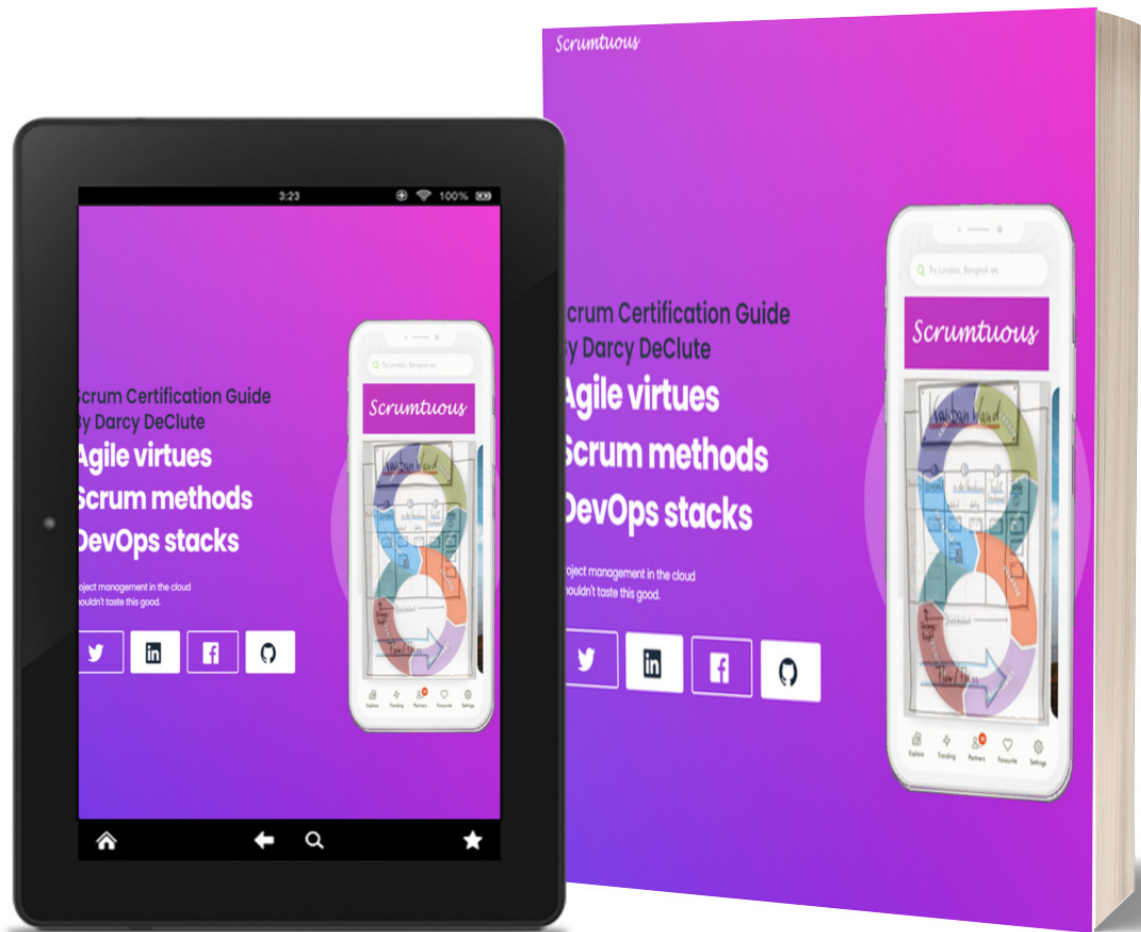# Scrum Certification Guide

Getting Started

I'm going to get you Scrum Certified. I hope you're ready.

In the process we're going to:

- Go through the whole 2020 Scrum Guide, one line at a time
- Extract as much meaning as possible out of the Scrum Guide's words
- Discuss how ideas in the Scrum Guide translate into exam questions
- Iteratively and incrementally test you on what you've learned

When you've finished this book, my goal is for you to be 100% ready to schedule and pass the Scrum Certification exam.

The Best Ways to Learn

Before you read too far ahead, let me recommend that you download and print out a couple of copies of the 2020 Scrum Guide.

The Guide is only 13 pages long, and that includes the cover page, introduction, and table of contents. You won't be destroying a forest by having a couple of printed copies by your side.

Grab a marker and a highlighter as well and markup that printed copy of the Scrum Guide as we go along. It will seriously help you learn.

Figure 1. Try to hit as many learning modalities as you can.

There are many different 'learning modalities' help people learn.

- Reading a book is one modality.
- Watching videos online is popular too.
- Attending lectures is a proven learning modality.
- The tactile experience of putting pen to paper is another.

Try and stimulate your learning in as many ways as you can.

Before We Begin

There is one required prerequisite before you dig into this book.

For this book to make sense, you must have read the 13-page 2020 Scrum Guide at least once.

This book is designed to give you a deeper understanding of the topics covered in the Scrum Guide, with the eventual goal of helping you get Scrum Certified.

However, this book assumes you have read the 2020 Scrum Guide at least once, hopefully twice.

Figure 2. Try to hit as many learning modalities as you can.

This book assumes you know the basic idea behind who a Scrum Master is, what a Product Backlog is, and the idea behind the Definition of Done.

We will often reference terms like 'Increment' or the 'Product Goal' because we assumed that you have read the Scrum Guide at least once and have some fundamental understanding of these concepts.

But once you've given the 2020 Scrum Guide a read or two, you're ready to jump into this Scrum Certification guide and get yourself on the path to career advancement.

Now let's get started with the Scrum Guide!

What is Scrum?

How would you define Scrum?

To be successful on the Scrum Master Certification exam you have to commit to the Scrum Guide's definition of Scrum, which means abandoning many biases and misconceptions you may have adopted over years of hearing people talk about Scrum or seeing Scrum implemented in a less than pure manner.

Here's the 2020 Scrum Guide's first sentence. How well does this definition of Scrum work with the way you previously perceived it? (And I say 'previously', because this is the definition you must commit to if you want to pass the Scrum Certification exam.)

> Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.
>
> — 2020 Scrum Guide page 3

Given what you know about Scrum, and taking into account any experiences you've had with Scrum, how would you rate this definition?

The Scrum Framework provides a powerful strategy for product development. Just don't call it a process or methodology!

image::images/scrum-incomplete.png

The Definition of Scrum

Whoever crafted that definition tried to make it as general and all-encompassing as possible, almost to the point where the definition doesn't provide much value.

- The term 'generate value' is very generic
- The term 'complex problems' could apply to anything
- The term 'adaptive solutions' sounds like marketing gibberish

But this is the definition we have, and this is the definition you will be tested on.

A couple of things to notice about the official definition of Scrum:

- The definition never mentions software development
- The definition calls Scrum a framework, not a process or methodology

Given that definition of Scrum, how would you answer the following question?

Test Yourself

The answer is false.

Scrum is not a process, nor does it specifically target software development.

You'll get beaten with a stick if any of the Scrum gatekeepers ever hear you call Scrum a process or a methodology. Scrum is a lightweight, incomplete framework.

- Scrum is not a process.
- Scrum is not a methodology.
- Scrum is purposefully incomplete.

Scrum is a Framework

Feel free to debate whether you believe Scrum is a process or a methodology on Twitter or in your favorite online forum. I know I have.

On the Scrum Certification exam? Scrum is a framework.

The stewards of the Scrum framework have also worked hard to position Scrum as a tool that can be applied in a variety of industries, not just software development.

If you ever see an option on the certification exam that asserts Scrum works exclusively in the domain of software development, avoid it, because it's wrong.

Test Yourself

Here's the type of trick question you'll see on the Scrum certification exam that attempts to trip you up on the incorrectly held belief that Scrum is only used in software development:

> True or False: Scrum is a lightweight framework used exclusively by software development teams to generate value through adaptive solutions to complex problems.

The answer is false because the question implies that Scrum is only applicable in the world of software development.

There is a big push in the Scrum community to gain acceptance outside of software development. Any certification questions that pigeonhole Scrum into a software development box will be wrong.

Test Yourself

> Which of the following statements most accurately reflects the definition of Scrum?
>
> ☐ Scrum is a software development methodology
> ☐ Scrum is an Agile process for teams and organizations to following
> ☐ Scrum is a lightweight framework to help teams tackle complex problems
> ☐ Scrum is a lightweight framework to help teams and organizations build software

Option C is correct.

The Guide very vaguely describes Scrum as a "lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems."

Any references to Scrum being a methodology, a process, or a framework that targets software development will always be a wrong answer on the Scrum Certification exam.

Iterative and Incremental

According to the Guide, here's a high-level overview of how Scrum is supposed to work.

> In a nutshell, Scrum requires a Scrum Master to foster an environment where:
>
> 1. A Product Owner orders the work for a complex problem into a Product Backlog.
> 2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
> 3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
> 4. Repeat
>
> — 2020 Scrum Guide page 3

The name 'Scrum Master' sounds intimidating.

People think that since the term 'master' is in the name, the Scrum Master controls everything.

The Scrum Master controls very little. Their only real job is to coach people on how Scrum works, or as this paragraph states, 'foster an environment' where this iterative set of steps is performed.

Test Yourself

Which of the following descriptions is true?

☐ Scrum describes an iterative process

☐ Scrum is an iterative framework

☐ Scrum generates value by repeatedly delivering usable increments to the stakeholders

☐ Scrum only allows stakeholders to inspect progress when the final product is delivered

Scrum describes a set of steps that are to be repeated iteratively. So Scrum is iterative. But it's an iterative framework, not an iterative process. So Option B is correct while Option A isn't.

Scrum is an incremental framework, which means it constantly tries to deliver something tangible and of value to the client at the end of every sprint. That way the stakeholders can regularly give feedback. If there's an issue, the Scrum team can then adapt.

That's in stark contrast to what is known as the Waterfall model where the client gets a complete product at the end of development. So Option C is correct while Option D is wrong.

Scrum is Simple

Many people overthink things in Scrum.

People think there are a bunch of rules they have to follow if they want to use Scrum.

The fact is, there are very few rules in Scrum. The brevity of the Scrum Guide is proof of that.

Scrum is pretty simple, and when problems arise, it's pretty pragmatic too.

> Scrum is simple.
>
> Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value.
>
> The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory.
>
> Scrum is built upon the collective intelligence of the people using it.
>
> Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.
>
> — 2020 Scrum Guide page 3

Figure 3. Scrum is a lightweight, incomplete framework that helps teams generate value as they work towards a Product Goal.

It's a Guide, not an Instruction Manual

People often look to the Scrum Guide for definitive answers to things. The Scrum Guide doesn't contain many definitive answers.

It's a guide, not a rulebook.

The Scrum Guide even promises *not* to be heavy on rules in this paragraph, saying that it promises not to 'provide people with detailed instructions.'

There are very few actual rules in the 13-page Scrum Guide.

Outside of the few rules Scrum has, the framework encourages people to discover strategies that work best for them.

Test Yourself

> True or False: Scrum is a complete and proven framework that helps teams achieve goals and create value.

This is false.

Scrum self-identifies as an incomplete framework.

This fact seems counter-intuitive to many. After all,

- Why would anyone want to use an incomplete framework?
- Wouldn't a complete framework be better?

The incomplete nature of Scrum is actually what makes it so attractive. Scrum provides only enough direction to be useful, but not so much direction that it is restrictive. Scrum teams are given all the leeway they need to find the processes and frameworks that work best for them.

Exposing Efficacy

One of the funny things about Scrum is that because it's so simple, it can expose practices and processes that are wasteful and non-productive. It also allows developers to focus on the practices that make them most productive.

> Various processes, techniques, and methods can be employed within the framework.
>
> Scrum wraps around existing practices or renders them unnecessary.
>
> Scrum makes visible the relative efficacy of current management, environment, and work techniques so that improvements can be made.
>
> — 2020 Scrum Guide page 3

Since Scrum is a framework, not a process, other processes can be used within it.

Combine the Scrum Framework with other Processes

For example, people often think Kanban is a competitor to Scrum, but there is nothing that says Scrum and Kanban can't be used together.

If you're not familiar with Kanban, don't worry. Kanban is never mentioned in the Scrum Guide and it will never be a 'correct answer' on the Scrum certification exam.

Test Yourself

> True or false: Scrum can be used alongside various processes and methodologies including Kanban and Lean.

This is true.

Scrum is not a process nor is it a methodology, and because of that, it can be used in conjunction with a variety of popular methodologies like Kanban and Lean.

The Scrum Certification Exam will not test you on the intricacies of Lean Manufacturing or Kanban. It's sufficient just to know that these are two processes commonly used in manufacturing and software development.

> True or False: When implemented properly, Scrum will expose ineffective management techniques that may not have anything to do with software development.

First of all, Scrum is not just about software development.

The people who oversee the Scrum framework are pushing hard to have Scrum used in all areas of industry and manufacturing. To be successful on the Scrum certification exam, get the idea out of your head that Scrum is just about software development. It's not.

And secondly, the iterative and incremental nature of Scrum, where there is constant inspection and adaptation is supposed to shine a light on practices external to Scrum that may be ineffective. That's what the Scrum Guide means when it says "Scrum makes visible the relative efficacy of current management, environment, and work techniques so that improvements can be made."

We're done with the definition of Scrum. Now on for a little overview of what Scrum Theory is and what it's based on.

TIP Scrum is not just for software development. It can be used in industry, manufacturing, construction, and even beekeeping. Sometimes, to understand Scrum, frame it as an industry other than software development.

