60 Scrum Master Interview Questions to Avoid Hiring Imposters

Proven questions you can ask when interviewing Scrum Master candidates

By Stefan Wolpers and Andreea Tomoiaga

Version 5.4 | 2022-01-10

Table of Content

Introduction

Maybe ‘Agile’ in general is a fad as opposed to a trend. Though whatever the case, we can say for sure that Scrum[1] is very popular in software development. Demand for seasoned Scrum practitioners and the entry of new professionals into the market are both on the rise.

If you are looking to hire a Scrum Master for your organization, you will find the following interview questions useful in identifying the right candidate. Being cognizant of what to listen for in a candidate’s answers to these questions will allow you, as an interviewer, to more quickly understand not only a candidate’s familiarity with Scrum — but also their agile mindset. Given the complexity of applying agile practices to any organization, multiple-choice questions are mostly insufficient when you need to discern a candidate’s agile mindset.

The authors, Stefan Wolpers and Andreea Tomoiaga, share a holistic view on agile practices:

Agile equals product discovery (what to build) plus product delivery (how to build it).

The examples and guidance provided in this book reflect this view and the personal experiences of the authors, and may not be valid for every organization. Please keep in mind that what works for another organization may not work for yours.

These interview questions are not intended to turn an inexperienced interviewer into an expert on agile software development. But in the hands of a seasoned practitioner, these questions will provide ample support for determining who among your candidates has actually worked successfully in the agile trenches — and who among these candidates are, in fact, imposters.

[1] “Scrum is a framework for developing, delivering, and sustaining complex products.”

Why these Questions

These questions are derived from Stefan Wolpers' fifteen years of practical experience with Kanban[2], Scrum, XP[3], and several product creation frameworks. Stefan has worked at different times as a Product Owner, Scrum Master, and agile coach with a variety of teams and organizations of all sizes and levels of maturity. On behalf of clients and employers he has interviewed dozens of candidates throughout his career for the role of Scrum Master. Today, Stefan is educating Scrum Masters and Product Owners in his capacity as Professional Scrum Trainer (PST) with Scrum.org.

Many of these questions were first introduced by a blog post written by Stefan on the Age of Product web site. The post led to a public discussion on LinkedIn, following which Andreea and he decided it would be helpful to create a handbook that provides examples of, and guidance interpreting, the answers that they believe would indicate suitable candidates for the role of Scrum Master. 54 Scrum Master Interview Questions to Avoid Hiring Imposters, now in its fifth edition, is the outcome of that.

A Message for Age-of-Product.com Subscribers

If you would like to be notified when the sixth edition of this book becomes available, please **do not unsubscribe from the newsletter**. Your subscription is the only way we can know that you once downloaded this book.

If you received a copy of this book from a friend or colleague and are interested in subscribing to the Food for Agile Thought newsletter, or if you'd simply like to learn more, visit us at Age-of-Product.com.

[2] Kanban is a flow management framework, and its principles are a model for introducing change through incremental improvements.

[3] Extreme Programming is a lightweight agile software development practice.

Set 1: Role of the Scrum Master

Background

- Scrum is not a methodology, but a framework. There are no rules that apply to each and every scenario — just good practices that have worked before in other organizations.
- The good practices of other organizations cannot simply be copied to your own. Every good practice requires a particular context to work.
- As somebody hiring for a Scrum Team, you need to determine for yourself what works for your organization — which is a process, not a destination.
- The role of a Scrum Master is primarily one of servant leadership and coaching. It is not a mere management role. (Although the Scrum Master role also has management aspects, for example, regarding the responsibility for promoting and supporting Scrum within the organization.)
- A Scrum Master should recognize that different stages of a Scrum Team's[4] development require different approaches: some, teaching; some, coaching; and some, mentoring.
- A Scrum Master should know of the Shu-Ha-Ri (Japanese martial arts) method of learning new techniques.
- A Scrum Master's principal objective should be to remove themselves from daily operations by enabling the Scrum Team to be self-organizing.
- Being a Scrum Master does not entail, and should never entail, enforcing processes.
- Scrum is not designed for bean counters, although some metrics are helpful in understanding the health of a Scrum Team. Generally, insisting that the team achieve specific KPI[5] (e.g. forecasts vs. velocity) does not help.
- Scrum doesn't elaborate on the process that enables a Product Owner to add valuable, usable, and feasible work items such as features to the Product Backlog[6]. Product discovery using the Design Thinking, Lean Startup, or Lean UX frameworks help, but in any case a good Scrum Master will want the Scrum Team to be a part of this process (whether by participating in user interviews or running experiments).

[4] The Scrum Team comprises of the Development Team members, the Product Owner, and the Scrum Master. It is cross-functional, self-organizing, and does the work necessary to create a Product Increment.

[5] Key Performance Indicators are metrics used to evaluate an organization's success at reaching targets.

[6] Product Product Backlog is a list of items to work on, such as features, bugs, technical work, and knowledge acquisition.

- A Scrum Team’s communication with stakeholders should not be run through a gatekeeper (e.g. solely through the Product Owner) because this hurts transparency and negatively affects the team’s performance. Sprint Reviews[7], conversely, are a good way to stay in close contact with stakeholders, and to present the value delivered by the Scrum Team during each previous Sprint[8].

Q 01: The Scrum Master Role as a Contradiction?

The Agile Manifesto infers people over processes. Isn’t a Scrum Master — whose role is meant to “enforce” the process — therefore a contradiction?

Scrum Masters do not wield any real authority but act as servant leaders. The Scrum Team does not report to them. This question is meant to help reveal whether your candidate understands that their role is to lead — as opposed to managing — the team. Asking this question is also likely to reveal why your candidate is interested in the role of a Scrum Master in the first place.

Acceptable answers should emphasize facilitation and support:

- “I am the servant leader for the Scrum Team. It’s my job to make them successful.”
- “I am neither a project manager nor a people manager. I support the Scrum Team in achieving self-organization. I do not tell people what to do.”
- “I am the Scrum Team’s facilitator as teacher, coach, or mentor, encouraging them to excel as a team.”

Q 02: Success Factors of “Agile”

What indicators might there be that demonstrate agile practices are working for your organization, and which of these would demonstrate your efforts are succeeding?

There is no standard or general definition of ‘agile success’ that can be used to measure an organization’s agility. Every organization must develop its own criteria. Increasing team

[7] Sprint Reviews involve discussion of work completed, planned work not completed in a Sprint, the state of the product or project, and how the market is developing. It is attended by the Scrum Team members as well as internal and external stakeholders.

[8] A basic unit of development in Scrum, the planned Sprint is restricted to a duration of no more than a month.

velocity[9] is usually not considered to be a meaningful indicator. (See also Question 39 for a discussion of team velocity topic.)

However, although mostly indirect, there are various indicators that may be useful in determining success:

- Products delivered to customers are resulting in higher retention rates, better conversion rates, increased customer lifetime value, and similar improvements to the business. (A successful Scrum Team provides a good return on investment to the business.)
- The improved organizational agility allows pursuing market opportunities successfully, which previously would have been considered futile.
- There has been a reduced allocation of resources to low-value products.
- Lead time, from validated idea to shipped product, has been reduced.
- The cycle time[10] for hypotheses validations has been reduced, speeding up the product discovery process.
- Improved team happiness is exhibited by reduced churn and an increase in the number of referrals from team members.
- Increased competitiveness in the war for talent can be demonstrated by an increase in the number of experienced people willing to join the organization.
- Increased software quality can be demonstrated by measurably less technical debt[11], fewer bugs, and less time spent on maintenance.
- There is greater respect among stakeholders for the product delivery teams.
- Stakeholders are increasingly participating in events, for example, during the Sprint Review[12].

Q 03: Impediment Remover

Should a Scrum Master remove impediments on behalf of the Scrum Team?

[9] Team velocity is a measurement of work completed within a given time period based upon relevant comparisons.

[10] The cycle time is the number of days passed between starting and ending an experiment suitable to validate or falsify the underlying hypothesis.

[11] Technical debt is the extra development work that arises when code that is easier or faster to implement in the short run is used instead of applying the best overall solution.

[12] Part of the Sprint Review, the Scrum Team presents the completed work to the stakeholders.

A Scrum Master should not be concerned with removing problems that the Scrum Team can solve themselves, no matter how often this requirement is mentioned in job advertisements. If a Scrum Master acts like a 'Scrum mom', their team will never become self-organizing.

A Scrum Team must learn to make its own decisions. This almost inevitably results in failures, dead-ends, and other unplanned excursions when the team is learning something new. Consequently, in the beginning, a team will need more guidance than usual from the Scrum Master — and of a different kind than exemplified by drawing offline boards (see Questions 31 and 32) or updating tickets in JIRA[13]. Such guidance should not, however, become an exercise in protective parenting — a team must be allowed to learn from their failures.

That being said, there is one area where the Scrum Master is indeed removing problems on behalf of the team. This applies when the Scrum Team cannot solve the problem by themselves, for example, because the issue is an organizational problem. Now we are talking about "impediments." Only in this situation, the Scrum Master becomes the impediment remover of the Scrum Team.

Read more: Scrum Master Anti Patterns: Beware of Becoming a Scrum Mom (or Scrum Pop).

Q 04: Communication between SM and PO

How should a Scrum Master communicate with a Product Owner?

Communicating honestly and openly is the best way for a Scrum Master to get the cooperation of a Product Owner. Both must serve as servant leaders without being authoritative, and each depends upon the other working reciprocally for a Scrum Team's success (e.g. accomplishing a Sprint's Goal). They are allies with respect to coaching the organization to become, and remain, agile.

A Product Owner is responsible for providing prompt feedback on product matters, clarifying goals, and for ensuring that the entire product delivery team[14] understands the product vision.

[13] A proprietary issues tracking and project management software system, JIRA® is a registered trademark of Atlassian Pty Ltd.

[14] A product delivery team comprises everyone involved in delivering a product to market, including the Scrum Teams working on the product.

A Scrum Master, in return, supports the Product Owner in building a high-value, actionable Product Backlog, and to this end must facilitate effective collaboration between the Product Owner and the Scrum Team.

Q 05: The Product Discovery Process

Should the Scrum Team become involved in the product discovery process and, if so, how?

There are two principal reasons why a Scrum Team should be involved in the product discovery process as early as possible:

1. The sooner engineers participate in the product discovery process, the lesser the chances solutions will be pursued that are technically not viable or would not result in a return on investment.
2. Involving a Scrum Team early on ensures that the team and its Product Owner develop a shared understanding and ownership of what will be built. This helps significantly with allocating resources to the right issues, maximizing value for the customer, and mitigating investment risk by maximizing the amount of low-value work not done.

Involving the Development Team members early in the process ensures their buy-in, and the team's willingness to participate in all phases of a product's development. This motivates the team to participate when making changes necessary to accomplish the Sprint Goals defined for each Sprint or product release.

Q 06: Supporting the Product Owner

The role of the Product Owner is a bottleneck by design. How do you support the Product Owner so that they maximize value?

This question revisits the previous. Again, your candidate should focus on explaining why involving the Scrum Team early in the product discovery process is beneficial for both the Product Owner and the organization.

Additionally, Scrum Masters can effectively support Product Owners by ensuring that the Product Backlog refinement process is continuous and of a high value regarding the Product Backlog. "Garbage in, garbage out" does apply to Scrum.

Essentially, the Scrum Team either wins together or loses together.

Q 07: Access to Stakeholders

How can you ensure that a Scrum Team has access to a product's stakeholders?

When answering this question, your candidate should explain that there is no simple way to ensure access to stakeholders.

For example, in larger organizations, functional silos, budgeting and governance practices, and the organizational hierarchies often effectively limit team members' access to stakeholders. Overcoming this organizational debt, thus building trust among all participants, is a prime objective for the work of Scrum Masters.

