

SCRUM MASTER ESSENTIALS

eGuide



3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

Scrum continues to be the most popular Agile approach, with 66% of Agile teams identifying it as the method they follow according to the [15th State of Agile Report](#). As Agile adoption itself is now mainstream, and with Scrum taking the biggest chunk of an ever expanding Agile pie, the role of the scrum master is more in demand than ever. But what does it truly take to be a good scrum master?

In this eGuide

8 Winning Qualities of a Scrum Master

A good leader keeps everything in order. In this fast-paced era where agile methodologies have become a more often used framework, a good scrum master is what is needed. But what makes a good scrum master?

3 Mistakes Teams Make When Choosing a Scrum Master

One big cause of agile project failure is choosing the wrong person to be your scrum master. While a bad scrum master is a problem for any team, it is particularly bad for teams new to agile, as the people on the team won't know they are being led down the wrong path. Let's review three mistakes organizations make when choosing scrum masters.

A Scrum Master's Superpower of Observation in Virtual Teams

Scrum Masters rely on observational skills, but does that mean we only use vision? What happens when we are all remote? Can we leverage other senses to "observe" how teams are surviving (or thriving)? We may not be able to sit together with our team for a long time, and it may never be the same due to the long-term effects of the coronavirus. What skills might we adapt or create for virtual teams? And, how can we still reflect back on the observations so the team can decide how to improve?

Retrospectives Without Action Items Mean Nothing Gets Done

The point of a retrospective is not just to complain about things we don't like or to congratulate ourselves on the end of a sprint. The goal is to identify opportunities to improve. Sometimes that is stopping bad practices and other times that is doing more of a good practice. Either way, in order to improve you need to make changes. In order to make changes someone has to actually do something.

Daily Standup Purpose: It's All About Flow, Not Status

The Daily Standup (DSU) is a key ceremony in the Scrum Framework. When run effectively, it can ensure the Scrum team focuses on what's important to meet the Sprint goal and commitment. When DSUs serve only as a status meeting, the value of the ceremony is lost.

How Testers Can Collaborate with the Scrum Master

Those in testing roles help build quality into the process and product early, so they're in a good position to collaborate with the scrum master to improve agile processes. We will review some ways testers can partner with, support, and assist the scrum master—and the rest of the team.

Additional Resources

8 Winning Qualities of a Scrum Master

By Dan Martin

A good leader keeps everything in order. In this fast-paced era where agile methodologies have become a more often used framework, a good scrum master is what is needed. But what makes a good scrum master?

A scrum master is responsible for making sure the scrum process is followed and that the team adheres to the rules and practices. A scrum master also coaches the team to improve and eventually eliminate impediments. With so much responsibility, a scrum master must possess winning qualities to perform the required duties. Here are the eight winning qualities of a scrum master.

1 Good Communication

Communication is always a key for a scrum master and is especially important in an organization first adapting the scrum framework. A scrum master should be exceptional in communication—both in verbal or written forms. He will do a lot of back and forth coordination with managers who he is reporting to down to team members in each process to make sure that scrum practices are followed.

Scrum communication isn't about giving orders to team members either. A scrum master must be able to carefully listen and comprehend when his members report on progress and blockers. Most of the time problems are raised to scrum masters for resolution. If one does not know how to listen and comprehend what the problem is, it is difficult to arrive at the right solution.

2 Responsible

A scrum master is often the representative of the team to management and other teams. A good one should be able to stand

up for them when something is needed or there is a crisis. The responsibility entrusted to the scrum master does not come with power. Their responsibility is to guide the team into the adoption of the scrum process and practice it. The advantage of being a responsible scrum master is you gain your members' trust. Once the scrum master has established that, members will listen more and thus commit fewer mistakes and will contribute more to the success of the team.

3 Meticulous

A scrum master has to be meticulous and have great attention to detail. Why? Because it is her responsibility to come up with ways and strategies on how to properly implement solutions to problems. The level of attention may vary depending on the difficulty of the problem.