Chapter 2A(new chapter name)

Scrum Theory

Scrum boasts of being built around the concepts of 'empiricism and lean thinking.'

The subject of 'Empiricism' is one of the 13 groups into which questions on the Scrum Master certification exam are categorized, so pay special attention to the concept.

> Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.
>
> — 2020 Scrum Guide page 3

I'm not a big fan of the word empiricism.

Fundamentally, empiricism means using your six senses to understand the situation you're in. I'm not sure how my sense of smell or taste helps me as a Scrum Master.

In the context of Scrum, empiricism means understanding the situation you are in based on knowledge, experience, and verifiable facts.

Empirical Example

Here's an example of the application of empiricism.

Imagine you have a burndown chart that says you are completing 10 stories a Sprint. How many stories do you think you'll complete in the next Sprint?

You'd probably say 10, but that's an answer based on a chart. Charts don't have six senses.

Now say you know half the developers have gone back to India to be with their families during the next Sprint?

You'll probably reduce your projection by 50 percent. That decision is based on empirical evidence that a chart simply couldn't provide.

Lean thinking is also important. Avoid waste and embrace minimalist thinking when you're working on a project. That's Lean thinking.

Empiricism and Lean Thinking

You are guaranteed to get two questions from this section on the Scrum Certification Exam. They are easy marks if you just understand what 'empiricism' and 'lean thinking' means.

Here are two multiple-choice questions for you.

Test Yourself

Scrum theory emphasizes empiricism. That means a decision made by Scrum teams:

- ☐ Should be based on factual evidence
- ☐ Should be based on intelligent assumptions
- ☐ Should be based on verifiable observations
- ☐ Should be driven by pragmatic speculation
- ☐ Should be based on experience

Options a, c, and d are correct.

Empiricism is all about using your experience, your gained knowledge, and verifiable observations (which is sorta the same as 'experience') to make decisions.

Concepts like 'speculation' and 'assumptions' run counter to the concept of empiricism, even if the speculation and assumptions are pragmatic and intelligent.

"In empiricism, knowledge is spoken of as a posteriori, or "from the latter," meaning gained from experience. Simply put, empiricism is the idea that all learning comes from only experience and observations.

The term empiricism comes from the Greek word for experience: empeiria. The theory of empiricism attempts to explain how human beings acquire knowledge and improve their conceptual understanding of the world."

https://www.techtarget.com/whatis/definition/empiricism

Test Yourself

Which of the following are characteristics of lean thinking?

- ☐ A focus on waste reduction and efficiency
- ☐ A command and control approach to problem-solving
- ☐ A minimalist approach that removes unnecessary steps
- ☐ A methodology that is made up of individualized, compartmental steps that flow into each other.

In this case, A and C are correct.

Efficiency, waste reduction, and the elimination of unnecessary steps within a process are all hallmarks of lean thinking.

By the way, the old Scrum Guide use to talk about the importance of servant leadership. That's still important, but it's just not part of the Guide anymore. The 'Command and Control' approach that perhaps the military takes is the opposite of 'servant leadership.'

The last option describes the Waterfall methodology, which is the antithesis of the Scrum

framework.

Empiricism and lean thinking lay the foundation for the Scrum Framework. Be comfortable with these two terms and understand their basic meaning to score a few easy marks on the Scrum Certification Exam.

Predictability and Risk

> Scrum employs an iterative, incremental approach to optimize predictability and control risk.
>
> — 2020 Scrum Guide page 3

This one sentence is a lot to unpack.

Scrum is iterative because it describes a set of steps to be repeated over and over again. The iterative sequence of steps as outlined in the first section of the Scrum Guide:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. Repeat

Scrum is incremental. That's referenced in the second step of the iterative process:

"The Scrum Team turns a selection of the work into an Increment of value during a Sprint."

The idea of Scrum being incremental means that small victories, small units of value, and small pieces of the final product get created and added together slowly over time until the product is finally finished. Piece by piece, through the delivery of value-added upon value, the product gets built. That's the incremental process.

Cross-Functional Teams

The idea that teams should be cross-functional and self-managed is a key concept in Scrum, and it is one that you will be tested on multiple times when you sit for the Scrum Certification exam.

> Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.
>
> — 2020 Scrum Guide page 3

This is another extremely loaded statement.

Understanding this paragraph will go a long way toward properly answering some of the most challenging questions on the Scrum Master Certification exam.

Scrum assumes that the Scrum Team has all the skills required to build the product being developed.

- Does your project need testers? Then those people are on the Scrum team.
- Does your project need an architect? That person is on the Scrum team.
- Does your project need a performance or security specialist? Then a person with those skills must be on the Scrum team.

And what if your Scrum team doesn't have those skills?

Then the people on the Scrum team better acquire them.

Test Yourself

> One of the items under development as part of your project is a spaceship to Mars, but nobody on your team knows how to build a Spaceship to Mars. What should you do?
>
> ☐ Outsource the development of a spaceship to a third party
>
> ☐ Remove the development of a spaceship from the project requirements
>
> ☐ Explain to the Product Owner that you do not have the skills to build a spaceship to Mars
>
> ☐ Get the team to start learning about how to build a spaceship for Mars

This question is silly to the extreme, but it makes a point.

According to Scrum, all of the skills required to build a project under development exist on your team, or your team will take it upon themselves to acquire the skills needed.

If your team outsources work to a third party, then the work in question is no longer within the control of the team, which means it is no longer part of the Scrum process. That's what the Scrum Guide means when it says "Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed."

The All-Encompassing Sprint

> Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint.
>
> — 2020 Scrum Guide page 3

This statement is the source of the most commonly asked trick questions on the Scrum Certification exam, which are:

- Which events happen after the Sprint finishes?
- Which events happen before a Sprint begins?
- Which events happen before or after a Sprint?
- When a Sprint ends, when does the next Sprint begin?

Scrum has four time-boxed events that happen within a fifth Scrum event known as a Sprint. Sprint Planning, the Daily Scrum, the Sprint Review, and the Sprint Retrospective all happen within the confines of a Sprint.

- None of the Scrum events happen after a Sprint
- None of the Scrum events happen before a Sprint.
- None of the Scrum events can be left out of a Sprint.

Everything in Scrum happens within the boundaries of a Sprint. As soon as one Sprint ends, the

next Sprint begins.

There is no buffer time between when one Sprint ends and the next Sprint starts where integration takes place, quality assurance happens or testing is done. If any of those things are part of the development of your product, all of those have to happen during the Sprint.

Don't get tripped up on any questions that ask what happens before or after a Sprint.

Inspection and Adaption

Notice how the Scrum Guide states that the higher purpose of the different Scrum Events, such as the Review, Retrospective, Planning meeting, and the Daily Scrum is to 'inspect and adapt.'

> Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint.
>
> — 2020 Scrum Guide page 3

You will often get questions on the Scrum Certification exam about what the purpose of the Sprint Retrospective is or what the purpose of the Daily Scrum is. If any of the options include the terms 'inspect' or 'adapt', those will likely be the correct answers.

Test Yourself

When does a new Sprint begin?

- ☐ When Sprint Planning is completed
- ☐ When the Sprint Review is completed
- ☐ When the Product Owner begins the Sprint in JIRA
- ☐ After Sprint Planning when the Scrum Master declares the start of the Sprint
- ☐ As soon as the previous Sprint ends

Option E is correct.

A new Sprint begins as soon as the previous Sprint ends.

Technically speaking, the last event in the Sprint is the Sprint Retrospective, and it is the end of this event marks the end of the Sprint. If we were to temporarily jump ahead 7 pages in the Scrum Guide, we'd see that stated in plain text.

> The Sprint Retrospective concludes the Sprint.
>
> — 2020 Scrum Guide page 10

Inspection and Adaption

Notice how the Scrum Guide states that the higher purpose of the different Scrum Events, such as the Review, Retrospective, Planning meeting, and the Daily Scrum is to 'inspect and adapt.'

> Scrum combines four formal events for **inspection and adaptation** within a containing event, the Sprint.

— 2020 Scrum Guide page 3

You will often get questions on the Scrum Certification exam about what the purpose of the Sprint Retrospective is or what the purpose of the Daily Scrum is. If any of the options include the terms 'inspect' or 'adapt', those will likely be the correct answers.

Test Yourself

What is the purpose of the Daily Scrum?

☐ For the Scrum Master to get daily status updates from the developers

☐ To allow the developers to inspect their progress toward the Sprint Goal

☐ For the Product Owner to track the development team's progress on Product Backlog Items

☐ To allow the developers to adapt their Sprint Plan as they work towards the Sprint Goal

Options B and D are correct.

From day to day and hour to hour, conditions change.

Scrum recognizes this reality, which is why it provides several events that allow for the inspection of progress along with the ability to adapt if necessary.

It should be noted that inspection and adaptation can happen at any time during the Sprint, not just during the official Scrum events.

If a computer hosting the Git repo catches fire, you don't wait until tomorrow's Daily Scrum to put it out, nor would you wait to tell the rest of the team about it.

A Word on Empiricism

Empiricism means decisions are made on factual evidence, verifiable observations, and most importantly, an experience that has led to greater knowledge and understanding of a given problem domain.

Empiricism is an approach to knowledge and decision-making based on empirical evidence and experience. In the context of Scrum, empiricism refers to the idea that decisions should be based on observations and experimentation rather than on assumptions and speculation.

Scrum is an empirical process framework that operates on the principles of transparency, inspection, and adaptation. This means that the Scrum Team continuously inspects and adapts its work based on the feedback and evidence it gathers from the ongoing work, to meet the goals of the project. The Scrum Team does this through regular events and activities, such as Sprint Planning, Daily Scrums, Sprint Reviews, and Sprint Retrospectives.

The three pillars of Scrum — transparency, inspection, and adaptation — embody the principles of empiricism in Scrum. They are:

Transparency: The Scrum Team and its stakeholders share a common understanding of the progress, goals, and risks associated with the project. Transparency is achieved through the use of artifacts such as the Product Backlog, Sprint Backlog, and Increment.

Inspection: The Scrum Team regularly inspects its progress toward the Sprint Goal, and uses the results to adapt its plan for the upcoming Sprint. Inspection is achieved through the use of events such as the Daily Scrum, Sprint Review, and Sprint Retrospective.

Adaptation: The Scrum Team adapts its plan based on the feedback and evidence gathered during the inspection. Adaptation is achieved through the use of the Sprint Retrospective and the adjustment of the Product Backlog and Sprint Backlog.

By using an empirical approach, Scrum helps teams to continuously improve their processes and deliver high-quality products that meet the needs of their stakeholders. This is why empiricism is an important part of Scrum Theory.

Chapter 3

Scrum Pillars and Values

'Scrum values' is one of the 13 categories upon which your Scrum Certification exam is graded. It's an imnportant topic.

The 'Scrum Pillars' often arise as answers to questions that ask why an event or artifact is important.

Memorize the values and pay close attention to the pillars. They are key to passing the Scrum Certification exam.

The Scrum Pillars

We've actually already seen a reference to the Scrum Pillars in the Guide:

> Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint.
>
> — 2020 Scrum Guide page 3

Here's the Scrum Guide's formal declaration of the three of them:

> These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.
>
> — 2020 Scrum Guide page 3

It' can't be emphasized enought. The three pillars of Scrum are:

- Transparency
- Inspection
- Adaptation

Everthing that happens in Scrum harps back to these values and pillars. The question as to why a particilar event happens or artifiact is required typically harps back to the Scrum Pillars, with the Scrum values providing further support.

Scrum Values

The five Scrum values that support the Scrum Pillars are:

- Commitment
- Focus
- Openness
- Respect
- Courage

Memorize the Scrum pillars and values, and don't confuse them with each other.

Test Yourself

What of the following are Scrum pillars?

☐ Comitment

☐ Inspection

☐ Openness

☐ Focus

☐ Adaptation

☐ Respect

Options B and E are correct.

The Scrum pillars are transparency, inspection and adaptation.

Inspection

One thing you will notice about the Scrum Pillars is that they feed into each other.

Transparency allows you to honestly inspect.

Inspection can reveal that the plan isn't working, which leads to adaptation.

As you adapt, you must be transparent about how well the adaptation has worked, which requires you to be transparent.

> The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems.
>
> To help with inspection, Scrum provides cadence in the form of its five events.
>
> Inspection enables adaptation. Inspection without adaptation is considered pointless.
>
> Scrum events are designed to provoke change.
>
> — 2020 Scrum Guide page 4

Inspection is all about taking a look at what you and your team is doing so you can figure out if it's working or not.

- Why do we do the Daily Scrum? We do it so the developers can 'inspect' their work and change change their plans if they need to.

- Why do we do the Sprint Review? We do it so the team and the stakeholders can 'inspect' what's been done and see if it's good or not.

- Why do we do the Sprint Retrospective? We do it to 'inspect' how the team worked together during the Spring.

- Why do we have Scrum Artifacts? So people can inspect them and use them as a basis for adaptation.

The need for inspection throughout a Sprint is a common justification for Scrum events and artifacts.

Test Yourself

> Inspection in Scrum is:
>
> ☐ Done frequently to detect potential problems
>
> ☐ Done infrequently to allow teams to concentrate on development
>
> ☐ Done to identify undesirable variances
>
> ☐ Done to ensure undesirable variances do not occur

Inspection is done frequently to detect potential problems and to identify any undesirable variances so that the team can address them.

