beek, J. J. W. van der. (2020). A Method for Combining Agile, Internal Control, and Stakeholders’ Needs [Master’s Thesis, University of Twente]. UT Student Theses. http://essay.utwente.nl/81057/
R11	Laitila, T. (2017). User-centered development and maintenance method for software teams [Master’s Thesis, University of Jyväskylä]. JYX Digital Repository. https://jyx.jyu.fi/handle/123456789/56647
R12	Rabon, B. M. (2014). <i>Scrum for the Rest of Us!: A Braintrust Field Guide</i> . Dog Ear Publishing, LLC.
R13	Toomey, E. (2021). The project management challenges, benefits, risks and limitations of adopting agile methodologies for a multiphase ERP program [Master’s Thesis, University College Cork]. CORA. https://cora.ucc.ie/handle/10468/12388
R14	Waerdt, T. van de. (2013). Evaluating and Improving a Scrum Project Management Tool Aimed at Distributed Teams [Master Thesis, Universiteit Utrecht]. Student Theses Repository. https://studenttheses.uu.nl/handle/20.500.12932/15021
R15	Winter, B. (2015). <i>Agile Performance Improvement</i> . Apress. https://doi.org/10.1007/978-1-4842-0892-2_6

TABLE B.1: MLR result IDs and references.

Appendix C

Literature Quality Scores

ID	Crit. 1	Crit. 1.1	Crit. 2	Crit. 3	Crit. 4	Crit. 5	Crit. 6	Total	Score
S1	2	1	1	1	2	2	2	11	0,92
S2	2	1	1	2	2	2	2	12	1,00
S3	2	1	1	1	1	2	1	9	0,75
S4	1	1	1	1	2	2	2	10	0,83
S5	2	1	1	2	2	2	2	12	1,00
S6	2	1	1	2	2	2	2	12	1,00
S7	2	1	1	2	2	2	2	12	1,00
S8	2	1	1	2	2	2	2	12	1,00
S9	1	1	1	1	2	1	1	8	0,67
S10	2	1	1	1	2	2	2	11	0,92
S11	2	1	1	2	2	2	1	11	0,92
S12	2	1	1	1	2	2	2	11	0,92
S13	2	1	1	2	2	2	2	12	1,00
S14	2	1	1	2	2	2	2	12	1,00
S15	2	1	1	2	2	2	2	12	1,00
S16	1	1	1	1	0	1	1	6	0,50
S17	1	1	1	1	1	1	0	6	0,50
S18	2	1	1	1	1	1	1	8	0,67
S19	1	1	1	1	1	1	1	7	0,58
S20	2	1	0	2	2	1	2	10	0,83
S21	2	1	1	2	2	2	2	12	1,00
S22	2	1	1	2	2	1	2	11	0,92
S23	2	1	1	2	2	2	2	12	1,00
S24	2	1	1	2	2	2	2	12	1,00
S25	2	1	1	1	2	2	1	10	0,83
S26	1	1	1	0	1	2	0	6	0,50
R1	1	0	0	0	0	0	0	1	0,08
R2	1	0	0	1	0	1	1	4	0,33
R4	2	1	1	2	2	2	2	12	1,00
R5	2	1	1	2	2	2	2	12	1,00
R6	1	1	0	1	0	1	1	5	0,42
R7	2	1	0	0	1	0	1	5	0,42
R8	1	1	0	1	0	1	1	5	0,42

TABLE C.1: Scientific literature quality scores.