4 Multitasker

A scrum master should be able to do multiple tasks at the same time even if it means doing tasks that he is not used to. However, a good one should be able to quickly adapt to keep everything according to schedule, thus he must be a multi-tasker. Problems and issues can occur when least expected, and more often than not, all at the same time.

Part of being a multitasker is being resourceful. To be able to reduce, if not eliminate impediments, he should be able to find quick ways to solve problems even with little to no resources. He should be able to take whatever it is around and look for clever ways to address the situation at hand. The scrum master should be able to quickly change gears when bombarded with such situations.

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

5 Committed

A scrum master's role may not involve around-the-clock work, but a full commitment is required. Her work does not end when a certain impediment is being solved. Also, some impediments take days or weeks to eliminate, so commitment is required to be able to follow through the progress of eliminating these impediments.

For example, when you ask your manager to allocate a huge resource for the whole team, you wouldn't be able to get it right away. It will involve a series of meetings and discussions between involved parties.

A scrum master should be able to commit to the full duration of the project. And besides, it is very disruptive for the team members to change scrum masters mid-project.

6 Influential

Why is being influential a winning quality for a scrum master? Sure, being a leader assumes a level of control in a team, but being influential will do the team much more good than just giving out commands. Avoid asserting yourself just because you are the leader. A scrum master must be able to influence both the team itself and those outside of it.

First, a scrum master needs to influence team members to give the scrum process a fair trial. This allows team members to behave more collaboratively. A scrum master may also influence the whole team to try new technical practices, such as pair programming and test-driven development.

Second, a scrum master needs to influence those outside the team. A scrum master may need to influence the operations manager to assign resources to the team or to explain to company executives the importance of scrum—and eventually convince them to try it.

However, scrum masters should know how to use their influence without having to resort to the “because I say so” style. The ideal

scrum master should come with a degree of corporate political skills, know how decisions are being formulated in an organization, and most importantly be an asset to the company.

7 Collaborative

A good scrum master ensures that a collaborative culture exists in the team. Team members should be able to freely raise issues and concerns that are up for open discussions, and they should feel supported in doing so. Also, team members should be able to freely express their ideas, which might give solutions to existing problems.

Furthermore, this type of environment will resonate with team members as they will feel confident and not be intimidated by their scrum master and other members, which allows them to give and accept criticism from other team members.

8 Composed at All Times, Despite the Chaos

You may be helping your members with the process now, solving problems later, and then resolving an impediment. Everything tends to be very chaotic at times! A good scrum master must be able to maintain composure despite the chaos.

A good scrum master does not shy away when problems arise. He must be quick in coming up with solutions even in the most chaotic times. Mind you, these are not just solutions that will stop the existing problem, rather they will prevent it from happening again. It would also be a good additional skill if a scrum master can foresee and stop problems from happening.

To put it into perspective, if the house is on fire, a scrum master should not be found running away from the house screaming his lungs out. Instead, he should be like a firefighter—always focused and composed in assessing the situation. A scrum master must then be quick to apply a solution where it's needed.

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

3 Mistakes Teams Make When Choosing a Scrum Master

By Jeffery Payne

One big cause of agile project failure is choosing the wrong person to be your scrum master. While a bad scrum master is a problem for any team, it is particularly bad for teams new to agile, as the people on the team won't know they are being led down the wrong path.