The concept that anything could ever be done to ensure variances never occur, as option D suggests, is just wishful thinking. In the real world, undesirable variances will always occur. Scrum recognizes that reality and helps teams deal with them early and adapt to them.

Transparency

To know exactly how product development is progressing, everyone in the process must be transparent about the work they are doing.

Empiricism doesn't work if we cannot look transparently into the results of the processes and methods we use within the Scrum framework.

> The emergent process and work must be visible to those performing the work as well as those receiving the work.
>
> With Scrum, important decisions are based on the perceived state of its three formal artifacts.
>
> Artifacts that have low transparency can lead to decisions that diminish value and increase risk.
>
> Transparency enables inspection. Inspection without transparency is misleading and wasteful.
>
> — 2020 Scrum Guide page 3

One of the Scrum values is openness. One of the ways to be open is to be transparent about the work being done and transparent about the progress being made.

The Sprint Backlog creates transparency because it lists everything the Scrum Team is working on, what their goal is during this Sprint and their plan for achieving that goal. If anyone wants to know what the Scrum Team is working on, they can look at the Sprint Backlog. It provides transparency.

What is the quality standard the team is using? Transparency into that is provided by the team's definition of done.

What will the team try to build next? Transparency into that is provided by the way the Product Backlog is prioritized.

Like inspection, the Scrum pillar of transparency is woven into all of the Scrum events and artifacts.

Test Yourself

Scrum allows important decisions to be made emprically by basing those decisions on:

☐ The Scrum values of Commitment, Focus, Openness, Respect, and Courage

☐ Lean thinking

☐ The Scrum pillars of transparency, inspection and adaptation

☐ The perceived state of the three formal artifacts.

Empiricism requires decicions to be made on facts and evidence. In Scrum, evidence comes from the state of Scrum's three formal artifacts, namely the Product Backlog, the Sprint Backlog and the Increment.

The state of these artifacts must be transparent to all, otherwise the team and the stakeholders don't have all of the facts they need to make the right decisions for the future of the product.

Adaptation

Things never go according to plan.

That's one of the reason we don't spend months planning things in Scrum.

It's more important to produce something of value than it is to waste time planning, because nothing ever goes according to plan.

And when plans do go awry, Scrum developers adapt. That's how they achieve their goals.

> If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted.
>
> The adjustment must be made as soon as possible to minimize further deviation.
>
> Adaptation becomes more difficult when the people involved are not empowered or self-managing.
>
> A Scrum Team is expected to adapt the moment it learns anything new through inspection.
>
> — 2020 Scrum Guide page 4

It's understood that in the world of software development things change quickly.

Things also change quickly in the field of construction, manufacturing, banking, etc.

The ability for teams to quickly adjust and change thier plan when things go sideways is a core tenent of Scrum. It's one of the reasons we have the Daily Scrum - it allows developers to collectively discuss problems and adapt as needed.

Always Adapt

The Scrum Guide provides a number of artifacts and time boxed events that provide an opportunity to adapt. However, these are not the only times the team is allowed to meet, speak and adapt.

If a problems comes up during the day, a developer doesn't have to wait until the next day's Daily Scrum to adapt. Nor does a developer have to wait until the next day's Daily Scrum to discuss issues with fellow developers.

If a problem arises, teams are encouraged to inspect and adapt immediately.

If the team's war room catches fire, don't wait until tomorrow's Daily Scrum to leave the building. Ongoing inspection and adaption is a requirement in Scrum.

Test Yourself

> If the process used to track development throughout the Sprint deviates outside of an acceptable limit, the Scrum development team should:
>
> ☐ End the Sprint early and begin a new round of Sprint Planning
>
> ☐ End the Sprint early and do a Sprint Review with all stakeholders
>
> ☐ End the Sprint early and do an internal Sprint Retrospective to see what went wrong
>
> ☐ Adapt during the Sprint and contunue to push towards the Sprint goal.

Scrum is all about adaptation. If things don't go according to plan, the team should adapt. They certainly shouldn't end the Sprint. For the most part, Scrum doesn't allow them to.

- The developers do not have the ability to end a Sprint early in Scrum.

- The Scrum Master does not have the ability to end a Sprint early in Scrum.

Only the Product Owner can do that, and only under the very special condition under which the Sprint Goal has become obsolete.

If things go sideways during a Sprint, the solution is not to cancel the Sprint or end the Sprint early. The solution is to adapt and continue to work towards the Sprint goal.

Sprints are short, typically between 2 to 4 weeks. Even if things go completely sideways, it won't be too long before a new Sprint begins, so continue to work hard towards the Sprint Goal. A new Sprint is always just around the corner.

Test Yourself

> A serious security related bug has appeared in the code written by a fellow developer and you need more details about it in order to fix it. When should this issue be discussed with the developer?
>
> ☐ When the Scrum Master can coordinate a meeting betweeen the two of you
>
> ☐ After the Quality Assurance (QA) team has time to investigate
>
> ☐ During the next scheduled Daily Scrum
>
> ☐ You should go over to the developer's desk and discuss it now

Option D is correct.

There are scheduled invents in Scrum that provide opportunities to inspect and adapt, but those should never be used limit communication and interaction between members of the team.

If a problem arises in Scrum, there's no requirement to wait until a Scrum event happens to address it. Address problems immediately and adapt.

'

Scrum Values

Scrum is a simple, incomplete framework that doesn't solve every possible problem a development team will encounter.

What Scrum does do is provide five values it believes are important. When problems arise, the best solutions will respect these five values.

> Successful use of Scrum depends on people becoming more proficient in living five values:
>
> - Commitment
> - Focus
> - Openness
> - Respect
> - Courage
>
> The Scrum Team commits to achieving its goals and to supporting each other.
>
> Their primary focus is on the work of the Sprint to make the best possible progress toward these goals.
>
> The Scrum Team and its stakeholders are open about the work and the challenges.
>
> Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work.
>
> The Scrum Team members have the courage to do the right thing, to work on tough problems.
>
> These values give direction to the Scrum Team with regard to their work, actions, and behavior.
>
> The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values,

> not diminish or undermine them.
>
> The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts.
>
> When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

— 2020 Scrum Guide page 10

Like the Scrum pillars, the Scrum values provide justification for doing the various Scrum events and the creation of various Scrum artifacts.

Quite often when a Scrum Master, Product Owner or Scrum Developer is faced with a difficult challenge, the answer to problem lies in how to conjur up a solution that is in line with these Scrum values.

For the exam, know the Scrum values and how each of them is defined. There is usually a question or two that will test to see if you know what the Scrum values are and how they are generally defined.

Test Yourself

According to the Scrum Guide, which of the following is not a Scrum Value?

- ☐ Agreeableness
- ☐ Commitment
- ☐ Conscientiousness
- ☐ Openness
- ☐ Respect
- ☐ Extroversion
- ☐ Focus
- ☐ Emotional stability
- ☐ Courage
- ☐ Honesty

The five Scrum values are Commitment, Focus, Openness, Respect and Courage.

Honestly is not one of them, but that's not to say you shouldn't be honest. Always be honest!

The other traits, extraversion (also often spelled extroversion), agreeableness, openness, emotional stability (neuroticism) and conscientiousness are together known as the Big 5 personality traits. Big 5 personality traits are worth looking into if you're into psychology and human behavior, but you won't be tested on them in the Scrum Master exam.

A Word on Iterative and Incremental Development

How does the incremental and iterative nature of Scrum optimize predictability and control risk?

The incremental and iterative nature of Scrum, along with it's short-sprints and empirical nature, helps optimize predictability and control risk in several ways:

Regular inspection and adaptation: Scrum provides regular opportunities for the Scrum team to inspect the work that has been done and to adapt their plan for the next iteration. This allows the team to make necessary adjustments to their process, product, and priorities to optimize their ability to deliver value and minimize risks.

Short Sprints: Scrum Sprints typically last 1-4 weeks, during which the team works to deliver a potentially releasable increment of the product. The short time-boxed nature of Sprints helps to control risk by reducing the amount of work in progress, thereby limiting the amount of unfinished work that could create risks and uncertainties.

Incremental development: Scrum emphasizes delivering the product in small increments, which helps to control risk by allowing the team to identify and address issues early on in the development process. This also helps to optimize predictability by allowing stakeholders to see working increments of the product more frequently, which can help reduce uncertainty and improve predictability.

Empirical process control: Scrum is based on the three pillars of transparency, inspection, and adaptation, which provide a framework for empirical process control. The team regularly inspects the work done and adjusts the plan for the next iteration, based on the feedback received. This empirical approach allows the team to optimize predictability and control risk by making data-driven decisions that are informed by their experience.

Overall, the incremental and iterative nature of Scrum helps to optimize predictability and control risk by providing opportunities for regular inspection and adaptation, limiting the amount of unfinished work, delivering the product in small increments, and providing an empirical process control framework.

Chapter 4

Scrum Teams

Two of the thirteen catogories in which Scrum Certification exam questions are grouped into include:

- Scrum Teams
- Self-Managing Teams

Pay close attention to how Scrum Teams are structured, the role self-management and self-organization has, and avoid clouding the simple approach Scrum takes to team dynamics with what you may have learned in environments where Scrum may not have been applied properly.

Composition of a Scrum Team

> The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers.
>
> — 2020 Scrum Guide page 5

This is another politically charged couple of lines that need unpacking.

Take note of the fact that a Scrum team is made up of three things: one Scrum Master, one Product Owner and multiple Developers.

Notice that I called them 'things.' People like to call them 'roles.' The Scrum Guide never uses the term 'role' once. Technically, these are not roles. The Scrum Guide later refers to them as 'accountabilities.' But the are tehnically not 'roles.'

Test Yourself

Here's a standard Scrum Certification Exam question to test your knowledge on this basic fact:

> What is the typical composition of a Scrum Team?
>
> ☐ A Scrum team is made up of one Scrum Master, many Product Owners and many Developers
>
> ☐ A Scrum team is made up of one Scrum Master, one Product Owner and many Developers
>
> ☐ A Scrum team is made up of many Scrum Masters, many Project Mangagers and many Developers
>
> ☐ A Scrum team is made up of one Scrum Master, one Project Mangager, one Product Owner and many Developers

The answer is C.

You will often see the term "Project Manager" appear in certification questions to throw you off. There is no "Project Manager" in Scrum. There is a Product Owner, but there is no such thing as a

Project Manager in Scrum.

Also, notice there are not testers, architects, engineers, plumbers, carpenters, rocket scientists or hairdressers listed in the composition of a Scrum team.

If people with those skills are required in order to build the product you're working on, then they all fall under the term 'developer.' There are no separate 'Quality Assurance (QA)' people or 'Business Analysts' in Scrum. There are no separate testing teams. If your project needs testers, blacksmiths or QA people, then they are considered developers on the team.

Test Yourself

> With this knowledge, answer this typical Scrum Master question:
>
> There are problems and concerns with the products, and some things are not getting done. Who should be doing the work that is problematic?
>
> ☐ The developers should do as much as they can and then pass their work off to QA people.
>
> ☐ The developers should just build the core features and then allow testers take care of non-functional aspects.
>
> ☐ The developers will hand off the work to an engineer on another team to complete the work.
>
> ☐ All work will be completed by the developers on the Scrum Team ensuring the Definition of Done is met.

The content of this questions is irrelevant. It's honestly not a great question. But what it does do a great job of is demonstrating how many Scrum Certification Exam questions will talk about professions like QAs, business analysts or engineers who are not on the team. Anyone doing work as part of the Scrum must be on the Scrum Team. And regardless of what their skills are, they are simply seen as a developer in the eyes of the Scrum Guide.

Options on the Scrum Certification Exam where they start talking about specific professions are usually wrong, because a Scrum Team only has three accountabilities: The one Scrum, the one Product Owner and the developers.

Sub-teams are Radioactive in Scrum

In Scrum, the development team is a collection of 'equals', regardless of their skills or experience.

In Scrum Team, there are no separate 'QA Teams' or 'Testing Teams', nor is there even a team lead.

Everyone pitches in to do the required work, and with te team's cross-functional skills, they can accomplish any specified testing or QA requirements.

> Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused

This is another important point you're likely to be tested on in the Scrum Certification exam.

There are no sub-teams. So there is no QA team to check for quality. There is no testing team to check for bugs. There is no performance team to do load testing.

If bugs need to be fixed, if quality needs to be assured or performance needs to be tested, then the people doing that work are part of the development team.

Test Yourself

Here's the structure of a prototypical Scrum Master Certification exam question that tests you on this point:

> There is an issue. What should you do?
>
> ☐ Talk to the senior developer on the Scrum Team about it
>
> ☐ Talk to someone on the QA team about it
>
> ☐ Talk to someone on the testing team about it
>
> ☐ Coach the developers around strategies of how to fix it

Notice how the first three options are automatically wrong because they talk about a senior developer (hierarchy) or a QA or testing team (subteams). Scrum doesn't have subteams or hierarchies.

Cross-functional Scrum Teams

The importance of the concept of cross-functional teams that can accomplish everything required of them without resorting to third parties our outsourced help can't be emphasized enough for people who wish to pass the Scrum Certification exam.

> Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint.

The cross-functional aspect of a Scrum Team confuses a lot of test takers.

If you bake a cake, you need all the ingredients before you begin, right?

And if you want to build a product, you need a team of developers who have all of the skills needed to build that product.

If your team is going to build, test, secure and load test an application, then your team has to have

developers on it with all of those skills.

