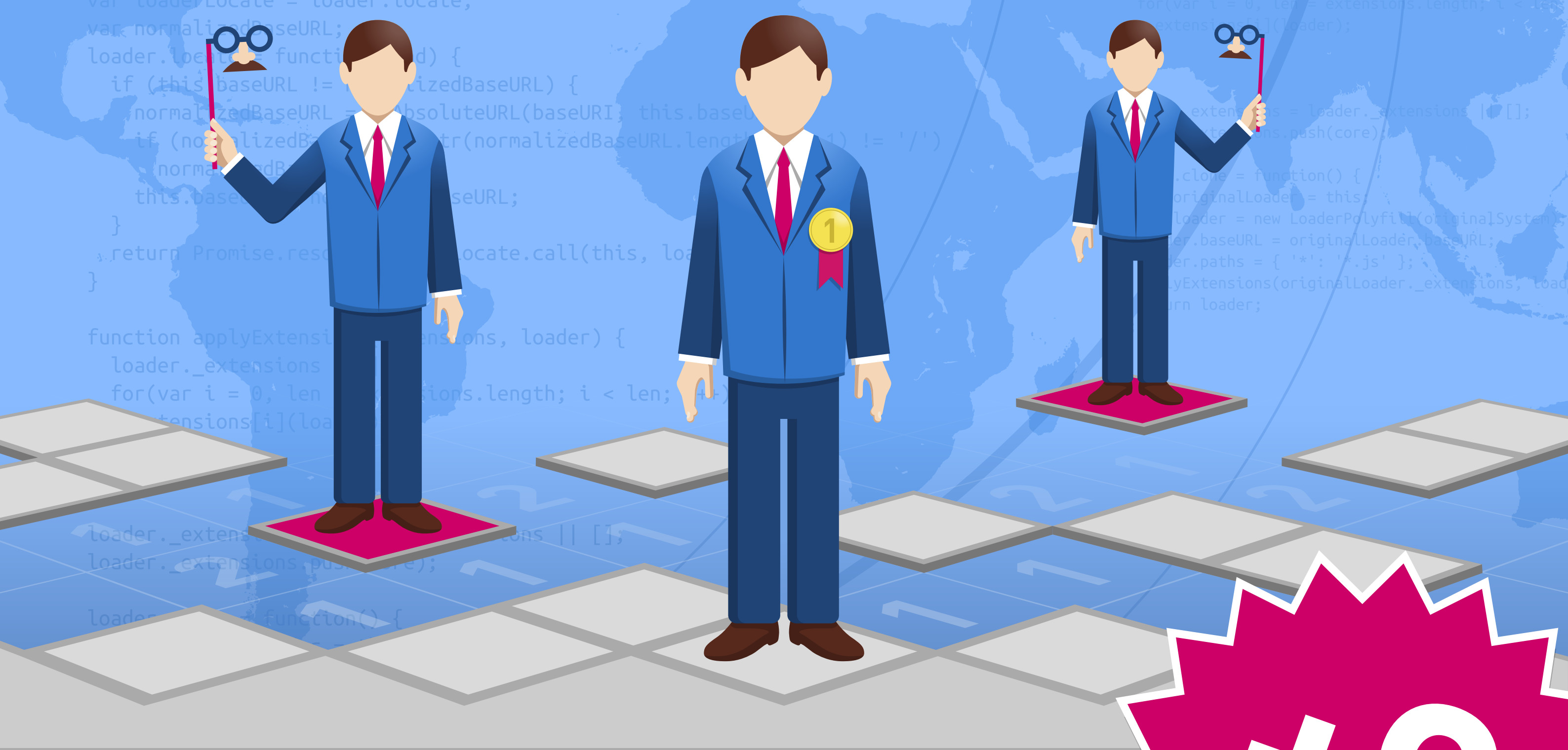
FOURTH EDITION

38 SCRUM MASTER INTERVIEW QUESTIONS

TO AVOID HIRING AGILE IMPOSTERS

Stefan Wolpers
with **Andreea Tomoiaga**



+9 new questions!

AGE of PRODUCT™
INVENT FOR YOUR CUSTOMERS

Stefan Wolpers
with **Andreea Tomoiaga**

38 SCRUM MASTER INTERVIEW QUESTIONS

TO AVOID HIRING AGILE IMPOSTERS

FOURTH EDITION

+9
new questions!

This book has been formatted to be viewed electronically.

Contents

Introduction

38 questions to identify the best candidate for your organization

Plus 9 new questions in this fourth edition

Maybe 'agile management' in general is a fad as opposed to a trend. Whatever the case, we can say for sure that scrum[1] is now 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're looking to hire a scrum master for your organization, you'll find the following 38 (plus nine) interview questions useful in identifying the right candidate.

---

1. This text assumes a familiarity with what scrum is. If you're unfamiliar with agile software development frameworks, and scrum in particular, read the Wikipedia entry here.

Being cognisant 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 to any organization, multiple choice questions are insufficient when you need to discern a candidate's agile mindset.

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

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 enough 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, agile imposters.

Why these questions

These questions are derived from Stefan Wolpers' twelve years of practical experience with kanban[2], scrum, XP[3], and several product discovery 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.

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. **38 Scrum Master Interview Questions to Avoid Hiring Agile Imposters**, now in its fourth edition, is the outcome of that.