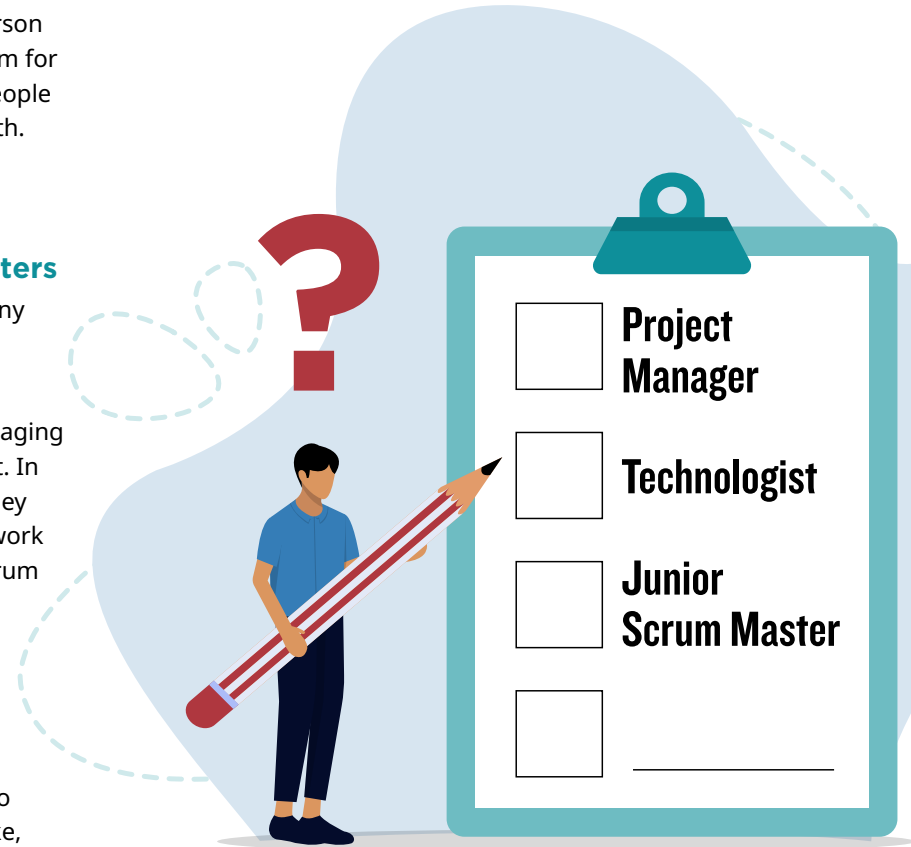
Here are three mistakes organizations make when choosing scrum masters.

1 Transitioning Project Managers to Scrum Masters

A scrum master is not a project manager. Unfortunately, many organizations assign their existing project managers to the scrum master role when moving to agile.

The problem is that most project managers are used to managing a team, not serving it. The distinction is subtle but important. In agile, teams makes their own decisions about the process they follow, the tools they use, how much time it will take to get work done, and who does what work in sprints. The role of the scrum master is to facilitate the conversations that allow the team to make good decisions and to remove barriers to progress.

Telling a team what to do is a scrum master's last resort, used only when a very experienced scrum master sees the team is about to make a decision that will cause irreparable harm. This unlikely scenario aside, the team is empowered to make decisions, learn from the inevitable mistakes they make, and continuously improve their capabilities.



3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

While it is certainly possible that an existing project manager has the servant leadership style necessary to be a successful scrum master, don't assume this is the case.

2 Making Technologists Scrum Masters

Agile relies on collaboration among everyone on the team much more than traditional software development. A good scrum master has strong communication skills they use to get everyone collaborating, to facilitate discussions, and to keep everyone on the team in the loop.

While it is tempting to promote a strong technical leader into the role of a scrum master, do so with caution. Having a technical background is definitely an advantage, as they are able to dig into issues and understand them better, but this should not be the overriding priority. Technical personnel are often introverts who are great problem-solvers but don't have the communication skills or aptitude to effectively lead collaboration efforts with a team.

Putting someone in the scrum master role who does not communicate well will often result in a significant amount of rework, redundant work, and frustration on the team.

A good scrum master is part coach, part mentor, part facilitator, part motivator, and sometimes even part therapist.

3 Hiring Junior Scrum Masters

There is no such thing as a junior scrum master. I've seen teams where the most junior (and nontechnical) person on a team is the scrum master, and this never works.

A good scrum master is part coach, part mentor, part facilitator, part motivator, and sometimes even part therapist. Individuals capable of playing this role are those with lots of leadership experience who know how to best handle a wide variety of personalities, situations, and challenges. If you hire a junior scrum master, all you will get is someone who checks boxes and asks for status updates, and that doesn't add any value to the team.