Does the product under development require a warp drive made from dilithium crystals? Then your cross-functional team better have someone with the skills to build that, or at the very least, they better be actively learning how to do it.

A Scrum team doesn't outsource their work, or hand over their work to a 'testing team' or a 'quality assurance' team. The team itself is cross-functional and capable of doing whatever is required to produce the product under development. That's what it means to be cross-functional.

Test Yourself

> The next feature required by the Product Owner is a Mars orbiter, but nobody on your team knows how to build a Mars orbiter. What should you, the Scrum Master, advise the team to do?
>
> ☐ Tell the dev team the Mars Orbiter will be removed from the required features list.
> ☐ Cancel the Sprint until the team finds someone who can build a Mars orbiter.
> ☐ Tell the dev team the Mars Orbiter feature will be outsoursed to Elon Musk and SpaceX.
> ☐ Coach the dev team on the importance of learning and acquiring the skills required to complete all of the required product features.

This may sound like a silly question, but what would happen if you were on a team that really needed to build a Mars Orbiter?

You'd have to research how to do it and figure it out. And if you were hired to build a Mars Orbiter, the assumptioun is that you are indeed someone smart enough to figure out how to solve that impossible problem.

What Does Cross-Functional Mean?

Here's the official, Scrum Guide definition of what it means to be a cross-functional team:

> In Scrum, the develepers are assumed to be cross-functional, and as a group, they will possess all of the skills needed to produce the product being built. That's what it means to have a cross-functional team.
>
> Scrum Teams are also self-managing, meaning they internally decide who does what, when, and how.
>
> — 2020 Scrum Guide page 5

Scrum self-describes as an 'incomplete framework.' That means it doesn't have a lot of rules or requirements.

Scrum practitioners are encouraged to 'find their own path' and do what works for them. The developers should be allowed to manage themselves as much as possible.

Test Yourself

> Your company just hired 50 new developers. How should they be put into teams?
>
> ☐ Have the Project Manager split them up to create balanced teams based on education, experience and skill level.
>
> ☐ Have the Scrum Master split them up into evenly divided teams where their skills match the their problem domain.
>
> ☐ Have Human Resources split them up into equally divided teams based on their personality types.
>
> ☐ Let the developers split themselves up into any number of groups in any way they like.

In this case, option D would be the correct answer. If the teams are truly self managed, they should be allowed to group together whichever way they see fit. Scrum does limit the size of a team to 10 individuals, so that is the only hard constraint that would be put on the grouping.

Test Yourself

> You, the Scrum Master, come to work on the first day of the new Sprint, and you find out from the Product Owner that the 8-member development team has split themselves up into one group of 3, and another group of 5. The Product Owner is worried about how this will impact the project. What would you, as the Scrum Master, do?
>
> ☐ Coach the development team on the importance of staying together as a single, cohesive team.
>
> ☐ Explain that groups can't be broken up midway through the development of a product.
>
> ☐ Have Human Resources talk to the developers about making rash, surprise decisions
>
> ☐ Allow the team to split up in whatever way they see fit, and coach the Product Owner on the importance of self-managed teams.

In this case, the development team wants to split up into two groups. If that's what they want, then let them do it. The development team is a group of self-motivated, highly motivated individuals with one overarching objective - to achieve the product goal. If the developers believe two teams would be more productive, then that's their decision. They get to self-organize in any way they see fit.

Test Yourself

Here's another self-management question that often catches test-takers off guard.

> Who has the right to remove a developer from a Scrum Team?

- ☐ Human Resources
- ☐ The Product Owner
- ☐ The Scrum Master
- ☐ The Development Team

Again, the development team is self-managing. If there is a developer that they believe is best removed from the team, then the developer should be removed from the team.

That doesn't mean the developer should be fired. Maybe they were under-utilized. Or maybe they somehow impeded progress. But if the development team believe that removing an individual from the team would better serve the Product Goal, then that's a decision the team should be allowed to make.

Why only 10 to a Scrum team?

The maximum size of a Scrum Team, including the Scrum Master and Product Owner, is 10. Here's why.

> The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive.
>
> — 2020 Scrum Guide page 5

You'll definitely be tested on the maximum size of a Scrum Team: it's 10 or fewer. That includes the Product Owner and the Scrum Master, so generally speaking, that implies 8 developers.

However, there are many instances where a Scrum Master or even a Product Owner might be doing development too. Which means there might be 10 developers, with two of them playing double-duty as a Scrum Master and Product Owner.

Can Developers also be Scrum Masters?

People often wonder if a Scrum Master can also be a developer, or if a Scrum Master can also be the Product Owner. There's nothing in the Scrum Guide that forbids it, so it's fair dinkum.

Is it ideal? Probably not.

Is it pragmatic?

If the Scrum Master gets sick, or a Product Owner has amazing skills with regards to a Product Backlog Item or two, why not allow them to contribute to the Increment? Again, there's nothing in the Scrum Guide that forbids it.

It's nice to think of the Scrum Master or Product Owner as a dedicated role. But on small startups, where there may only be five people in the company, you may have all of them doing some sort of development during a Sprint. In that case, the person who took on the Scrum Master accountability

also takes on accountability as a developer.

There's absolutely nothing in the Scrum Guide that says a Scrum Master can't also do development, and pragmatically speaking, for many small companies, it's a reality. But regardless of how many people are wearing multiple hats, the total number of people on a Scrum Team should be 10 or less.

Test Yourself

What is the optimal size of a Scrum team?

☐ 3-9 people

☐ 7 plus or minus 2

☐ 10 people or less

☐ Scrum lets self-managed teams determine their own size

Normally Scrum allows self-managed teams to decide what is best for them. But when it comes to team size, Scrum recommends they are made up of 10 people or less.

I believe SAFe uses 7 plus or minus 2, and the old Scrum Guide said 3-9, but this isn't a SAFe certification guide, and you're not being tested on what the old guide said.

10 people or less is the correct size of a Scrum Team on the 2020 Scrum Certification exam.

Test Yourself

The Scrum team is kept small because:

☐ Small teams are easier to reloace

☐ Small teams communicate better

☐ Small teams are more productive

☐ Small teams demand fewer benefits

According to the Scrum Guide, small teams communicate better and are more productive.

One Product with Multiple Scrum Teams

Ever wonder how Scrum manages a big project with multiple development teams working on it?

The Scrum Guide doesn't say much about the issue, but it does outline a few important rules to follow when more than one Scrum Team works on the development of the same product.

> If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

You can expect 5 or 6 questions on this one paragraph on the Scrum Certification Exam. It's significant.

The first takeway from this paragraph is that you can have multiple Scrum teams working on the same product. In fact, Scrum advises you to break up your teams if they become too big. And that doesn't even mean if they are over 10 people.

Maybe 9 people on the Scrum Team is too large to be nimble and productive? In that case, if the self-managed team thinks splitting into two groups of 4 and 5 developers makes sense, then they should be encouraged to do it.

Also notice that when multiple teams work on the same product, they share the same Product Goal, Product Backlog and Product Owner. Commit those facts to memory, because you are guaranteed to see a question like this on the exam:

Test Yourself

> Five Scrum Teams are working on the same product. Which of the following statements are not true?
>
> ☐ Each team will share the same Product Owner
>
> ☐ Each team will share the same Scrum Master
>
> ☐ Each team will share the same Product Goal
>
> ☐ Each team will share the same Product Backlog

Options A, C and D are correct.

When multiple teams work on the same product, they share the same Product Owner, Product Goal and the same Product Backlog.

To Share or Not to Share a Scrum Master?

Notice the same rule about multiple teams sharing the same Product Owner do not apply to the Scrum Master when multiple teams work on the same product.

Multiple teams on the same project might share the same Scrum Master, or each team might have their own Scrum Master. The Scrum Guide doesn't care either way.

Test Yourself

Also, given the fact that you know one Product Owner may need to be shared across five or even ten teams, how would you answer these two questions:

> • True or False: A Product Owner must dedicated 100% of their time to a single Scrum

> - Team.
> - True or False: A Scrum Master must dedicate 100% of their time to a single Scrum Team.

Both of those statements are false.

We know the first statement is false, because the Scrum Guide insists that if there are five teams working on a single product, the Product Owner must be split across all five teams, which makes it impossible to be dedicates 100% to just one.

Similarly, a there is no rule that says a Scrum Master must be 100% dedicated to a single Scrum team. It's quite common for one Scrum Master to work on multiple Scrum teams within an organization.

Scrum Team Responsibilities

You will often see questions on the Scrum Certification exam about who should be responsible for maintenance or research or user acceptance testing (UAT) or quality assurance (QA).

An 'incorrect answer' will suggest a sub-team or external resource should to UAT or QA. That's wrong.

If QA or UAT or maintenance is a Product Backlog Item or part of the Definition of Done, then it's the Scrum Team this is responsible for it.

> The Scrum Team is responsible for all product-related activities including:
>
> - stakeholder collaboration,
> - verification,
> - maintenance,
> - operation,
> - experimentation,
> - research and development,
> - and anything else that might be required.
>
> — 2020 Scrum Guide page 5

This statement reinforces the idea that a Scrum team is cross functional. Every skill required to build your product, at least within the the scope of your Scrum project, must exist on your team, or at least, your team should be able to acquire those skills. There are no separate research or development or quality assurance teams. The Scrum team is responsible for all of the work on its plate.

Test Yourself

Take this question for example:

In Scrum, who is responsbile for for quality control and testing before an Increment is released into production?

☐ The Quality Assurance Team

☐ And external user acceptance testing (UAT) team

☐ The Product owner

☐ The Development team

There are no external teams when it comes to Scrum and the product your team is building. All of the skills required to incrementally build the product the Scrum Team is working towards must exist amongst the Scrum Team's developers.

Sustainable, Self-Managed Teams

> They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

— 2020 Scrum Guide page 5

This section is interesting because it speaks to the responsibilities an organization has to the Scrum team:

The organization allows the Scrum Team to manage their own work The Scrum team is allowed to work at a sustainable pace

These points will arise on the Scrum Master Certification exam in questions like these:

Test Yourself

How does the organization help to manage the work of the development team?

☐ The organization reads daily reports created by the Scrum Master

☐ The organization regulary sends a manager to attend the Daily Scrum

☐ The organization pull developers into a weekly meeting to hear status reports

☐ The organization allows Scrum teams to manage themselves

The last answer is correct. The best thing an organziation can do to help manage a Scrum Team is to not manage a Scrum Team.

By the way, the other three options are all Scrum anti-patterns. Daily reports, status meetings and sending managers to the Daily Scrum are things that should not happen in the Daily Scrum.

Test Yourself

When should the Scrum Team add a new developer to the project?

☐ When the Project Manager provides enough funding for a new developer

☐ When the development team repeatedly fails to meet the Sprint Goal

☐ When the development team fails to estimate the amount of work they can complete in a Sprint

☐ When a new developer is required to maintain a sustainable pace of development

Motivated developers will work hard to achieve the Sprint Goal and the Product Goal. But they should not be overworked.

If developers are getting overworked and are no longer working at a sustainable pace, it is the organization's responsibility to hire new developers to help the team.

Increments Must Be Valuable and Useful

Every Sprint the Scrum Team must create an increment that is valuable and useful.

That means you can't have an 'infrastructure sprint' that just gets things organized.

Even if the increment is small and minor, each Sprint should try to create something that is valuable and will be used in the final product.

> The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint.
>
> — 2020 Scrum Guide page 5

There's a couple of important points nested in this sentence.

First, every Sprint must create at least one valuable, useful Increment. Notice that an increment must be something 'useful.' The means simply setting up infrastructure doesn't count as a 'useful' Increment. The Increment must be useful in terms of the usability of the end product.

Secondly, notice how the Guide says the entire team is responsible for the creation of at least one, valuable, useful increment each Sprint - not the Scrum Master, not the Product Owner and not the developers alone, but the whole team is responsible for the creation of a useful increment.

Test Yourself

You're guaranteed to get a question like this on the exam:

Who is accountable for the creation of a useful Increment eashc Sprint?

☐ The stakeholders

In this case, it's the last option: the Scrum Team as a whole.

Scrum Accountabilities

Scrum plays a few word games with the 'accountabilities' it defines.

Notice that the Scrum Guide never mentions the term 'role' or 'job.' When it comes to the Developers, Product Owner and the Scrum Master, it is very careful to use the much more abstract term 'accountability' as opposed to 'job' or 'role.'

The term 'role' is not actually found in the 2020 Scrum Guide even once.

> Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.
>
> — 2020 Scrum Guide page 5

This sentence reiterates the opening paragraph of the section of the Scrum Team. Again, note that Scrum describes the developers, Scrum Master and Product Owner as accountabilities, not roles. The term 'role' is never mentioned in the Scrum Guide.

Test Yourself

The Scrum Guide defines:

☐ 3 Roles

☐ 4 Roles

☐ 3 Accountabilities

☐ 4 Accountabilities

The answer to this question is C. Scrum defines 3 accountabilities, not roles.

Chapter 5

The Developers

Scrum has its roots in software development, but the stewards of the framework have worked hard to push Scrum into other domains including:

- Construction
- Manufacturing
- Engineering
- Bee Keeping

To pass the Scrum certification exam, you must shake yourself out of the mindset that Scrum is all about software development.

A big mental transition many software developers have difficulty with is realizing that in terms of Scrum, a 'developer' isn't necessarily a 'software developer.'

> Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

— 2020 Scrum Guide page 5

Who is a Developer?

