What is Ready in a DoR? Rationales, Responsibility & Rules in using a Definition of Ready

Mark van Riesen[1], Gerard Wagenaar[1,*]

[1] *Utrecht University, Heidelberglaan 8, Utrecht, 3584 CS, the Netherlands*

Abstract

In agile software development, a Definition of Ready (DoR) is used to indicate conditions a Product Backlog Item (PBI) has to meet before accepting it into a Sprint Backlog. This research aims (1) to identify those conditions, (2) to investigate why Scrum teams do or do not use a DoR, and (3) to identify which (Scrum) role is responsible for drafting a DoR. Research questions are answered by a literature review and interviews with Scrum team members. Results show that conditions vary, but common elements are include INVEST criteria, clearly defining, and prioritizing a PBI. Reasons to use a DoR include a positive impact on workflow, like an efficient process or increased quality of software. Teams might not use a DoR, because of overhead of writing it or encouraging a less 'agile way of thinking'. Both Product Owner and Scrum Master can be involved in defining and maintaining a DoR, and it is considered good practice to involve all team members. Our findings also reveal that use of a DoR is not intuitive per se.

Keywords

Agile Software Development, Definition of Ready, DoR, Scrum, User story, INVEST criteria

1. Introduction

In Scrum software development, or agile development in general, a Definition of Ready (DoR) may be used to specify conditions a Product Backlog Item (PBI) has to meet before accepting it into a Sprint Backlog [4]. The original Scrum guide does not mention the concept [12] and few initiatives have been undertaken to consider its role and importance. According to one of few studies, ready means that a PBI has to be sufficiently prepared, so that a team can start working on it [11]. While a Scrum development team is responsible for meeting a Definition of Done, the Product Owner is responsible for PBIs meeting the DoR [10]. The DoR thus serves as a checklist for a Product Owner before declaring an item is ready to be pulled in by the team. A user story or feature does not have to be 100% defined, but it has to be ready enough for the team to successfully deliver it.

In: D. Mendez, A. Moreira, J. Horkoff, T. Weyer, M. Daneva, M. Unterkalmsteiner, S. Bühne, J. Hehn, B. Penzenstadler, N. Condori-Fernández, O. Dieste, R. Guizzardi, K. M. Habibullah, A. Perini, A. Susi, S. Abualhaija, C. Arora, D. Dell'Anna, A. Ferrari, S. Ghanavati, F. Dalpiaz, J. Steghöfer, A. Rachmann, J. Gulden, A. Müller, M. Beck, D. Birkmeier, A. Herrmann, P. Mennig, K. Schneider. Joint Proceedings of REFSQ-2024 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2024. Winterthur, Switzerland, April 8, 2024.

[*] Corresponding author.

m.vanriesen2@students.uu.nl (Mark van Riesen); g.wagenaar@uu.nl (Gerard Wagenaar)

0000-0001-6982-7282 (G. Wagenaar)

Little is known about DoR usage in practice. This research aims to understand a DoR's role in Scrum software development. We formulate our main research question as: **What constitutes a DoR in Scrum software development?**

To answer the main question, we ask three sub-questions: (SQ1) What are reasons for Scrum teams (not) to use a DoR?, (SQ2) Which (Scrum) role is responsible for a DoR?, and (SQ3) What conditions must a PBI meet to be accepted into a Sprint?

2. Research method

Our research involved literature review and interviews. The former part is inspired by a systematic literature review [7], but we did not use its full systematic approach. We selected several data sources (Google (Scholar) & Scopus) as we were already aware that previous work on the DoR would be scarce and we would need to access grey literature. We defined our search string as: ("*definition of ready*") AND ("*agile*" OR "*scrum*" OR "*user story*"). Inclusion/exclusion criteria included language (English or Dutch) and applicability to DoR. Quality assessment involved type of publication, authorship, citations, and publication date.

In the empirical part, we conducted a small 'case study' with software engineering teams. We are hesitant to use the label 'case study', because our context would at best be partially real-life nor do we use multiple sources [16]. Engineering teams were bachelor computer science students working on Scrum projects as their final thesis. They might be familiar with the concept 'DoR', but were certainly not trained. They are familiar with Scrum software development, its roles, artifacts, and events. This setting provided an unique opportunity to investigate a DoR's 'intuitiveness'. We interviewed Product Owners and Scrum Masters from three teams, where semi-structured interviews explored themes with regard to (1) their involvement with the Product Backlog, (2) familiarity with a DoR, and (3) their use of readiness of PBIs for a Sprint.