3

8 Winning Qualities
of a Scrum Master

5

3 Mistakes Teams
Make When Choosing
a Scrum Master

7

A Scrum Master's
Superpower of
Observation in
Virtual Teams

10

Retrospectives
Without Action
Items Mean Nothing
Gets Done

12

Daily Standup
Purpose: It's All About
Flow, Not Status

14

How Testers Can
Collaborate with the
Scrum Master

16

Additional Resources

A Scrum Master's Superpower of Observation in Virtual Teams

By Mark Kilby, Savita Pahuja

Scrum Masters rely on observational skills, but does that mean we only use vision? What happens when we are all remote? Can we leverage other senses to “observe” how teams are surviving (or thriving)? We may not be able to sit together with our team for a long time, and it may never be the same due to the long-term effects of the coronavirus. What skills might we adapt or create for virtual teams? And, how can we still reflect back on the observations so the team can decide how to improve?

Listening as Primary Skill

First, listening becomes a primary observation skill with online teams. Listening occurs on many levels, but it should always be

“active listening.” Active listening involves full concentration to hear the speaker, and the speaker knows you are listening to what they say. This skill develops with continuous practice. Understanding the words used, tone of voice, and what may not be spoken can tell the skillful scrum master other ways to assist their team.

For example, in sprint planning, a developer doesn't ask any questions but nods his head. Others may assume he understands the backlog item. Later, when the developer is assigned the story, he may ask many questions. Were they listening during sprint planning? What might have been a distraction?

To avoid these types of challenges, you may ask questions that don't have simple answers.

For instance, in the above-mentioned situation, a scrum master may ask the developer: “Do you understand the backlog item?” and get a “Yes” answer. It won't avoid clarifying the assumption well. Instead, the scrum master should ask an open-ended question such as “Would you explain what's your understanding of this backlog item?” or “Based on what we've discussed, what challenges do you anticipate?”

Open-ended questions require careful thought. One type, known as reflective coaching questions, help to mirror back observations. For example:

- At what point did you notice...?
- Describe what happened when...?
- Tell me more about...



3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

These types of questions help everyone share the observations and insights about the work.

Paraphrasing is another technique of active listening and allows a listener to summarize what they have heard from the speaker in their own words. It helps the listener confirm they understand what the speaker says.

For example, while facilitating retrospectives, it becomes important to paraphrase the speaker's challenge before asking further questions. In this way, you make sure that you are asking the right question to generate insights into the problem.

With these types of questions, we may also start to explore the context of the issue being discussed.

Observations on Context

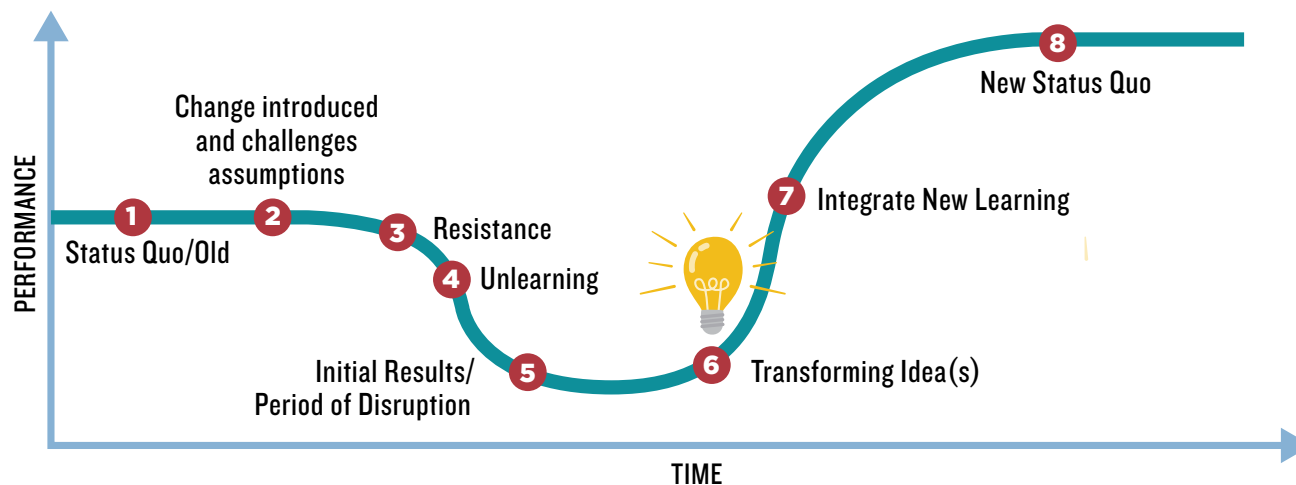
Second, we may need to make observations about the context surrounding a change to determine how team members process the change. For instance, do they have a good balance when they live at work and work at home as remote team members? Can they accept help when a change has been forced upon them, such as when the coronavirus forced many out of the office with little warning? Do they accept new ideas or does the stress of the change make it diffi-

