

# Effects of Daily Scrum Meeting on Software Quality – Open Questions

Marija Katic

*Independent Researcher, Serbia*

## Abstract

In recent years, the agile method Scrum has been not only widely researched but also widely practiced in software development. At the same time, not all of its effects on software quality have been clearly understood neither by researchers nor by practitioners. Moreover, some advocate for its benefits, while others claim that they struggle with the application of Scrum thus negatively affecting the software quality. In other words, there is no agreement on the effects of Scrum on software quality. A Daily Scrum meeting or a Daily Stand-Up meeting is one of the five events in Scrum that teams need to take part in, and it happens on a daily basis. The Daily Scrum meeting is widely negatively perceived, especially by senior team members. On the other hand, paradoxically, it is widely accepted in the practice of software engineering. This paper discusses the positive and the negative sides of the Daily Scrum meeting as perceived in the practice of software engineering and proposes the questions that still need to be answered in order to shed more light on the effects of the Daily Scrum meeting on software quality.

## Keywords

daily stand-up meeting, software quality, scrum, daily scrum, agile

## 1. Introduction

Scrum is one of the agile methods for software development emerged as a response to inability of traditional development methods such as waterfall to cope with the changing requirements in dynamic markets [1]. Although there were mentions of Scrum earlier, the method was first formally presented at the OOPSLA'95 [2]. Following that, there were published five versions of Scrum Guide that provide the definition of Scrum. The first version was published in 2010 [3] and the last version was published in 2020 [4].

Scrum is a framework within which various processes and techniques can be employed [3]. It consists of the three phases: *pre-game*, *game* and *post-game* [2]. In the pre-game phase, there is planning and defining of a new release, along with creating a high-level design. The game refers to the development part in which the system is being developed throughout iterative cycles so called *sprints*. In this phase, *quality* is mentioned as a variable that needs to be constantly monitored. The preparation for the release which includes testing and other activities required for system delivery are accomplished in the post-game phase.

In order to create regularity in a process, the key elements of Scrum are time-boxed [3]. Those elements (Fig. 1), which are referred to as events in the latest Scrum definitions [4], are the *Release Planning Meeting* (optional, and used to define an overall goal and outcomes), the *Sprint Planning Meeting* (iteration planning), the *Sprint*, the *Daily Scrum Meeting*, or simply the *Daily Scrum*, (daily meeting that takes place throughout the Sprints), the *Sprint Review* (at the end of the Sprint, usually a demo of what was done and what could be done), and the *Sprint Retrospective* (prior to the next Sprint Planning meeting, to discuss what went well and what could have been done better).

The Daily Scrum, also frequently called the Daily Stand-Up meeting or just the Stand-Up meeting, is expected to improve "*communications, eliminate other meetings, identify and remove impediments*

---

*SQAMIA 2024: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 9–11, 2024, Novi Sad, Serbia*

✉ marija.katic@gmail.com (M. Katic)

ORCID 0000-0001-8813-5281 (M. Katic)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to development, highlight and promote quick decision-making, and improve everyone's level of project knowledge" [3].

However, in spite of all those expectations, the existing research [5], [6], [7], [8] shows that the Daily Scrum generates a lot of negative feedback, while paradoxically [7], still being extensively practiced. Andersson [7] conducted a literature survey of the use of Daily Stand-Up and stated that the feedback was mostly positive in the early days following the introduction of agile methods, while later negative feedback appeared ("when skimming through studies from the past 10 years, one cannot avoid a clear shift in practitioners' opinion towards daily stand-up meetings"). Some of the benefits of the Daily Scrum, that Andersson reported, are the improved communication, transparency and trust, and knowledge sharing, while some of the challenges include that meetings do not start on time and take longer than 15 minutes, that decisions are not documented which can be problematic for traceability, that some team members need to attend multiple daily stand-up meetings, and the misuse of daily stand-up for a status report meeting. Singh and Strobel [8] state that their "results indicate that despite growing literature in daily stand-up meetings, the problem with daily stand-up meetings being perceived as irrelevant and wastage of time, still exists". Further, Stray et al. [5], through observations and interviews with teams in 40 companies and 5 countries, concluded that negative sides of the Daily Scrum can be adjusted so that the Daily Scrum is useful to all team members. For example, they recommend to reduce the frequency of the Daily Scrum if meeting for 5 days in a week is too much. In addition to research work, there are blog posts and forum discussions around positives and/or negatives of the Daily Scrum [9], [10], [11], [12].