A message for Age of Product subscribers

If you'd like to be notified when the fifth 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

2. Unlike scrum, kanban is not a framework but a methodology, and much less structured than scrum. Often used together with scrum, kanban introduces the concept of a 'kanban board' to provide a system for introducing change through incremental improvement.
3. XP (Extreme Programming) is a lightweight software development methodology.

subscribing to the Age of Product newsletter, or if you'd simply like to learn more, visit us at www.age-of-product.com.

1

The role of a scrum master

Background

- Scrum is not a methodology, but a framework. There are no rules that apply to each and every scenario — just best practices that have worked before in other organizations.
- The best practices of other organizations cannot simply be copied to your own. Every best practice requires a particular context to work.
- A scrum team [redacted] is an agile team. As somebody hiring for an agile team, you

4. Typically a software development team, a scrum team comprises a scrum master, product owner, and less than 10 members who are cross-functional and can do the work necessary to create a product increment.

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 leadership and coaching. It is not a management role.
- A scrum master should recognize that different stages of a scrum team's development require different approaches: some, teaching; some, coaching; and some, mentoring.
- A scrum master would do well to 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 and self-managing.
- 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 (e.g. commitments vs. velocity) does not help.

- Scrum doesn't elaborate on the process that enables a product owner to add valuable, usable, and feasible user stories to the product backlog. Product discovery using the Design Thinking, Lean Startup, or Lean UX methodologies may help, but in any case a good scrum master will want the scrum team to be a part of this process (whether participating in user interviews or running experiments).

- 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, conversely, are a good way to stay in close contact with stakeholders, and to present the value delivered by the team during each previous sprint.

---

5. KPI (Key Performance Indicators) are metrics used to evaluate an organization's success at reaching targets.
6. A product backlog is a list of items to work on, such as bugs, technical work, and knowledge acquisition.
7. A sprint is the basic time-measured (and time-restricted) unit of development in scrum. A sprint is always particular to a scrum team.
8. Sprint reviews are the meetings that follow sprints. During a sprint review, work planned for the sprint that was not completed during the sprint is discussed, and work that was completed is demonstrated for the project's stakeholders.

Question 01

> The Agile Manifesto infers people over processes. Isn't a scrum master — whose role is meant to enforce the process — therefore a contradiction?

A scrum master does not wield any real authority. 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 manage — 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 facilitator 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-management. 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 an agile team."

Question 02

> What indicators might there be that demonstrate agile practices are working for your organization, and which of these would demonstrate your efforts at agile 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. A growing team velocity[9] is usually not considered to be a meaningful indicator (see Question 40 for a discussion of team velocity).

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

- Improved team happiness is exhibited by reduced churn and an increase in the number of referrals from members.
- Increased competitiveness in the battle for talent can be demonstrated by an increase in the number of experienced people willing to join the organization.
- Products delivered to customers are resulting in higher retention rates, better

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

conversion rates, increased lifetime value, and similar improvements to the business.

- Increased software quality can be demonstrated by measurably less technical debt, fewer bugs, and less time spent on maintenance.
- Production time, from validated idea to shipped product, has been reduced.
- The cycle time for hypothesis validation has been reduced.
- There has been a reduced allocation of resources to low value products.
- There is greater respect among stakeholders for the IT team.
- Stakeholders are increasingly participating in agile meetings, especially during the sprint demo.

Question 03

> Should a scrum master remove impediments on behalf of the scrum team?

A scrum master should not be concerned with "removing impediments on behalf

10. 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.
11. Cycle time is the number of days between starting and ending an experiment suitable to validate or falsify the motivating hypothesis.
12. Part of the sprint review, the sprint demo is the event during which the scrum team presents work completed during the sprint to its stakeholders.

of the scrum team", 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 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.

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

13. JIRA® is a proprietary issues tracking and project management software system published by Atlassian Pty Ltd.

Question 04

> 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 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, for clarifying goals, and for ensuring that the entire product delivery team[14] understands the product vision.

A scrum master, in return, supports the product owner in building a high-value product backlog, and to this end must facilitate effective collaboration between the product owner and the scrum team.

14. A product delivery team comprises everyone involved in delivering a product to market, including the scrum teams involved.

Question 05

> 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:

01 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.

02 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.

Involving a scrum team's engineers 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 goals defined for each sprint or product release.

Question 06

> The role of 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. Essentially, the team either wins together, or loses together.

Question 07

> How can you ensure that a scrum team has access to a project's stakeholders?

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

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.

Question 08

> 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:

- Most importantly, a scrum master should live and breathe the principles of 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, in presentations or otherwise, 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 prediction 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 meetings, 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.

  Read more: App Prototyping with Absolute Beginners — Agile Experiments.

Question 09

> 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. At the beginning of a transition any organization shows inertia to change, so to overcome this resistance executives and stakeholders need to know how agile 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 C-level management. 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 methodologies if organized as a scrum team.

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

Question 10

> You've already provided your project'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 agile 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 agile failure patterns at Age of Product.

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.

Backlog refinement and estimation

Background

- Estimation and backlog refinement 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 team to do so.

- A cross-functional and 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) and deliverables from the UX or UI department.

- There are two essential ingredients for good scrum team performance:

01 **Writing the user story as a team.** When something 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 corresponding and 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 defining the 'what'), and a shared sense of ownership among team members.