According to Scrum, the term 'developer' refers to anyone on the team who is adding value and working towards the creation of a usable and valuable increment of work.

- In construction, a developer could be the person painting the walls
- For castaways on a desert island, a developer could be the person digging a trench for irrigation
- In bee keeping a developer might be the person who disposes of the slumgum

As you read this section about the 'developer', it is easy to fall into the trap of thinking that Scrum is specifically talking about a 'software developer.'

They're not. And quite often, framing a scenario or exam question outside of the world of software development will help you choose the correct option .

Aspects vs Accountability

The Scrum Certification exam will likely ask you one of these two questions:

- Who is accountable for creating a useful Increment?
- Who is committed to creating aspects of a usable Increment?

The two questions sound very similar, and if you're unprepared, the nuance between the job of the Developers and the duties of the team might lose you a mark.

The Scrum Team as a whole is accountable for creating a useful Increment, not just the Developers.

However, it is the developers who create the various **aspects** of the usable Increment.

Compare the quote above to the one below and notice the difference the Scrum Guide makes between the Scrum Team and the Developers when it comes to the Increment:

> The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint.
>
> — 2020 Scrum Guide - Scrum Team Section

> Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.
>
> — 2020 Scrum Guide - Developer Section

Complicated Brevity

The whole point of Scrum is to deliver value to the stakeholders, and it is the developers who create all aspects of the Increment, which is the additive unit of value in Scrum.

Yet despite the importance of the Developer to the framework, the Scrum Guide only dedicates 84 words to them.

The remaining verbiage about the Developers in Scrum simply outlines the tasks they are accountable for:

> The specific skills needed by the Developers are often broad and will vary with the domain of work.
>
> However, the Developers are always accountable for:
>
> - Creating a plan for the Sprint, the Sprint Backlog;
> - Instilling quality by adhering to a Definition of Done;
> - Adapting their plan each day toward the Sprint Goal; and,
> - Holding each other accountable as professionals.
>
> — 2020 Scrum Guide - Developer Section

Developer Accountabilities

Let's take a look at the accountabilities of the Developer one point at a time.

> Developers are always accountable for creating a plan for the Sprint: the Sprint Backlog.
>
> — 2020 Scrum Guide - Developer Section

There are two backlogs in Scrum:

- There's the Product Backlog, which is owned by the Product Owner, and it outlines all of the work necessary to build the product
- and there's the Sprint backlog, which is owned by the developers.

The Sprint Backlog is the developer's plan of what then intend to get done during the Sprint.

The developers own it, and nobody tells them how to manage it, not even the Scrum Master.

> Who on the Scrum Team has the right to add new items to the Sprint Backlog?
>
> ☐ The Scrum Master
>
> ☐ The Product Owner
>
> ☐ The Stakeholders
>
> ☐ The Developers

The answer is D.

Only the Developers have the authority to add or remove items from the Sprint Backlog.

In contrast, only the Product Owner can add or remove items from the **Product** Backlog.

Instilling Quality

> Developers are always accountable for Instilling quality by adhering to a Definition of Done.
>
> — 2020 Scrum Guide page 5

Notice the connection between quality and the Definition of Done.

Any time the Scrum Master certification asks about how to enforce quality, the answer will usually get mapped to the application of the Definition of Done.

> The gatekeeper of quality for the Scrum Framework is:
>
> ☐ The Scrum Master
>
> ☐ The Product Owner
>
> ☐ The Product Goal
>
> ☐ The Definition of Done

The answer is D. The Definition of Done is the quality gate all increments must pass through to become valid.

Daily Adaptation

> Developers are always accountable for adapting their plan each day toward the Sprint Goal.
>
> — 2020 Scrum Guide page 5

It's assumed that things will change and plans will go awry during a Sprint.

Developers are expected to constantly adapt their plan to adjust to changes. And since the developer's plan is the Sprint Backlog, that means the developers will be tinkering with, adding, deleting and managing their Sprint Backlog constantly throughout a Sprint.

When are developers allowed to adapt their plan by changing the Sprint Backlog

☐ Any time during the sprint

☐ During Sprint Planning

☐ During the Sprint Retrospective

☐ During the Daily Scrum

Option A is correct.

If the developers see an opportunity to adapt to new changes or information, they are encouraged to update their plans immediately. In Scrum, the Sprint Backlog is the developer's plan. The developers can update the Sprint Backlog anytime during the Sprint if they feel it is necessary.

Interpersonal Accountability

> Developers are always accountable for holding each other accountable as professionals.
>
> — 2020 Scrum Guide page 5

You are guarnateed to get a question on the Scrum Certificaiton exam about who holds the developers accountable for monitoring or tracking their progress.

The answer is the developers. The developers hold themselves accountable.

Who holds the development team accountable for their work?

☐ The Scrum Master

☐ The Product Owner

☐ The Management Team

☐ The Development Team as a whole

The answer is D. The development team holds themselves accountable for their progress.

This four item list actually ends the 'Developers' section in the Scrum Guide. This covers the basic responsibility as far as the Scrum Guide goes.

What's not said about Developers

About 90% of the question on the Scrum Certification exam come directly out of the Scrum Guide. However, there are a few questions that require some additional knowledge.

Two topics you'll want to familiarize yourself with in regards to Developers is:

- The fact that adding developers sometimes slows down the team's velocity
- New developers can be added any time you need them.

Adding New Developers

When should you add new developers to a Team?

- During Sprint Planning?
- During the Sprint Review?
- During the Sprint Retrospective?

The Scrum Guide doesn't say anything difinitive about this subject, but it also doesn't need to.

You can add new developers to a project any time you need them.

If you're half way through a Sprint, and the HR team has just hired four new programmers that you've wanted for months, it wouldn't make sense to just keep those developers on the bench for two weeks so they can start at the beginning of the next Sprint. Start them right away!

So you can add new developers to the team any time they become available.

Sustainable Development

And when should you add developers? Scrum says developers should always work at a sustainable pace. So any time the pace starts to become unsustainable, add some new developers to the team.

The Scrum Certification exam might talk about budget money or internships or something like that, but those factors don't have anything to do with Scrum.

In terms of Scrum, developers can be added to a project any time they are needed. They can be removed at any time too.

The Impact of Adding Developers

One reality of adding new developers to a team is that it tends to slow the rest of the team down.

When new developers are onboarded, they usually take other developers away from their work, as they are given help setting up printers, finding out where the washrooms are, learning about the project and figuring out how to connect to GitHub and kick of a continuous integration build.

That's just the reality of adding new people to the team. For the short term, the productivity of individual members of the team will dip, although over time, it will go back to normal.

Just be aware of that fact on the Scrum Certification exam.

Chapter 6

The Product Owner

It all starts with the Product Owner (PO).

Seriously, the whole idea of Scrum is for a team to work towards the creation of a product that makes stakeholders and customers happy.

- Who knows what the product's supposed to do?
- Who knows what features the product needs?
- Who has the product vision?
- Who knows what features matter the most?
- Who can quickly answer questions about the product?
- Who do the stakeholders trust to make decisions about the product?

The answers to all of those questions is 'The Product Owner.'

Without a Product to build, there's no point in putting together a Scrum Team.

Product Owners Maximize Value

The Product Owner knows everything there is to know about the product the team's trying to build.

> The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.
>
> — 2020 Scrum Guide page 5

The Product Owner knows all about the Product. But what is the Product Owner most accountable for?

The Scrum Guide makes the Product Owner's job clear: to maximuze the value of the product being created.

Interestingly, the Scrum Guide doesn't make it clear as to how the Product Owner goes about doing that. Fortunately, I can help you out here.

The PO maximizes the value of the work done by the Scrum Team by making sure the developers are working on the most important and most valuable features of the product.

- If the PO gets the developers to work on a bunch of features nobody is going to use, the PO is not maximizing value.
- If the PO gets the developers to create features the clients are dying to get their hands on, they the PO is maximizing value.

That's typically how the PO maximizes the value of the stuff the developers are creating.

Product Backlog Management

What we are essentailly describing here with regards to prioritizing the most important features is one of the four key elements of Product Backlog management. The Scrum Guide describes Product Backlog management in more detail here:

> The Product Owner is also accountable for effective Product Backlog management, which includes:
>
> • Developing and explicitly communicating the Product Goal;
>
> • Creating and clearly communicating Product Backlog items;
>
> • Ordering Product Backlog items; and,
>
> • Ensuring that the Product Backlog is transparent, visible and understood.
>
> — 2020 Scrum Guide page 6

Take these four points seriously, because Product Backlog Management is one of the 13 categories in which the questions on the Scrum Certification exam are categorized.

The exam will try to catch you off guard with false statements that seem to make sense.

For example, the exam will assert in a True or False question that the stakeholders or customers are responsible for communicating the Product Goal. On the surface that almost sounds like it makes sense.

After all, the product is for the customers, right? So shouldn't they communicate the Product Goal? Don't fall for the tricker. The Product Owner communicates the Product Goal.

The same tricker goes for communicating Product Backlog items (PBIs). It sounds logical that a customer or stakeholder might be the one creating or communicating Product Backlog items, after all, the Product Backlog items represents features and functionality that the customers want, right?

Again, don't fall for it. It's the Product Owner's job to create and clearly communicate Product Backlog items.

When it comes to the Product Backlog and the Product Goal, the responsibility falls on the shoulder of the Product Owner, nobody else.

Test Yourself

Who is responsible for managing the Product Backlog and maximizing the value of of the product produced?

☐ The Scrum Master

☐ The Product Owner

☐ The Management Team

☐ The Development Team as a whole

As we just learned, the Product Owner manages the Product backlog, and it's also the Product

Owner's job to 'maximize the value of the product resulting from the work of the Scrum Team.'

Test Yourself

Which of the following statements are true?

- ☐ The Scrum Master can add items to both the Sprint Backlog and the Product Backlog
- ☐ The Product Owner can add items to both the Sprint Backlog and the Product Backlog
- ☐ The Product Owner is fully accountable for all items added to the Product Backlog
- ☐ The Scrum Master is fully accountable for all items added to the Sprint Backlog

Only Option C is correct.

The developers have full control over the **Sprint** Backlog and they are the only ones who can add to it.

The Product Owner has full control over the **Product** Backlog and only the PO can add items to it.

Notice in the question above the term 'accountable' was used.

The Product Owner can actually delegate any of the work they are accountable to anyone they deem fit. However, the Product Owner remains fully accountable for the results, regardless of who does the work.

Who Does the Work?

In Scrum, the developers do the developing.

In Scrum, the Scrum Master does the Scrum Mastering.

But according to the Scrum Guide, the Product Owner can get someone else to do the 'Product Owner gruntwork' for them, just so long as they remain responsible and accountable for the final results.

> The Product Owner may do the Product Backlog management work, or they may delegate the responsibility to others.
>
> Regardless, the Product Owner remains accountable.
>
> — 2020 Scrum Guide page 6

Product Ownership and Trust

Perhaps the most important quality of the Product Owner is that they have the complete and total trust of the organization to make decisions about what's best for the Product being built.

The importance of PO trust can't be understanted.

- Scrum teams move quickly
- Conditions change from day to day
- Adaptation should happen daily
- Backlog item clarity is of utmost importance

When the developers have questions about the work they are doing, they need answers that are clear and difinitive.

The Product Owner must be able to respond quickly, and the development team needs to know that the decisions of the Product Owner will not be second guessed by management.

If someone else in the organization keeps overriding the decisions of the PO, the team will quickly lose trust in the Product Owner, and the whole Scrum Framework falls apart.

Test Yourself

Important decisions that pertaining to the future direction of the product being built must be made by:

☐ The Scrum Master

☐ The Product Owner

☐ The Product Owner along with a stakeholder committee

☐ The Scrum Team as a whole

The answer to this question is the Product Owner.

In fact, the Scrum Guide states quite emphatically that the product based decisions are not to be made by a committe.

A Single Product Owner

The job of the product owner cannot be shared amongst multiple individuals. There can't be two co-Product Owners on a team.

As the Scrum Guide stated earlier, the Product Owner can delegate some of Product Backlog management work to a cohort, but in the end, all of the accountabilities associated with being a Product Owner will fall on the shoulders of only one person.

> The Product Owner is one person, not a committee.

> The Product Owner may represent the needs of many stakeholders in the Product Backlog.

> — 2020 Scrum Guide page 6

Notice how the Scrum Guide states that the Product Owner may represent the needs of many stakeholders. Implied in that statement is that different stakeholders might have different interests with regards to which features or enhancements should be prioritized.

When it comes to competing interests, the Product Owner is responsible for managing the expectations of the stakeholders.

Test Yourself

There are many stakeholders groups with many competing interests, including the priority of features, the cost of the project, the release date and community outreach. How should Scrum ensure the interests of each of these groups is recognized and respected?

- ☐ Have the Scrum Master represent the interests of each of these groups
- ☐ Have multiple Scrum Masters, with each Product Owner assigned to each stakeholder group
- ☐ Have one Product Owner represent the interestes of each of these groups
- ☐ Have multiple Product Owners, with each Product Owner assigned to each stakeholder group

There is only one Product Owner on a Scrum Team, and that one Product Owner represents the interests of all of the stakeholders.

Keeping Tabs on the Product Owner

Stakeholders will always be interested in how the product is progressing. They will constantly want to know what's been created, and what the team will be working on next.

The Product Backlog and inspectable Increments are how stakeholders get the answers to these question.