That paradox of the Daily Scrum coming from its popularity despite many negative connotations from practitioners, first reported by Andersson [7], is the main motivation for investigating the effects of the Daily Scrum on software quality. We claim that more research is needed to confirm whether the Daily Scrum is wastage of time or not, and that research needs to be done from the point of view of understanding the effects of the Daily Scrum on software quality. We claim that it is important to take into account the power of the negative connotations that the Daily Scrum meeting provokes in people, as those may have a stronger impact (consequently on software quality) than comparable good experiences of the Daily Scrum meeting. This claim is based on the research of the power of bad events over good events done by Baumeister et al. [13]. While the existing research is more oriented towards investigating the impact of the whole Scrum on the software quality [14], because of the greater power of bad over good [13], we claim that it is needed to focus on a single element of Scrum that generates a lot of negative connotations, and investigate its impact on the software quality.

Software products with inappropriate level of software quality are generally difficult to maintain and hard or even impossible to change in order to fulfill new customer requirements. On the other hand, replacing a whole product might be too expensive endeavor for fulfilling a potentially simple new requirement. This emphasizes how important is for the practice of software engineering to deliver the product with the adequate levels of quality. Therefore, in order to ensure adequate levels of software quality for shipped products, examining the impact of various methods, techniques or even social interactions on software quality has been in the research focus so far [15], [16], [17], or [18].

When measuring the impact of some activity on software quality, researchers look at quality characteristics such as maintainability, reliability and functionality. These characteristics are defined as part of the industry standard ISO/IEC 25010 [19]. In their research about the impact of Scrum on software quality, Alami and Krancher [14] found that those study participants who perceived Scrum to have the positive impact on software quality emphasized its positive impact on internal quality (these refer to low-level characteristics of a product such as code readability and maintainability as mentioned in the study, but can also be coupling and cohesion), while the focus of those participants who had negative experiences with Scrum was more on the external quality characteristics (such as functional suitability and usability). In this paper, we refer to the potential impact of Daily Scrum on software quality in general, which might include internal or external quality characteristics, or both. The main contributions of this paper are the discussion of the positive and negative sides of the Daily Scrum as perceived in the practice of software engineering and a list of open questions with respect to the impact of Daily Scrum on software quality.

First, in Section 2 we give a background of the Daily Scrum, then in Section 3 we discuss the Daily

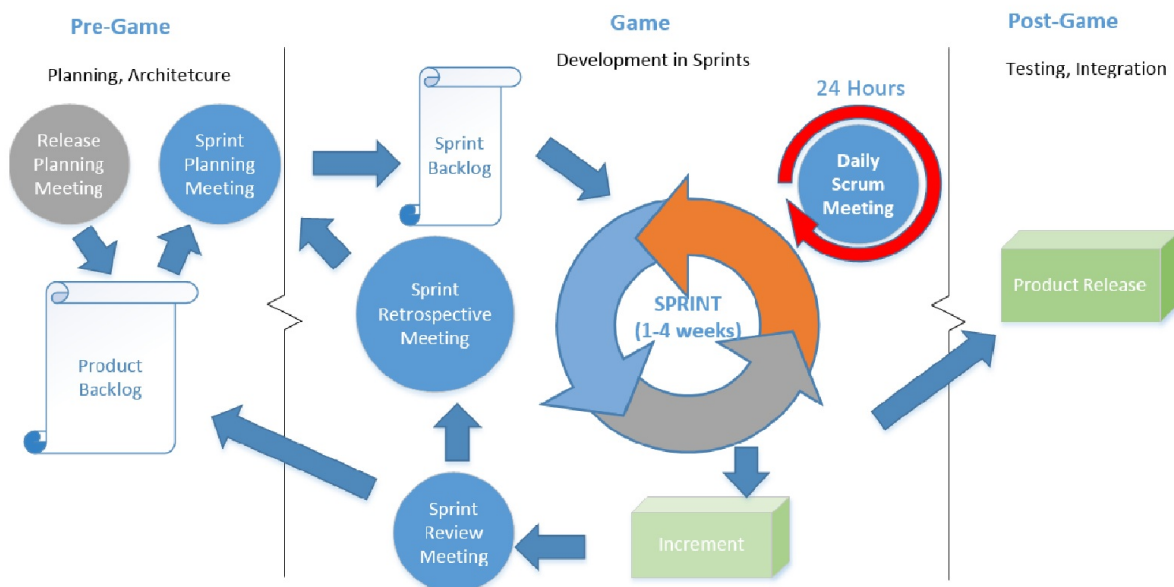
Scrum as perceived in practice. Section 4 lists the proposed open questions on the impact of Daily Scrum on software quality and Section 5 concludes the paper.