02 **Sharing a definition of ready.** In order to ensure a flow of well-drafted

---

16. An application programming interface (API) endpoint is a URL, and the commands that may be issued through it, for use within software to instruct other software.
17. UX, or user experience design, is a design practice that focuses on optimizating products for user satisfaction by designing with consideration for all perceivable aspects. UX encompasses all human-software interfaces, including visual.
18. UI, or user interface design, is the predominantly visual presentation and interactivity of a software product, and the design practice concerned with this.
19. An agile software development tool, a user story is a description of the desired functionality for a requirement.

user stories for the development process, the scrum team and the product owner need to agree on a 'definition of ready' (see Question 15) for these stories. This definition is an agreement about what needs to be provided for a user story to be considered ready for estimation. If even one of the defined requirements is not met, a user story isn't ready for estimation. A user story without a previous estimation is an unknown entity and, therefore, not ready to be made part of a sprint backlog[redacted] because a scrum team can't commit to an unknown entity in a sprint. Consequently, the scrum team must learn to say "No".

- A well-groomed product backlog probably has user stories detailed for about two or three sprints, and probably less than half of these stories conform to the scrum team's definition of ready. There may also be additional user stories that no one except the product owner is working on at the moment.

20. The sprint backlog is the prioritized list of tasks to be completed during a sprint.

Question 11

> 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 never turn requirements documents[21] received from stakeholders into tickets, and a scrum master should never accept such a procedure. It's nothing more than a waterfall process[22] dressed-up in a pseudo-agile methodology.

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.

21. Requirements documents might include, for example, software requirements specifications (SRS).
22. A waterfall process is a sequential design process that generally adheres to the waterfall model traditionally used in software development.

Question 12

> 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?

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.

Question 13

> Who should be writing user stories?

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

Question 14

> 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.

Question 15

> What should a definition of ready consist of?

A 'definition of ready' is an agreement between the scrum team and the product owner about what must be included in a user story (before the story can be considered ready for estimation). It defines what a good user story looks like.

The discussion at Question 14 includes an outline of what a good user story should include. Another approach is to use a framework for user stories — such as the INVEST mnemonic by Bill Wake:

- **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.

Question 16

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

Estimating user stories in man-hours is never 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.

Story points[23] are much better suited to estimating than man-hours in all situations, but especially in tricky situations, because they accurately reflect both the

23. Story points are units of measure expressing estimates of the overall effort required to fully implement a product backlog item.

complexity of the task and the effort required to complete it. Using man-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.

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' (e.g. #noestimates) concept.

Question 17

> The product owner for 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 tickets in various stages. What are your thoughts on this? Can a scrum team work on 200 tickets?

Any product backlog larger than the scope of two or three sprints is not manageable. Misusing a 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 backlog avoids misallocating resources; a larger backlog is an indication of waste.

Your candidate should make it clear that they would support a product owner in managing with the size of the product backlog, and with processing input from stakeholders.

3

Sprint planning

Background

- It used to be that a product owner would explain high value user stories in a product backlog to the scrum team during sprint planning. The team would then turn these into more detailed user stories, and estimate the subsequent stories. There is now, however, a consensus among agile practitioners that working on these high-level user stories in separate backlog refinement and estimation meetings — before sprint planning — actually improves the quality of the stories and thus the outcome of the team's work.

- Sprint planning can create a sense of ownership among a scrum team's members by enabling them to make a valid commitment to the items in the sprint backlog. But this only happens if a team's uncertainty about the quality of the user stories they're receiving is eliminated. To be certain that their team can be certain, a scrum master should run weekly product backlog refinement and estimation sessions, only allowing into sprint planning those user stories that meet the team's definition of ready standard.

- Sprint planning should normally be divided into two parts:

  Sprint planning I: During the first part of sprint planning, a product owner presents to the scrum team the product owner's choice of the most valuable user stories from the product backlog as a ranked list. The team then selects from the top of the list down those stories it can commit to delivering by the end of the sprint — taking into consideration their present constraints including, for example, available capacity, or the required technical tasks that need to be addressed during the same sprint.

  Sprint planning II: During the second part of sprint planning, the scrum team adds detail to the user stories in the sprint backlog (e.g. splitting the stories into tasks, identifying parts of the stories that need further clarification, and agreeing on who will be working on what tasks). The product owner does not necessarily need to participate in this second part of sprint planning, but does need to be available to answer questions that the team may have.

- If user story preparation is handled well, an entire sprint planning session might be completed within less than 2 or 3 hours.

- 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 new tasks — including user stories, technical tasks, bugs, and probably spikes. Flow theory shows that a 90% or higher allocation of available capacity will not lead to a team achieving their peak performance.

- Bugs, refactoring, and research require regular attention in order to avoid building-up technical debt. An effective scrum team allocates at least 25% of their capacity to these tasks.

- Incomplete and poorly prepared user stories seriously hamper the effectiveness of a scrum team. These stories should never be selected for the sprint backlog, but instead sorted out during backlog refinement and estimation meetings.

24. A spike is a small task done to reduce uncertainty about a larger task.
25. Flow theory is the theory that an optimal psychological state can be experienced which results in immersion and concentrated focus on a task.

Question 18

> 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 scope of an upcoming sprint by identifying and prioritizing 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, the best way for a scrum master to ensure that a scrum team is working on the most valuable user stories is

01 to ensure that the scrum team is involved in the product discovery process at an early stage;
02 to ensure that the product backlog refinement process is well understood by both the scrum team and the product owner (this should be supported, for example, by the creation of a definition of ready standard for user stories); and
03 to ensure that all user stories are created in a collaborative effort between the product owner and the scrum team (the goal being a shared understanding of the user stories and thus joint ownership).