3. Findings

Two studies stand out in literature when it comes to DoR usage. The most elaborate example is a case study at Cisco [10]. This research describes using a DoR for (1) user stories, (2) sprints, and (3) releases. The choice to use a baseline (DoR) was made because of interdependencies across teams, and challenges and impediments to flow of the organization. At Cisco, work items were considered ready when stories, acceptance criteria, and dependencies were defined, a story was sized, user experience artifacts were done, architecture criteria (performance, security) were identified, the person who would accept the user story was identified, the team had reviewed the user story, and the team knew what it would mean to demo the user story.

The second one describes the introduction of a DoR as part of an agile transition [5]. The DoR demands primarily that the role related to a story must be specified and the user story should follow the user story template The quantitative and qualitative documentation was reported to improve by specifying documentation needs in the DoR.

3.1. Literature: Reasons (not) to use a DoR

Literature mentions reasons (not) to use a DoR, although its use is often recommended. Positive effects include (1) acting as a filter, stabilizing team working environment, and preventing issues, e.g. wasted time or delays [10,13], and (2) reducing defects, improving documentation, speeding up delivery, and increasing high-quality user stories [4,5,8].

Opponents of a DoR raise challenges in implementation and concerns about potential conflicts with agile principles [1,9], suggesting that it may lead to Waterfall thinking [3,10]. Some argue that details can be best sorted out during a Sprint [3]. Therefore, it is crucial to strike a balance between providing sufficient information without determining implementation details [8].

3.2. Literature: DoR responsibility

Attribution of responsibility for a DoR is not uniform. Some assign it to the Product Owner [10,11], while others suggests that the task belongs to the Scrum Master [6]. During backlog refinement, the team must work with the Product Owner to help them get the stories in actionable shape [13].

3.3. Literature: Backlog item conditions

Features do not have to be 100% defined, but sufficiently enough for successful delivery or to establish a common understanding of risks [3,10]. Criteria such as the "As a role, I want function, so that reason" template [5], and the INVEST criteria (Independent, Negotiable, Valuable, Estimated, Sized appropriately, and Testable) [1,2,3,4,8,13] can be used for screening stories entering a Sprint.

Ready stories must be clear, concise, and actionable, with the latter the most important [13]. Furthermore, a user story should clearly state the resulting business value, allowing the Product Owner to prioritize. Finally, ready stories must meet the INVEST criteria, and be absent of external dependencies, in the sense that there is nothing beyond the teams control that must be done first in order to complete the user story.

A checklist for readiness may also be used with elements such as clarity, testability, feasibility, defined acceptance criteria and dependencies, and being sized appropriately by the development team [15]. The IBPM Story Check was transformed into another DoR checklist: a story in the 'role, what, reason' template is prioritized and acceptance criteria are defined [14].

Overall, literature suggests a DoR varies based on project needs but commonly includes clarity, prioritization, size, and team understanding. The INVEST criteria are frequently mentioned, and a DoR should be tailored and regularly reviewed.

3.4. Definition of Ready in practice

Members from three teams were interviewed. None of them was directly familiar with the concept 'DoR'. Only one team stated that it used a set of constraints that a PBI should adhere to before it could be used in a Sprint, but it did this implicitly.

Three interviewees shared an example of a PBI, which they thought was "*ready*" to be used in a Sprint. Examples show that PBIs should have clear value for the client, be easily understandable, could easily be picked up by another member, clearly state technical details, not be too large, and be independent of other features.

None of the teams used a DoR, at least not explicitly, so responsibility for a DoR was not assigned. However, for all teams, both Product Owner and Scrum Master were involved with Product Backlog management. Team A's Scrum Master believed the advantages of a DoR outweigh its disadvantages by, for example, increasing the team's ability to work independently on assigned tasks. Tasks of Product Owner and Scrum Master always included creating, prioritizing, and assigning PBIs, which makes both roles eligible to determine whether a PBI is ready for a Sprint.