cult for them to let go of assumptions?

A tool like the Satir Change Model reveals how the team navigates the change and provides clues on the impact of their context. As shown in the diagram, each stage of the model shows how each team member individually navigates the change. Once things move from (1) Status quo to (2) The change being introduced, each team member could be at any of the following stages:

3. Resisting by holding onto old assumptions
4. Letting go of those assumptions (unlearning)
5. Struggling with the change because they don't have a model or assumptions to deal with the new context
6. Discovering new ideas (either individually or from other team members) that transform their thinking
7. Integrating their new ideas into their way of working through experimentation
8. Deciding on their new standard ways of working based on the new ideas

A scrum master may privately note where they feel each team member sits on the change curve based on their observations. Optionally,



3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

Date of Meeting/ Discussion	Behavior What are the key behaviors of the team? (team's energy, collaboration, openness, etc.)	Action Key reaction/comment from the team	Insights SM's insight about the discussion
<p>Common Patterns (Find out any common behavioral or action patterns):</p> <ol style="list-style-type: none"> 1. 2. 3. 			
<p>Action Plan (When and how you would reflect the patterns to the team to bring improvement. Some common events to bring reflections are one-on-one coaching sessions, group coaching sessions, or retrospectives:</p>			

the scrum master may ask each team member to indicate where they are on the change curve in a retrospective or an anonymous poll.

Rarely will everyone be at the same point on the curve during the change process. It becomes important for the team to be aware of where fellow team members sit on the change curve to determine if they need help, time to think through the change on their own, or if they are ready to settle into new ways of working. You can read more about the Satir Change Model at <https://www.markkilby.com/navigating-the-change-curve-basics/>.

Observation Log

Third, when scrum masters learn to reflect back, they may use this new tool of observation too often. scrum masters should learn when and how they can create the space for self-learning for the team. Sometimes a scrum master consciously takes a step back to

help the team take their own actions in the improvement journey. An observation log becomes another tool for the scrum master to notice observations about the team and share it with the team at an appropriate time. Consider the sample observation log below.

Observational skills remain critical even for remote teams and their scrum masters. Other senses and skills help scrum masters notice what their remote team may be struggling with.

The scrum master notices what team members say and how they respond to changes in their context. Based on that, scrum masters can share their observations with the team.

Amplifying listening skills, using frameworks like the Satir Change Model, and capturing and sharing observations in a log help remote scrum masters use the superpower of observation in building a high performing team.

3

8 Winning Qualities
of a Scrum Master

5

3 Mistakes Teams
Make When Choosing
a Scrum Master

7

A Scrum Master's
Superpower of
Observation in
Virtual Teams

10

Retrospectives
Without Action
Items Mean Nothing
Gets Done

12

Daily Standup
Purpose: It's All About
Flow, Not Status

14

How Testers Can
Collaborate with the
Scrum Master

16

Additional Resources

Retrospectives Without Action Items Mean Nothing Gets Done

Every agile coach and agile book worth its salt will preach the benefits of holding regular retrospectives. Most teams I work with do their best to hold them, but many claim they just don't work. They say this important agile ceremony often turns into a finger-pointing session, or perhaps worse, a meeting where everyone celebrates a job well done while ignoring issues that are holding the team back from being even better.