> These decisions are visible in two ways:
>
> 1. through the content and ordering of the Product Backlog
> 2. through the inspectable Increment at the Sprint Review
>
> Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

> — 2020 Scrum Guide page 6

Trust, Transparency and the Prouduct Backlog

The PO will have the complete and total trust of the organization with regards to product

ownership. But trust is a two-way street.

If the Product Owner is to be trusted, the Product Owner must also be transparent about what they're doing.

So how does the Product Owner make their decisions transparent?

They do so by making the product backlog visible and available to all stakeholders.

- Want to know what the Product Owner is building? Look at the Product Backlog.
- Want to know what the Product Owner has prioritized? Look at the Product Backlog.
- Want to know what the Product Owner wants to build next? Look at the Product Backlog.
- Want to know how features are described? Look at the Product Backlog.
- Want to know the vision and goal for the product? Look at the Product Goal which is part of the Product Backlog.

Inspection of the Increment

Furthermore, at the end of every Sprint, a Sprint Review takes place where stakeholders inspect what's been done. That's known as the Increment, which represents an valuable and usable piece of the product puzzle.

If a stakeholder wants to know what's being done, they look at the Product Backlog.

If a stakeholder wants to know what's been done, they attend the Sprint Review and inspect the increment of work that's been produced in the Sprint.

The Product Owner's commitment to visibility and transparency with regards to these things is what makes it all work.

Negotiating Product Features

Not everyone will agree with what should be built next, what features should be prioritized, or how product development should be managed.

- Stakeholders might disagree.
- The Scrum Master might disagree.
- The Developers might disagree.

Disagreement is expected, especially in a fast moving environment where things change quickly.

If anyone wants to change the Product Backlog, update the Product Backlog, delete something from the Product Backlog or add something to the Product Backlog, they go through the Product Owner.

When it comes to the Product, and the Product Backlog that describes what's being built, the Product Owner has full control.

> How does ensure their decisions are transparent and open?

☐ By making their decisions visible in the Product Backlog

☐ By allowing stakeholders to see how the Product Backlog has been ordered and prioritized

☐ By sending regular status updates to stakeholders.

☐ By scheduling weekly meetings between the Scrum Developers and Stakeholders

☐ By having the stakeholders inspect a usable Increment of work at the Sprint Review

In this case, options A, B and E are correct.

Remember that in Scrum we always shun 'more meetings.' The whole point of the various Scrum events, namely the Sprint Review, Sprint Retrospective and the Daily Scrum is to remove the need to schedule other, time-wasting meetings.

Test Yourself

The CEO has told you, the Scrum Master, that if a key feature isn't added to the product sometime within the three months, the project will be cancelled. What action should the Scrum Master take?

☐ Add the feature as a Sprint Backlog item so developers can start working on it immediately

☐ Add a new item to the Product Backlog to represent the feature

☐ Cancel the Sprint and have the developers shift their focus onto this new feature

☐ Inform the Product Owner and facilitate a conversation between the Product Owner and the CEO

If a change needs to be made to the backlog, it's the Product Owner who does it. If the CEO needs a feature prioritized, the CEO has to go through the Product Owner. Nobody else has the right to do it.

Chapter 7

The Scrum Master

People new to Scrum often assume that the Scrum Master has a great deal of power over other members of the Scrum Team.

They don't.

Scrum Masters don't have much power at all.

Most of what a Scrum Master does revolves around coaching and educating people about the Scrum framework.

- The Scrum Master doesn't make decisions about the Product
- The Scrum Master doesn't tell the developers how to manage their work
- The Scrum Master can't make changes to the Product Backlog
- The Scrum Master can't make changes to the work planned for the Sprint
- The Scrum Master can't hire or fire people

Even when the Scrum Master does do something of substance, such as helping to remove impediments or acting as a servant-leader, they typically achieve this by educating others about how Scrum is supposed to work, and coaching people about how to properly embrace the ideas expressed in the Scrum Guide.

Scrum Requires a Scrum Master

> The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide.
>
> They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.
>
> The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.
>
> — 2020 Scrum Guide page 6

Test Yourself

Given how the Scrum Guide describes the Scrum Master, how would you answer the following question?

> The developers say they don't feel ready to start development and would like to do an infrastructure Sprint to get everything set up first. However, the Product Owner wants to get the first Sprint, and the first increment, under the team's belt. What should the Scrum Master do?
>
> ☐ Cancel the first Sprint.
>
> ☐ Tell the developers there will not be an 'infrastructure Sprint.'

> ☐ Tell the Product Owner there will be no Increment on the first Sprint.
>
> ☐ Help the Scrum Developers understand the importance Scrum places on creating an increment in every Sprint.

Option D is correct.

The job of the Scrum Master is largely to coach, facilitate and help both the Scrum Team and the organization understand how Scrum works.

A Scrum Master does not:

- Manage the project
- Assign tasks to members of the team
- Tell devleopers what to do
- Cancel, delay or extend Sprints
- Assign developers into teams
- Manage budget money

Scrum Masters and Servant Leadership

Previous editions of the Scrum Guide talked about servent leadership. Yet the 2020 Scrum Guide removes all references to the term 'servant leader.'

Instead, the 2020 Scrum Guide includes this line that spins the idea of servant leadership around a bit:

> Scrum Masters are true leaders who serve the Scrum Team and the larger organization.
>
> — 2020 Scrum Guide page 6

The Scrum Guide always tries to be very precise in its working.

The stewards of the Scrum framework want everyone to know that a Scrum Master is a leader first, and then a servant to the team. They are a leader who serves, not a servant who leads.

It's a persnikity distinction, but an important one nevertheless.

Test Yourself

> Which of the following statements is most true about a Scrum Master?
>
> ☐ The Scrum Master manages their team with a command and control approach to leadership
>
> ☐ The Scrum Master is a leader first, and a servant to their Scrum Team second
>
> ☐ The Scrum Master is a servant-leader on the Scrum Team
>
> ☐ The Scrum Master is a leader and not a servant

Option B is correct.

The Scrum Master is a leader first who also serves the other member of their team.

How Scrum Masters Serve the Scrum Team

> The Scrum Master serves the Scrum Team in several ways, including:
>
> • Coaching the team members in self-management and cross-functionality;
>
> • Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
>
> • Causing the removal of impediments to the Scrum Team's progress; and,
>
> • Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
>
> — 2020 Scrum Guide page 6

The first two bullet points emphasize the point made by the Scrum Guide earlier that the Scrum Master helps the team by coaching them on how Scrum work, and how Scrum Teams should behaves within the scope of the Scrum framework.

Notice the use of terms such as 'coaching' and 'helping.'

On the Scrum Certification exam, if you ever see an option that implies a Scrum Master will force somebody to do something, or tell somebody to do something, that option will typically be wrong.

A Scrum Master is a coach, a diplomat, a leader who serves and a facilitator. They are not managers or commanders who coerce others into productivity through intimidation or Macheavellian tactics.

Test Yourself

Blah, blah, blah Developer problem. Blah, blah, blah what should the Scrum Master do?

☐ Tell the Developer to blah, blah, blah...

☐ Coach the Developer on blah, blah, blah...

☐ Instruct the Developer to blah, blah, blah...

☐ Facilitate the Developer with blah, blah, blah...

☐ Force the Developer blah, blah, blah...

Options B and D are correct.

There will be questions on the Scrum Certification exam where some of the options include the Scrum Master telling, instructing, insisting or demanding a Developer does something. Answers like that will always be wrong, and can be disqualified immediately.

A Scrum Master is a coach, facilitator and a leader who serves. They are not managers or army generals.

Removal of Impediments

The Scrum Guide says the Scrum Master should work to cause the removal of impediments to the team's progress.

> The Scrum Master serves the Scrum Team by causing the removal of impediments to the Scrum Team's progress
>
> — 2020 Scrum Guide page 6

So what's an impediment?

It's important to keep in mind that we are talking about impediments within the context of the Scrum framework.

For example, if a manager starts asking for daily status reports from the developers detailing what they're working on, that infrines on the development team's ability to self-manage. The need to needleslly fill out status reports that zap the dev team's productivity is an impediment that the Scrum Master should work to remove.

In this case, the Scrum Master would coach management on the need to allow developers to manage themselves, while at the same time showing the managers how the Sprint Backlog and the Sprint Review are much better ways to gain insight into the progress the development team is making.

Similarly, if a manager is trying to attend the Daily Scrum, or the Product Owner is not clearly communicating the requirements of their Product Backlog items, these are impediments the Scrum Master can work to remove.

In the Context of Scrum

Impediments that fall outside of the context of Scrum are not typically things the Scrum Master would be expected to solve.

For example, if a developer can't figure out how to book a conference room for the Daily Scrum, that's not exactly a blocker for the Scrum Master to solve. Or if the computer hosting the Git repository goes up in flames and the team needs a new computer, that's not a blocker the Scrum Master would be expected to address either.

The Scrum Master works to cause the removal of blockers that slow the team down, but only within the context of the Scrum Framework, which tyipcally involves coaching and educating others about Scrum.

Test Yourself

A developer who has taken on the task of booking a conference room for the daily scrum has found the company's booking system to be an impediment, and they can't manage to book the same room every day of the week. What should the Scrum Master do in this circumstance?

Option C is correct.

The booking system may be a blocker or impediment to the developer, but it's not an impediment in terms of the Scrum framework. It's not the Scrum Master's job to teach a developer how to use their tools.

Perhaps a Scrum Master could advise the developer on how to find training that would teach them how to better use the tool, or facilitate a learning session with other developers where knowledge sharing about the booking system could happen. But actually doing the developer's work for them is not what the Scrum Guide means when it talks about removing impediments.

In this case, the only correct option is for the Scrum Master to coach the developer on the importance of having Scrum Events take always take place at the same time and location, so long as it's reasonable to do so.

The types of impediments a Scrum Master is expected to help solve are typically things that go beyond issues that self-managed and self-organized teams can solve on their own, while at the same time fall within the context of Scrum. Examples include:

- Availability of the Product Owner

- The team being pushed beyond a sustainable pace

- External attempts to micro-manage the team

- Management scheduling needless meetings

- Poorly described backlog items

- Unclear Sprint and Product Goals

All of these issues are impediments for which the Scrum Master should be able to help remove the cause through coaching, facilitation and educating people about the proper application of the Scrum framework.

Positive, Productive and Timeboxed

The final point here emphasizes the Scrum Master's role in making sure all Scrum Events happen throughout the Sprint, that they are productive, and they are time boxed.

> The Scrum Master serves the Scrum Team by ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

When answering questions on the certification exam, remember that according to this point:

- It's the Scrum Master's job ensure that all Scrum Events take place
- It's the Scrum Master's job ensure that all Scrum Events fall within their timebox
- It's the Scrum Master's job ensure all Scrum Events are positive

Positivity, Occurrence and Timeboxing

There will be questions on the exam that state the developers don't want to do the Daily Scrum every day, or they want to hold the Daily Scrum in different locations on a Friday.

In these situations, the Scrum Master much coach the team on the importance of taking part in all Scrum events, and making sure they take place at the same time and in the same location every time.

There will also be questions about whose job it is to keep Scrum Events within their allotted time. In situations like these, it is the Scrum Master's job to coach the team about the importance of keeping within the timebox.

And other questions will say that developers are not enjoying the Daily Scrum, or don't want to take part in the Sprint Review or Retrospective. In these situations, it's the Scrum Master's job to coach the team about the importance of these events and try to facilitate a strategy that will make the events more positive and rewarding for those who attend them.

Test Yourself

The Scrum Developers are frustrated, not clear on what they are developing and do not want to attending the Daily Scrum. What should the Scrum Master do?

- ☐ Cancel the Daily Scrum so the developers can focus on their work.
- ☐ Coach the team on the value the Daily Scrum brings to the team.
- ☐ Reprimand developers who don't attend the Daily Scrum
- ☐ Facilitate the creation of a clearer Sprint goal during the next Sprint Planning event

The Scrum Master can't cancel any Scrum Events.

The Scrum Master must coach the team ensure all Scrum Events take place, are positive and are productive, so option A is incorrect.

The Scrum Master also doesn't have the authority to reprimand anyone, so option C is incorrect.

B and D are correct. The Scrum Master should coach the team on the value of the Daily Scrum, and also help the team create clearer goals so they have more focus on future Sprints.

Scrum Master and the Product Owner

> The Scrum Master serves the Product Owner in several ways, including:
>
> - Helping find techniques for effective Product Goal definition and Product Backlog management;
> - Helping the Scrum Team understand the need for clear and concise Product Backlog items;
> - Helping establish empirical product planning for a complex environment;
>
> — 2020 Scrum Guide page 6

Notice how the term 'helping' is used repeatedly with regards to how the Scrum Maser serves the Product Owner. Furthermore, each item the Scrum Master his helping with is an element of Scrum, such as:

- The Product Goal

- The Product Backlog

- Empiricism

Furthermore, helping the Product Owner with these things works to remove impediments for the team.

For example, poorly defined Product Backlog items make it difficult for the developers to understand what is required of them. So encouraging the Product Owner to create clear Product Backlog items removes and impediment from the team.

Product Owner Impediments

A poorly defined Product Goal means it's unclear as to what Scrum Team's efforts are aiming at. By helping the Product Owner effectively define the Product Goal, another impediment to progress is removed.

A lack of empirical product planning can lead to budgetary problems, timelines getting overshot and development grinding to a halt because of unsatisfied dependencies on other teams.

These are all impediments to progress that can be resolved if the Scrum Master helps the Product Owner to more effectively do what the Scrum Guide requires of them.

