



# Agile Record

The Magazine for Agile Developers and Agile Testers





# Daily Scrum meetings: What it is and what not, and what can be improved!

by Chetan Giridhar & Vishal Kanaujia

According to Wikipedia, “Scrum is an iterative, incremental framework for project management often seen in agile software development, a type of software engineering.

Although the Scrum approach was originally suggested for managing product development projects, its use has focused on the management of software development projects, and it can be used to run software maintenance teams or as a general project/program management approach.”

Before discussing daily scrum meetings in detail, let’s get an insight into the scrum process.

## What happens in Scrum?

In Scrum, three roles are defined:

- **Project Manager:** represents both development and quality assurance teams who are responsible for requirement analysis, design & development, and testing.
- **Product Owner:** represents the stakeholders and the business development professionals (also a Product Manager in product-based companies).
- **Scrum Master:** is responsible for running the complete program (typically a Program Manager); the one who resolves impediments faced by Project Management’.

A prioritized set of product requirements provided by business analysts or the product development team is referred to as ‘product backlog’. Based on the priority of these requirements, the ‘sprint backlog’ is formed. Decision on what requirements shall feature in sprint backlog items is taken in the ‘sprint planning’ meeting which involves the Scrum Master, representatives from the project management and Product Owner. In a sprint (with a typical duration of two to four weeks), the team delivers the sprint backlog items, which are essentially the product requirements. Subsequent sprints focus on the delivery of new requirements and the existing feature set is incrementally improved. Once a

sprint is completed, a ‘Sprint review’ meeting is conducted to understand what was delivered in the sprint and what not. This is followed up with a ‘Sprint retrospective’ meeting that discusses the possible improvements the team might consider so that they can have a better sprint (in terms of more product requirements, product quality, better development/QA interactions and many such factors).

Each day during the sprint, a standup meeting, the so-called ‘Scrum meeting’, takes place. This article discusses the good and not so good elements of Scrum meetings. We also suggest useful practices that teams could follow for effective Scrum meetings.

## What are Scrum meetings?

### A stand-up meeting

Every member in the team has to stand up to answer three crucial questions:

- What did I do yesterday, OR what did I do since the last time we met?
- What am I going to do today?
- Am I blocked?

The standup meeting typically lasts 15-20 minutes; longer discussions among engineers are generally avoided. Status is tracked with the help of burn down charts, in which the outstanding work (or backlog) is plotted versus time.

### Easy flow of information across teams

Product Management, Project Management (Development and QA) and Program Management (Scrum Master) all take part in the Scrum meetings; hence requests for information needed from teams or program level communication are easily achieved.

For instance, a development engineer may be facing issues implementing a backlog item as he is blocked on certain hardware

or a third party software; such issues get raised quickly and are easily shared across teams.

Product Management can float information on the new features or customer's feature enhancement requests to the Scrum Master, development and QA.

Another example could be where the Scrum Master uses this forum to communicate changes to the product backlog and consequent impact on the sprint backlog.

#### **"Feel good" factor on project status**

As Product Management, Project Management (Development and QA) and Program Management (Scrum Master) all take part in the Scrum meetings, all the stakeholders and the project contributors are synchronized with the complete program at a given time. This "feel good" factor pays dividends as all the teams strive to meet the schedule, which in turn helps the program to become successful.

#### **Improved Development/QA rapport**

While presenting their status during a Scrum meeting, Development and QA get an opportunity to be synchronized on the technical details of the backlog assigned to them.

For example, a QA engineer may have very little idea on the implementation of a backlog item and hence is unable to write tests. Talking or listening to the Development engineer's scrum meeting status for that backlog item, the QA engineer would either get his answers or he could get his questions clarified offline (after the meeting) on mail or on call.

#### **Substantiates the global working model**

With teams working on a single project scattered across different geographic locations, it becomes difficult to get everybody synchronized all the time. Mail communication may sometimes be ineffective.

Scrum meetings scheduled for a common time that is suitable to teams across different locations will help in setting better communication channels. Teams can discuss and collaborate on their views on the urgent and important items at hand fairly quickly. This solves the problem of potentially unclear mail communications.

#### **Importance of All!!**

It is imperative that all the concerned teams (Development, QA, Product teams, Scrum Master) be present at the Scrum. Even just one of these teams is not well represented, the purpose of Scrum gets lost.

#### **What are NOT Scrum meetings?**

Often Scrum meetings turn out the way they should not have been! Here are some situations teams should avoid:

#### **Scrum is NOT for detailed discussions**

Often during Scrum meetings, developers get involved in archi-

tectural or implementation discussions about certain backlog items, or Development and QA unknowingly utilize this opportunity for synchronizing with each other on a backlog item. This may at times be helpful to other team members as they may get some benefit from these discussions with regard to product knowledge, but Scrum is not the right forum for these discussions. Scrum teams should avoid such situations; instead a separate knowledge-sharing session for these discussions might be appropriate.