ID	Crit. 1.1	Crit. 1.2	Crit. 2	Crit. 3.1	Crit. 3.2	Crit. 4	Crit. 5.1	Crit 5.2	Total	Score
G1	1	1	1	1	1	1	1	1	8	1,00
G2	1	1	0	1	0	0	1	1	5	0,63
G4	1	0	1	0	1	1	0	1	5	0,63
G5	1	1	0	1	1	1	0	1	6	0,75
G6	1	0	1	1	1	1	0	1	6	0,75
G7	1	0	1	0	1	1	0	1	5	0,63
G8	1	0	1	1	1	1	0	1	6	0,75
G9	1	0	1	1	1	1	0	1	6	0,75
G10	1	1	1	1	1	1	0	1	7	0,88
G11	1	0	0	1	0	1	1	1	5	0,63
G12	1	0	1	1	1	1	1	1	7	0,88
G13	1	0	1	1	1	1	1	1	7	0,88
G14	1	1	1	1	1	1	1	1	8	1,00
G15	1	0	1	1	1	1	0	1	6	0,75
G16	1	1	0	1	0	1	0	1	5	0,63
G17	1	0	1	1	0	1	0	1	5	0,63
G18	1	0	1	1	1	1	0	1	6	0,75
G19	1	0	1	1	1	1	1	1	7	0,88
G20	1	1	1	1	1	1	0	1	7	0,88
G21	1	0	1	0	1	1	0	1	5	0,63
G22	1	0	1	1	1	1	1	1	7	0,88
G23	1	1	1	0	0	1	0	1	5	0,63
G24	1	0	1	1	1	1	0	1	6	0,75
G25	1	0	1	1	1	1	1	1	7	0,88
G26	1	0	1	1	1	1	0	1	6	0,75
R9	1	0	1	1	1	1	0	0	5	0,63
R10	1	0	1	1	1	1	0	0	5	0,63
R11	1	0	1	0	1	1	0	0	4	0,50
R12	1	1	0	0	0	1	0	0	3	0,38
R13	1	0	1	1	1	1	0	0	5	0,63
R14	1	0	1	1	1	1	0	0	5	0,63
R15	1	1	0	1	0	1	0	0	4	0,50

TABLE C.2: Grey literature quality scores.

Appendix D

Data Extraction Forms

ID	Research Type	SQ1	SQ2	SQ3	SQ4
S1	Evaluation Research	"The Definition of Ready (DoR) is an agreement defined by the Scrum team that describes the criteria a backlog item must satisfy before it can be added to a sprint"		Respondents agreed for all the sustainability dimensions (Economic, Environmental, Technical, Social, Individual) that the DoR should define that impacts of a backlog item should be assessed before addition to a sprint. Likewise, sustainability impacts should not only be assessed before development but also that there should be a commitment for such an assessment."	
S2	Systematic Mapping Study			"Similarly, product owners can adopt practices such as specifying QRs in Definition of ready of user stories, to manage QRs better."	To what extent should the criteria be met? A threshold (or tolerance level) for each of the six criteria should be introduced. This would help to reduce subjectivity in the use of this set of evaluation criteria. The INVEST grid in Appendix A can be used as a template when the customer and provider meet to evaluate a US.
S3	Proposal of Solution	Before an evaluation session of a set of US for new sprints, the customer and provider can set acceptance thresholds for INVEST. These thresholds can be called a profile. If thresholds are not met after the session, the remaining distance in terms of effort constitutes the improvement axis. On this axis work can be started to determine what should be done and how much effort may be needed to meet the thresholds.	"... to reduce the estimation error as much as possible. This error is often a consequence of incomplete requirements, requirements provided late (scope creep), or not having the right level of detail for expressing what is known in the Functional Size Measurement (FSM) community as the elementary process concept..."	"Until INVEST criteria have been met in a US, that US should not be transferred to a Sprint."	
S4	Evaluation Research	".. Some criteria already existed in the minds of team members. The two practices (DoR & DoD) helped to make these individual assumptions and criteria explicit and made the team reflect on additional criteria that need to be taken into account."	The quantitative and qualitative documentation was improved. The DoD and DoR also reduced communication overhead since the need for further inquiries was significantly reduced.	"... (1) the role related to a story must be specified and that (2) the US should follow the US template "As a *role*, I want *what*, so that *reason*", which is the most commonly used template. ... (3) the concrete instance of the DoD ..."	All criteria should be present.
S5	Proposal of Solution	For user stories incompleteness and incorrectness is inherent, compensated by extensive informal communication between stakeholders represented by the product owner and team members. Teams using user stories should decide which criteria to apply to their practice in a Definition of Ready. Some criteria are applicable to fully documented requirements but not to face-to-face discussion of user story comments.	"This question assumes that correctly specified agile requirements contribute to a higher final software product quality, an assumption that has been considered to hold for traditional requirements ..." "All participants agree that agile requirements should fulfill certain quality criteria. This helps the understanding within the team and is important for traceability or accountability towards the rest of the organization"	"VC1 Completeness 1.1 Basic Elements 1.2 Required Elements 1.3 Optional Elements VC2 Uniformity 2.1 Use of Tool 2.2 Necessity of Comments 2.3 Follow Template 2.4 Uniform Models VC3 Consistency & Correctness 3.1 No Contradiction 3.2 No Contradicting Comments 3.3 Correct Language 3.4 Specify Problem 3.5 INVEST 3.6 Correct Summary 3.7 Atomic 3.8 Glossary 3.9 No Duplicates 3.10 Navigable Links" ... "process-oriented criteria like "have been approved by the product owner"	"It is the reviewer or author of the requirement that can still decide how serious a violation of one of the checks is in the given situation. The above example with the missing 'Rationale' for obvious requirements could for instance be solved by marking this check as 'N/A' in the Excel checklist. This would avoid the automatic 'LOW' score"