Your candidate should note that although the product owner defines the scope of the sprint (and the sprint's goal), it is the prerogative of the scrum team to address technical debt and bugs during the same sprint (a team should be able to allocate up to 25% of their available capacity for this).

Question 19

> 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

- revenue increases,
- cost cutting benefits achieved by internal process improvements,
- increases in customer satisfaction rates (NPS[26]),
- increases in signups for new products, or
- positive customer feedback received by the customer care team.

26. NPS® (Net Promoter Score) is a customer loyalty metric and registered trademark of Fred Reichheld, Bain & Company, and Satmetrix Systems.

Question 20

> How do you facilitate user story selection in a way that the most valuable stories are chosen without overruling the scrum team's prerogative to define their own commitment?

If a scrum 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 guidance to see that the most valuable stories are chosen. Most teams will support the product owner's choice of user stories for a given sprint.

If a team resorts to cherry picking — choosing user stories only to satisfy personal preferences — during sprint planning, the backlog refinement process needs to be seriously inspected. In all likelihood the product owner is choosing user stories that are not maximizing customer value.

Question 21

> How much of a scrum 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,
- 10% of a team's capacity to bugs, and
- 5% of a team's capacity to explorative spikes (when potentially helpful).

A scrum team may, of course, deviate from this when it comes to individual sprints. But, generally, consistently making these allocations will satisfy both the code quality and maintenance requirements of most software applications.

Question 22

> Should a product owner assign user stories or tasks to individual members of a scrum team?

A product owner individually assigning user stories to members of a scrum team is not agile, and if a product owner is doing this they need to be stopped. Scrum teams are supposed to be self-organizing. The assignment of user stories and the distribution of tasks among the members of a scrum team is the prerogative of the team itself. Preventing this error should be one of the scrum master's most pressing concerns.

Question 23

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

A scrum 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.

Question 24

> A 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 team's present concerns and experience with the circumstances that caused the story to not meet their definition of ready. 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 team agrees to it — then an exception may be acceptable.

Beware that exceptions have a tendency to become accepted practices. An organization intent on being agile should not be allowed to bypass the backlog refinement and sprint planning process. Your candidate should be aware that such situations are not tenable. Furthermore, if implementation of a user story 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 agile process itself as having failed.

Your candidates may either accept or reject exceptions to the agile 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.

Question 25

> 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 a member of a scrum team does not want to participate in sprint planning and considers the meetings a waste of time, they're exhibiting a type of passive-aggressive behaviour. Although not particular to scrum, this is a problem because the underlying attitude is toxic and will affect both team-building and team performance.

When the member of a scrum team behaves as described, the team's scrum master needs to take action. Counterproductive behaviour can neither be ignored nor tolerated if the team is to continue functioning. Effective action is likely to require a series of escalating steps:

01 The scrum master should start by addressing the team member privately to discuss their reservations and, perhaps, more coaching or a longer training period.

02 Following private discussion, the entire team can be involved by making the team member's reservations a topic of discussion during one or more retrospectives (see Set 5). This enables the team to offer their support.

03 If there is still no change in the team member's attitude, a meeting with the team member and their manager is advisable.

04 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.

4

Standups

Background

- Standups (also known as 'daily scrums') are meetings well suited to discuss a current sprint's progress: is all going as planned, or does the scrum team need to adjust?
- Standups are a convenient time for a scrum team to meet and communicate with a project's stakeholders.
- Standups cannot 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.
- Standups are valuable if the scrum team is already collaborating well and the basics — such as the product backlog, and sprint planning — are in order.

- The more experienced a scrum team, and the better the internal communications, the more a standup will seem a time consuming ritual of little value.

- An advanced scrum team may consider virtual meetings instead of real meetings using, for example, a Slack channel.

- A two person scrum team does not necessarily need a formal standup — meeting for coffee would be a practical alternative.

- There is something wrong with a scrum team who do not communicate impediments to their scrum master prior to each standup. It's possible they're acting more like a group of friends than a scrum team.

- Standups are not reporting sessions for the benefit of product owners or participating stakeholders.

- Offline boards are valuable: physically taking a card and moving it instills a certain ownership of a user story. If you have to let go of either an online or offline board and you're a co-located team, consider letting go of the online board.

27. Slack is a popular online messaging software for team communication and collaboration.

Question 26

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

In answering this question, your candidate should exhibit common sense regarding formal standups. Standups are an important part of scrum, but not all standups need to be formal. A small, experienced, and co-located team may use a coffee break for their standup.

However, taking a relaxed, informal approach to standups for a large team with several junior members would probably achieve nothing — if it doesn't first descend into chaos. For large teams, a formal meeting is needed to provide format and guidance.

For distributed teams who can't easily meet for coffee, a formal standup is necessary to accommodate technical constraints, and must be scheduled and conducted in an organized fashion.

Question 27

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

When impeded, members of a scrum team should never need to wait, neither for a standup 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 standup before either asking for help or themselves dealing with an impediment, the scrum master has team-building work to do.

Question 28

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

There are no **official** leadership roles in scrum. However, it's not uncommon for some members of a scrum 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 scrum 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 standups and otherwise — in the spirit of scrum.

Question 29

> How do you manage team members who consider standups to be a waste of time and are therefore either late, unco-operative, or simply don't attend?

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

Question 30

> Your team's standups are not attended by any stakeholder. How do you change that?