#### **You don't report to the Scrum Master**

This is a serious problem Scrum meetings are plagued with. Scrum meetings are held for the team and not for reporting the status to the Scrum Master. Sometimes Development and QA fall into the trap of reporting the status in a way the Scrum Master needs it, which defeats the purpose of Scrum.

#### **It's a standup; it's time-bound**

The term standup meeting itself suggests that the meeting would be held for a time frame that the team can stand for. Scrum meetings shouldn't take more than 15 to 20 minutes and shouldn't be used for solving problems.

#### **Not a bug scrub session**

Scrum meetings also have a tendency of getting converted to bug scrub sessions. Bug scrub is a process where, every week, Development and QA get together to discuss bugs logged by the QA team in that week. After listening to view points from both teams, bugs can get deferred, marked as duplicates or the severity of bugs could be modified. The outcome of this process is to ensure that Development and QA are on same page for all defects concerned. In Scrum, bugs from the previous sprint tend to become backlog items for the current sprint. Status reporting in Scrum meetings on these bugs by Development and QA can take a bug scrub form.

Similarly, QA may file a bug during a sprint on the backlog assigned to him/her. The development engineer may not agree and start asking more questions or execution logs pertaining to the bug during the Scrum meeting.

These situations should be avoided; an explicit bug scrub session on a weekly basis should help.

#### **Do not discourage engineers**

The purpose of Scrum meetings is lost when engineers start feeling discouraged or unmotivated. Sometimes engineers may not be able to make any progress on certain backlog items and report the 'No Progress' status during Scrum meetings. Questions from fellow engineers, who are dependent on these backlog items, or from Product Management or from the Scrum Master may make these engineers feel discouraged as they have not been able to make decisive progress on the backlog items. There could, of course, be genuine reasons for 'No Progress' being made, and constant questioning may frustrate the engineer for the wrong reasons.

A suggestive tone and genuine technical help in such situations will help a lot!

### **Improper use of communication channels**

In a global setup, where teams are placed across different geographic locations, it's often observed that the status reported from one location to another is not properly conveyed because of intermittent line breaks and voices in the background (if someone has dialled into the meeting from an outside office). Also, as teams are not meeting face-to-face, the facial gestures that are often used for effective communication are lost.

At times, crucial pieces of information get lost because of these reasons, which should be avoided.

A video conference from office conference rooms, where teams can see and hear each other, would be a definite alternative.

### **Incorrect reporting on backlog items**

For reporting of status, teams fall into the trap of using burn down charts. These charts typically are plotted as time (spent and left in a Sprint) v/s backlog items (completed or in progress or not started). Often, however, these charts fail to map time for a set of sub-features in a backlog item. So essentially, a backlog item is only marked complete by a developer when all sub-features of a backlog are developed. This is problematic for QA and for the stakeholders who are kept worrying about the completion of the backlog item. It can happen that even if 90% of work is done, the status of the backlog item is seen as 'In Progress', which doesn't present the correct picture of work completed.

The use of Kanban Task Boards, where backlog items are divided into sub-features and the status is tracked with respect to time, will improve reporting significantly.

### **Conclusion**

It's important to remember that Scrum meetings are a means to achieve something and not the objective itself. Of course, these meetings are helpful, but it's crucial to understand they are not imperative. If they don't suit your needs, please don't fall into the trap of using them.

Don't jump to conclusions too quickly. It's often observed that teams find the benefits of Scrum meetings from Sprint 3 or Sprint 4. A start-up time where teams get used to this change is often required.

As always, you know best what's good for you!

### **References**

Wikipedia.org, the free encyclopedia that anyone can edit

7 Tips for Improving the Daily Scrum, <http://agilesoftwaredevelopment.com/blog/artem/7-tips-daily-scrum>

## **> About the author**



### **Chetan Giridhar**

(<http://technobeans.com/about>) has more than 5 years of experience working as a software engineer in research and product organizations. Chetan is a technology enthusiast and runs a website TechnoBeans (<http://technobeans.com/>) that he updates with

tools developed, books, articles and publications written by him. You can reach him at [cjgiridhar@gmail.com](mailto:cjgiridhar@gmail.com).



### **Vishal Kanaujia**

is a Member of Technical Staff at NetApp, Bangalore, India. He has written articles for the "Linux-For-You" magazine on topics such as virtual machines, Android OS, and Python. He has interests in compiler development, performance tuning, and algorithms. You can reach him at [vishalkanaujia@gmail.com](mailto:vishalkanaujia@gmail.com).

algorithms. You can reach him at [vishalkanaujia@gmail.com](mailto:vishalkanaujia@gmail.com).