<p>S6</p> <p>Validation Research</p>	<p>The list of quality criteria can be used as a checklist to determine when a user story is ready. JIT RE practices solve the initial vagueness of requirements by face-to-face communication or prototyping, but many situations require fully documented requirements which increases the importance of correctness.</p>	<p>"As said, there is some evidence that early verification of requirements quality contributes to a higher software product quality!" "According to the practitioners, good quality agile requirements help the understanding within the team and are important for traceability or accountability towards the rest of the organization."</p>	<p>This paper adds to the 2014 version: "C Creation-time. This criterion should hold at soon as the requirement or the requirement part is created. (1.1, 1.2, VC2, 3.3, 3.4, 3.6, 3.7, 3.8, 3.9, 3.10). "J Just-in-time. This criterion does not necessarily have to hold when the requirement (part) is created. However, it should hold at a later moment, just-in-time for a certain step in the development process. (1.1, 1.2, 1.3, 3.1, 3.2, 3.5).</p>	
<p>S7</p> <p>Systematic Literature Review</p>	<p>It is advised that a checklist of quality criteria is established based on the presented list, for a project, team or environment. For each criterion it should be decided at what point in time it should hold. This checklist can be seen as the DoR.</p>	<p>"The goal of INVEST is to divide the system to be developed in small deliverables that can be delivered independently, one of the key principles of agile development."</p>	<ul style="list-style-type: none"> • Priority • Unique Identifier • Non-functional requirements • Acceptance criteria • INVEST • Understandability • Forward traceable <p>(These were discussed in text, see figure 2 for the full list)</p>	
<p>S8</p> <p>Evaluation Research</p>		<p>"Using the "definition of ready" for user stories [43] forced customers to provide detailed requirements with clear business drivers. The definition of ready complemented the existing Scrum definition of done [13]."</p>		
<p>S9</p> <p>Personal Experience Paper</p>	<p>The DoR was agreed on and revised several times by the PO and the team. The PO should sign off on the DoR.</p>		<p>"The definition of ready required that key stakeholders, including a product manager representing the business perspective, should sign off on the detailed requirements of features that were included in a sprint as ready to work on for developers"</p>	
<p>S10</p> <p>Proposal of Solution</p>		<p>Using a Definition of Ready contributes to the agile feature "customer satisfaction" as part of the principle that the utmost priority is to satisfy the customer through early and continuous delivery of software. This is part of the Agile value "Working software over comprehensive documentation".</p>		