Your candidate might suggest encouraging stakeholders to engage in effective (transparent, helpful) communication. Sprint Reviews are a useful venue for this, and the interaction often promotes better relationships between different departments and business units.

Q 08: Stakeholders and the Agile Mindset

How do you promote an agile mindset across departmental boundaries and throughout an organization and, in pursuit of that, what is your strategy when coaching stakeholders not familiar with IT?

There are various tactics a Scrum Master can use to engage stakeholders with Scrum, for example:

- Most importantly, a Scrum Master should live and breathe the principles of the Scrum Guide and the Agile Manifesto. They should talk to everyone in the organization involved in building the product, and they should be transparent about what they do. (**Read more**: 10 Proven Stakeholder Communication Tactics During an Agile Transition.)
- Product and engineering teams can produce evidence proving to stakeholders that Scrum is significantly reducing the lead time from idea to product launch.
- Product and engineering teams can demonstrate that Scrum mitigates risk (i.e. the forecast of when new features could be made available), thus contributing to other departments' successes in planning and execution.
- A Scrum Team can be transparent with respect to their work and proactively engage stakeholders by inviting them to Sprint Reviews and other events where the team communicates their activity or progress.
- Training for everyone in the organization, particularly the stakeholders, is important. One hands-on approach is to organize workshops designed to teach agile techniques for non-technical colleagues.

Q 09: Scrum and Senior Executives

How would you introduce Scrum to senior executives?

This is a deliberately open question meant to encourage discussion. In answering this question, your candidate should elaborate on how they would spread an agile mindset throughout an organization or, ideally, and more specifically, how they would create a learning organization that embraces experimentation in order to identify the best product for its customers.

A good candidate is likely to talk about the necessity of 'selling' agile to the organization in order to win the hearts and minds of the stakeholders. They will also point at the necessity to find a high-ranking executive to sponsor the transformation.

At the beginning of a transition any organization shows inertia to change, so to overcome this resistance executives and stakeholders need to know how Scrum will benefit them before they're likely to make a commitment. (**Read more**: The Big Picture of Agile: How to Pitch the Agile Mindset to Stakeholders.)

One practical approach when introducing Scrum to senior executives is to organize workshops for higher management levels. Applying Scrum at the executive level has been successful in the past. Executives, and potentially even key directors, can gain first-hand experience with agile practices if organized as a Scrum Team.

There are no right or wrong answers to this question. Good practices need to take into consideration an organization's culture, size, product maturity, legal and compliance requirements, and the industry it is operating in.

Q 10: Overcoming Stakeholder Resistance

You've already provided your product's stakeholders with training in Scrum. After the initial phase of trying to apply the concepts, when the very first obstacles are encountered, some of these stakeholders begin to resist continued adoption. What is your strategy for and experience in handling these situations?

This question is meant to encourage an exchange of ideas about, and lessons learned when overcoming resistance to Scrum within an organization. Familiarity with agile failure patterns that are common to many organizations will demonstrate your candidate's experience. (We have published a list of several agile failure patterns.)

Your candidate should also be familiar with the particular challenge middle managers face in any transition to agile practices. Moving from a command-and-control style (i.e. managing people and telling them what to do) to a servant-leadership style — thus abandoning Taylor's principles[15] — is not for everyone.

Read more: Why Agile Turns Into Micromanagement.

[15] F.W. Taylor's principles of scientific management are an industrial era organization and management theory according to which workers are seen as commodities and should be managed as such.

Set 2: Product Backlog Refinement and Estimation

Background

- Product Backlog refinement and estimation are essential tasks for every Scrum Team. Although the Product Owner (at least officially) is in charge of keeping the Product Backlog at 'peak value delivery', they need the assistance of the entire Scrum Team to do so.
- A cross-functional — be it distributed or co-located — Scrum Team working independently of other teams is an ideal scenario. The reality is that most Scrum Teams will often be dependent upon deliveries from other teams (e.g. API endpoints[16]) and deliverables from the UX[17] or UI[18] people if those are not embedded within a Scrum Team.
- There are two essential ingredients for good Scrum Team performance:
    01. **Writing user stories[19] as a team.** When a new feature should be built, the Product Owner first explains why, and provides the necessary background (i.e. market intelligence, results from experiments, user interviews, statistical data). Writing user stories, then, is a collaborative effort involving the entire Scrum Team. The process should create a shared understanding of what will be built and for what reasons (the Product Owner providing the 'why', the Scrum Team detailing the 'how', both negotiate the 'what'), and a shared sense of ownership among team members. A good team will always challenge the Product Owner whether a proposed functionality is indeed the best use of the Development Team's time. (Please note that not all Product Backlog items are user stories. There are, for example, also bugs, spikes, or non-functional requirements that do not fit into the user story template.)
    02. **Keeping technical debt at bay.** When a weak Development Team meets a commanding Product Owner, focusing on shipping new features, the team

[16] An Application Programming Interface endpoint is a URL, and the commands that may be issued through it, for use within software to instruct other software.

[17] User Experience is a process of tasks focused on the optimization of a product for improving user satisfaction.

[18] User Interface design complements the UX in the presentation and interactivity of a product.

[19] An Agile software development tool, a user story talks about the desired functionality of a requirement.

may end up as a feature factory, churning out new code while neglecting the technical foundation. A good Scrum Team pays attention to the preservation of an application's technical health to ensure the Scrum Team is ready to actually pursue an opportunity in the market. (Read more: Technical Debt & Scrum: Who Is Responsible?)

- A well-refined Product Backlog probably has work items detailed for about two or three Sprints in various refinement stages. There may also be additional work items that no one except the Product Owner is working on.
- Product Backlog refinement is a continuous process involving the Product Owner, the Development Team members and probably subject matter experts or stakeholders.
- A Product Backlog is "actionable" if the Scrum Team can organize a successful Sprint Planning at a moment's notice.

Q 11: External Requirement Documents

The Product Owner for your Scrum Team frequently turns requirements documents received from stakeholders into tickets, and asks you to estimate each. How do you feel about this procedure?

A Product Owner should not take this shortcut and turn requirements documents[20] received from stakeholders into work items, and a Scrum Master should never accept such a procedure. It's nothing more than a waterfall process[21] dressed-up as a pseudo-agile practice.

If an organization is supposed to focus on delivering value to its customers, it is essential that any process involving 'requirements' being handed down to its engineers by a project manager be abandoned. It makes no difference if the project manager is posing as a Product Owner. Instead, the organization should start including everyone in the product discovery process, thereby ensuring a shared vision of what needs to be built.

Q 12: PO Anti-Pattern

What kind of information would you require from the Product Owner in order to provide your team with an update on the product and market situation?

[20] Requirements documents might include, for example, software requirements specifications (SRS).

[21] The waterfall model is the sequential design process traditionally used in software development.

Information that a Scrum Master might require from a Product Owner when wanting to update their team on the product, or a market's reaction to it, would include any information that could provide the Scrum Team with an understanding of why something is of value to customers. Such information may be of a quantitative nature (e.g. analytical data describing how a process is utilized) or of a qualitative nature (e.g. transcripts, screencasts, or videos from a user testing session).

An excellent suggestion on the part of your candidate would be for the Scrum Team to participate in gathering qualitative signals by taking part in user interviews.

Please note: Normally, the Product Owner would provide this information during Sprint Reviews or the refinement process. Noting that the question Q12 itself is pointing at an anti-pattern, that would make a good topic for a Retrospective, is a bonus for the candidate.

Q 13: Writing User Stories

Who should be writing user stories?

Writing user stories should be a joint effort by all members of a Scrum Team—card, conversation, confirmation. If it's not, the team might not feel that they have ownership of the work items — inevitably leading to less or no commitment, reduced motivation, and ultimately a lower-quality product.

Additionally, handing down user stories reduces the accuracy of forecasts by the Development Team members as the joint creation process creates the shared understanding necessary.

Q 14: A Good User Story

What does a good user story look like? What is its structure?

A good user story:

- Includes a description,
- Has acceptance criteria defined,
- Can be delivered within a single Sprint,
- Has all UI deliverables available,
- Has all (probable) dependencies identified,
- Has performance criteria defined,
- Has tracking criteria defined, and
- Is estimated by the Scrum Team.

Q 15: INVEST

What does the acronym INVEST mean?

The INVEST acronym was coined by Bill Wake and describes the characteristics of a good user story:

- **Independent.** The user story should be self-contained, in a way that there is no inherent dependency on another user story.
- **Negotiable.** Until becoming part of an iteration, user stories can always be changed and rewritten.
- **Valuable.** A user story must deliver value to the end-user.
- **Estimable.** You must always be able to estimate the size of a user story.
- **Small.** User stories should not be so big as to become impossible to plan, task, and prioritize with some certainty.
- **Testable.** The user story (or its related description) must provide the necessary information to make test development possible.

Q 16: Person-hour Estimations

Why aren't user stories simply estimated in person-hours?

Estimating user stories in person-hours is rarely a good idea. It intentionally diverts the emphasis away from the true purpose of the estimation process: to create a shared understanding of the task ahead among all members of the Scrum Team. Ergo, the estimate itself is just a byproduct.

Estimating is often tricky when:

- Legacy software is involved,
- A team is facing significant technical debt, or
- A team is composed of mostly junior members.

Hence story points[22] are much better suited to estimating than man-hours in all situations, but especially in tricky situations, because they reflect both the complexity of the task and the effort required to complete it.

[22] Story points are units of measure expressing estimates of the overall effort required to fully implement a Product Backlog item.

Using person-hours instead of story points typically shifts the focus from value creation for customers to the more traditional project management of costs and budgeting, effectively imposing a waterfall process. Also, estimating in person-hours suggests an unmerited level of precision.

A good candidate would mention the ongoing discussion in the agile community as to whether estimations are useful in general. They would also likely point to the ‘no estimates’ concept.

Q 17: Cluttering the Product Backlog

The Product Owner of your Scrum Team tends to add ideas of all kinds to the Product Backlog as a reminder to work on them at a later stage. Over time, this has led to over 200 items in various stages. What are your thoughts on this? Can a Scrum Team work on 200 Product Backlog items?

Any Product Backlog larger than the scope of two or three Sprints is barely manageable. Misusing a Product Backlog by adding hundreds of items to it is a clear sign that the Product Owner needs help from the Scrum Team or the Scrum Master to better cope with an influx of ideas, suggestions, and requirements. A smaller Product Backlog avoids misallocating resources; a larger Product Backlog is an indication of waste.

Your candidate should make it clear that they would support a Product Owner in dealing with the size of the Product Backlog, and the ideation process in general, for example, processing input from stakeholders and customers.

Set 3: Sprint Planning

Background

- It used to be that a Product Owner would explain high-value user stories to the Scrum Team during Sprint Planning. The Scrum Team would then turn these into more detailed work items and probably the subsequent task. There is now, however, a consensus among practitioners that working on these high-level user stories continuously in separate Product Backlog refinement process — during the Sprint — actually improves the quality of the items and thus the outcome of the team's work.
- Sprint Planning creates a sense of ownership among a Development Team's members by enabling them to make a valid forecast regarding the Sprint Goal and subsequently the composition of the Sprint Backlog. But this only happens if a Scrum Team is certain about the quality of the Product Backlog items in question.
- It is the prerogative of the Development Team members to pick the Product Backlog items that compose the Sprint Backlog. No one can force work upon the Development Team.
- If Product Backlog refinement is handled well, an entire Sprint Planning session might be completed within than 2 or 3 hours.
- A productive Sprint Planning requires a healthy Scrum Team. Dysfunctional teams will not achieve the level of cooperation required. Sprint Planning with dysfunctional teams will only result in a futile and painful exercise.
- A Scrum Team should usually avoid allocating more than 80% of their capacity to working on new tasks — including user stories, technical tasks, bugs, and probably spikes[23]. Flow theory[24] shows that a 90% or higher allocation of available capacity will not lead to a team achieving their peak performance. A high-performing Scrum team needs slack time to be prepared for the unexpected or share knowledge among themselves.
- Bugs, refactoring, and research require regular attention in order to avoid building-up technical debt. An effective Scrum Team allocates approximately 20 to 25% of their capacity to these tasks.