## 2. Daily Scrum Background

The Scrum guide prescribes 15 minutes for a Daily Scrum meeting that needs to happen at the same time and place every day [3], [4]. While the latest guide [4] does not mention the exact questions that need to be answered by each team member in the meeting, thus being less prescriptive than earlier guides, it still provides the guidelines for the meeting that would account for the exact questions listed in all the earlier versions of the guide. Those are the following three questions [3]:

- *What did I do yesterday that helped the Development Team meet the Sprint Goal?*
- *What will I do today to help the Development Team meet the Sprint Goal?*
- *Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?*

Figure 1 illustrates the overall Scrum and the place in it where the Daily Scrum occurs. The meeting is a way to track and plan the progress within the 24 hours and adjust the work without delays, where the Sprint Backlog (list of items being worked in the Sprint) can be updated if required.



**Figure 1:** Scrum framework (picture created based on [2] and [3]).

To describe who participates in the Daily Scrum, we need to describe roles of a project team in Scrum. Schwaber [2] defines the project team in Scrum as a team that includes not only developers, but also customers/users. The team consists of the management team with a Product Manager role who manages the Backlog, and of the development team(s) with developers, documenters and quality control roles. The Scrum guides [3], [4], which build further on the Schwaber's introduction of Scrum [2], state that the Scrum team, which is cross-functional and self-organizing, consists of one Scrum Master, one Product Owner and Developers. Scrum Master ensures that the Scrum Guide is followed, Product Owner manages the Backlog and Developers deliver the work. With respect to the skill for software quality, it is expected that the Developers have specialized skills necessary for quality control and, eventually, they are *accountable for instilling quality by adhering to a Definition of Done* [4].

While the Daily Scrum is a meeting for the Developers, others, including customers, can participate where necessary. Usually, it is the Scrum Master who participates regularly because the role of the Scrum Master is to facilitate the meeting. The Scrum Master is not expected to respond to the questions

unless he/she works on the items in the Sprint Backlog. For that reason, the Daily Scrum is in a danger of becoming a status report meeting where the Developers report to the Scrum Master, which is the case that happens in practice [5].

### 3. Daily Scrum in Practice

In this section, we describe the results of our limited literature survey about the practice of Daily Scrum. Also, we summarize the results of a small analysis of our selection of online public conversations about the Daily Scrum meeting in practice. This would provide us with a better insight into how the Daily Scrum is perceived in practice both positively and negatively, so that we can pose questions/problems that need to be investigated to better understand the effects of the Daily Scrum on software quality.

We searched for research papers that, in their results, include positive and/or negative practitioners' perceptions of the Daily Scrum. We selected two papers where those perceptions are more positive or neutral [20], [21], and two papers where perceptions are more negative [6], [8]. The results are presented in Table 1 and Table 2.

To select online public conversations, we entered the following statement in the Google search engine: "**as a programmer, what do you think about daily standups**". From the obtained results, we selected the three conversations available in the three well-known forums: Software Engineering StackExchange<sup>1</sup>, Quora<sup>2</sup> and Reddit<sup>3</sup>. Table 3 describes the summary of our discovery for online public conversations.