Probably the single biggest cause that I see of ineffective retrospectives is the lack of clear action items to come out of the meeting. The point of a retrospective is not just to complain about things we don't like or to congratulate ourselves on the end of a sprint. The goal is to

The point of a retrospective is not just to complain about things we don't like or to congratulate ourselves on the end of a sprint. The goal is to identify opportunities to improve.

identify opportunities to improve. Sometimes that is stopping bad practices and other times that is doing more of a good practice. Either way, in order to improve you need to make changes. In order to make changes someone has to actually do something. Action items are things for people to do.

At the end of every retrospective, make sure that there are clear activities that individuals are going to do to change the way the team operates. Sometimes these will be individual actions. Other times they may be team actions. Regardless, they need to be clearly captured as a part of the meeting.



3

8 Winning Qualities
of a Scrum Master

5

3 Mistakes Teams
Make When Choosing
a Scrum Master

7

A Scrum Master's
Superpower of
Observation in
Virtual Teams

10

Retrospectives
Without Action
Items Mean Nothing
Gets Done

12

Daily Standup
Purpose: It's All About
Flow, Not Status

14

How Testers Can
Collaborate with the
Scrum Master

16

Additional Resources

I've worked with teams that despite doing a great job of creating action items nothing changed. This was usually because they wrote the action items on a white board or a set of sticky notes, agreed to them, congratulated themselves on a successful retro, and then promptly left the items in the room never to be seen again. Make sure you record the action items and store them in a place that is visible and where the team can hold themselves accountable to completing them.

A few tips for managing action items:

- Co-located teams can post the retrospective action items in a visible place within the team room. This way everyone sees them when they are in the room. If you track work on a physical Kanban board, create an action item story and put it in the backlog for the following sprint.
- Use your project management tool (e.g., Jira) to manage the action items. Just like a physical board, create a story for each action item and add it to the team backlog. In Jira, you could even create a custom issue type and track how many action items the team is working on to get a sense of how much time they are spending on improvement activities.
- If you don't want to create stories, a Wiki that the team uses regularly can be a good alternative. There is a feature that I like in Confluence where you can create individual retrospective pages and add action items to those pages. You can then create a summary page for all the retrospectives that can show all the open action items with either the date or the sprint in which they were opened. That way it is easy to see if the team is letting items linger over time without resolving them.
- However you track the action items, the first thing a scrum master should do at a retrospective is review the open action items from previous retrospectives and discuss why they haven't been closed.

One final note is that you can have too much of a good thing. Recently I was asked to facilitate a retrospective for a team who's previous retrospective had resulted in over 100 action items. They even recorded them and knew where they were. Guess how many

...commit to only the top five or so. You don't even need to save the others; if they remain important they will come up again at the next retrospective.

had gotten done when I came in about a month later? Basically zero. That many action items was just overwhelming.

Keep your action item list to a manageable number—no more than a handful. Even if you do identify 100 potential items, commit to only the top five or so. You don't even need to save the others; if they remain important they will come up again at the next retrospective.

When I held that retrospective we had a very healthy discussion but I limited the team to four high-priority action items. Within a week they had tackled at least some piece of each of them, a vast improvement over zero in a month.

Remember, retrospectives give a team the opportunity to change and improve. Clear, visible, and manageable action items identify what needs to be done to change. Retrospectives without action items mean nothing will change.

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

Daily Standup Purpose: It's All About Flow, Not Status

By Rich Stewart

The new 2020 version of The Scrum Guide has thankfully removed the “three questions” from the Daily Scrum/ Daily Standup (DSU) section of the guide. Those who understand that The Scrum Guide is just that—a guide—and not a prescribed methodology have never felt bound to use the three questions anyway. The questions are:

- What did you accomplish yesterday?
- What do you commit to completing today?
- Are there any obstacles in your way that may prevent you from meeting your commitments?

While answering these three questions may lead managers to feel the team members are working hard, they do not provide actionable insight into how well the team is progressing against their sprint goal.

For many years in my role as an agile coach, I've served as Scrum Master for 1-2 teams. This has kept me working day-to-day as a practitioner. In my current organization that I've been a part of for nearly five years, I've had the opportunity to partner with an executive director of development with whom I'm simpatico on how to run a DSU. Our focus has always been on the flow of work (user stories and bugs) from the In-Progress state (or In-Development state if you prefer) to the Accepted state during a Sprint.