<p>S11 Proposal of Solution</p>	<p>Non-functional requirements and constraints regarding data protection need to be integrated into the DoR. An example of this: "</p> <ul style="list-style-type: none"> • Have the relevant roles (data subject, controller, processor) been identified? • Have the personal data been identified which are to be processed? • Are only those personal processed that are genuinely needed for the functionality to be developed? • Has the legal basis for processing been identified? What exactly is included, what is not?" 	<p>Expected Implementation Duration (EID) for measuring granularity, which corresponds to 'small' from INVEST.</p> <p>Only focuses on information derivable from User Story text.</p> <ul style="list-style-type: none"> • Syntactic • Atomic • Minimal • Well-formed <p>Semantic</p> <ul style="list-style-type: none"> • Conflict-free • Conceptually sound • Problem-oriented • Unambiguous <p>Pragmatic</p> <ul style="list-style-type: none"> • Complete • Explicit dependencies • Full sentence • Independent • Scalable • Uniform • Unique"
<p>S12 Proposal of Solution</p>	<p>Using the treatment's 5 criteria (of 13 total) decreased number of user story quality defects with 43,1%, however participants did not perceive meaningful change in user story quality. The participants did perceive improved communication frequency and effectiveness. However, measurements of perception and software process metrics indicated no significant differences.</p>	<p>INVEST (n=43) or self-defined quality guidelines (n=60) were reported. There were also respondents which did not use quality guidelines (n=72).</p>
<p>S13 Evaluation research</p>	<p>INVEST can be a checklist but interviews do not use it as such. The six characteristics are internalized by the team, and when violated a team member mentions this. For self-defined quality guidelines interviews do not have a well-defined, list for user stories. They rely on experience of the writer and peer review. Interviewees that do not use guidelines attribute this to unawareness.</p>	<p>INVEST (n=43) or self-defined quality guidelines (n=60) were reported. There were also respondents which did not use quality guidelines (n=72).</p>
<p>S14 Evaluation Research</p>	<p>INVEST can be a checklist but interviews do not use it as such. The six characteristics are internalized by the team, and when violated a team member mentions this. For self-defined quality guidelines interviews do not have a well-defined, list for user stories. They rely on experience of the writer and peer review. Interviewees that do not use guidelines attribute this to unawareness.</p>	<p>INVEST (n=43) or self-defined quality guidelines (n=60) were reported. There were also respondents which did not use quality guidelines (n=72).</p>
<p>S15 Evaluation Research</p>	<p>INVEST can be a checklist but interviews do not use it as such. The six characteristics are internalized by the team, and when violated a team member mentions this. For self-defined quality guidelines interviews do not have a well-defined, list for user stories. They rely on experience of the writer and peer review. Interviewees that do not use guidelines attribute this to unawareness.</p>	<p>INVEST (n=43) or self-defined quality guidelines (n=60) were reported. There were also respondents which did not use quality guidelines (n=72).</p>

S16	Evaluation Research	"for each project, the Scrum team should have a clear definition of which features (i.e. user stories) are ready to be implemented. Typically, the whole team collaboratively has to agree that the requirement is clear." "DoR applies no matter what process the team is using. Teams have a discussion about Definition of Ready, making sure they understand each point, and the responsibilities this creates for their team. This discussion generally takes place at the start of a new release, or when a new team is formed. " .. "definition of ready is a checklist"	Adhering to ISO 9001 requirements Design Review results (7.3.4) and Design Change Reviews (7.3.7).	<ul style="list-style-type: none"> • "User Story defined • User Story Acceptance Criteria defined • User Story dependencies identified • User Story sized by Delivery Team • User Experience artifacts are Done, and reviewed by the Team • Architecture criteria (performance, security, etc.) identified, where appropriate • Person who will accept the User Story is identified • Team has reviewed the User Story • Team knows what it will mean to demo the User Story" 	"Being "ready" does not mean the user story or feature must be 100% defined; it needs to be "ready enough" so that the team is confident they can successfully deliver the user story, or that everyone has a common understanding of the risks. "
S17	Personal Experience Paper	responsibilities this creates for their team. This discussion generally takes place at the start of a new release, or when a new team is formed. " .. "definition of ready is a checklist"	Not being ready causes estimation problems, waste, working on the wrong things, and rework. The biggest gains of deploying a DoR are reduced churn and delays and a smoother workflow.	<ul style="list-style-type: none"> • "User Story defined • User Story Acceptance Criteria defined • User Story dependencies identified • User Story sized by Delivery Team • User Experience artifacts are Done, and reviewed by the Team • Architecture criteria (performance, security, etc.) identified, where appropriate • Person who will accept the User Story is identified • Team has reviewed the User Story • Team knows what it will mean to demo the User Story" 	"Being "ready" does not mean the user story or feature must be 100% defined; it needs to be "ready enough" so that the team is confident they can successfully deliver the user story, or that everyone has a common understanding of the risks. "
S18	Philosophical Paper	The DoR is a set of Simple Rules adopted by Agile teams to remember what needs to be done on an item before sprint inclusion. Product owners are responsible for making sure work items meet the DoR. This social contract focuses on the product owner to meet the contract's terms, and the team agreeing to support.			
S19	Personal Experience Paper	"Brainstorming sessions to explicitly fix the "definition of done" and "definition of ready" contracts between team and PO"			
S20	Evaluation Research		Establishing a DoR helps solve a lack of understanding of the requirements at the planning meeting, which results in changes in the ongoing sprint.		
S21	Evaluation research			<ul style="list-style-type: none"> • "User Story defined • User Story Acceptance Criteria defined • User Story dependencies identified • User Story sized by Delivery Team • User Experience artifacts are Done and reviewed by the Team • Architecture criteria (performance, security, etc.) identified, where appropriate • Person who will accept the User Story is identified • Team has reviewed the User Story • Team knows what it will mean to demo the User Story" 	"Being "ready" does not mean the user story or feature must be 100% defined; it needs to be "ready enough" so that the team is confident they can successfully deliver the user story, or that everyone has a common understanding of the risks. "