**Table 1**

Literature review - neutral or slightly more positive than negative

Reference	Outcome	Summary
[20], 2016, Journal	On average almost neutral, slightly more positive than negative	This study analyses daily stand-up meetings in software teams in three companies in Malaysia, Norway, Poland and the UK. The authors interviewed 60 people, observed 79 daily stand-ups, and also gathered data via questionnaires. They explored how meetings are conducted, as well as people's opinions and attitudes. The authors state that results are surprising because, on one hand, daily stand-ups are popular among agile practitioners, while, on the other hand, the results evidence a lot of negative opinions and attitudes, with overall " <i>slightly more satisfied than dissatisfied</i> ". Getting information about activities of team members and discussing and solving problems are mentioned on the positive side. On the negative side there are progress reporting and long meeting duration as well as the frequency of the meetings.
[21], 2017, Conference	While on average neutral, the majority were either positive or negative	This study analyzed responses from 221 professional developers, gathered via survey posted on Reddit. The aims of the study were to understand the "perceived value of daily stand-up meetings", and "the adoption rate of the practice". The results show that junior developers perceived it positively, while senior developers and members of large teams perceived it negatively. However, it is important to note that those who responded positively about daily stand-ups attended a smaller number of meetings per day.

The StackExchange conversation is one of the first that appeared soon after following the formalized introduction of the Daily Scrum in the first Scrum Guide in 2010. The question behind this conversation asks if anyone already practices the Daily Stand-Up and what their opinion about the meeting is. The answers to the question consist of around 40 responses and comments, together, with both positive

<sup>1</sup><https://softwareengineering.stackexchange.com/>

<sup>2</sup><https://www.quora.com/>

<sup>3</sup><https://www.reddit.com/>

**Table 2**

Literature review - slightly more negative

Reference	Outcome	Summary
[6], 2020, Conference	Deviations can have positive or negative impact, while here there are mostly those with the negative impact	In the practice of Scrum, this study investigates deviations from the Scrum guide. The aim is to understand the reasons and consequences of identified deviations. The authors apply observations, online survey and interviews, and the study is conducted in two companies in Sweden. The four deviations for the Daily Scrum meeting identified are: (1) developers do not answer all of the three prescribed questions, mostly skipping the one about impediments, (2) many team members, who attend, do not contribute to the meeting, (3) the meeting duration is longer than 15 minutes, and (4) there is no fixed time for the Daily Scrum meeting. From the point of impact on software quality, it is interesting to mention the deviation when not all questions were answered by the developers in the Daily Scrum meeting, there were reported additional bugs in the new features.
[8], 2022 available online, Journal	Negative, with some positive	The study describes the experiences of 19 professional developers (from US and India) with Daily Scrum meetings. The in-depth interviews were conducted with 14 senior developers and 5 junior developers, asking them about their thoughts and feelings on daily stand-ups, what they like, or dislike, and if there is anything they would do differently. The conclusions of the study are that daily stand-ups were too short to clearly identify or solve problems (contrary to the results of Stray et al. 2016) and "did not have a meaningful outcome". Junior and senior developers experienced them differently. While juniors were afraid of the meetings but also liked them because of learning opportunities, some senior developers found the meetings useful for tracking progress, but others found information shared as irrelevant. We give here, one positive and one negative statement recorded in the interviews. <i>+"Stand-ups help to resolve a lot of issues and queries during a particular sprint which we otherwise would be scrambling for in the end. And we may end up having a lot of issues and bugs in our applications. So, stand-ups help a lot to keep us on track with our product releases."</i> <i>-"I find that stand-ups have no outcome and are a big waste of time. Managers are not very comfortable with the new technologies, anyways. When members in the stand-up propose a strategy or say that a task would take 20 days to complete, when in reality the task is a half-an-hour work; the managers would simply accept their proposal without any questions asked."</i>

and negative feedback. Additionally, in some of the answers, there are proposals of what needs to be improved for the meeting to work better, such as defining tasks small enough so that daily conversations are easier to follow and understand. This conversation indicated that the Daily Scrum, in coming time, would induce the opposite perceptions on its purpose or usefulness. For example, one answer, not in favor of the Daily Scrum, states: *"I've had much better experiences with ad-hoc communication between team members than formal stand-ups. If someone cares about what you're working on or your progress, they'll ask you or you'll tell them. If you have a problem or are blocked, you make sure the people who need to know about it know"*.