[23] A spike is a small task done to reduce uncertainty about a larger task.

[24] 'Flow' is an optimal psychological state experienced, resulting in immersion and concentrated focus on a task.

- Incomplete and poorly prepared work items seriously hamper the effectiveness of a Scrum Team. These items should never be selected for the Sprint Backlog, but instead sorted out during Product Backlog refinement meetings.

Q 18: A Scrum Master's Contribution to the Sprint Planning

How can a Scrum Master contribute to Sprint Planning in a way that enables the Scrum Team to work only on the most valuable user stories?

It is the prerogative of the Product Owner to define the business objective of an upcoming Sprint by identifying and ranking the most valuable user stories in the Product Backlog, and it is the duty of the Scrum Master to support the Product Owner in this. Pursuant, a suitable way for a Scrum Master to support a Scrum Team's strive to work on the most valuable Product Backlog items is:

1. To ensure that the Scrum Team is involved in the product discovery process at an early stage;
2. To ensure that the Product Backlog refinement process is well practiced by both the Development Team members and the Product Owner; and
3. To ensure that all user stories are created in a collaborative effort between the Product Owner and the Development Team members (the goal being a shared understanding of the user stories and thus joint ownership).

Your candidate should note that although the Product Owner practically outlines the scope of the Sprint, it is the prerogative of the Development Team to address technical debt and bugs during the same Sprint. A Development Team should be able to allocate up to 25% of their available capacity for this. (**Read more**: Technical Debt & Scrum: Who Is Responsible?)

Q 19: Assessing the Value of a User Story

With what metrics would you assess the value of a user story?

There are quantitative as well as qualitative measurements that may be used to assess the value of a user story or whether the investment is worthwhile. These may include, for example:

- Revenue increases,
- Cost cutting benefits achieved by internal process improvements,

- Increases in customer satisfaction rates (NPS[25]),
- Increases in signups for new products, or
- Positive customer feedback received by the customer care team.

Q 20: Selecting the Most Valuable User Stories

How do you facilitate user story selection in a way that the most valuable stories are chosen without overruling the Development Team's prerogative to select the Sprint Backlog?

If a Development Team is involved early enough in either user story selection (preferably by jointly creating the stories with the Product Owner) or product discovery, a Scrum Master will probably not need to provide any guidance to see that the most valuable stories are chosen.

If a Development Team resorts to cherry-picking — choosing user stories only to satisfy personal preferences — during Sprint Planning, the Product Backlog refinement process needs to be seriously inspected. In all likelihood, the Product Owner is probably focusing on Product Backlog items that are not maximizing customer value.

Q 21: Time Allocation During a Sprint

How much of a Development Team's capacity during a regular Sprint would you consider adequate for refactoring? Fixing important bugs? Exploring new technologies or ideas?

Apart from Sprints during which there are critical and urgent tasks to address (such as fixing a problem that has taken the web site offline), a good rule of thumb is a 15-10-5 allocation of a Scrum Team's capacity to refactoring, fixing, and research. Specifically, this means dedicating:

- 15% of a team's capacity to technical debt and refactoring,
- 10% of a team's capacity to bugs, and
- 5% of a team's capacity to explorative spikes (when potentially helpful).

A Development Team may, of course, deviate from this rule of thumb depending on the context. Generally, consistently making these allocations will satisfy both the code quality and maintenance requirements of most software applications and build trust among stakeholders regarding the Scrum Team's capability to deliver valuable product Increments.

[25] Net Promoter Score ® is a customer loyalty metric and registered trademark of Fred Reichheld, Bain & Company, and Satmetrix Systems.

Q 22: Assigning User Stories

Should a Product Owner assign user stories or tasks to individual members of a Development Team?

A Product Owner individually assigning user stories to members of a Development Team is not Scrum, and if a Product Owner is doing this they need to be stopped. Development Teams are self-organizing. The assignment of user stories and the distribution of tasks among the members of a Development Team is the prerogative of the team itself. Preventing this anti-pattern should be one of the Scrum Master's most pressing concerns.

Q 23: Cherry-Picking Items

How do you deal with team members cherry-picking tasks?

A Development Team has autonomy in how its members choose to distribute tasks, so it may be that a presumed cherry-picking of tasks by individual team members is in fact a valuable and crucial part of the team's path to performance.

However, if team members are complaining about how the others are choosing their tasks, the Scrum Master needs to address the issue. Additional training might help some team members accommodate a greater variety of tasks. Or, perhaps, other team members may need to be gently pushed out of their comfort zone so that they will more readily choose different kinds of tasks over what they've become accustomed to. Pair programming may be a suitable first step in that direction.

An anti-pattern of this behavior is when specific tasks, such as quality assurance, are regularly left to the same team members. This pattern reintroduces sub-roles to the Development Team — and needs to be addressed by the Scrum Master.

Q 24: The Almost Ready User Story

A valuable user story is lacking the final user interface designs, but the design team promises to deliver on day two of the upcoming Sprint. The Product Owner for your team is fine with that and pushes to have the user story added to the Sprint Backlog. What are your thoughts on this scenario?

Whether an incomplete user story should be added to the Sprint Backlog depends upon the Development Team's present concerns and experience with the circumstances. In the case of an incomplete or missing user interface (UI) design, for example, if the design team is almost certain to deliver because they have done so in the past, and if the user story is high value, and if the story can be accomplished within the Sprint despite its UI deliverables arriving late, and if the Development Team agrees to it — then an exception may be acceptable.

Beware that exceptions have a tendency to become accepted practices. An organization's intent on being agile should not be allowed to bypass the Product Backlog refinement and Sprint Planning process. Your candidate should be aware that such situations are not tenable. Furthermore, if the implementation of a work item subjected to such an exception fails, no one will bother to read the fine print and acknowledge that an exception had been made. Instead, they will most likely view the Scrum process itself as having failed.

Your candidates may either accept or reject exceptions to the process. But they should also be able to analyze situations in which exceptions have been made, and explain the collateral damage that the Scrum Team may be exposed to.

Q 25: Sprint Planning Is a Waste of My Time

A member of the Scrum Team does not want to participate in Sprint Planning and considers the meetings a waste of time. How do you deal with this attitude?

If s Developer does not want to participate in Sprint Planning and considers the meetings a waste of time, they're exhibiting a type of passive-aggressive behavior. Although not particular to Scrum, this is a problem because the underlying attitude is toxic and will affect both team-building and team performance as now the knowledge of a team member will not be available at Sprint Planning.

When an individual behaves as described, the team's Scrum Master needs to take action. Counterproductive behavior can neither be ignored nor tolerated if the team is to continue functioning. Effective action is likely to require probably a series of escalating steps:

1. The Scrum Master may start by addressing the team member privately to discuss their reservations and, perhaps, more coaching needs or a longer training period.
2. Following private discussion, the entire Scrum Team can be involved by making the team member's reservations a topic of discussion during one or more Sprint Retrospectives. This enables the Scrum Team to offer their support to their teammate.
3. If there is still no change in the team member's attitude, a meeting with the team member and their line-manager is advisable.
4. If no change can be achieved, it might be possible to reassign the team member to another (probably non-agile) team, or to a Kanban team unlikely to force the team member out of their comfort zone.

Situations such as described highlight how Scrum is not meant for everybody.

Set 4: Daily Scrum

Background

- Daily Scrums events are essential to discuss a current Sprint's progress: is all going as planned, or does the Development Team need to adjust its approach to accomplish the Sprint Goal?
- Daily Scrums cannot fix — and is not supposed to fix—, among other things: a dysfunctional organization, a dysfunctional Scrum Team, an inadequate Product Backlog, a Sprint Planning session gone wrong, low-quality user stories, or a missing product vision.
- It is the prerogative of the Development Team to decide on the best way of handling their Daily Scrum event.
- The more experienced a Scrum Team, and the better the internal communications, the more a Daily Scrum might be seen as a time-consuming ritual of little value.
- An advanced Scrum Team may consider virtual meetings instead of real meetings using, for example, a Slack[26] channel.
- Just saying: A two-person Scrum Team probably does not need a formal Daily Scrum — meeting regularly for coffee would be a practical alternative.
- There is something wrong with a Scrum Team who does not communicate impediments to their Scrum Master prior to each Daily Scrum.
- Daily Scrums are not reporting sessions for the benefit of the Product Owner or participating stakeholders.
- Offline boards are valuable: physically taking a card and moving it instills certain ownership of a work item.

Q 26: The Formal Daily Scrum

Would you recommend formal Daily Scrums for all teams, no matter the size or experience level?

In answering this question, your candidate should exhibit common sense regarding "ritualized" Daily Scrums. Daily Scrums are an important part of Scrum, but not all Daily Scrums need to be formal — a Development Team should not have a Daily Scrum for the

[26] Slack is a popular messaging software for team communication and collaboration.

sake of having it; it serves a different purpose than ticking off a box on a checklist. A small, experienced, and co-located team may use a morning coffee break for their Daily Scrum.

Nevertheless, the Daily Scrum is the essential inspect & adapt event of the Development Team: are we still on track accomplishing the Sprint Goal? Or have we learned something since the previous Daily Scrum that requires to change our plan of how to achieve the Sprint Goal?

Q 27: Impediments

Do you expect experienced team members to wait until the next Daily Scrum in order to ask for help overcoming an impediment?

When impeded, members of a Scrum Team should never need to wait, neither for a Daily Scrum nor any other event, to ask for help. A team waiting to ask for help is a team delaying progress. If the more experienced members of a Scrum Team are waiting for the next Daily Scrum before either asking for help or themselves dealing with an impediment, the Scrum Master has team-building work to do.

Q 28: Leading a Daily Scrum?

How do you handle team members who 'lead' Daily Scrums, turning the event into a reporting session for themselves?

There are no leadership roles in the Development Team. However, it's not uncommon for some members of a Development Team to assume leadership. This typically happens when a particular team member possesses superior (technical) expertise, communication skills, or simply a greater level of engagement.

All teams go through Tuckman's stages of group development: forming, norming, storming, and performing. Scrum Teams are no exception.

It's important that when a member of a Development Team assumes leadership this does not result in other members reporting to them. A Scrum Master must be vigilant and intervene if necessary to ensure that all team members communicate and work together — during Daily Scrums and otherwise — in the spirit of Scrum.

Q 29: Waste of My Time?

How do you manage team members who consider Daily Scrums to be a waste of time and are therefore either late, uncooperative, or simply don't attend?

Refer to Question 25, where addressing this similar attitude and behavioral problem is discussed at length. Your candidate's answers should address similar points.

Q 30: Stakeholder Attendance

Your team's Daily Scrums are not attended by any stakeholder. Should that change?

Asking this question can easily spark a philosophical discussion about whether stakeholders should be allowed to participate in a Development Team's Daily Scrums. Try to avoid this.

If stakeholders participate in a Development Team's Daily Scrums, is it likely to result in a form of reporting that circumvents Scrum rules? Not necessarily. It's good if some adaptation of Scrum can be made to work for an organization. Allowing stakeholders to participate in Daily Scrums need not be ruled out if the Development Team finds it acceptable. In fact, if stakeholders attend Daily Scrums regularly, this invariably and significantly improves communication between a team and their stakeholders.

So shall a Scrum Master encourage stakeholders to attend Daily Scrums? That depends on the context; your candidate should not rule out their participation immediately.

Q 31: Daily Scrum with Distributed Teams

How do you approach Daily Scrums with distributed teams?