Of course, it's not easy to focus on the workflow without a visual way to see the work. My teams always use a kanban board for this



purpose. Using a kanban board (or Scrum board in Jira terminology) is critical to visualizing the teams' workflow within the Scrum Framework. However, using a visual board is not enough if it only includes work states and not queue states. Consider a kanban board that only consists of the following work states:

- Sprint Backlog
- In-Progress
- In-Test
- Done

With this minimal set of states, there is no insight into the wait times between work states. For example, suppose a user story moves from In-Development to In-Test. In that case, we have no insight into how much of the time spent in the In-Test state is spent testing versus waiting for a QA engineer to be available to test. We, therefore, have

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

no insight into the flow efficiency (expressed as a percent) of the workflow process, defined as:

$$[\text{Work time} / (\text{Work time} + \text{Wait time})] \times 100$$

Breaking it down further, In-Progress for most development teams consists of two separate sub-work states: In-Development and In-Review (by a peer). Thus, a complete set of states for a typical development team might include the following:

- Sprint Backlog/Ready to Pull (queue state)
- In-Development (work state)
- Ready for Review (queue state)
- In-Review (work state)
- Ready for Test (queue state)
- In-Test (work state)
- Ready for Acceptance (queue state)
- In Acceptance Review (work state)
- Accepted

With this or a similar set of states for a particular workflow, it would be possible to accurately measure the team's **flow efficiency** and identify bottlenecks based on the proportion of time work items spend in a specific queue state compared to the corresponding work state.

Note: It's also useful to set work-in-progress (WIP) limits for particular work states to discourage context switching between work items, though very mature teams may **evolve beyond WIP limits based** on the application of the **Theory of Constraints**.

In practice, most development teams will push back from using nine distinct states based on practical concerns over frequent state management of work items and fitting so many states on a kanban board without scrolling horizontally. My teams have compromised and use the following states for work items within a sprint:

- Defined (queue state)
- In-Progress (work state)

- Dev Complete (work state)
- Ready for Test (queue state)
- In-Test (work state)
- Completed (work state for product owner acceptance)
- Accepted

While this set of states often doesn't allow us to rigorously measure overall flow efficiency, it provides enough information for useful DSUs that are workflow and work item-centric instead of being team member-centric. With my globally distributed teams, I display the kanban board in our video call and "walk the board," which means moving from right to left, discussing every work item in the Completed to In-Progress states. We start from the right because our focus is getting work items to the Accepted state, and those closest to Acceptance have the best chance of getting accepted within the sprint.

This format, focused on flow, should include every team member's discussion because all work done by a Scrum team should be visible and represented on the board. Most leading agile tools, such as Jira, Rally, and Azure DevOps, allow work items to include a visual indicator of how many days a work item has been in a particular state. Thus, the discussion of a work item stuck in a specific state for multiple days should include an open dialog on why it's not progressing, including any obstacles the team may be able to **swarm** to resolve.

My teams also change the owner of a work item based on its state—with the owner removed for queue states. As the facilitator, in real-time during a DSU if I see a work item in a work state without an owner, I'll ask the team to whom it should be assigned. Changing the owner based on the work state makes it clear while walking the board who should speak up on the work item's progress within a particular state.

Focusing on the flow of work instead of team member status can lead to much more efficient use of time and impactful discussion with improved transparency on the sprint's health during the DSUs. Seemingly small changes to the Scrum process, such as the focus on flow, can significantly improve value delivery.

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

How Testers Can Collaborate with the Scrum Master

By Michael Sowers

Most of the organizations I work with are embracing agile or some variant of lean development and delivery. While agile has been around for a while, there are still several organizations that are just beginning their agile transformation journey or are continuing to refine their agile practices.

In the Scrum implementation of agile, we have three roles: the product owner, scrum master, and team members. The team works together to define the need or problem, develop the solution, and test the business value to be delivered before, during, and after development.

The scrum master serves the team by providing facilitation and coaching, including helping the team remove roadblocks, orchestrating the daily standups, working with the team to define and refine their practices, and working with the product owner to prioritize stories.