Asking this question can easily spark a philosophical discussion about whether stakeholders should be allowed to participate in a scrum team's standups. Try to avoid this.

If stakeholders participate in a team's standups, 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 standups need not be ruled out if the team finds it acceptable. In fact, if stakeholders attend standups regularly, this invariably and significantly improves communication between a team and their stakeholders.

So how does a scrum master encourage stakeholders to attend standups? By making it worth their while. The scrum master might achieve this in any number of ways: for example, they might offer stakeholders the opportunity to learn early details of a new product or feature, or they might choose to give stakeholders the opportunity to ask questions of the engineers directly (without otherwise going through the product owner).

Question 31

> How do you approach standups with distributed teams?

Standups for scrum teams whose members are distributed between different offices or working remotely are not much different to standups for scrum 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, 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 Skype or Google Hangouts[28] call will likely be enough for a distributed team to have their standup.

28. Skype and Google Hangouts are popular computer-based video telephony applications.

Question 32

> Can you draw an example of a scrum team's offline kanban board, right now?

In this question, the qualifier 'kanban' is a teaser. Anyone interviewing for the role of scrum master should be able to draw a simple offline board.

There are usually five columns (or rows) on an offline board:

01 Backlog
02 In progress
03 Code review
04 Quality assurance
05 Done

Additional information may be included on or attached to any kind of board, e.g.

- sprint or meeting dates,
- user acceptance tests (UAT),
- a definition of ready,
- a burndown chart (progress and work remaining over time), and
- a parking lot (topics for future discussion).

Your candidate should mention that a scrum master is not obligated to provide the scrum team with an offline board. A board is the responsibility of the team working with it. The scrum master should, however, provide an introductory workshop on the subject if no member of the team is familiar with offline boards.

Read more: How to Build Offline Boards.

5

Retrospectives

Background

- Retrospectives should encourage self-expression, thereby making it easier for a scrum team to uncover the concerns and frustrations that its members may be harbouring so that strategies may be devised to overcome them.

- 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 not helpful. During a retrospective, the members of a scrum team should focus on how to improve a situation — and avoid blaming one another.

29. A retrospective, also known as a 'sprint retrospective', is a meeting held at the end of a sprint during which the members of a scrum team may discuss the sprint.

- Some scrum teams always include the product owner in their retrospectives, while other teams insist that the product owner should be expressly invited.

- It's best not to hold 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 retrospectives should be changed regularly. The same format should not be run more than twice.

- Smartphones, tablets, and laptops should not be permitted at retrospectives so that the members of the scrum team are not distracted, and can focus on contributing to the meeting.

- 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.

- Retrospectives must produce answers to certain questions. The '**classic**' set of questions includes

  - What went right?
  - What went wrong?
  - What is there to improve?

An alternative set of questions is the '**starfish**' retrospective:

- What to introduce?
- What to keep doing?
- What to stop doing?
- What to do more of?
- What to do less of?

- An alternative to asking questions at a retrospective is to employ the **Mad Sad Glad** 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.

- According to Diana Larsen and Esther Derby in their book Agile Retrospectives: Making Good Teams Great, there are five stages to running a retrospective: setting the stage, gathering data, generating insights, deciding what to do, and closing the retrospective.

30. Mad Sad Glad is a retrospective exercise designed to elicit feedback and possible corrective actions.

- A retrospective should set **SMART**[31] 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 an estimate of when results can be expected.
    - Action items should be placed on a board to make tracking progress visual and more prominent.
- Every new retrospective should start with reviewing the status of the action items decided upon during the previous retrospective.

Question 33

> Who should participate in a retrospective?

Only the immediate members of a scrum team should participate in that team's

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

retrospectives. Especially important is that the managers of a team's members not be present.

The only exception is the product owner. It's generally a good idea to include the product owner in a scrum team's retrospectives because the product owner is a crucial member of the larger team. But it's not mandatory. Some teams may prefer that the product owner not participate — and a team's wishes must always be considered.

Question 34

> Should you check a team's health during a 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.

One effective method of measuring the health of a scrum team is to circulate a multiple choice questionnaire at the team's retrospectives. A questionnaire that requires just two minutes to complete and uses a simple scale for each of the questions, e.g.

01 terrible
02 poor
03 neutral
04 good
05 excellent

is usually best.

During the retrospective, upon completing the questionnaire, the team should discuss the results with an aim to uncover any concerns or frustrations they may be harbouring.

Question 35

> What retrospective formats have you used in the past?

There are various 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:

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 retrospective

- Start doing ...
- Do less of ...
- Do more of ...
- Stop doing ...
- Continue doing ...

The Diana Larsen and Esther Derby format

- Set the stage
- Gather data
- Generate insights
- Decide what to do
- Close the retrospective

32. This is the format described in the book Agile Retrospectives: Making Good Teams Great by Diana Larsen and Esther Derby.

Question 36

> How do you prevent boredom during retrospectives?

When required to attend a boring retrospective, members of a scrum team will become bored.

There are many possibilities for variation that can be used to prevent a 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 acknowledgement. Web sites like Retromat offer hundreds of different games and exercises to make 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.

Question 37

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

During a retrospective, the members of a scrum team are usually expected to pick a series of action items (tasks to be done). 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 must 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 team. In this latter case, the scrum master needs to provide the offending team members with sufficient encouragement or motivation to overcome the problem — and then see that they deliver on their commitments.

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 team will suffer as a result.

Question 38

> How would you recommend following-up on action items?