The conversation from Quora is the shortest among the three selected for this paper, with only 10 answers. However, we selected it because it points to the fact that there are occurrences of the usage of the Daily Scrum in the practice which are not what the Scrum Guide defines. That is when the Daily



Scrum is being used for teams where each member works on a different project. Perhaps that makes some sense if different projects are sub-projects of one big project, or for some other reason. However, it is worth mentioning that Stray et al. [5], in their paper on breaking the rules from the Scrum Guide, found that in large teams there will be "*less need for mutual adjustment among all the team members*" and thus recommended to create smaller teams from one large team and conduct the Daily Scrum with smaller teams. We can only assume that the attempt to run the Daily Scrum with such diverse teams is a consequence of the popularity of Scrum and the tendency to try using it wherever there is a need to promote greater collaboration between people to improve communication, knowledge sharing, software quality or similar. It is worth emphasizing that we have done very limited research with the online discussions and it would be needed to further explore when and why such occurrences of the Daily Scrum in practice appear.

**Table 3**  
Online public conversations from three forums

Source	Year Asked	Summary of Content	Feedback Summary
Software Engineering StackExchange [9]	2011	Question asking for opinion about how valuable or not are Daily Scrum meetings, and if anyone has already practiced it	<ul style="list-style-type: none"> <li>• good idea with extreme frequencies</li> <li>• tasks need to be defined small enough</li> <li>• too many people in standup is a bad idea</li> <li>• meetings are valuable</li> </ul>
Quora [10]	2021	Question asking for opinion on how efficient are stand-ups in groups where each person work on a different project	<ul style="list-style-type: none"> <li>• waste of time</li> <li>• it should work if people are from the same team</li> <li>• it is a chance of dysfunctional team</li> <li>• why stand-up is done in this case</li> </ul>
Reddit [11]	2023	Software engineer with 5 years of software development career tells that he/she was lucky enough to be a junior that did not need to have stand-up every day and further states: I'm so fed up of telling people what I did yesterday and what I'm doing today. I just want to show up to work like everyone else and do my job.	<ul style="list-style-type: none"> <li>• reduce stand-ups to three times a week</li> <li>• stand-up contribute to micromanagement</li> <li>• management want long stand-ups</li> <li>• there are 2 stand-ups in a day, team stand-up and dev stand-up</li> <li>• if stand-up is pointless, then it is done wrongly</li> <li>• stand-ups must not exceed 15min or last even less than that</li> <li>• some people love being in meetings</li> <li>• if working remotely, stand-up is needed</li> </ul>

The Reddit conversation, which was created in 2023 and generated 283 comments, shows that the disagreement over the purpose and the usefulness of the Daily Scrum still exists even it has passed over a decade since the appearance of the first formal Scrum Guide and since the appearance of similar discussions like the conversation we found on Software Engineering StackExchange. The Reddit conversation reveals the case where software engineers are asked to attend two Daily Stand-Up meetings, the one for the whole team (*team stand-up* in Table 1) and the one for the development team (*dev stand-up* in Table 1). We can question the application of Scrum in such cases, especially if team members work on different projects as mentioned in the Quora conversation. Interestingly, in order to improve the execution of the Daily Stand-Up for large teams, Stray et al. [5] recommended to create small teams from one large team and conduct two separate meetings, one meeting, daily, with small teams, and one meeting, less frequently, with large teams. While this approach can potentially make the Daily Stand-Up more relevant to all attendants, it actually adds to the overall number of meetings

that team members need to attend.

Nowadays, it is clear that the Daily Scrum, as part of the agile movement, affected the approach to software development, thus potentially affecting software quality. For example, Stray et al. [22] state that the Daily Stand-up meeting is a venue where many project decisions are made, some of which might have major consequences. The teams that they observed did not document the decisions in their Daily Stand-Up meetings. This is unlike to the traditional software development approaches that are not flexible enough to support frequent changes of requirements like agile methods, and where the important decisions would not normally be done on the daily basis, and they would normally be documented. In order to identify the effects of the Daily Scrum on software quality, we need to relate positive and negative sides of the Daily Scrum with the software quality characteristics. Mortada et al. [6] suggested to measure software quality attributes/characteristics and relate them to individual issues of the Daily Stand-Up.

#### 4. Open Questions on the Impact of Daily Scrum on Software Quality

We already know that software quality (in terms of introduced bugs) can be related to the day of a week in which changes inducing bugs are introduced. Śliwerski et al. [23] found that the likelihood for a change to introduce a bug is highest on Friday. We also know that the Daily Scrum can interrupt work at the sensitive times of a day (such as sometime between early morning and lunch) and affect the daily routine. To conduct Daily Scrum, Stray et. al. [5] recommended to find the least disruptive time in the day, such as right before lunch. Furthermore, there are also positive perceptions of the effects of the Daily Scrum on software quality. For example, Alami et al. [14], in their study about the impact of Scrum on quality, presented that many participants reported about making better coding and design decisions with the collective intellectual effort. They also referred to the Daily Scrum positively from the collaboration and knowledge sharing point of view. Taking all of this into account, we can pose the following question(s).