Daily Scrums for Development Teams whose members are distributed between different offices or working remotely are not much different from Daily Scrums for Development Teams whose members are co-located. The exception is that distributed teams sharing board activity may require video conferencing when working with offline boards that mirror each other.

If a Scrum Team is using online task management or planning software like JIRA[27], the team's boards can be online and updates can take place on-screen. This generally makes it easy for members of a distributed team to follow board activity. With online boards in place, a Zoom or Google Hangouts call will likely be enough for a distributed team to have their Daily Scrum.

[27] JIRA ® is a proprietary issue tracking and project management software, and a registered trademark of Atlassian Pty Ltd.

Alternatively, the Development Team may try an asynchronous Daily Scrum by utilizing messenger software like Slack. It is the prerogative of the Development Team to decide on the best way of handling their Daily Scrum event.

Q 32: The Scrum Board

Can you draw an example of a Scrum Team's Kanban board — right now?

In this question, the qualifier 'Kanban' is used as a teaser. Anyone interviewing for the role of Scrum Master should be able to draw a simple Sprint board.

The columns of a Sprint board usually include columns such as :

1. Backlog of tasks,
2. Task In progress,
3. Code review,
4. Quality assurance,
5. Done.

Additional information may be included on or attached to any kind of board, for example:

- Scrum Team members,
- Sprint or event dates,
- Definition of "Done,"
- A burndown chart (progress and work remaining over time).
- A parking lot (topics for future discussion).

Your candidate should mention that a Scrum Master is not obliged to provide the Scrum Team with a Sprint board. A board is the responsibility of the Development Team working with it. The Scrum Master should, however, support the effort with an introductory workshop on the subject if no member of the team is familiar with offline boards.

Read more: How to Build Offline Boards.

Set 5: Sprint Retrospectives

Background

- "The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint." (Source.)
- Retrospectives should encourage self-expression, thereby making it easier for a Scrum Team to uncover the concerns and frustrations that its members may be harboring so that strategies may be devised to overcome them. (Learn more on Retrospectives.)
- Retrospectives will only improve a team's collaboration and performance if the team considers these meetings a safe place to provide honest and constructive feedback.
- The blame game is hence not helpful. During a Sprint Retrospective, the members of a Scrum Team should focus on how to improve a situation — and avoid blaming one another.
- There are various dedicated applications available to support Retrospectives with distributed Scrum Teams.
- Alternatively, a Scrum Master can at any time handcraft a Retrospective format using Liberating Structures — which works well for co-located and distributed Scrum Teams.
- It's best not to hold Sprint Retrospectives at a team's workplace. Distance makes it easier for team members to reflect on the Sprint. It's also helpful to regularly change locations for the meeting. Being in a new locale helps to prevent boredom (and team members 'checking out' completely).
- The format for a Scrum Team's Sprint Retrospectives should be changed regularly. (**Read more**: Retrospective Exercises Repository.)
- For co-located Scrum Teams, Smartphones, tablets, and laptops should not be permitted at Sprint Retrospectives so that the members of the Scrum Team are not distracted, and can focus on contributing to the meeting.
- According to Diana Larsen and Esther Derby in their book Agile Retrospectives: Making Good Teams Great, there are five stages to running a Sprint Retrospective: setting the stage, gathering data, generating insights, deciding what to do, and closing the Sprint Retrospective.
- All issues, concerns, and frustrations, should be documented — even if just temporarily using sticky notes. Though it's always better to keep a formal document or file. (Limit access to these documents to the Scrum Team members, though.)
- Retrospectives shall produce answers to certain questions. The **'classic'** set of questions includes
  - What went right?
  - What went wrong?

    - What is there to improve?
- An alternative **'classic'** set of questions is the **'starfish'** Sprint Retrospective:
    - What to introduce?
    - What to keep doing?
    - What to stop doing?
    - What to do more of?
    - What to do less of?
- Yet another alternative to asking questions at a Sprint Retrospective is to employ the Mad Sad Glad[28] technique. This technique works best following either:
    - A long interval (e.g. at the end of the year),
    - A major change,
    - A major drawback,
    - Unusual pressure, or
    - An outstanding achievement made by the team.
- A Sprint Retrospective should set SMART[29] goals for action items (the tasks to be done):
    - Action items should be specific and measurable ("do X more often" does **not** meet that criteria).
    - A single member of the Scrum Team should be made responsible for each action item.
    - Each action item should include a forecast of when results can be expected.
    - Action items should be placed on a board to make tracking progress visual and more prominent.
    - Every new Sprint Retrospective should start with reviewing the status of the action items decided upon during the previous Sprint Retrospective.
- Don't forget to includes stakeholders in meta-level Retrospectives from time to time.

Q 33: Participants of a Retrospective

Who should participate in a Sprint Retrospective?

Only the immediate members of a Scrum Team — Development Team members, Product Owner, and Scrum Master — should participate in that team's Sprint Retrospectives.

[28] Mad Sad Glad is a Sprint Retrospective exercise designed to elicit feedback and possible corrective actions.

[29] SMART is a mnemonic for various acronyms that generally provide guidelines to be used during the process of setting goals.

Especially noteworthy is that the line-managers of a Scrum Team's members not be present. Also, they should not be allowed access to the minutes of any Sprint Retrospective.

Q 34: Team Health

Should you check a team's health during a Sprint Retrospective, or is doing so unnecessary? If you do, how would you go about it?

Measuring the health of a Scrum Team — that is, getting an idea about current levels of engagement and satisfaction — is useful for identifying trends that may affect productivity and team cohesion.

For example, one effective method of measuring the health of a Scrum Team is to circulate an anonymous multiple-choice questionnaire before the team's Sprint Retrospectives. A questionnaire that requires just two minutes to complete and uses a simple scale for each of the questions — from 1 (terrible) through 2 (poor), 3 (neutral), 4 (good), to 5 (excellent) — is usually well-suited.

During the Sprint Retrospective, the team should discuss the results with an aim to uncover any concerns or frustrations they may be harboring. (See above, gathering data.)

Q 35: Retrospective Formats

What Sprint Retrospective formats have you used in the past?

There are various Sprint Retrospective formats in common use, and each is meant to accommodate different situations. Your candidate should have experience applying more than one of these formats and should be able to share their logic for having done so. Some basic formats for Retrospectives include:

The classic format:

- What did we do well?
- What should we have done better?

The boat format:

- What's pushing us forward?
- What's holding us back?

The starfish Sprint Retrospective:

- Start doing…

- Do less of…
- Do more of…
- Stop doing…
- Continue doing…

You can embed all of these formats in the general Sprint Retrospective format popularized by Diana Larsen and Esther Derby:

- Set the stage.
- Gather data.
- Generate insights.
- Decide what to do.
- Close the Sprint Retrospective.

There are several websites available that help Scrum Masters to customize Retrospectives to the needs of the Scrum Team, such as Retromat or Tasty Cupcakes. Alternatively, Liberating Structures provide excellent tools, too.

Suitable candidates will elaborate passionately about their preferred ways and tools for delivering Retrospectives. Candidates that provide only mechanical answers require more scrutiny as the Sprint Retrospective is a key event from a Scrum Master's perspective.

Q 36: Retrospective Fatigue

How do you prevent boredom during Sprint Retrospectives?

When required to attend a uninspiring Sprint Retrospective, members of a Scrum Team will become bored.

There are many possibilities for variation that can be used to prevent a Sprint Retrospective from being boring, and team members from becoming bored. A different location, a different format, and shortening or lengthening the allotted time box are just some of the variations that can be tried.

Scrum Masters might also use a team's choice of action items to encourage and structure discussions around issues that matter to the team, thus creating engagement through acknowledgment. Web sites like Retromat offer hundreds of different games and exercises to make Sprint Retrospectives enjoyable and valuable for the whole team.

There is no single solution, and consequently no single correct answer, to either boredom or this question. What's important is that your candidate acknowledges that boredom with routine might become an issue and that there are ways to deal with it.

Read more: How to Curate Retrospectives for Fun and Profit With Retromat.

Q 37: Not Delivering on Action Items

If your team is picking reasonable action items but not delivering, how would you address the situation?

During a Sprint Retrospective, the members of a Scrum Team would usually pick some action items — tasks to be done — and include them in the upcoming Sprint Backlog. If these action items are subsequently not completed in a timely manner, the Scrum Master needs to follow up.

A team might not be completing the action items they've picked because they've run into an external impediment. If this is the case, the Scrum Master has to address the cause, and the team can then catch up during a later Sprint.

However, if there is no external impediment, the problem is likely due to motivation, attitude, or personal issues within the Scrum Team. In this latter case, the Scrum Master needs to provide the team members with sufficient encouragement or motivation to overcome the problem — a Scrum Team is self-organizing.

If a team is not completing the action items they've picked and the problem ultimately cannot be resolved, picking action items becomes a useless exercise and the Scrum Team's continuous improvement effort will suffer as a result.

Q 38: Follow-up on Action Items

Would you recommend following up on action items? If so, how would you do that?

The Scrum Team is self-organizing. However, there are always moments when working on improving its practices is less of a Scrum Team's priority. In this situation, a Scrum Master should follow up on the action items — tasks to be done — that members of a Scrum Team pick during their team's Sprint Retrospective to remember everyone that Scrum is not working without self-organization.

A good way for a Scrum Master to do this is to start talking about the status of the action items picked during the last Sprint Retrospective before picking new ones by initiating a discussion at the beginning of each new Sprint Retrospective. (Note: This is not meant to be a reporting session but practical help to get self-organization going with the Scrum Team.)

Suppose this discussion uncovers action items picked during a previous Sprint Retrospective that haven't been completed as expected. In that case, the team needs to understand why this happened and offer its support to prevent it from happening again.

Set 6: Agile Metrics

Background

- The purpose of metrics, generally, is to understand a current situation better and gain insight on how it’s likely to change over time.
- A good metric is a leading indicator for a pattern, providing an opportunity to analyze the cause for change — and act appropriately in due course.
- Metrics in an agile context are not used to manage, and certainly not micromanage, an individual (particularly the creative worker) — contrary to traditional command-and-control management structures.
- Metrics in an agile organization should be used to provide the Scrum Team with insights on how to continuously improve, helping them achieve their goals:
    - Agile practitioners strive for autonomy, mastery, and purpose as explained by Daniel Pink.
    - Agile practitioners address personal development with metrics by applying practices like Objectives and Key Results (OKR).
- The experienced agile practitioner realizes that autonomy and responsibility are equally important for self-organized Scrum Teams. Without metrics, both autonomy and responsibility are limited.
- The metrics most suitable reflect either a team’s progress in becoming agile or the organization’s progress in becoming a learning organization.
- Both qualitative and quantitative metrics may be used for Scrum:
    - Qualitative metrics typically reveal more than quantitative metrics when applied to the Scrum Team.
    - Quantitative metrics provide more insight than qualitative metrics when applied to the organization.
- Any agile metric used must be tailored to the organization.
- The metrics that Scrum Masters should track are only those that apply to the Scrum Team as a whole. Metrics that measure the individual should be ignored.
- A metric’s context should always be recorded to avoid misinterpretation.
- Parameters that are easy to follow should not be measured for that reason alone — especially if a report is readily available in the project management software being used.

Q 39: Volatile Velocity

Your Scrum Team is consistently failing to meet forecasts and Sprint Goals, and their velocity is volatile. What are the probable reasons for this problem, and how would you address it with the team?

If a Scrum Team is exhibiting a volatile velocity, consistently failing to meet their forecasts, it suggests that velocity is being used as the prevalent metric for measuring that team's progress.

Your candidate should mention this, and talk about the notoriety of 'velocity' as the industry's most prevalent metric for measuring a team's progress. They should further be able to explain why velocity is altogether a doubtful agile metric, and point out that quantitative metrics are not ideally suited to measuring a team's progress in mastering Scrum.