A scrum master is expected to follow up on the action items (tasks to be done) that members of a scrum team pick during their team's retrospectives. 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 retrospective before picking new ones by initiating a discussion at the beginning of each new retrospective. If this discussion uncovers action items picked during a previous retrospective that haven't been completed as expected, the team needs to understand why — and prevent it from happening again.

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 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 micro-manage, 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 — agile practitioners all — with insights on how to continuously improve, helping them achieve their goals:

- Agile practitioners strive for autonomy, mastery, and purpose as explained in this video by Daniel Pink.
- Agile practitioners address personal development with metrics by applying methods like Objectives and Key Results (OKR).

The experienced agile practitioner realizes that autonomy and accountability are equally important for self-organized scrum teams. Without metrics, both autonomy and accountability are limited.

- The metrics most suitable to agile 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 agile:

    - 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 metric used for agile must be tailored to the organization.

- The metrics that the scrum master should be tracking are only those that apply to the scrum team. 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.

Question 39

> Are there any standard metrics that you would track? If so, which, and for what purpose?

When tracking metrics at the organizational level, the effects of any process or change can be measured quantitatively with a metrics scoring model. The effects measured would include

- the ability to respond to change and produce valuable code (e.g. the capacity to break down features);
- the duality of planning at both release and sprint;
- the flexibility to adapt to changing facts, time boxes, and continuous delivery;
- the frequency with which scrum teams are bidding on stories, and whether the teams are exercising any freedom in their approach to solving them;
- the creation and growth of a culture of shared learning; and
- the continuity with which features are delivered.

The design of a metrics scoring model should take into account the agile maturity of the organization such that qualitative aspects may be quantified, and thus compared. If the metrics scoring model can be designed before introducing an agile framework into an organization, the status quo should be surveyed in order to establish a baseline against which to measure these effects and track their evolution over time.

Any metrics useful to measuring the effects of a relevant process or change should be recorded regularly, throughout the agile journey. Surveying the members of an organization's scrum teams is a good start.

Question 40

> Your scrum team is consistently failing to meet commitments, and their velocity is volatile. What are the most probable reasons for this problem, and how would you address it with the team?

If a scrum team is exhibiting a volatile velocity and consistently failing to meet their commitments, 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 an industry 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 may 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 priority bugs.

Another common cause for a scrum team to consistently fail in meeting their commitments is that the team's commitments are frequently too aggressive. This might indicate that the user stories are being poorly prepared (e.g. not meeting the team's definition of ready), thus making the stories 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 an agile adoption is working only because a scrum team's commitment and velocity are aligned. Cooking the agile books is easy to do!

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

Question 41

> What qualitative agile metrics would you consider tracking?

The purpose of qualitative metrics in agile 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 — the Scrum Checklist by Henrik Kniberg is a good example. The interval to test via self-assessment is every 4–12 weeks, with teams of lesser maturity 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[33] 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

33. An agile practice map is a method of organizing user stories to prevent failures that may be caused by incremental delivery.

- 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 talking points during a team's 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 generally.

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

34. A product delivery organization is effectively everyone within an organization who's involved with getting a product to market.

7

How to kick-off a transition to scrum

Background

- There is no checklist or master plan readily available, or that could be made readily available, that would ensure a successful transition to scrum.

- The 'best practices' of and 'lessons learned' by other organizations during their transition to scrum may indicate a direction to take when transitioning, though the context of their transition may not be comparable: what worked for Spotify[35] may not work for General Motors[36].

35. Spotify is a popular online music subscription service.
36. General Motors, the car manufacturer.

- 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 will be affected by a successful 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, it is likely to deliver a lesser return on investment.

Question 42

> 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, quality assurance (QA) professionals, 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 eight 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.

Question 43

> 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 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, and creating offline boards.

Question 44

> 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 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 backlog into a well-refined, current product backlog such that a new scrum team (including the product owner) could work with.

Candidates should mention that running a product backlog refinement workshop creates a good opportunity to provide a new scrum team and product owner hands-on training with scrum. This is because a 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 scrum.

Read more: Product Backlog Refinement.

Scrum anti-patterns

Background

- Humans — including scrum masters — are fallible. Given the human propensity for error, there will always be room for (professional) improvement.

- Anti-patterns will emerge when core principles (as explained in the Manifesto for Agile Software Development and the Scrum Guide) are ignored, made to fit existing structures, or watered down.

- A 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 agile and scrum anti-patterns is fundamental to any effort to achieve serious, continuous improvement.
- Anti-patterns can be identified by observation, retrospectives, and other forms of feedback-generating activities.
- **Read More**: The Scrum Anti-Patterns Guide is another free ebook in Age of Product's collection of practical guides from the trenches. It discusses more than 140 scrum anti-patterns that can block your team's improvement.

Question 45

> What anti-patterns might a scrum master fall into during a sprint?

Anti-patterns that a scrum master might fall into during a sprint will impede a team's productivity. The most typical of these are summarized here. It is the scrum master's obligation to prevent these anti-patterns from manifesting themselves.

✔ Flow disruption

The scrum master allows stakeholders to disrupt the workflow of the development team during the sprint.

There are several possible ways stakeholders can interrupt the flow of a 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.

✓ 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.

✔ **#NoRetro**

The scrum master does not gather data during the sprint that would support the team in the upcoming retrospective.

This is self-explanatory.

Question 46

> What anti-patterns do you know of that can happen during a retrospective?