S22	Evaluation Research		<p>"There is a shared understanding of the quality demands, supported by practices such as Coding Styleguides, Definition of Ready and Definition of Done."</p>		
S23	Evaluation Research		<p>The DoR can be used in Product Backlog Refinement sessions, by executing them in a parallel fashion to a running sprint, upcoming sprint planning sessions are supposed to be relieved from lengthy discussions and unready or ambiguous user stories (waste).</p>		
S24	Evaluation Research		<p>The DoR is used as an artifact as the result of adopting Agile Software Development, in this study called the 'agility rationale'.</p>		
S25	Proposal of Solution			<p>"Hence, for Definition of Ready, there should be an understanding of the impact of the proposed change not only for the implementation, but also for the assurance cases."</p>	
S26	Proposal of Solution	<p>Develop and share a core set of definitions across the project or program, so that attribute values of backlog items are consistent and can be used for decision making and reporting. Keep the definitions in a shared space easily accessible by all team members, like a project wiki. Make sure the definitions are actively used because team members contributed to them, agree with them, and know where they are.</p>			

ID	SQ1	SQ2	SQ3	SQ4
G1	<p>The product owner is held responsible for bringing user stories into the definition of ready, but there is no mention of discussing a DoR with the team.</p> <p>"... it is crucial that teams themselves create their DoR and DoD, own them, and adhere to them. In creating these definitions, two things are crucial: consulting Agile (SAFE) principles and ensuring team's full agreement."</p>	<p>"Starting the implementation with too many open questions may reduce development speed considerably and cause delays against forecasts."</p> <p>"... reinforce Transparency, assure Built-In Quality, and set the right expectations for the work items to be planned, developed, and completed during an Agile product development."</p>	<p>The INVEST criteria and Cohn's User Story Template are prescribed.</p> <p>An example DoR: "</p> <ul style="list-style-type: none"> Has Acceptance Criteria that can be tested objectively Estimated by the entire development team Socialized with the entire team Has the right size that can be completed within a Sprint, preferably 2-3 days or not bigger than [certain] story points Complies with the INVEST Model (Independent, Negotiable, Valuable, Estimable, Small, Testable) and has no external dependencies Uploaded/ created in the team's Agile tool/ environment Written in the user voice format " 	<p>"The Pareto principle can be used in assessing when this point has been reached: requirements must not be defined 100% perfectly, but well enough to address the team's key questions and clear enough allowing for the implementation effort to be estimated"</p> <p>"The items these definitions are applied to must have binary outcome: Done or Not-Done. Avoid the temptation of an "almost done", "kind of done", or "99% done"</p>
G2		<p>Not having or meeting a DoR can cause an imprecise set of backlog items, needing clarification before sizing or adding a DoD.</p>	<ul style="list-style-type: none"> Backlog item is clear and understood by team Backlog item has acceptance criteria that is defined, so team agrees on DoD Backlog item should be testable Backlog item is feasible to implement within a sprint 	
G3		<p>"If work items are not precisely understood, development effort (and time) tend to balloon, which in turn cause the Sprint to fall."</p>	<ul style="list-style-type: none"> The work to be done has value It has been estimated It is testable and the tests have been defined The pieces are sized appropriately" <p>Apply the INVEST criteria.</p>	
G4	<p>A team should have a common DoR in the form of a checklist, which is explicitly visible to all team members. The DoR should be actively used in sprint planning, and its effectiveness should be discussed at the sprint retrospective leading to adjustments.</p>	<p>The DoR can be used to identify defects in the story before working on it, reducing defects earlier when they are less costly to address. Also, high-quality user stories increase the likelihood of successfully implementing it.</p>		<p>"Avoid rules that require full compliance to DOR at all times, allowing for exceptions based on specific attributes of the user story"</p>
G5		<p>"To prevent duplication of work or work in completely different direction, a user story must meet a set of criteria before it is ready for iteration in the next sprint."</p>	<ol style="list-style-type: none"> Immediately actionable by the team Approved by the product owner Have value Estimable by the team by size 	
G6				