**Q1: Paradoxically, does Daily Scrum positively affect software quality (internal or external) that enough, so that it justifies its existence in spite of all of its negative connotations?** For example, the improvement could be that Developers get relevant information timely, which helps them to improve their designs or to avoid misinterpreting user requirement(s). But, is that impact significant when looked in the actual project data? In other words, can we find less bugs, flexible designs that allow non-bug inducing changes, increased readability, and similar?

All agile methods, including Scrum, share specific values and principles that make them suitable for achieving success in software development projects in modern environments where changes are common. Those values and principles are encompassed within the Agile Manifesto [24], which appeared in 2001, following the introduction of Scrum. There are four values and twelve principles behind the Agile Manifesto. They describe an environment in which collaboration not only between team members, but also with customers is the key of success. In such an environment, there is no or little documentation, and the key measure of progress is just software that works.

In spite of this agile value where "*working software is over comprehensive documentation*", some of the events of Scrum such as Sprint Planning or Sprint Retrospective could be documented, at least via the tools used to manage Scrum such as Jira or Azure DevOps. On the other hand, Daily Scrum meetings are rarely documented, unless when they happen in form of online chats. This is probably justified, because they do not take long and their purpose is to promote collaboration and eliminate impediments. However, in such an environment it can happen that some important design decisions, made during those meetings, that are relevant or critical for future system changes might stay undocumented thus making it difficult and/or expensive to change the system in future.

Alami et al. [14] reported a response from one participant from their study that nicely summarizes what happens in the practice of software engineering nowadays: "*I mean, that in Agile, you have one*

team, and all knowledge is circulated within this team, and there is no need to pass this information. It's inside, it might be no formal documents, it might not be some website on Confluence, or wiki. And you store that information inside as a team, and there is not any information loss when you move further." But, what if a company loses the whole team (for whatever reason like change of management direction which can provoke decisions for people to leave the company) or just the key players in the team? Obviously, that would be an expensive information loss.

**Q2: How the lack of documentation from Daily Scrum meetings affects maintainability?**

Alami et al. [14] state that Scrum teams fail to improve quality when there are inconsistencies with the prescribed Scrum process. For Daily Scrum that is when that meeting becomes the status report meeting. Stray et al. [5] concluded that in case when team members are expected to address a meeting facilitator, the Daily Scrum meeting tends to become a status report meeting. For that reason, the Daily Scrum is perceived negatively, where people's feelings are expressed as if having oral exams every day [5]. Stay et al. [5] also documented one interesting response from the participant of their study: "*Today Scrum Master was not there, and we probably had our best daily meeting. Because, usually, the meetings are focused on information useful to the Scrum Master, but not for us. He is also more interested in what we have done than what we are going to do*". But, is it only addressing the meeting facilitator the reason for such negative feelings, which can potentially affect quality, and is it possible to avoid that without avoiding the whole meeting?

**Q3: Is there a difference in the impact of the Daily Scrum on software quality between when the Daily Scrum is run exactly as prescribed in the Scrum Guide as opposed to when the Daily Scrum is run in a form of a status report meeting?**

Junior team members in Scrum are more keen to attending Daily Scrum than senior team members [5]. Andersson [7] report that Buchan et al. [25] discovered that Daily Scrum is beneficial to new team members with the on-boarding process. But, could that activity be done separately from the official Scrum events such as Daily Scrum and, eventually, last a limited period of time while still providing a desired outcome where new team members learn about the team they joined to? Furthermore, it is important to differentiate between newcomers and novice programmers when evaluating the quality of software they create. That is because a period in which someone is a novice programmer is not the same as a period during which someone is just a new team member.