Typical retrospective anti-patterns include

✔ **Waste of time**

The team does not collectively value the retrospective.

If some team members consider the retrospective to be of little or no value, it is most often the retrospective itself that's the problem. Is it the same procedure every time, ritualized and boring? Have a meta-retrospective on the retrospective itself. Change the venue. Have a beer- or wine-driven retrospective. There are many things a scrum master can do to make retrospectives interesting and valuable, reducing the absence rate. Furthermore, it's good to remember that (in our experience) introverts, not just extroverts, like to take part in retrospectives.

✔ Prisoners

Some team members only participate because they are forced to team up.

Don't pressure anyone to take part in a retrospective. Instead, make it worth their while. The drive to continuously improve as a team needs to be fueled by intrinsic motivation, not by fear or command. Tip: Retromat's Why are you here? exercise is a good opener for a retrospective from time to time.

✔ Groundhog day

The retrospective never changes in composition, venue, or length.

In this case, it's likely that the team will revisit the same issues over and over again — like groundhog day, without the happy ending.

✔ Let's have it next sprint

The team postpones the retrospective until the next sprint.

Beyond the "inspect and adapt" task, a retrospective serves as a moment of closure, helping reset everybody's mind so that the team can focus on the next sprint's goal. That's the reason we have a retrospective before planning the follow-up sprint. Postponing a retrospective until the next sprint can also interrupt the team's flow, thus delaying possible improvements. For these reasons it's important to hold a retrospective before planning the follow-up sprint.

✔ #NoDocumentation

No one is taking minutes for later use.

A retrospective is a substantial investment for many reasons and should be taken seriously. Taking notes and photos supports the process.

✔ No psychological safety

Each retrospective is an endless cycle of blame and finger pointing.

The team wins together, or the team loses together. Unfortunately, the blame game indicates both the failure of the scrum master as the facilitator of a retrospective as well as a team's lack of maturity and communication skills.

✔ Bullying

One or two team members are dominating the retrospective.

This communication behavior is often a sign of either a weak or disinterested scrum master. The 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 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 retrospective.

There are plenty of scrum ceremonies that address the communication needs of stakeholders: the sprint review, the product backlog refinement, the daily scrums — not to mention opportunities for conversation at water coolers, over coffee, or during lunch. If those possibilities for communicating are still not sufficient, hold additional meetings. The retrospective is for the team, and should be off-limits to stakeholders.

✔ Passivity

Team members are present, but are not participating.

There are plenty of reasons for such behavior: team members regard the retrospective as a waste of time, it's unsafe, or the participants are bored by its predictiveness. Team members may also fear negative repercussions should they be absent... or maybe a 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 retrospective will work best in their organization's context.

Question 47

> How can you (as a scrum master) identify where you need to improve?

How can a scrum master identify where they need to improve? It's a simple question, with a simple answer: a scrum master should regualrly ask their team and stakeholders how they think he or she might improve.

Scrum masters can run retrospectives on themselves. A dedicated retrospective is much more effective than asking for hints at how one might improve during the last minutes of the regular team retrospectives.

Conclusion

How to use these interview questions

At 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 methodologies 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 60 minute conversation.

Scrum has always been a hands-on business, so your candidate needs 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, the more levels of management there are, the more likely there will be resistance or possibly even failure when applying agile. In these circumstances you would be wise to choose the pragmatic veteran who has experienced failure at other organizations (and who carries the scars to prove it) over a junior scrum master.

Being a Certified ScrumMaster[redacted] — or having any such kind of designation — does not guarantee success.

37. An industry certification provided by (and trademarked by) Scrum Alliance, Inc.

About the authors

@stefanw

/in/stefanwolpers

stefan@age-of-product.com

Stefan Wolpers

Author

Stefan has worked many years as a product manager, product owner, and agile coach (Scrum, LeSS[38], Lean Startup, Lean Change). He's founded multiple companies, and has led the development of B2C and B2B software, primarily for startups, but also for other organizations — including a former Google subsidiary. He is a steward of the XSCALE Alliance and an XBA Exponential Business Agility Coach (XBAC).

Despite originally studying chemistry Stefan has never worked in a laboratory, and instead

38. LeSS (Large-Scale Scrum) is a product development framework that extends scrum with scaling rules and guidelines.

continued his education in business administration and law. Following school he discovered a passion for software and, in 1996, launched the first online ecommerce platform to feature SAP R/3 connectivity — only to learn that the early bird does not necessarily catch the worm. After moving from his home town of Hamburg to Berlin, Germany, he created Susuh GmbH, a marketplace for local services. Other ventures followed, and in 2011 he founded Startup Camp Berlin — one of the largest German startup conferences today.

Stefan's latest project, Age of Product, focuses on the exchange of knowledge between the people involved in product development: product managers, product owners, scrum masters, designers, and developers. The goal is to help those involved in product development with lessons learned and best practices for continuous agile product discovery and delivery.

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

/in/andreeatomoiaga

Andreea Tomoiaga

Contributor

Andreea is a product manager who is passionate about designing and optimizing B2B integrations. She is keenly interested in strategy and prioritization 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 architect, project manager through product manager — successfully delivering projects of different sizes by combining a variety of managerial methods. 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® and PRINCE2® Practitioner. She likes to write about various product and project management topics on her personal website.