<p>The DoR is used together with the creation of acceptance criteria in a pre-refinement step by product owner, quality member and tech lead in this medical software development methodology.</p>			
<p>G7 At StepStone the Definition of Ready is not strictly enforced, relying on informal communication and agreement between PO and developers. Later in the case study the DoR was adjusted and enforced.</p>			
<p>G8 Respondents from globally distributed teams: "Their notion of working towards a shared goal is all about their shared understanding of the definition of Ready."</p>			
<p>G9</p>			
<p>G10</p>	<ul style="list-style-type: none"> • "The user story has been estimated • There is a clearly defined set of acceptance criteria • The user story has been uniquely ordered within the product backlog • An appropriate and applicable level of extended documentation exists (such as mock-ups and wireframes if they are necessary) • Based on the initial estimate, it should appear that the user story will comfortably fit within a single sprint" 	<ul style="list-style-type: none"> • "written in this format "As a <kind of user> I want <feature> so that <benefit>"; • includes the associated Definition of Done criteria; • written in a way that can be understood by the Sprint Team; • includes both functional and non-functional requirements; • has an associated level of priority; • has unique identifiers to maintain traceability and mapping of closely related User Stories; • has been sized and allocated a number of Story Points by the Sprint Team; and • it is agreed how the User Story will be tested and demonstrated." 	<p>The DoR can help in identifying a fixed price for the development of the user story. This allows to use traditional contractual rights and mechanisms to allocate risk to the supplier, holding them to development.</p>
<p>G11</p>		<p>Increase customer involvement in development by having a standard (DoR) the customer requirements should adhere to before the developers will start working on it.</p>	<p>Initially the DoR was used by a modular team focussing on getting items ready, before being handled by another team working on getting them done. Merging the teams improved morale and stabilized results.</p>
<p>G12</p>		<p>"They describe the decrease in the amount of costly back-and-forth discussion or rework and possibility for the team to "pushback" as the benefits of using the "Definition of Ready."</p>	
<p>G13</p>	<ul style="list-style-type: none"> • A user story exists, pointing out the actor, describing the feature and the purpose for it • The formulation of the story is general enough so that the team has the flexibility to deliver it in increments • The story is small enough to fit inside a sprint • Each story has its own unique priority in relation to other stories in the product backlog, to ensure that most important things are implemented first • The story has a description of how it can be tested or demoed, giving the reader a good sense of when story is completed • The story has been estimated" 		