There are many factors that make a Scrum Team's velocity volatile:

- New team members being onboarded;
- Experienced members leaving the team;
- The team working in uncharted territory;
- The team working with legacy code, probably undocumented;
- The team running into unexpected technical debt;
- Holidays and sick leave reducing the team's capacity;
- An executive intervention changing a Sprint's scope; and
- The team addressing unplanned priority bugs.

Another common cause for a Scrum Team to consistently fail in meeting their forecasts is that the team's Product Backlog items are being poorly prepared, thus making the work items difficult for the team to estimate. Conversely, the projects being given the team might suffer from poorly documented legacy code, excessive technical debt, or just too much buggy and poorly written code — all of which make estimation a gamble.

Your candidate should not align themselves with the fallacy that a team's adoption of Scrum is working only because a Scrum Team's forecasts and velocity are aligned. Cooking the agile books is easy to do!

Read more: Scrum: The Obsession with Commitment Matching Velocity.

Q 40: Suitable Agile Metrics

What suitable agile metrics have you used in the past?

This question is an invitation to the candidate to share lessons learned from the successful application of metrics to help a Scrum Team improve continuously.

Suitable agile metrics follow three rules:

- The first rule of tracking meaningful metrics is only to track those that apply to the team. Ignore those metrics that measure the individual.
- The second rule of tracking metrics is not to measure parameters just because they are easy to follow. This practice often is a consequence of using various agile tools that offer out-of-the-box reports.
- The third rule of tracking metrics is to record context as well. For example, data without context, the number of available team members, or the intensity of incidents during a sprint may turn out to be nothing more than noise.

Examples of suitable agile metrics are:

- Lead time,
- Cycle time,
- Number of defects escaping to production, or
- Ratio of fixing work to creating new value.

Good candidates should be aware of the evidence-based management concept.

Q 41: Qualitative Metrics

What qualitative agile metrics would you consider tracking?

The purpose of qualitative metrics is to gain insight into how one or more of an organization's Scrum Teams are progressing with agile.

There are several self-assessment tests available that a Scrum Team can regularly run to collect qualitative metrics about their implementation of Scrum — Hendrik Kniberg's Scrum Checklist is a good example. The interval to test via self-assessment is every 4–12 weeks, with teams of lesser fluency running their tests at the lower end of this range. The individual values recorded by these tests are not very important, but the trend over time is. To visualize these trends, a Scrum Master will need to aggregate the results — in the case of Henrik Kniberg's checklist, an agile practice map may be created over time.

While self-assessment tests like Henrik Kniberg's checklist are usually team exercises for recording **implementation** metrics, **sentiment** metrics are best captured by running anonymous opinion polls to ensure the participation of the more introverted team members.

Using opinion polls, typical questions for recording sentiment metrics include:

- What value did the team deliver in the last Sprint?
- Has the level of technical debt increased or decreased during the last Sprint?
- Are you happy working with your teammates?
- Would you recommend your employer (or client) to a friend seeking a new job?

It’s best to run opinion polls after every Sprint; these polls should only require a few seconds to complete. As with the self-assessment tests, the individual values recorded by running anonymous opinion polls are not very important — it’s the trend over time that matters. Trends derived from these polls are great points for discussion during a team’s Sprint Retrospectives.

Concerning metrics in general, your candidate should support the Agile Manifesto and its principle of transparency: all metrics should be available to all members of a Scrum Team, and largely also to those working in the product delivery organization[30] generally.

Read more: Agile Metrics — The Good, the Bad, and the Ugly.

[30] A product delivery organization is essentially everyone within an organization who’s involved with getting a product to market.

Set 7: How to Kick-off a Transition to Scrum

Background

- There is no checklist or master plan readily available that would ensure a successful transition to Scrum.
- The ‘best practices’ of and ‘lessons learned’ by other organizations may indicate a direction to take when transitioning, though the context of their transition may not be comparable: what worked for Spotify may not work for General Motors.
- Every transition to Scrum should start with understanding the ‘why’: why should the organization become agile?
- Reasons typically given by management for transitioning to Scrum and other agile practices include:
  - Making the organization more efficient;
  - Helping the organization deliver faster; and
  - Improving the predictability of delivery dates.
- The recognized benefits of transitioning to Scrum and other agile practices are:
  - Outperforming competitors by creating a learning organization;
  - Creating a great workplace culture by providing room for autonomy, mastery, and purpose; and
  - Mastering continuous product discovery and delivery (thus minimizing risk).
- Agile and its benefits need to be sold to an organization before beginning its transition to Scrum — Agile is not everybody’s darling, and personal agendas of individuals will likely affect a transition.
- A transition to Scrum will encounter inertia and resistance to change directly proportional to the size of the organization.
- How a transition to Scrum should be undertaken depends upon many factors, including: an organization’s industry, regulations and compliance rules, the size and age of the organization, workplace culture, the maturity of an organization’s products and services, team size, and current project management practices.
- How a transition to Scrum is undertaken should be determined by the goals of the organization — what is hoped to be achieved.
- A successful transition to Scrum requires the backing of C-level executives; a bottom-up approach is futile.
- The first step of any transition to Scrum is the creation of the first Scrum Team.
- Transitioning to Scrum requires training and educating the entire organization — not just future Scrum Team members — in agile practices and principles. Training and education are essential for a successful transition.
- There is a huge difference between ‘doing Agile’ and ‘being agile’. Transitioning to Scrum successfully means becoming — and being — agile.

- In an organization transitioning to Scrum, future Scrum Masters should be agents of change rather than drill sergeants — this is by design, given their lack of proper authority.
- Creating a ‘happy agile island’ for the product and engineering department is a valid objective. However, in comparison to breaking up functional silos and creating a learning organization, a team of teams, it is likely to deliver a lesser return on investment.

Q 42: Kicking off Scrum

How would you prepare to kick off a transition to Scrum?

If you don’t know where you are going, any road will get you there. Your candidate should understand that an agile transition needs to have an objective and a goal — which means planning ahead.

To prepare for kicking off a transition to Scrum is to listen and observe: your candidate should express interest in interviewing as many team members and stakeholders as possible, before jumping into action. These interviews should include everyone, no matter their role — engineers, QA professionals[31], UX and UI designers, product managers — in order to identify the patterns underlying current problems, failures, and dysfunction within the organization. Merging those patterns with the most pressing technical and business issues will identify the most likely objectives for the first Scrum Teams. This observation phase, during which a Scrum Master performs their interviews, will typically require between four and twelve weeks depending upon the size and structure of the organization.

The training of future team members and stakeholders should commence and run parallel to the interviews.

Creating the first Scrum Teams from the existing engineering and product departments is the second step in kicking off a transition to Scrum.

Your candidate should be able to sketch the rough plan of a transition, and address common issues that might arise during kickoff.

Read more: How to Kick-off Your Agile Transition.

[31] Quality assurance professionals focus on quality earlier in the development process during defect prevention.

Q 43: Creating the First Scrum Team

How would you create the first Scrum Team?

When an organization is transitioning to Scrum and at the same time dealing with significant organizational, business, and technical problems, the founding members of its Scrum Teams should be volunteers who fully understand the challenge ahead of them, rather than people pressed into service. The best volunteers are those eager to prove that becoming agile is the most effective way to reach an objective.

Candidates for the role of Scrum Master should be astute enough to suggest inviting every member of the product delivery team, as well as the C-level executives sponsoring the transition, to a kickoff meeting. The objective of a transition kickoff meeting is to support the members of the engineering and product teams in how they choose to self-organize into the first cross-functional Scrum Teams. Transition kickoff meetings can last a few hours or several days, depending upon the circumstances of a particular organization.

Despite the importance of the kickoff meeting to a Scrum transition, going much deeper into its structure will take too much time from the interview. It's more important that your candidates embrace the idea of team self-selection and present a brief roadmap of what should happen next for the newly formed Scrum Teams.

Although somewhat dependent upon the existing skills, experience, and training of the members of an organization's new Scrum Teams, your candidates should anticipate having to teach the very basics of Scrum following a kickoff meeting. They might propose doing this through a series of workshops or on-the-job training with exercises in Product Backlog refinement, writing user stories, estimating, creating boards, and setting up collaboration software.

Q 44: First Steps of a New Scrum Team

What do you recommend a newly formed Scrum Team works on first?

The first critical issue for the majority of newly formed Scrum Teams is the existing legacy Product Backlog. Answers to this question need not reference Tuckman's team development stages (see Question 28), additional team building exercises, or any kind of Scrum training or workshop not concerned with the Product Backlog.

It is a rare occasion for a Scrum Master to start from scratch with a brand new team and no existing product — even more so in a nascent organization like a startup. Most often, it's an existing product delivery organization with existing products and services who will 'go agile'.

For these cases your candidate should point out that refining the legacy Product Backlog is the practical first step.

The legacy Product Backlog per se is an interesting artifact because it provides comprehensive insight into the product delivery organization's history: this particular Product Backlog allows for identifying organizational debt, process insufficiencies, questionable product decisions, and other anti-patterns.

Looking at a legacy Product Backlog, an excellent candidate will be able to point out some of these anti-patterns (e.g. outdated or poorly maintained tickets), and provide a good idea about how to transform the legacy Product Backlog into a well-refined, current Product Backlog such that a new Scrum Team could work with.

Candidates should mention that running Product Backlog refinement workshops creates a good opportunity to provide a new Scrum Team and Product Owner hands-on training with Scrum. This is because a Product Backlog refinement workshop will typically cover user story creation, knowledge transfer among team members, the estimation process (if applicable), introductory agile metrics, technical debt analysis, and other topics critical to success of Scrum.

Read more: Product Backlog Refinement.

Set 8: Scrum Anti-Patterns

Background

- Humans are fallible, so with this propensity for error there will always be room for (professional) improvement – including Scrum Masters.
- Anti-patterns will emerge when core principles (as laid out in the Manifesto for Agile Software Development and the Scrum Guide) are ignored, made to fit existing structures, or watered down.
- The deterioration of principles may be a deliberate process (creating a form of cargo cult agile), unintentional, or a result of good intentions applied in the wrong way.
- Whatever the deterioration process, emerging anti-patterns will prevent an organization from reaping the benefits of agile software development.
- Recognizing Scrum and agile anti-patterns is therefore fundamental in the effort for serious, continuous improvement.
- Anti-patterns can be identified by observation, Sprint Retrospectives, and other forms of feedback generating activities.

Read More: The "Scrum Anti-Patterns Guide" is another free ebook from the "Hands-On Agile" series of practical guides from the trenches. It covers more than 160 Scrum anti-patterns that can block your Scrum Team's improvement.

Q 45: Scrum Master Anti-Patterns

What Scrum Master anti-patterns do you know?

Typical Scrum Master Sprint anti-patterns are below. Any of these behaviors will impede the team's productivity. It is the Scrum Master's obligation to prevent them from manifesting themselves. Some of the Scrum Master anti-patterns are:

- **Keeping the Scrum team dependent:** In this scenario, the Scrum Master pampers the team to a level that keeps the team dependent on his or her services: organizing meetings, purchasing stickies and sharpies, taking notes, updating Jira—you get the idea of this service level. More critical, however, is when the Scrum Master decides to keep the team in the dark about principles and practices to secure his or her job. This behavior is only a small step away from the dark side.
- **Flow disruption**: The Scrum Master allows stakeholders to disrupt the workflow of the Development Team during the Sprint. There are several possibilities on how stakeholders can interrupt the flow of the team during a Sprint:

    - The Scrum Master has a laissez-faire policy regarding access to the Development Team.
    - The Scrum Master does not object when management invites engineers to random meetings as subject matter experts.
    - Lastly, the Scrum Master allows either the stakeholders or managers to turn the daily Scrum into a reporting session.
- **Lack of support**: The Scrum Master does not support team members who need help with a task. Development teams often create tasks an engineer can finish within a day. However, if someone struggles with a task for more than two days without voicing that they need support, the Scrum Master should address the issue. Importantly, this is also the reason for marking tasks on a physical board with red dots each day if they haven't been moved on to the next column.
- **Turning a blind eye to micromanagement**: The Scrum Master does not prevent the Product Owner – or anyone else – from assigning tasks to engineers. The Development Team normally organizes itself without external intervention. And the Scrum Master should act as the shield of the team in this respect.
- **Focusing on team harmony:** The Scrum Master sweeps conflict and problems under the rug by not using Sprint Retrospectives to address those openly. This behavior is often a sign of bowing to politics and instead of using manipulation to meet organizational requirements that are opposing Scrum values and principles. If the organization values its underlings for following the 'rules' instead of speaking the truth why would you run Retrospectives in the first place? A 'Scrum Master' participating in cargo-cult Scrum is again a supervisor than an agile practitioner.