**Q4: Is there a difference between the impact of the Daily Scrum on the software quality of novice programmers and on the software quality of senior programmers?**

Stray et al. [5] listed diversity in roles, tasks and seniority as one of the main problems of the Daily Scrum meeting. We can notice that the Scrum Guide [4] prescribes that the Daily Scrum "*is not for anyone but the people transforming the Product Backlog items into an increment*", and that is developers who need to have all the required skills to do that. Those skills include programming, quality control, business analysis, architecture, user interface design, or data base design [3]. Therefore, in reality there could be Junior and Senior Developers, Quality Assurance staff, Business Analysts, Data Analysts, Architects and Designers who attend Daily Scrum every day. Those roles, indeed, will have different activities to work on a daily basis, and although those activities will be more or less related, they may not be relevant to everyone in a 24-hour period.



**Q5: Does software quality improve if a Daily Scrum meeting is conducted only by members who work on closely related activities that impact each other in a shorter timeframe?**

To answer that, one would first need to answer what would be the approach to identify those closely related activities in a shorter timeframe and how to organize multiple Daily Scrums so that everyone gets opportunity to talk to someone. Eventually, one might ask if we could achieve more by doing less and are we overdoing it with Daily Scrum?

The state-of-the-art research in agile software development organizations includes understanding the hybrid work in such organizations [26]. Hybrid work refers to a possibility of an employee to work remotely, which is usually working from home, but generally we could say working from outside the office. Existing research shows that Daily Scrum meetings can be challenging to conduct in such a context [5], and especially when team members are distributed globally in different time zones [27]. Therefore, there are different approaches used to improve conducting Daily Scrum, such as using video instead of using only audio [5] or taking into account the overlapping time zones by shifting work to different working hours [27].

**Q6/Q7: What impact does the Daily Scrum meeting have on software quality in a hybrid working environment / in an environment where team members are distributed globally in different time zones?** If it has a negative impact, is it possible to improve or is it better to abandon the Daily Scrum? Otherwise, if it has a positive impact, one could further question the recommended approaches to conduct Daily Scrum meetings in such an environment with the best possible outcome for software quality.

## 5. Conclusion

Finally, the Daily Scrum has been widely practiced in spite of a lot of negative feedback, surprisingly coming from the practice, about its suitability for software development. This paradoxical nature of Daily Scrum from the software quality point of view has not been confirmed. It is still not clear whether the Daily Scrum positively affects software quality regardless of all the complaints coming from the practice about the effects of its execution (interruptions, waste of time and similar). In this paper, we discussed the Daily Scrum, its positive and negative sides and listed the questions to answer so we could make final conclusions about the impact of Daily Scrum on software quality. In future research, we plan to expand the literature review on the impact of the whole Scrum on software quality and map that to the questions we listed as well as to identify any additional ones. Additionally, we plan to expand our research on Daily Scrum in practice, which is currently limited.

## References

- [1] K. Fertalj, M. Katic, An Overview of Modern Software Development Methodologies, Faculty of Organization and Informatics, University of Zagreb, 2008, pp. 633–639.
- [2] K. Schwaber, SCRUM Development Process, in: OOPSLA Business Object Design and Implementation Workshop, Austin, Texas, 1995.
- [3] K. Schwaber, J. Sutherland, SCRUM, 2010. URL: <https://www.qagile.pl/wp-content/uploads/2017/01/Scrum-Guide-Feb-2010.pdf>.
- [4] K. Schwaber, J. Sutherland, The Definitive Guide to Scrum: The Rules of the Game, 2020. URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>.
- [5] V. Stray, N. B. Moe, D. I. Sjoberg, Daily stand-up meetings: Start breaking the rules, IEEE Software 37 (2020) 70–77. doi:10.1109/MS.2018.2875988.