<p>"... there is no separate quality assurance function that checks that these gates may be passed but the person responsible for the work also checks that it has been completed and the criteria are satisfied. Social pressure within the development team ensures that this is done properly, rather than some external mechanism."</p> <p>G14</p>	<p>"In order to manage the Sprint planning efficiently, all features which are likely to be added to the department Sprint must meet the Definition of Ready"</p>	<p>An example DoR: must be broken down into sufficiently small tasks so that it can be implemented within one sprint, or it must have an adequate number of acceptance criteria."</p>	<p>"... it</p>
<p>"...the DoR is likely to change and evolve through the release, but there should be only one that is used for all user stories. If the DoR is properly defined, everybody on the team should be comfortable with it when it is pulled into any Sprint."</p> <p>G15</p>	<p>INVEST, and an example DoR: <ul style="list-style-type: none"> • The title should be meaningful, short and to the point • The description should be understandable by everyone on the team • The PO's acceptance criteria that must be met • The dependencies and/or prerequisites required to start or complete story • The story has been sized • The story is owned by someone on the team • The story is demonstrable to stakeholders at review • The story contains requirements for QA testing • The story contains requirements for documentation </p>	<p>An example DoR: <ul style="list-style-type: none"> • Can the item be finished within sprint that we are going to start? • Has this item been estimated by the development team? • Is this item adding value by itself? Or it must be done with other items on the list? </p>	<p>Should be determined according to team needs fitting their context, INVEST is mentioned as possible DoR criteria.</p>
<p>At Vopera the DoR is important. It is defined by the product owner, and becomes an issue of communication between the client-side represented PO and the team. After retrospective meetings the initial DoR can be tweaked and adapted to the needs of the team and client.</p> <p>G16</p>	<p>Teams should establish a definition of ready as a checklist of criteria, however the two teams in the case study did not have one.</p> <p>G17</p>	<p>"A strong definition of ready will substantially improve the Scrum team's chance of successfully meeting its sprint goal."</p> <p>G18</p>	<p>Should be determined according to team needs fitting their context, INVEST is mentioned as possible DoR criteria.</p> <p>G19</p>
<p>"...there is no separate quality assurance function that checks that these gates may be passed but the person responsible for the work also checks that it has been completed and the criteria are satisfied. Social pressure within the development team ensures that this is done properly, rather than some external mechanism."</p> <p>G20</p>	<p>"A strong definition of ready will substantially improve the Scrum team's chance of successfully meeting its sprint goal."</p>	<p>Should be determined according to team needs fitting their context, INVEST is mentioned as possible DoR criteria.</p>	<p> <ul style="list-style-type: none"> • Business value clearly articulated • Details understood by the team • Dependencies identified • No blocking external dependencies • Team staffed appropriately to complete item • Item is estimated • Item is small enough • Acceptance criteria • Performance criteria • Demonstrable </p>

	<p>For distributed systems development: "Definition Of Ready (DOR) should take in consideration the system distribution characteristics of the project in question. For example, it could include "complete data contract being available" and/or "data sample being available".</p>		
G21		<p>The DoR is perceived by practitioners as important for the Agile principles: "Deliver frequently with shorter timescale" and "Customer satisfaction/outcome".</p>	
G22		<p>"Stories committed to in the Sprint are not completed, work is done on features outside the scope of the sprint, and additional downstream work is needed during testing."</p>	<ul style="list-style-type: none"> • How to test feature, including acceptance test • Description of how to demo story • Effort estimation • Dependencies to other stories need to be included • External dependencies should be resolved before inclusion in sprint • Subtasks • Delivery dates
G23		<p>" determining the "definition of ready", which defines the readiness of the requirement, would help with the approval of the features."</p>	
G24		<p>The rationale behind the DoR is Agility (comes with adopting an ASD method) and governance (self-imposed by team or external/organisation).</p>	
G25	<p>In interviews it was found that teams have a DoR but it is not often used due to most team members not being aware of it, or due to time constraints.</p>		<ul style="list-style-type: none"> • "User Story is defined and complete • User Acceptance Criteria are defined • User Story dependencies are defined • User Story is sized correctly by the Development Team • The person who will implement the User Story is identified; the team members who will conduct the technical and hygiene reviews are identified. • The team has a good idea what it will mean to Demo the story"
G26			

Appendix E

Interview Question Protocol

Introduction

- *Thank the interviewee for their time and the opportunity to interview them.*
- *Introduce myself and ask who the interviewee is.*
- I am researching the Definition of Ready (DoR) in Agile software development for my master's thesis, this interview is part of a multiple-case study.
- I mentioned the informed consent in our previous contact, do you agree if you have not already given your permission?
- For clarification, the interview will be recorded and transcribed anonymously, meaning no personal information or information identifying the organisation is in the transcript. The recordings are deleted after completion of the master's thesis research project.
- *Start the recording.*
- I am very interested in your perspective on the DoR; how you are using it in your team and what you think of it?