Read more: Scrum Master Anti-Patterns — 20 Signs Your Scrum Master Needs Help.

Q 46: Product Backlog-Related Scrum Anti-Patterns

As a Scrum Master, what are some of the Product Backlog-related Scrum anti-patterns that you need to keep at bay?

Garbage in, garbage out: No matter how well your team is self-managing, how low your level of technical debt is, or how well your team is collaborating with stakeholders in general, your team will be measured primarily by one criterion: can the Scrum team regularly deliver valuable, done Product Increments? The key to live up to that expectation is an actionable Product Backlog which is compact, concise, continuously refined in a team effort, and focussed on delivery of valuable Increments.

According to the Scrum Guide, Scrum Masters support the Product Owner in many ways to ensure this level of Scrum fluency:

- *Page 7: The Scrum Master serves the Product Owner in several ways, including helping find techniques for effective Product Goal definition and Product Backlog management.*

- *Page 7: The Scrum Master serves the Product Owner in several ways, including helping the Scrum Team understand the need for clear and concise Product Backlog items.*

- *Page 7: The Scrum Master serves the Product Owner in several ways, including helping establish empirical product planning for a complex environment.*

- *Page 7: The Scrum Master serves the Product Owner in several ways, including facilitating stakeholder collaboration as requested or needed.*

Source: Scrum Guide 2020. (The aggregation is taken from the Scrum Guide 2020 Reordered.)

Typical examples of how organizations, Scrum team, or team members fail the principles mentioned above include:

1. **Prioritization by proxy**: A single stakeholder or a committee of stakeholder prioritizes the Product Backlog. (The strength of Scrum is building on the strong position of the Product Owner. The Product Owner is the only person to decide what tasks become Product Backlog items. Hence, the Product Owner also decides on the ordering of the work items. Take away that empowerment, and Scrum turns into a pretty robust waterfall 2.0 process.)

2. **100% in advance**: The Scrum team creates a Product Backlog covering the complete project or product upfront because the scope of the release is limited. (Question: how can you be sure to know today what to deliver in six months from now?)

3. **Over-sized**: The Product Backlog contains more items than the Scrum team can deliver within three to five Sprints. (This way the Product Owner creates waste by hoarding issues that might never materialize.)

4. **Storage for ideas**: The Product Owner is using the Product Backlog as a repository of ideas and requirements. (This practice is clogging the Product Backlog, may lead to cognitive overload and makes alignment with the 'big picture' at portfolio management and roadmap planning level very tough. It also may lead to less collaboration as team members may consider the Product backlog to be 'complete.')

5. **Copy & paste PO**: The Product Owner creates Product Backlog items by breaking down requirement documents received from stakeholders into smaller chunks. (That

scenario helped to coin the nickname “ticket monkey” for the Product Owner. Remember: Product Backlog item creation is a collaborative team exercise.)

6. **What team?** The Product Owner is not involving the entire Scrum team in the refinement process and instead is relying on just the “lead engineer” (or any other member of the Scrum team independently of the others).

7. **Submissive team**: The Developers submissively follow the demands of the Product Owner. (Challenging the Product Owner whether their selection of issues is the best use of the Scrum team’s time is the noblest obligation of every team member: Why shall we do this? Is this the best use of our time from a customer perspective?)

8. **No time for refinement**: The Scrum team does not have enough refinement sessions, resulting in a low-quality Product Backlog. (The Scrum Guide advises to spend sufficient time on refining the Product Backlog continuously. Which is a sound business decision: Nothing is more expensive than a feature that neither delivers value to customers nor supports the organization’s strive to create a sustainable business.)

9. **Too much refinement**: The Scrum team has too many refinement sessions, resulting in a too detailed Product Backlog. (Too much refinement isn’t healthy either; you are overinvesting in something potentially wasteful.)

Learn more: 28 Product Backlog and Refinement Anti-Patterns.

Q 47: Sprint Planning-Related Scrum Anti-Patterns

As a Scrum Master, what are some of the Sprint Planning-related anti-patterns that you need to avoid as a team?

According to the Scrum Guide, the Sprint Planning plays a vital role in the value creation process of the Scrum team:

- *Page 8: Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint.*

- *Page 8: This resulting plan is created by the collaborative work of the entire Scrum Team.*

- *Page 8: The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal.*

- *Page 8: [Sprint Planning: Why is this Sprint valuable?] The Product Owner proposes how the product could increase its value and utility in the current Sprint.*

- *Page 8: [Sprint Planning: Why is this Sprint valuable?] The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders.*

- *Page 8: [Sprint Planning: Why is this Sprint valuable?] The Sprint Goal must be finalized prior to the end of Sprint Planning.*

Source: Scrum Guide 2020. (The aggregation is taken from the Scrum Guide 2020 Reordered.)

Therefore, it should be of the highest priority of any Scrum team to perform the best possible Sprint Planning. It is the last moment where the Scrum team can change direction; once the Sprint Goal is defined, the investment decision is made. In my experience, the following six Sprint Planning anti-patterns have the most negative impact on a Scrum team's value creation:

- **What are we fighting for?** The Product Owner cannot align the business objective of the upcoming Sprint with the overall product vision. (A serious business objective answers the "What are we fighting for?" question. A good goal derived from this alignment is focused and measurable, as the goal of the upcoming Sprint — based on the business objective — and Developers' forecast goes hand in hand.)

- **No business objective, no Sprint Goal**: The Product Owner proposes Product Backlog items that resemble a random assortment of tasks, providing no cohesion. Consequently, the Scrum Team does not create a Sprint Goal. (If this is the natural way of finishing your Sprint Planning, you probably have outlived the usefulness of Scrum as a product development framework. Depending on the maturity of your product, Kanban may prove to be a better solution. Otherwise, the randomness may signal a weak Product Owner who listens too much to stakeholders instead of ordering the Product Backlog appropriately.)

- **Unfinished business**: Unfinished work items from the last Sprint spill over into the new Sprint without any discussion. (There might be good reasons for that, for example, a task's value has not changed. It should not be an automatism, though, remember the sunk cost fallacy.)

- **Last-minute changes**: The Product Owner tries to squeeze in some last-minute Product Backlog items that are not ready yet. (Principally, it is the prerogative of the Product Owner to make such kind of changes to ensure that the Developers are working only on the most valuable tasks at any given time. However, if the Scrum Team is otherwise practicing Product Backlog refinement sessions regularly, these occurrences should be a rare exception. If those happen frequently, it indicates that the Product Owner needs help with ordering the Product Backlog and team communication. Or the Product Owner needs support to say ‘no’ more often to stakeholders.)

- **Output focus**: The Product Owner pushes the Developers to take on more tasks than it could realistically handle. Probably, the Product Owner is referring to former team metrics such as velocity to support their desire. (This is also a road to becoming a feature factory and deserves attention from the team’s Scrum Master. It is violating the Developers’ prerogative to pick Product Backlog item for the Sprint Backlog as well as Scrum Values.)

- **No preparation**: The Product Owner does not prepare the Product Backlog to provide useful Product Backlog items for selection by the Development Team. (Product Backlog needs to represent the best possible use of the Developers’ work from a customer value perspective at any given moment. In other words, your Scrum Team’s Product Backlog has to be actionable 24/7. By my standards, that means that you need to be capable of running a meaningful Sprint Planning instantly. Preparing a few basic Product Backlog items an hour before the beginning of the Sprint Planning is not enough.)

Q 48: Sprint Review-Related Scrum Anti-Patterns

As a Scrum Master, what are some of the Sprint Review-related anti-patterns that you need to avoid as a team?

According to the Scrum Guide, the Sprint Review is an essential moment of collaboration with internal and external stakeholders to reassure that the Scrum team is still on the right track:

- *Page 9: The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations.*

- *Page 9: The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.*

- *Page 9: During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next.*

- *Page 9: The Product Backlog may also be adjusted to meet new opportunities.*

- *Page 9: The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.*

Source: Scrum Guide 2020. (The aggregation is taken from the Scrum Guide 2020 Reordered.)

Some of the most damaging Sprint-Review anti-patterns in my experience are as follows:

- **Following a plan**: The Scrum Team does not use the Sprint Review to discuss the current state of the product or project with the stakeholders. (Again, creating transparency and receiving feedback is the purpose of the exercise. A we-know-what-to-build attitude is bordering on hubris. **Read More**: Sprint Review, a Feedback Gathering Event: 17 Questions and 8 Techniques.)

- **Death by PowerPoint**: Participants are bored to death by PowerPoint. (The foundation of a successful Sprint Review is "show, don't tell," or even better: let the stakeholders drive the discovery.)

Death by Slide Deck (#3)
Bla, bla
bla, bla
bla, bla... and: bla, bla
PO
Hands-on Agile Webinar #9
Sprint Review Anti-Patterns
November 6th, 2018, at 6:15 pm CEST
(12:15 pm EDT or 9:15 am PDT)

- **Sprint accounting**: Every task accomplished is demoed, and stakeholders do not take it enthusiastically. (My suggestion: Tell a compelling story at the beginning of the review to engage the stakeholders. Leave out those user stories that are probably not relevant to the story. Do not bore stakeholders by including everything that was accomplished. We are not accountants; the output is less relevant by comparison to the outcome from a customer or value creation perspective.)

- **Cheating**: The Development Team shows items that are not "done." (There is a good reason to show unfinished work on some occasions. Partially finished work, however, violates the concept of "Done," one of Scrum's first principles.)

- **Scrum à la Stage-Gate®**: The Sprint Review is a kind of Stage-Gate® approval process where stakeholders sign off features. (This Sprint Review anti-pattern is typical for organizations that use an "agile"-waterfall hybrid. However, it is the prerogative of the Scrum team to decide what to ship when.)

- **No stakeholders**: Stakeholders do not attend the Sprint Review. (There are several reasons why stakeholders do not participate in the Sprint Review: they do not see any value in the event, or it is conflicting with another important meeting. They do not understand the importance of the Sprint Review event. No sponsor is participating in the Sprint Review, for example, from the C-level. To my experience, you need to "sell" the event within the organization, at least in the beginning of using Scrum.)

- **No customers**: External stakeholders—also known as customers—do not attend the Sprint Review. (Break out of your organization's echo chamber, and invite some paying users to your Sprint Review. Everyone will be grateful for that.)

- **Side gigs**: The Development Team was working on issues outside the Sprint Goal, and the Product Owner learns about those for the first time during the Sprint Review. (For the sake of transparency, openness, and respect: There is no room for side gigs when using Scrum.)

Learn more: 15 Sprint Review Anti-Patterns Holding Back Scrum Teams.

Q 49: Sprint Retrospective Anti-Patterns

What anti-patterns do you know of that can happen during a Sprint Retrospective?