Among the challenges that scrum masters face are prioritizing work in collaboration with the team; helping team members decide

who is best suited for tasks; keeping the daily standup focused and timeboxed; facilitating communication across the team, particularly if the team is distributed and in different time zones; keeping the emphasis on quality practices and outcomes; and dealing with senior management who may not yet have fully internalized the agile principles of team empowerment.

Those in testing roles help build quality into the process and product early, so they're in a good position to collaborate with the scrum master to improve agile processes.



3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

Here are some ways testers can partner with, support, and assist the scrum master—and the rest of the team:

Lead a team discussion with the scrum master on the team's quality approaches and provide mentoring and coaching on quality assurance and testing practices

Help the scrum master and team understand the benefits of continuous quality and implement approaches for verification at every step, such as grooming the backlog and testing user stories as they are being developed

Advise the scrum master on risks and focus on completing higher-risk stories first and implementing mitigation strategies for identified risks

Be a model team member in demonstrating communication and collaboration within and across teams and with distributed team members

Educate the scrum master on the many dimensions of functional and nonfunctional testing that may be necessary in sprints, such as testing APIs, data, mobile interfaces, application security, compatibility, performance, and so forth

Provide the scrum master visibility into the benefits of static and dynamic analysis and help the team implement tools, automation, and reporting for these early defect-prevention techniques

Contribute to the definition and refinement of "done"

Communicate the value and opportunities for automation to the scrum master and team, not just for testing, but for the entire software development lifecycle

Remind the scrum master of the importance of a lightweight set of measures and metrics and the criticality of rapid feedback, and help implement the appropriate team dashboard

Help the scrum master be alert to cross-dependencies with other teams

In the most effective agile teams I've worked with, those in testing roles become true and trusted partners to the entire team. Partnering and collaborating with the scrum master helps balance the focus on development and quality, yielding improved business value.

3

8 Winning Qualities of a Scrum Master

5

3 Mistakes Teams Make When Choosing a Scrum Master

7

A Scrum Master's Superpower of Observation in Virtual Teams

10

Retrospectives Without Action Items Mean Nothing Gets Done

12

Daily Standup Purpose: It's All About Flow, Not Status

14

How Testers Can Collaborate with the Scrum Master

16

Additional Resources

Additional Resources

MORE INFORMATION FOR SOFTWARE PROFESSIONALS



The TechWell Hub is a great resource to get your questions answered, help others with problems they're stuck on, and engage with experts in software. Follow channels like #agile, #agile_leadership, and more.

JOIN HERE

Narrow your search to a specific type of resource:

AgileConnection Community

AgileConnection brings you the latest in agile and DevOps principles, practices, and technologies. Check out articles and interviews from experienced software professionals and thought leaders, and join the community to gain access to member-exclusive content such as conference presentations, weekly newsletter updates, Q&A discussions, and more.

Interviews

Each year, TechWell interviews dozens of software professionals, including well-known thought leaders, seasoned practitioners, and respected conference speakers. **Click here** to read, listen to, and watch interviews with agile and software engineering experts.

Scrum Articles

AgileConnection is home to thousands of Scrum software resources, including articles, archived *Better Software* magazine articles, conference presentations, and interviews with industry notables. Check out these Scrum articles to read about shifting left, scrum tips, and better agile collaboration.

Agile + DevOps Conferences

The Agile + DevOps conferences bring you the latest in the methods, process, technology, tools, and leadership principles to deliver software with greater speed and agility while meeting quality and security demands. No matter your level of agile or DevOps adoption, you will find thought provoking keynotes, deep-dive training classes, engaging tutorials and session topics that will guide and inspire you.



Our partner, Coveros, has significant Scrum experience and can help organizations implement or improve agile processes. Coveros offers courses on agile, SAgile, Scrum, and more—all of which include best practices taught by industry leaders. Whether you're looking to get hands-on experience for yourself, your team, or your organization, Coveros has a learning solution for you.

[Agile, SAgile, and Scrum Courses](#) | [Agile Testing and Automation Courses](#) | [Agile Dev Services](#) | [Transformations](#)