Introductory Questions

Number	Question
IN1	What is your role within the team?
IN2	What ASD method is used in the team?
IN3	What type of software does the team develop?
IN4	Does your team work on one product, multiple but established products or varying products?
IN5	Does your team develop software for in-house usage or for clients?
IN6	Is your team part of a larger project? I.e. is there cross-team development?

TABLE E.1: Introductory interview questions.

Main Questions

Number	Question	SQ1	SQ2	SQ3	SQ4
M1	Do you use criteria for determining when a backlog item is ready for sprint planning, and if so do others in the team as well?	X		X	
1.1	Are you aware of the DoR concept and can you explain it?	X			
1.2	Are these criteria the same for everyone in the team, formalised in a DoR?	X			
1.3	In what format is the DoR used in the team?	X			
M2	Is the DoR an agreement between people in the team and/or with external stakeholders?	X			
2.1	How is backlog item adherence to the DoR enforced?	X			
M3	Why do you use the DoR?		X		
3.1	What is the reason the DoR was adopted?		X		
3.2	What consequences of using a DoR do you perceive?		X		
M4	What criteria are used in the DoR?			X	
4.1	How are the criteria used in the DoR decided?	X		X	
M5	To what degree is the DoR adhered to when an item is moved into the sprint?				X
5.1	To what extent should the DoR be adhered to?				X
5.2	Does each criterion carry the same weight?			X	X
5.3	Have there been consequences resulting from a lack of DoR adherence?		X		X

TABLE E.2: Main interview questions.

Conclusion

- Is there additional information you might add which has not been discussed?
- Do you have documentation on the DoR which you could share with me after the interview?
- This interview closes my case study on your team, at the end of my research I am willing to share the full master's thesis or a summary of the findings, would you be interested in this?
- *Thank the interviewee and close the interview.*

Appendix F

Case Study Data

The case study interview transcripts and thematic coding are made available as an NVIVO file (Phase2_CaseStudy.nvp, 10.6mb) through an Open Science Framework project:

https://osf.io/5n8ad/?view_only=513ba66a8d2e469da67341235cbf64fe

Appendix G

Expert Opinion Protocol

Introduction

- *Thank the interviewee for their time and the opportunity to interview them.*
- *Introduce myself and ask who the interviewee is.*
- I am researching the Definition of Ready (DoR) in Agile development for my master's thesis, this interview is part of the expert validation of the framework I developed.
- I mentioned the informed consent in our previous contact, do you agree if you have not already given your permission?
- For clarification, the interview will be recorded and transcribed anonymously, meaning no personal information or information identifying the organisation is in the transcript. The recordings are deleted after completion of the master's thesis research project.
- *Start the recording.*
- How many years of experience do you have with ASD and the DoR?
- Do you have certifications in the ASD domain?
- I am very interested in your perspective on the DoR implementation framework, and your expectations of it.

Main Questions

- *Introduce the framework, focus on the activities, sequencing, and on the aspects noted in the lower half.*
- Is the framework clear to you as an expert?
- Do you recognize the noted activities, sequencing and aspects in practice?
- Do you expect practitioners to understand the framework?
- Is the framework sufficient to enable a well-informed adaptation and adoption of a DoR?

Conclusion

- Is there additional information you might add which has not been discussed?
- At the end of my research I am willing to share the full master's thesis or a summary of the findings, would you be interested in this?
- *Thank the interviewee and close the interview.*

Appendix H

Expert Opinion Data

The three Expert Opinion interview transcripts are made available as .pdf files through an Open Science Framework project:

https://osf.io/5n8ad/?view_only=513ba66a8d2e469da67341235cbf64fe