Typical Scrum Sprint Retrospective anti-patterns are:

- **Waste of time**: The team does not collectively value the Sprint Retrospective. If some team members consider the Sprint Retrospective to be of little or no value, it is most often the Sprint Retrospective itself that sucks. Is it the same procedure every time, ritualized and boring? Have a meta-Sprint Retrospective on the Sprint Retrospective itself. Change the venue. Have a beer- or wine-driven Sprint Retrospective. There are so many things a Scrum Master can do to make Sprint Retrospectives interesting and valuable again, reducing the absence rate. Furthermore, it is good to remember that (in my experience) introverts like to take part in Sprint Retrospectives also.
- **Prisoners**: Some team members only participate because they are forced to team up. Don't pressure anyone to take part in a Sprint Retrospective. Instead, make it worth their time. The drive to continuously improve as a team needs to be fueled by intrinsic motivation, neither by fear nor by order. Tip: Retromat's "Why are you here?" exercise is a good opener for a Sprint Retrospective from time to time.

- **Groundhog day**: The Sprint Retrospective never changes in composition, venue, or length. In this case, the is that the team will revisit the same issues over and over again – it's like groundhog day without the happy ending.
- **Let's have it next Sprint**: The team postpones the Sprint Retrospective into the next Sprint. Beyond the "inspect & adapt" task, the Sprint Retrospective serves as a moment of closure, helping reset everybody's mind so that the team can focus on the new Sprint goal. That is the reason why we have the Sprint Retrospective before the planning of the follow-up Sprint. Postponing it into the next Sprint may also interrupt the flow of the team, and delay tackling possible improvements by up to a Sprint. This is why it is important to have the Sprint Retrospective before the planning of the follow-up Sprint.
- **#NoDocumentation**: No one is taking minutes for later use. A Sprint Retrospective is a substantial investment for many reasons and should be taken seriously. Taking notes and photos supports the process.
- **No psychological safety**: The Sprint Retrospective is an endless cycle of blame and finger pointing. The team wins together, the team loses together. Unfortunately, the blame game documents both the failure of the Scrum Master as the facilitator of the Sprint Retrospective as well as the team's lack of maturity and communication skills.
- **Bullying**: One or two team members are dominating the Sprint Retrospective. This communication behavior is often a sign of either a weak or uninterested Scrum Master. The Sprint Retrospective needs to be a safe place where everyone–introverts included–can address issues and provide their feedback free from team members who are dominating the conversation, bullying or intimidating other teammates. The failure to provide a safe place will result in participants dropping out of the Sprint Retrospective and render the results obsolete. It is the main responsibility of the Scrum Master to ensure that everyone can be heard and has an opportunity to voice their thoughts. According to Google, equally distributed speaking time fosters and signifies a high-performing team. **Read More**: "What Google Learned From Its Quest to Build the Perfect Team".
- **Stakeholder alert**: Stakeholders participate in the Sprint Retrospective. There are plenty of Scrum events that address the communication needs of stakeholders: the Sprint Review, the Product Backlog refinement, the Daily Scrum events – not to mention opportunities of having a conversation at water coolers, over coffee, or during lunchtime. If that spectrum of possibilities is still not sufficient, feel free to have additional meetings. However, the Sprint Retrospective is off-limits to stakeholders.
- **Passivity**: The team members are present but are not participating. There are plenty of reasons for such a behavior: they regard the Sprint Retrospective a waste of time, it is an unsafe place, or the participants are bored to death by

its predictiveness. The team members may also fear negative repercussions should they be absent, or maybe an homogenous group of introverts were unwittingly hired. Whatever the reason, there is likely no quick fix. The Scrum Master needs to determine what style of Sprint Retrospective will work best in their organization's context.

Read more: 21 Sprint Retrospective Anti-Patterns Impeding Scrum Teams.

Q 50: Improving as a Scrum Master

How can you (as a Scrum Master) identify where you need to improve?

This is a simple question: Regularly ask your team and stakeholders how you can improve as a Scrum Master.

Why not run a Sprint Retrospective on yourself? A dedicated Sprint Retrospective is much more effective than spending five minutes, asking for hints at how you might improve, at the end of each regular team Sprint Retrospective.

Good candidates also note that they proactively provide user manuals on how to work with themselves to other team members and the organization.

Set 9: Scrum Success Principles & Indicators

Background

- We do not get paid to practice Scrum according to the letters of the Scrum Guide. We are getting paid to solve customer problems, thus generating a sustainable business on the organization's side.
- Scrum is not the one-size-fits-all approach to creating valuable products and services. There are many alternative practices organizations/product teams can choose instead of Scrum.
- Scrum is not well-suited for either the simple or chaotic domain. Moreover, consider wisely before choosing Scrum for an endeavor located in the complicated domain.
- Scrum thrives when applied to solve complex, adaptive problems.
- No matter how diverse the stakeholders of your Scrum Team are, they are all united in one expectation: Your Scrum team delivers—with the precision of a Swiss clockwork—a Done, potentially releasable, valuable Increment every single Sprint.
- There indicators for all three Scrum accountabilities that the respective individuals are successful in practicing their Scrum role.

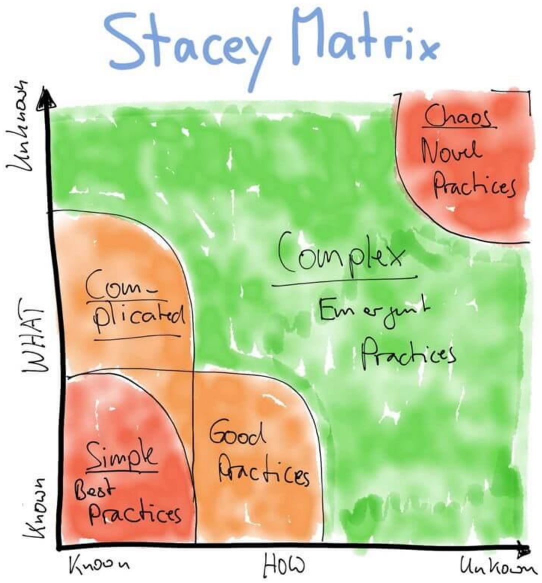
Q 51: When to Use Scrum and what to Look for

Can we use Scrum to solve any problem, task, or challenge? Or do you think that Kanban or even Waterfall may be better solutions in some cases? Moreover, if we choose Scrum, what principles and success indicators shall we observe?

Scrum is not the Swiss Army knife for any problem a product team may be facing. Throwing Scrum at all problems indiscriminately will likely be an ineffective strategy. However, when Scrum is chosen for the proper purpose, four first principles support Scrum Masters to help their teams deliver:

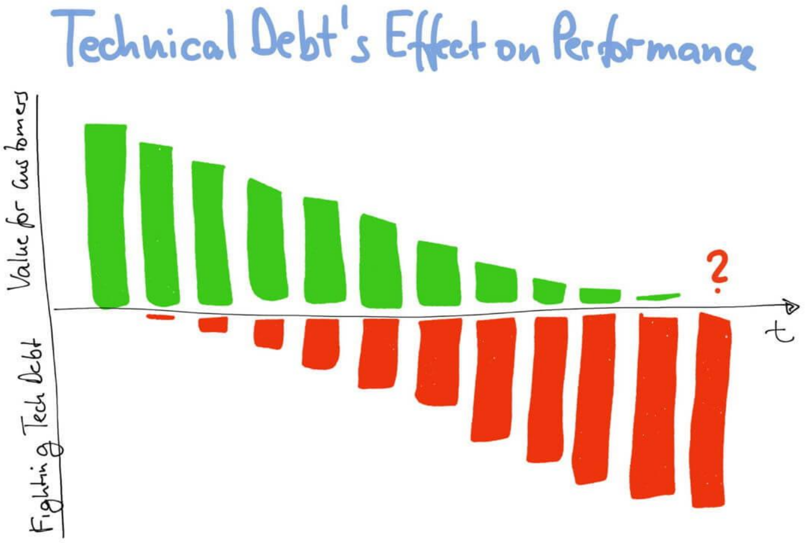
Choose Scrum for the Right Purpose:

Choosing the appropriate application area for Scrum is essential. Referring to the Stacey Matrix, applying Scrum to the areas "Chaos" and "Simple" is a waste. Scrum is best used in the "Complex" area. Here, empirical process control thrives, applying transparency, inspection, and adaptation to iteratively, incrementally developing valuable product Increments, thus mitigating risk.



Strive for High Product Quality:

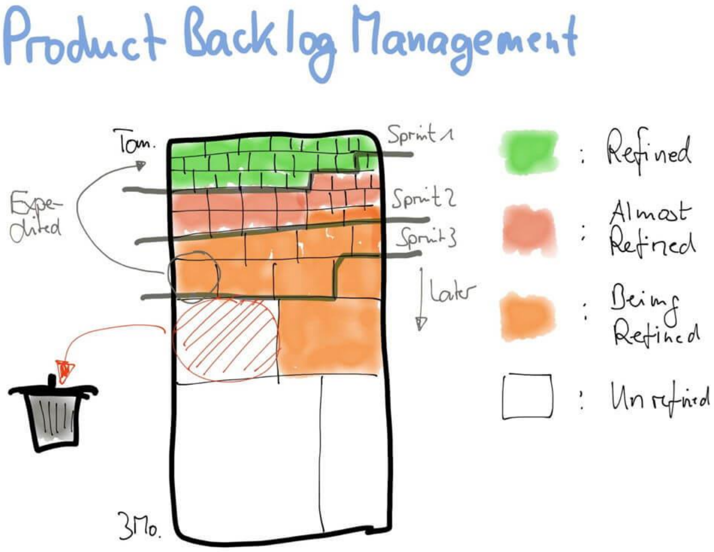
From day one, keep technical debt small and work continuously on high product quality, reflected in the Scrum Team's Definition of Done. Achieving business agility requires dedication to product quality and excellence at the technical level. (Learn more: Technical Debt & Scrum: Who Is Responsible?



Create and Maintain an Actionable Product Backlog:

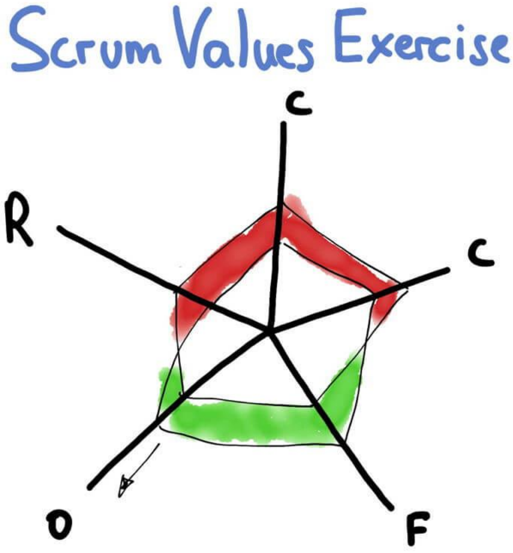
Garbage in, garbage out: No matter how your Scrum Team is everything else, a sub-standard Product Backlog will diminish all other team achievements. Hence, it would be best to support the Product Owner and the Developers to maintain a permanently "actionable" Product Backlog. By "actionable," I am referring to a refinement level of the

Product Backlog that would allow a Scrum Team to run a meaningful Sprint Planning at a moment’s notice. (Learn more: 28 Product Backlog and Refinement Anti-Patterns.)



Embrace Self-Management and Take It to the Scrum Team:

Restrain from solving problems that your teammates can solve themselves. I know it feels good to be helpful; however, it is not your job as a Scrum Master to become the team’s helping hand in all matters. Instead, make self-management our number one priority and ensure that everyone lives Scrum Values. Be a servant-leader at heart and, therefore, a good role model for the Scrum team.



Note: All sketches are taken from a previous Professional Scrum Master class; check out my upcoming training classes here.

Q 52: Scrum Master Success

How could you measure the success of a Scrum Master?