All teams used MoSCoW to assign priorities to their PBIs. Independence between PBIs was also deemed important to avoid delays and dependencies in the workflow. Respondents highlighted the importance of clear specifications, and stated that well-defined items prevent misunderstandings also contributing to the final product meeting requirements. Additionally, breaking down a PBI into smaller, more manageable tasks is also recommended, as it makes it easier for team members to understand progress.

4. Conclusions

What are reasons for Scrum teams (not) to use a DoR? Literature states that using a DoR positively impacts project workflow, such as increasing speed, promoting efficiency, and improving product quality. However, potential downsides exist: overhead and a 'Waterfall' approach. To use or not to use a DoR depends on situational factors of a team. From practice, we learn the applying a DoR is not that intuitive. Out of three teams, only one used it more or less, implicitly.

Which (Scrum) role is responsible for a DoR? We find in practice both Product Owners and Scrum Masters being involved in Product Backlog management. This is more or less in line with literature, which also emphasized involvement of all team members in the process around a DoR.

What conditions must a PBI meet to be accepted into a Sprint? Literature provides us with many answers. The criteria vary widely but commonly include INVEST criteria. Additional criteria may be project-specific. By lack of an explicit DoR within the teams, this could not be verified.

What constitutes a DoR in Scrum software development? The answer is situational, dependent on team, project, and organization. One thing is for sure, a DoR is not the first thing junior software engineers think of when doing Scrum software development.

5. Limitations & future research

Using interviews partly makes research difficult to replicate. All Scrum teams were part of the same population. Therefore, generalizability of this research has to be considered relatively low. Furthermore, mainly because of time limitations, we did not have access to more experienced Scrum teams of, for example, software companies. We would very much like to interview more experience Scrum team members, not only to improve the reliability of our research, but also to dive further into the intuitiveness of a DoR. Do such teams use it earlier or better, or are they sometimes simply obliged to? Finally, further research could focus on examining the effects of using a DoR within teams. Do the benefits, from literature, occur in practice?

We end by stating that DoR usage in agile software development is an under-researched area. Our limited study contributes some elements, but more research is called for.

References

[1] W.-J. Ageling, The rise and fall of the Definition of Ready in Scrum. URL: medium.com/serious-scrum/the-rise-and-fall-of-the-definition-of-ready-in-scrum-2407c6f1c455 (2021).

[2] R. Austin, Definition of ready. URL: www.leadingagile.com/2015/07/definition-of-ready.

[3] N. Butler, Definition of ready and definition of done: What's the difference? URL: www.boost.co.nz/blog/2022/06/definition-ready-definition-done (2022).

[4] J. Dalton, Definition of ready, in: Great Big Agile, pages 163–164. Springer (2019).

[5] P. Diebold, S. Theobald, J. Wahl, Y. Rausch, An agile transition starting with user stories, DoD & DoR in: Proceedings of the 2018 International Conference on Software and System Process, pages 147–156 (2018).

[6] A. Hooles, How to contract successfully for agile software development, in: Int'l. In-House Counsel J., 11:1 (2017).

[7] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering (2007).

[8] C. Kleczewski, Wat is de definition of ready? URL: agilescrumgroup.nl/wat-is-definition-of-ready.

[9] I. Mitchell, Walking through a definition of ready. URL: www.scrum.org/resources/blog/walking-through-definition-ready (2017).

[10] K. Power, Definition of ready: An experience report from teams at cisco, in: International Conference on Agile Software Development, pages 312–319. Springer (2014).

[11] K. Power, Social contracts, simple rules and self-organization: a perspective on agile development, in: International Conference on Agile Software Development, pages 277–284. Springer (2014).

[12] K. Schwaber, J. Sutherland, The scrum guide. Scrum Alliance, 21(19):1 (2011).

[13] J. Sutherland, Definition of ready. URL: www.scruminc.com/definition-of-ready.

[14] C. Thiemich, F. Puhlmann, An agile bpm project methodology, in: Business process management, pages 291–306. Springer (2013).

[15] B. Will, Definition of ready (dor) vs. definition of done (dod). URL: www.linkedin.com/pulse/definition-ready-dor-vs-done-dod-brian-will (2017).

[16] C. Wohlin, Case Study Research in Software Engineering - It is a Case, and it is a Study, but is it a Case Study?, Information and Software Technology, 133, 106514 (2021).