The Scrum Master and the Stakeholders

Many people approach Scrum with the assumption that the framework employs some type of command and control structure that rigidly defines hierarchies and sets boundaries around who can talk to each other and when.

Nothing could be further from the truth.

Developers, Scrum Masters, Product Owners, stakeholders and even customers are encouraged to talk to each other. There are absolutely no rules in Scrum about stakeholders only talking to Product Owners, or developers not being allowed to talk to customers.

Scrum promotes the exact opposite type of approach. Scrum values opennes and transparency.

In fact, one of the Scrum Master's main responsibilities is to remove barriers between the stakeholders and the Scrum Team, helping to open up channels of communication and enhance transparency in the process.

Removing Barriers Between Stakeholders and Teams

Imagine a small startup where the office has 8 desks for the following people:

- The two owners of the startup, who are also the stakeholders in the product being built
- One Product Owner
- One Scrum Master
- Four developers

Do you think the stakeholders and the developers should never talk to each other in that situation? Do you think there should be some rule that says the owners should but noise cancelling headsets on any time two developers talk about the project?

Or do you think it would improve transparency if the developers and the stakeholders talked to each other all the time?

In fact, one of the jobs of the Scrum Master is to remove barriers between stakeholders and Scrum Teams. It's always preferable tho have open and healthy lines of communication between everyone on a project.

> The Scrum Master serves the organization by removing barriers between stakeholders and Scrum Teams.
>
> — 2020 Scrum Guide page 6

Some people also get the incorrect impression that developers can only talk to stakeholders during the Sprint Review.

The Sprint Review is a scheduled event that makes sure the stakeholders and the Scrum Team get a chance to talk, but it is not the only time such communication is allowed. Openness and transparency is always better than secrecy and ambiguity.

Planning and Advising

The Scrum Guide doesn't speak much about activities that happen outside of a Sprint. However, the Scrum Guide does say the Scrum Master is expected to help an organization get up and running with Scrum by helping to plan the adoption of Scrum.

> The Scrum Master serves the organization by planning and advising Scrum implementations within the organization
>
> — 2020 Scrum Guide page 6

And along with helping to get Scrum up and running, the Scrum Master is also expected to help the organization as a whole by coaching others about how Scrum works and how Scrum's empirical approach to problem solving can aid in product development.

> The Scrum Master serves the organization in several ways, including:
>
> - Leading, training, and coaching the organization in its Scrum adoption;
> - Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
>
> — 2020 Scrum Guide page 6

The job of the Scrum Master is to teach people how to do Scrum, how to implement Scrum, and how to improve on Scrum based practices.

Test Yourself

> The management wants regular status updates on the progress of the project, so they have scheduled a standup meeting every Friday afternoon.
>
> How should you, as the Scrum Master, handle this situation?
>
> ☐ Tell the development team they need to attend a Friday afternoon standup meeting
>
> ☐ Replace Friday's Daily Scrum with management's standup meeting
>
> ☐ Have the Product Owner speak to the management about the need for the Friday meeting
>
> ☐ Speak personally to the management about the need for the Friday meeting

Option D is incorrect.

In this case, there is an issue between the Stakeholders and the Scrum Team.

Hopefully management can be coached on the transparency built into Scrum, and gain enough value from the Sprint Review to remove the request for additional meetings with the Scrum Teams.

Also, by keeping the Scrum Developers out of an unneeded meeting, the Scrum Master has removed a potential impedemint for the developers.

Chapter 8

Scrum Events

There are four events in Scrum that happen within the confines of a Sprint:

- The Sprint Planning event
- The Daily Scrum
- The Sprint Retrospective
- The Sprint Review

> The Sprint is a container for all other events.
>
> — 2020 Scrum Guide page 7

The fact that the four events mentioned above all happen within a fifth even, known as the Sprint, will likely show up in five or six question on the Scrum Master Certification exam.

The Sprint Container

The Sprint is a container for all other events.

That means the Sprint Review, Spring Planning, the Sprint Retrospective and the Daily Scrum all happen within the Sprint.

- None of the Scrum events happen before the Sprint.
- None of the Scrum events happen after the Sprint
- Nothing in Scrum happens in the moment where one Sprint ends and the next Sprint begins

> A new Sprint starts immediately after the conclusion of the previous Sprint.
>
> All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.
>
> — 2020 Scrum Guide page 7

Everything in Scrum happens inside the confines of the Sprint.

Test Yourself

The team wants to update their Definition of Done. When is the right time to make this change?

- ☐ Before the first Sprint starts.
- ☐ After the current Sprint finshes
- ☐ Before the next Sprint begins
- ☐ During the current Sprint's Sprint Retrospective

Since nothing happens before or after a Sprint, the first three options are incorrect.

The correct answer is D. The Sprint Retrospecive is the right time to update the team's Definition of Done.

> Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt.
>
> — 2020 Scrum Guide page 7

Inspection and adapatation are important concepts in the Scrum Framework. The Scrum events provide the opportunity for that to happen.

Test Yourself

> True or False: Only the Sprint Review provides the opportunity for the team to inspect and adapt.

This is false.

All of the Scrum events provide the opportunity for the team to inspect their work and adapt their plans.

Regularity and consistency

Scrum doesn't place restrictive controls over how development takes place, but it does request that certain patterns be followed, and that those patterns happen with a predictable regularity.

> Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum.
>
> — 2020 Scrum Guide page 7

Many people complain that Scrum has too many meetings. It really doesn't.

- There is one timeboxed meeting to plan what to do during the Spring
- There's a quick, 15 minute Daily Scrum to allow developers touch base
- There is a Sprint Review at the end of the Sprint where the stakeholders are invited
- There's a Sprint Retrospective where the team talks about what they could do better

That doesn't sound like too many meetings to me. It actually sounds like it's exactly the right number of meetings a team should have.

In fact, the meetings that Scrum does have are intended to have a maximum effectiveness on transparency, while at the same time eliminating the need for other meetings.

Teams that complain about Scrum having too many meetings probably aren't doing Scrum

properly.

Test Yourself

> The CTO wants to schedule afternoon status meetings with the dev team to monitor the progress of an important feature. What should the Scrum Master do?
>
> ☐ Allow the afternoon meetings to take place until the feature is complete
>
> ☐ Invite the CTO to participate in the Daily Scrum in place of the afernoon meetings
>
> ☐ Coach the CTO on how Scrum provides transparency through existing Scrum events and artifacts
>
> ☐ Have the Product Owner send reports from the Daily Scrum to the CTO to avoid the extra meetings

When stakeholders attempt to manage the Scrum Developers, it is often because they want more transparency into what is happening in terms of product devleopment. Scrum already provides enough meetings, in the form of events, to allow for transparent inspection of progress. And the Scrum artifacts like the Product Backlog, also provide transparency and opennes. If a stakeholder is concerned about transparency, a Scrum Master should coach the individual on how existing events and artifacts should provide all the transparency they need.

Same Time, Same Place

While it's not always pragmatically possible, Scrum asks that all of the Scrum Events happen at the same time and the same place.

- The Daily Scrum takes place at the same time and location
- The Sprint Retrospective takes place at the same time and location
- The Sprint Review takes place at the same time and location
- Sprint Planning takes place at the same time and location

That makes the Scrum events more predictable, easier to plan around and more likely to have full participation from everyone involved.

> Optimally, all events are held at the same time and place to reduce complexity.
>
> — 2020 Scrum Guide page 7

It should be noted that Scrum isn't completely unreasonable and unforgiving when it comes to a rule like this. The time and place is allowed to be adjusted for pragmatic reasons.

I mean, if the War Room is being fumigated, it's okay to move the Daily Scrum to Conference Room B. Just try to keep the time and place as unchanged as possible.

Test Yourself

The development team wants to move the Daily Scrum, which takes place at 1pm, to 8am on Fridays so developers can leave early. How do you respond as the Scrum Master?

☐ Respect the self-managing Scrum Team and reschedule Friday's Daily Scrum

☐ Change the time of the Daily Scrum to 8am for every day of the week

☐ Explain to the team that the Daily Scrum must always take place at the same time and location

☐ Ask the Product Owner if it's agreeable to changing the Daily Scrum to 8am on Friday

Since the Daily Scrum is a Scrum Event, and since Scrum Events are supposed to take place at the same time and location every day, the Scrum Master would need to explain to the team that it can't change arbitrarily change the time the Daily Scrum takes place on Fridays.

Chapter 9

The Sprint

All of the development, planning, retrospectives and reviews that happen in Scrum happen within the scope of a Sprint. It's the most important part of Scrum.

> Sprints are the heartbeat of Scrum, where ideas are turned into value.
>
> — 2020 Scrum Guide page 7

Fixed Length Sprints

A Sprint has a fixed length of one month or less.

Not to sound pedantic, but 'fixed' means the length of a Sprint can't be changed once it's been started.

> Sprints are fixed length events of one month or less to create consistency.
>
> — 2020 Scrum Guide page 7

Sometimes the Scrum Certification exam will ask you when it's possible to change the length of a Sprint.

The temptation is to say 'during Sprint planning,' but Sprint planning happens during the Sprint, so if you were to change the duration of the Sprint, it would be the length of the next Sprint you'd be changing, not the current one.

So discussion about the length of the Sprint and whether the Sprint should be shortened or extended are best to take place during the Sprint Retrospective. That way if you do decide to change the Sprint length, the change can apply to the Sprint that starts after the Retrospective finishes.

Having said that, the length of the Sprint shouldn't be changed too often. It should remain relatively consistent throughout the project.

One Month or Less

The other key point here is the fact that a Sprint is a month or less.

- It's not 28 days
- It's not 4 weeks

The length of a Sprint is one month or less.

You'll be tested on that, so commit it to memory.

The Start of the Sprint

> A new Sprint starts immediately after the conclusion of the previous Sprint.

We've already discussed, ad nauseum, about how the next Sprint starts immediate after the current Sprint ends.

The Scrum Guide re-states this point again here, which is a good clue for you to pay attention to it. You'll be tested on this fact several times on the exam.

Test Yourself

> True or False: Sprint Zero can last for up to 49 days.

This is false for two reasons.

First, there is no such thing as 'Sprint Zero' as far as the Scrum Guide goes.

- There are no Release Sprints in Scrum
- There are no UAT Sprints in Scrum
- There are no QA Sprints in Scrum
- There are no Infrastructure Sprints in Scrum
- There is no Sprint Zero in Scrum

Whenever you see the term 'Sprint Zero' on the Scrum Master Certification exam, it's usually a red herring.

More to the point, this statement is false becuase the maximum length of a Sprint in Scrum is one month. That's the max. Often times it is shorter.

Test Yourself

> True or false: A Sprint can be any length a team decides is best, so long as it is less than a month.

This is true.

The Scrum Guide does not specify the minimum length of a Sprint, and when the Scrum Guide doesn't explicitly define something, then it's up to Scrum Teams to figure things out for themselves.

I could not imagine fitting a Sprint Planning meeting, Sprint Retrospective and Sprint Review into a 24 hour Sprint, but nothing in the Scrum Guide forbids it.

What Happens During the Sprint?

It's always assumed that things will change unpredictably from day to day during a Sprint. Which is why Scrum advocates for constantly inspecting and adapting.

> All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.
>
> During the Sprint:
>
> - No changes are made that would endanger the Sprint Goal;
> - Quality does not decrease;
> - The Product Backlog is refined as needed; and,
> - Scope may be clarified and renegotiated with the Product Owner as more is learned.
>
> — 2020 Scrum Guide page 7

However, there are some things that should not change during the Sprint

1. No changes are made that put the Sprint Goal at risk

2. No changes are made to decrease the quality of the existing product

The Scrum Guide doesn't have many hard and fast rules, but those are two of them.

Test Yourself

During development, the Scrum Team has found a Product Backlog Item is much more complex than initially anticipated.

The developers believe this PBI should actually be broken into multiple Product Backlog Items that should be scheduled across multiple Sprints. What should you as the Scrum Master do?

☐ Have the Scrum Developers clarify the requirements and renegotiate the scope of the Product Backlog Items with the PO

☐ Expand the length of the Sprint so the Development team can complete the complex Product Backlog Items

☐ Break the Product Backlog Item into multiple PBIs and add them to the Product Backlog

☐ Rewrite the User Story so the work releated to the Product Backlog Item can be completed within the current Sprint

A is correct.

If the developers have issues with the size, scope, complexity and clarity of the work they are doing, they simply contact the Product Owner so requirements can be clarified and scope can be renegotiated.

Sprints are a fixed length, so they can't ever be extended, which makes B incorrect.

The Scrum Master can never add items to the Sprint Backlog or the Product Backlog, so C is incorrect.

And the Scrum Guide makes zero references to User Stories, so any talk of User Stories on the Scrum Master Certification Exam will likely be a red herring.

Why we have Short Sprints in Scrum?

Some people often wonder why Scrum emphasises short sprints.

The answer is simple. The longer the Sprint, the lest predictable things become.

It's hard enough to predict four days into the future, let alone four weeks.

Short Sprints make things more predictable, and they also allow more regular interactions with stakehoders as Sprint Reviews, which occur at the end of the Sprint, take place more often.

> Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month.
>
> When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase.
>
> Shorter Sprints can be employed to generate more learning cycles and limit risk of cost and effort to a smaller time frame.
>
> Each Sprint may be considered a short project.
>
> — 2020 Scrum Guide page 7

The Scrum Guide places a great deal of emphasis on a short Sprint, and there are plenty of reasons for that.