- [6] M. Mortada, H. M. Ayas, R. Hebig, Why do Software Teams Deviate from Scrum? Reasons and Implications, in: 2020 IEEE/ACM International Conference on Software and System Processes (ICSSP), 2020, pp. 71–80.
- [7] M. Andersson, Paradox of the daily stand-up meetings in agile software development context, Bachelor’s Thesis, University of Oulu, 2022. URL: <https://urn.fi/URN:NBN:fi:oulu-202206213095>.
- [8] K. Singh, J. Strobel, Exploring lived experiences of agile developers with daily stand-up meetings: a phenomenological study, *Behaviour & Information Technology* 42 (2023) 403–423. URL: <https://doi.org/10.1080/0144929X.2021.2023636>, publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/0144929X.2021.2023636>.
- [9] ..., Daily standups – yea or nay, 2011. URL: <https://softwareengineering.stackexchange.com/questions/2948/daily-standups-yea-or-nay>.
- [10] What do you think about daily standup meetings for software engineers?, 2021. URL: <https://tinyurl.com/2ka5xf82>.
- [11] Daily standup and the amount of pointless meetings is killing my love for software development and it needs to stop, 2023. URL: [https://www.reddit.com/r/cscareerquestions/comments/15bkbbg/daily\\_standup\\_and\\_the\\_amount\\_of\\_pointless/](https://www.reddit.com/r/cscareerquestions/comments/15bkbbg/daily_standup_and_the_amount_of_pointless/).
- [12] Y. Bugayenko, Daily Stand-Up Meetings Are a Good Tool for a Bad Manager, 2024. URL: <https://www.yegor256.com/2015/01/08/morning-standup-meetings.html>.
- [13] R. F. Baumeister, E. Bratslavsky, C. Finkenauer, K. D. Vohs, Bad is Stronger than Good, *Review of General Psychology* 5 (2001) 323–370. URL: <https://doi.org/10.1037/1089-2680.5.4.323>. doi:<https://doi.org/10.1037/1089-2680.5.4.323>.
- [14] A. Alami, O. Krancher, How Scrum adds value to achieving software quality?, *Empirical Software Engineering* 27 (2022) 165. URL: <https://doi.org/10.1007/s10664-022-10208-4>.
- [15] M. Katic, I. Boticki, K. Fertalj, Impact of aspect-oriented programming on the quality of novices’ programs: a comparative study 37 (2013).
- [16] N. Bettenburg, A. E. Hassan, Studying the impact of social interactions on software quality, *Empirical Software Engineering* 18 (2013) 375–431. URL: <https://doi.org/10.1007/s10664-012-9205-0>.
- [17] P. Thongtanunam, A. E. Hassan, Review Dynamics and Their Impact on Software Quality, *IEEE Transactions on Software Engineering* 47 (2021) 2698–2712. doi:10.1109/TSE.2020.2964660.
- [18] N. Ashrafi, The impact of software process improvement on quality: in theory and practice, *Information & Management* 40 (2003) 677–690. doi:[https://doi.org/10.1016/S0378-7206\(02\)00096-4](https://doi.org/10.1016/S0378-7206(02)00096-4).
- [19] ISO/IEC 25010:2011(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, 2024. URL: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>.
- [20] V. Stray, D. I. K. Sjøberg, T. Dybå, The daily stand-up meeting: A grounded theory study, *Journal of Systems and Software* 114 (2016) 101–124. doi:<https://doi.org/10.1016/j.jss.2016.01.004>.
- [21] V. Stray, N. B. Moe, G. R. Bergersen, Are Daily Stand-up Meetings Valuable? A Survey of Developers in Software Teams, in: H. Baumeister, H. Lichter, M. Riebisch (Eds.), *Agile Processes in Software Engineering and Extreme Programming*, Springer International Publishing, Cham, 2017, pp. 274–281.
- [22] V. G. Stray, N. B. Moe, A. Aurum, Investigating Daily Team Meetings in Agile Software Projects, in: 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012, pp. 274–281. doi:10.1109/SEAA.2012.16.
- [23] When do changes induce fixes?, MSR ’05, New York, NY, USA, ??? URL: <https://doi.org/10.1145/1083142.1083147>.
- [24] Manifesto for Agile Software Development, The Agile Alliance, 2001. URL: <https://agilemanifesto.org/>.
- [25] J. Buchan, S. G. MacDonell, J. Yang, Effective team onboarding in Agile software development: techniques and goals, in: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2019, pp. 1–11. doi:10.1109/ESEM.2019.8870189.

- [26] D. Khanna, E. L. Christensen, S. Gosu, X. Wang, M. Paasivaara, Hybrid Work meets Agile Software Development: A Systematic Mapping Study, in: Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering, CHASE '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 57–67. URL: <https://doi.org/10.1145/3641822.3641863>, event-place: Lisbon, Portugal.
- [27] V. T. Faniran, A. Badru, N. Ajayi, Adopting Scrum as an Agile approach in distributed software development: A review of literature, in: 2017 1st International Conference on Next Generation Computing Applications (NextComp), 2017, pp. 36–40. doi:10.1109/NEXTCOMP.2017.8016173.