There are several indicators of the success of a Scrum Master, for example:

- The Scrum team regularly meets Sprint Goals and delivers valuable, done Increments.
- They have an excellent understanding of the Scrum framework and the challenges it poses to individuals and organizations.
- They are prepared to step into the background when the Scrum team is successful.
- The successful Scrum Master strives to become redundant concerning the daily operations of the Scrum Team. (A successful Scrum Master can take a holiday at any time, just saying.)
- They spend more and more time on working with the organization while the Scrum team is self-managing.
- The Scrum team has high morale; rarely, a team member leaves the team, but others want to join it.
- The whole Scrum team is dedicated to continuously improve their skills and capabilities, branching out into adjacent areas of the organization in the process.

Q 53: Scrum Product Owner Success

How could you measure the success of a Product Owner?

There are several indicators of the success of a Product Owner, for example:

- The Scrum team regularly meets Sprint Goals and delivers valuable, done Increments, see above.
- The successful Product Owner aligns stakeholders and team member regarding product vision and Product Goal.
- They are obsessed with creating value for the customers while creating a sustainable business for the organization.
- Successful Product Owners are data-informed, not data-driven. Progress is made through applying empiricism.
- They include the Developers in the product discovery process early.
- Product Owners expect to be challenged by the Developers during Product Backlog refinement regarding their choices.
- They are transparent and outstanding communicators.

Q 54: Scrum Developer Success

How could you measure the success of the Scrum Developers?

There are several indicators of the success of Developers on a Scrum Team, for example:

- The Scrum team regularly meets Sprint Goals and delivers valuable, done Increments; again, see above.
- Developers actively engage in self-managing during the Sprint to meet the Sprint Goal.
- They take control over the Sprint Backlog.
- Developers accept the collective responsibility for quality.
- They uphold product quality by regularly inspecting the Definition of Done in collaboration with the Product Owner.
- Developers keep technical debt at bay by allocating sufficient time to refactoring and bug-fixing every Sprint.
- They take their commitment to continuous improvement as a team seriously by acting on actions items from Retrospectives.
- Successful Developers have a collaborative mindset; for example, they share knowledge, pair program, or swarm to support other Developers accomplishing their tasks.
- They identify gaps in their knowledge or experience and reach out to others to help fill them.

Set 10: How to Make Your Scrum Master Fail

Background

- Empathy is an essential trait of a successful servant leader.
- Being able to walk in the shoes of a stakeholder as a Scrum Master significantly supports any change, communication, and collaboration effort.
- Not everyone in the organization will be thrilled to become "agile."
- Many people, particularly at the middle management level, have a vested interest in maintaining the status quo.
- Becoming "agile," for example, by introducing Scrum, may devalue their accumulated career equity.
- Expect resistance from these stakeholders when embarking on the journey of making Scrum work in your organization.
- Resistance manifests itself in many different shapes and forms, from direct, outspoken opposition to backroom dealings and covert actions.
- The question at the beginning of a transformation to Scrum is not if resistance to the effort will happen, but only when.

Q 55: How To Mess with the Scrum Framework in General

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You're not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you mess with the Scrum framework itself?

The first category of how to best sabotage a Scrum Master is generally about disqualifying Scrum itself as a helpful framework or introducing changes to conflict with the first principles of Scrum. Effective examples are:

- Place the blame on Scrum whenever you can, even if it is technically unrelated.
- If Scrum uncovers an obstacle in the organization, blame that on Scrum.
- Find examples of where Scrum failed in other companies to spread around.

- Talk disrespectfully in the coffee breaks with developers and the other middle managers about Scrum and the role of the Scrum Master.
- Challenge anything the Scrum Masters try to say or do.
- Ignore the Scrum Master's offer to learn about Scrum.
- Create an ego-centric incentive system.
- Install multiple Product Owners in a Scrum Team.
- Place a proxy Product Owner in the Scrum Team and overrule all decisions.

Q 56: Employing Unsuitable Metrics & Reporting

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You're not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you use excessive reporting requirements or unsuited metrics to pursue your agenda?

The next bucket of useful sabotage practices are metrics, OKRs, KPIs — you name it. Just turn your Scrum Master into a glorified data-entry clerk with a challenging reporting burden. Successful approaches are:

- Ask the Scrum Master to prove their value with metrics.
- Create performance KPIs for each team member.
- Ask the Scrum Master to collect all working hours of the team members.
- Ask for individual performance metrics for every Sprint.
- Tie team member performance reviews with their average story points per Sprint using the Bell curve.
- Calculate a Scrum team budget and insist that the utilization rate of team members needs to be higher.
- Demand estimates and treat them as commitments.

Q 57: Messing with Scrum Team Building & Line Management

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You're not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you interfere with the team building by utilizing your line management prerogatives?

If a challenging reporting burden does not help, see Q 56, sabotage your Scrum Master by actively undermining their activities to turn a group of people into a cross-functional Scrum Team. Examples to consider are:

- Only add members to the Scrum team that don't live any Scrum values.
- Recommend the most bull-headed senior developer as the engineering team lead.
- Constantly switch Developers from one project to another, claiming emergencies that require swift action.
- Have Scrum team members regularly work on multiple different Scrum teams.
- Add in new people to the Scrum team without prior consultation.
- Alternatively, slow down the hiring or replacement processes.
- Promote a member within the Scrum team to act as a proxy manager.

Q 58: Interfere with Work Organization

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You're not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you impede the Scrum team's self-management and work organization?

If you are already messing with the Scrum Team team-building process, why not place a few obstacles into the Scrum Team's way of working? Sabotage your Scrum Master by creating unachievable objectives while meddling with the very foundation of Scrum:

- Define unachievable objectives for the Scrum Team.
- Overload the Scrum team with requests, then complain to others that you do not get results on time.
- Hand over only fixed price, time, and scope projects to Scrum Team.
- Change requirements during the Sprint.
- Insist on hard deadlines.
- Ask the Scrum Master to provide a product roadmap with deadlines.
- Make the Scrum Master responsible for accomplishing deadlines.
- Outsource part of the product roadmap creation to an off-site team in a completely different timezone.
- Request work that would switch focus away from the Sprint Goal directly to Developers.
- Assign tasks directly to Scrum team members.
- Don't allow Scrum Team members to speak to the customer; act as the single point of contact.
- Create unnecessary organizational bottlenecks outside of Scrum, for example, approvals gates, etc.
- Only provide inadequate equipment and tools to the Scrum team.

Q 59: Manipulating the Flow of Information

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You're not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you manipulate the flow of information to the Scrum team?

Does your Scrum Master have an insatiable appetite for data, information, and knowledge? Well, keep them out of the loop then. What could be an easier way of sabotaging Scrum:

- Claim that everyone already knows what to do. There is hence a need for alignment or a Scrum Master.
- Restrain from sharing essential or valuable information with the Scrum Team.
- Encourage silo thinking by promoting a strict “need to know” basis for sharing information and knowledge.

Q 60: Use other Meetings to Interfere with Scrum Events

You are a middle manager in the IT organization, and you believe this Scrum thingy is a fad and will go away—with a little help from your side.

Come up with ideas on how to best sabotage the new Scrum Master of the first Scrum Team in your organization.

You’re not allowed to use any form of illegal activity. So, outsourcing the task to a bunch of outlaws is out of the question. Instead, you are only allowed to use practices that are culturally acceptable within your organization.

How can you use other events and meetings to make planning and communication harder for the Scrum team?

Finally, ensure that everyone on the Scrum Team understands that your events are more important than theirs:

- As a manager, claim that there are too many Scrum events that require too much time. Instead, suggest skipping some of them.
- Require to be present at every Scrum event.
- Exclude Scrum Master from important meetings outside the Scrum Team’s events.
- Constantly pull Scrum team members into long unnecessary meetings during their Scrum team events.
- Be very understanding of the needs of the Scrum Team; for example, that stakeholders shall participate in the Sprint Review. However, never join any Scrum event yourself.

Conclusions

During the interview, move as fast as possible from the theoretical to the practical. Be careful not to waste too much time discussing the advantages of agile frameworks or other (likely) opinionated topics. Two or three questions from each of the sets in this handbook will provide more than enough ground for an engaging sixty-minute conversation.

Scrum has always been a hands-on business, so your candidate will need to have a passion for getting their hands dirty if they're going to be successful. Although the rules are basic, building an effective team from a group of individuals with different backgrounds, levels of engagement, and personal agendas is a complex task – as is often the case when people and communication are involved.

The larger an organization and more levels of management, the more likely there will be resistance or possibly even failure when applying agile. In these circumstances it would be wise to choose the pragmatic veteran, who has experienced failure at other organizations (and may carry the scars to prove it), over a junior Scrum Master.

Regarding certifications of candidates, I recommend looking out for those with PSM I, PSM II, and particularly PSM III certificates from Scrum.org.

Suppose you want to suggest preparation for Scrum Master candidates in advance. In that case, I suggest pointing them to the Open Assessments of Scrum.org, a set of free online self-assessment tests for Scrum Masters, Product Owners, and Development Team members.

About the Authors

Stefan Wolpers

Author

Stefan is a Professional Scrum Trainer with Scrum.org, an Agile Coach, and Scrum Master.

PST | Professional Scrum Trainer Scrum.org

He is specializing in coaching agile practices for change, for example, agile software development with Scrum, LeSS, Kanban, and Lean Startup, as well as product management.

He also serves as one of the XSCALE Alliance stewards and coaches organizations in business agility. Additionally, he is a licensed facilitator of the Agile Fluency™ Team Diagnostic.

He has served in senior leadership positions several times throughout his career. His agile coaching expertise focuses on scaling product delivery organizations of fast-growing, venture-capital funded startups, and transitioning existing product teams in established enterprise organizations.

Stefan is also curating the popular ‘Food for Agile Thought’ newsletter for the global Agile community with 34,000-plus subscribers. He blogs about his experiences on Age-of-Product.com and hosts the most significant global Slack community of agile practitioners with 11,000-plus members.

His ebooks on agile topics have been downloaded more than 75,000 times. Lastly, Stefan is the organizer of the Agile Camp Berlin, a Barcamp for 200-plus agile practitioners, and the Hands-on Agile Meetup community with 4,000-plus members globally.

Read more about Stefan at Scrum.org, and connect with him via LinkedIn, or Twitter, or privately via email.

Andreea Tomoiaga

Contributor

Andreea is a product manager who is passionate about designing and optimizing B2B integrations. She is keenly interested in strategy and prior-itization techniques, and uses agile principles to build and grow motivated teams that deliver products successfully — whether working locally or around the globe.

Throughout her career Andreea has worked in many different roles — from developer through ar-chi-tect, project manager through product man-ager — successfully delivering projects of different sizes by combining a variety of mana-gerial meth-ods. She sees the combination of methods in the right balance and at appropriate moments as an important key to product success.

Andreea is a Certified ScrumMaster[32] and PRINCE2[33] Practitioner. She likes to write about various product and project management topics on her personal website.

Read more about Andreea at Scrum Alliance, and connect with her via LinkedIn.

[32] Certified ScrumMaster ® is a registered trademark of Scrum Alliance, Inc.

[33] PRINCE2 ® is a registered trademark of AXELOS Limited.

Legal

60 Scrum Master Interview Questions to Avoid Hiring Agile Imposters

By Stefan Wolpers. Published and distributed by Berlin Product People GmbH. First published on February 15, 2017.

Edited by Candace Chapman and Jourdan Ritchey

Copyright © 2017-2022 by Berlin Product People GmbH

The advice and information included in this publication was carefully researched and written and believed true and accurate as of the date first published. However, neither the author nor the editor nor the publisher can accept legal responsibility for the accuracy thereof, or for any errors or omissions that may have been made. The publisher makes no warranty, express or implied, with respect to the material herein.

Berlin Product People GmbH

Borsigstrasse 8
10115 Berlin
Germany

Registered at: AG Charlottenburg (HRB 160341 B)
Managing Director: Stefan Wolpers
VAT identification number: DE227895748

Website: https://berlin-product-people.com

Email: info@berlin-product-people.com