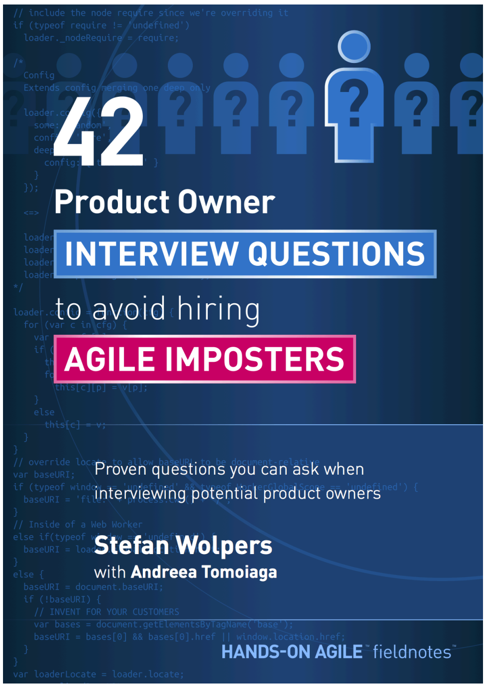
Read more about Andreea at Scrum Alliance, and connect with her via LinkedIn.

Other publications

The 2017 Scrum Master Salary Report

Age of Product's comprehensive report on world-wide scum master demographics, educations, and salaries, the **2017 Scrum Master Salary Report** is the industry's first and only report to reveal the earning potential of scrum masters worldwide.

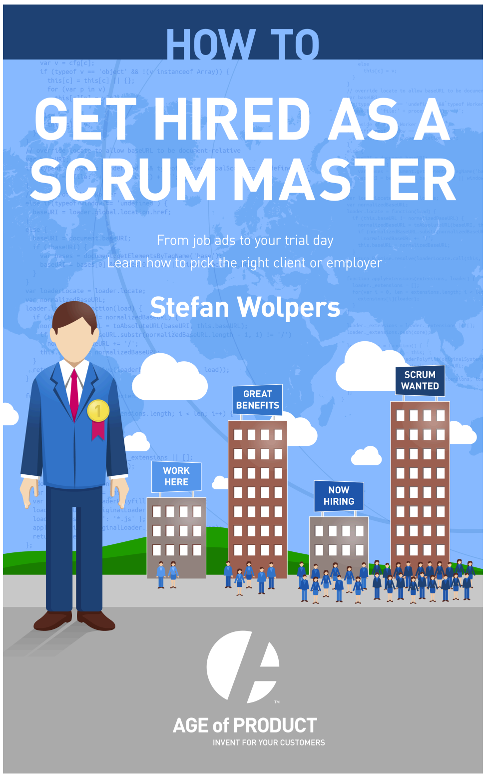
Download this comprehensive report for free



42 Product Owner Interview Questions to Avoid Hiring Agile Imposters

It takes more than a scrum master to successfully create an agile organization. **42 Product Owner Interview Questions to Avoid Hiring Agile Imposters** provides the interview questions you need to identify the right candidate for the role of product owner in your organization.

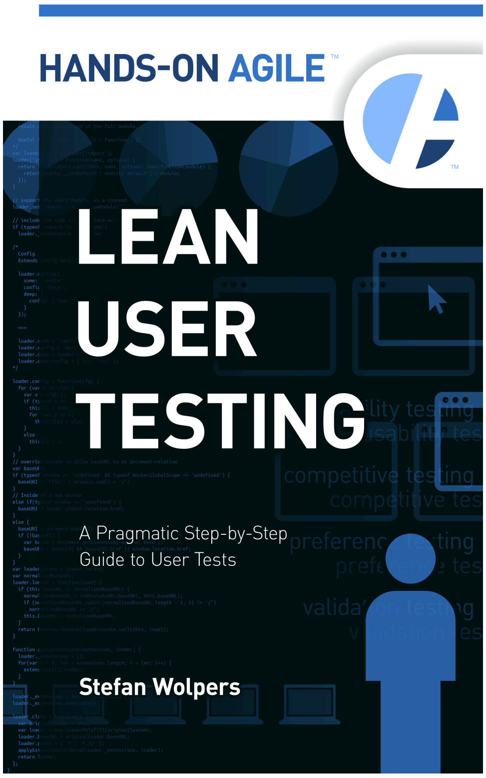
Download this 63 page book for free



How to Get Hired as a Scrum Master

From job ads to your trial day, **How to Get Hired as a Scrum Master** teaches you from a candidate's perspective how to select the right client or employer, and helps you prepare to take on the job.

Coming soon! Subscribe to Age of Product's weekly newsletter to be among the first to know when the book becomes available.



Lean User Testing: A Pragmatic Step-by-Step Guide to User Tests

Lean User Testing: A Pragmatic Step-by-Step Guide to User Tests is a practical guide for everyone who wants to create, design, or market software: designers, developers, entrepreneurs, product managers, marketers, and salespeople.

Buy and download the book from Amazon

AGE of PRODUCT

INVENT FOR YOUR CUSTOMERS

www.age-of-product.com

38 Scrum Master Interview Questions to Avoid Hiring Agile Imposters, Fourth Edition

Written and compiled by Stefan Wolpers and Andreea Tomoiaga. Published and distributed by Berlin Product People GmbH. First published 2017 December 15.

Edited by Candace Chapman and Jourdan Ritchey. Typesetting and design by Jourdan Ritchey and Roman Morhunov.

Berlin Product People GmbH

Borsigstrasse 8
10115 Berlin
Germany

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

info@berlin-product-people.com

TM