Remember, a Sprint Review happens at the end of a Sprint. With a 2 week Sprint, the stakeholders get to review the product twice. With a 4 week Sprint, they only get to see it once.

A short Sprint makes it easier to manage risk. If the team is doing something wrong, they'll find out sooner from the stakeholders in a 2 week Sprint than in a month long one.

It's arguably easier to estimate work for two weeks, as opposed to three or four, and the project is less likely to go completely sideways in two weeks as opposed to two months.

Test Yourself

The development team is working with a new technology, which has introduced a great deal of risk and uncertainty into the project.

The Product Owner wants to reduce the Sprint time down to 2 or 3 weeks from the current Sprint length of one month.

What should the Scrum Master do?

The correct answer is B. As a Scrum Master, your job is to teach people about Scrum practices and Scrum Theory, and coach them on how to apply Scrum in a manner that works best for them.

Neither the Scrum Master or the Product Owner can arbitrarily set the length of a Sprint in Scrum. This is a decision that must come about through the concensus of the team.

C is incorrect because the length of the Sprint can be changed if the team agrees a change is needed. However, the Sprint length cannot be changed mid-sprint.

D is incorrect because there are no 'learning Sprints' in Scrum. Furthermore it is a requirement that every Sprint work towards the creation of an Increment. So Sprints are planned where the end result is not a usable increment.

Empiricism Trumps Scrum Metrics

> Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows.
>
> While proven useful, these do not replace the importance of empiricism.
>
> In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.
>
> — 2020 Scrum Guide page 7

The Scrum Guide is light on metrics.

The only Scrum metrics referenced in the guide are mentioned here:

- Burn-down charts
- Burn-up charts
- Cumulative flows

What's more, the guide kinda talks them down, saying that while they are useful, they're not as good as empirical analysis, which is a bit of a diss.

Still, for the exam, you do need to know what each of these charts do. You don't need to understand them in depth, but questions will arise about what they can tell you.

Burn-Down Chart: A burn-down chart is a graphical representation of the amount of work

remaining in a project versus time. It tracks the progress of a project by showing the remaining work that needs to be completed on the vertical axis and the time on the horizontal axis. The chart starts with the total amount of work to be completed at the beginning of the project, and as work is completed, the line on the chart gradually moves down until it reaches zero at the end of the project.

Burn-down charts are useful because they help teams track progress over time and adjust their approach as necessary. For example, if the team is falling behind schedule, they can identify this early on and make changes to get back on track.

Burn-Up Chart: A burn-up chart is similar to a burn-down chart but shows progress in a different way. Rather than showing the remaining work on the vertical axis, a burn-up chart shows the amount of work completed over time. The chart starts with zero completed work at the beginning of the project, and as work is completed, the line on the chart moves up until it reaches the total amount of work to be completed at the end of the project.

Burn-up charts are useful because they show progress towards a goal and can help teams identify whether they are on track to complete the project on time.

Cumulative Flow Chart: A cumulative flow chart is a graphical representation of the flow of work in a project. It shows the amount of work in progress at any given time, as well as the rate at which work is being completed. The chart has a horizontal axis representing time and a vertical axis representing the number of tasks.

The chart starts with a backlog of tasks at the beginning of the project and shows how tasks move through different stages of completion over time. For example, a task might start in the "to do" column, move to "in progress", and then finally move to "completed".

Cumulative flow charts are useful because they help teams identify bottlenecks in their workflow and optimize their processes to increase efficiency. By tracking the rate of completion over time, teams can also identify whether they are on track to complete the project on time.

You Can't Predict the Future with Scrum

Managers and team leads always want metrics that will help predict when a given project will be finished, or when a product will hit a given milestone. Scrum is quick to downplay the various charting techniques used to predict the future.

The Scrum Guide is quick to point out that there are limits to the accuracy of charts and metrics, and there is no replacement for empiricism.

> While proven useful, these do not replace the importance of empiricism.
>
> In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.
>
> — 2020 Scrum Guide page 7

The Scrum Guide is quick to point out that there are limits to the accuracy of charts and metrics, and there is no replacement for empiricism.

Empiricism insists that experience and evidence should for the basis for belief and knowledge; moreso than a chart does.

Importance of Empiricism

For example, a chart might say a team is completing 20 backlog items a week. Does that mean you complete 20 backlog items next week?

You might empirically know that two developers will be on vacation, or perhaps one developer is coming down with the flu.

The trend on the chart may indicate that next week the team will complete 20 backlog items next week, by you emprically know that sickness or vacation times will likely cut that number in half.

That's an example of empirical knowledge being a much better, and a much more informed indicator of future performance than a burn-down chart.

Who Can Cancel a Sprint?

The Scrum Certification guide will definitely ask you who has the power to cancel a Sprint, and under what conditions the Sprint can be cancelled.

> A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

— 2020 Scrum Guide page 7

These two points surprise people for the following three reasons.

1. It is indeed possible for an active Sprint to be cancelled
2. The only reason to cancel a Sprint is if the Sprint Goal becomes obsolete
3. Only the Product Owner can cancel a Sprint, not the Scrum Master or any stakeholder

Just by virtue of the authoritative nature of the name, people think a Scrum Master would have the authority to cancel a Sprint.

Scrum Masters don't.

A Scrum Master can't cancel a Sprint. Only the Product Owner can.

And there is only one reason why a Sprint can be cancelled: the Sprint Goal has become obsolete.

That means none of the following reasons are justifications to cancel the Sprint:

- The lead developer quit
- The building flooded and there's no power
- The Scrum Master has gone on vacation
- The CEO wants a different feature developed

There are plenty of reasons why someone in the organization might want to cancel the Sprint, but

the only reason a Sprint can be cancelled is if the Sprint Goal has become obsolete, and the only person allowed to make that call is the Product Owner.

What isn't said about the Sprint

People want the Scrum Guide to answer all of their questions about how to run a big project. Scrum doesn't do that.

Scrum provides a set of basic rules written out in a guide that's less that 4000 words. I wrote college papers longer than that.

If Scrum doesn't provide a rule or guideline around something, then it's up to the group of self-managing, highly motivated professionals on that team to figure it out on there own.

That can be frustrating.

When to Start the Sprint?

The Scrum Certification exam will try to trick you into applying some arbitrary rule that doesn't exist.

For example, say a product has two Scrum Teams working on it, which happens quite regularly.

- Should those two teams start their Sprints on the same day?
- Should those two teams end their Sprints on the same day?
- Should those two teams have Sprints of equal length?

On the surface, all three of those assertions seem reasonable. It seems like a nice way to keep the two Scrum Teams in sync.

At the same time, that approach might create chaos. Imagine stakeholders having to go to two Sprint Reviews on the same day. That might be too much for them.

Or maybe two team who finish their Sprints on the same day would end up pushing their increments into production on the same day, which might cause panic for system administrators?

So maybe it'd actually better to stagger the Sprints? Have one team start a 4 weeks Sprint on the first of the month, and have the other start the Sprint in the middle of the month?

No Rule Means No Rules

The fact is, the Scrum Guide doesn't care what those two teams do. The teams are allowed to figure it out on their own.

If the Scrum Guide doesn't speak directly to the topic, then there are no rules about it.

The Scrum Guide **does** say two teams working on the same project must share the same Product Backlog and the same Product Owner.

- Should their Sprints be the same length?

- Should their Sprints start on the same day?

Scrum doesn't provide any guidance on these topics either way. It's assumed your team will use empirical measures and apply lean thinking and come up with a plan that is best for them. Scrum wants you to figure out what's best for you.

Chapter 10

Sprint Planning

Scrum is simple. It really is.

I often hear people say they hate Scrum because it's too complicated, but I just don't see how that's possible.

At the heart of it, Scrum basically just says this:

- Teams should take a bit of time every month or so to plan out the work they're going to do

- Developers should meet for 15 minutes every workday just to get the ball rolling again

- At least once a month developers should share what they've done with stakeholders

- After talking to stakeholders, the team should talk a bit about what did and didn't work

That's it. That's the basics of Scrum right there.

I honestly don't know how anyone could say a strategy like that is a bad thing. It seems pretty basic, pretty reasonable and pretty straight-forward to me.

And it all starts off with Sprint Planning.

> Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.
>
> — 2020 Scrum Guide page 8

Harping back to a theme we've crowed about before, note that Sprint Planning happens within the Sprint, not before the Sprint begins.

Many questions on the Scrum Master certification exam will suggest that Sprint Planning happens before the Sprint, but that's always a red herring.

None of the Scrum events happen before or after a Sprint - they all take place within the Sprint.

Who is Accountable for Maximizing Value?

> The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal.
>
> — 2020 Scrum Guide page 8

We learned earlier in the section about the Product Owner that the PO is the one accountable for maximizing the value of the product created by the Scrum Developers.

Here the Scrum Guide re-emphasizes that point, stating that the Product Owner comes to the Sprint Planning meeting with a list of the most important features for the team to work on.

Who's Invited to the Sprint Planning Meeting?

> The Scrum Team may also invite other people to attend Sprint Planning to provide advice.
>
> — 2020 Scrum Guide page 8

Notice how anyone can be invited to participate in the Sprint Planning meeting. That's a significant point, and a concept you'll be tested on often.

There are often multiple Scrum teams working on a single project.

Remember, the maximum size of a Scrum team is 10 people, but it's not unusual for an enterprise software development project to have thirty or forty developers working on it. That's means there will be at least four or five Scrum teams allo working on the same product at the same time.

Minimizing Inter-team Dependencies

So how do multiple teams avoid stepping on each other's toes and working on the same backlog items? How to teams avoid being blocked by a dependency their chosen work has on something that has to be completed by another team first?

One way to manage these issues is to have representatives from other Scrum Teams attend your Sprint Planning meeting. Representatives from other teams can share their progress, insights and even deliverable dates for dependencies your project might have on them.

Or better yet, if you know that certain Product Backlog items have dependencies on work that might not get completed by another team, then you can just pick different backlog items altogether. The fewer the dependencies your work has on other teams the better.

Test Yourself

The Scrum Developers might want to invite someone outside the Scrum Team to attend the Sprint Planning meeting to discuss pertinent issues.

Which of the following is true about people from outside the Scrum Team participating in the Sprint Planning meeting?

- ☐ Outsiders can discuss how Product Backlog Items align with corporate strategy and business goals
- ☐ Outsiders can discuss how certain Product Backlog Items may be of importance to customers
- ☐ Outsiders can discuss dependencies on other Product Backlog Items under development by other teams
- ☐ Outsiders can discuss risks that might be encountered in the current Sprint
- ☐ Scrum Developers cannot invite anyone outside the team to the Sprint Planning event

Option E is correct.

There is nothing in the Scrum Guide that forbids Scrum Developers from inviting people outside the team to the Sprint Planning meeting.

If someone outside the Scrum Team can provide guidance on how to best plan the Sprint, they are more than welcome to speak up during Sprint Planning.

Test Yourself

> True or False: It is the Scrum Master's responsibility to ensure all Sprint Planning attendees are ready and prepared to discuss the Produt Backlog.

This is false.

It's actually the Product Owner who ensures attendees are prepared to discuss how Product Backlog items map to the Product Goal.

Answering Why, What and How

The goal of Sprint Planning is to come up with good answers to the questions of why, what and how. You'll be tested on these questions on the exam.

> Sprint Planning addresses the following topics:
>
> * Topic One: Why is this Sprint valuable?
> * Topic Two: What can be Done this Sprint?
> * Topic Three: How will the chosen work get done?
>
> Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint.
>
> For shorter Sprints, the event is usually shorter.
>
> — 2020 Scrum Guide page 8

Time Boxing Sprint Planning to 8 Hours

To pass the Scrum certification exam have to know the time boxes for Scrum events like the back of your hand:

* Sprint Planning is time boxed to a maximum of 8 hours for a one-month Sprint
* The Daily Scrum is time boxed to a maximum of 15 minutes
* The Sprint Review is time boxed to a maximum of 4 hours
* The Sprint Retrospective is time boxed to a maximum of 3 hours

Test Yourself

Which of the Scrum events can last the longest?

☐ Daily Scrum

☐ Sprint Review

☐ Sprint Planning

☐ Sprint Retrospective

☐ The Review, Planning and Retrospective are all time boxed to 4 hours

☐ There is no time limit for the Sprint Planning

The answer to this question is C, Sprint Planning.

Sprint Planning is timeboxed to a maximum of 8 hours. Hopefully your team can get it done a bit faster.

Why is the Sprint valuable?

> The Product Owner proposes how the product could increase its value and utility in the current Sprint.
>
> — 2020 Scrum Guide page 8

It is the Product Owner who is responsible for ensuring the work of the Scrum Team produces the greatest amount of value.

The Product Owner knows what needs to be built to in order to make the product better.

Negotiating Product Backlog Item Selection

Note that the Product Owner simply makes a proposal with regards to what should be built next. The Scrum Team may have pragmatic objections and push for other Product Backlog items to be made part of the current Sprint.

Just think about a Scrum Team constructing a house. The Product Owner would likely want the kitchens and the bathrooms done, but the foundation of the house may not be laid. In that case, the developers would need to explain how the kitchen and the bathrooms will need to wait for a future Sprint.

Test Yourself

Who on the Scrum Team is responsible for maximizing the value of the work performed by the developers?

☐ The Scrum Master

☐ The Product Owner

Option B is correct.

Maximizing the value of the work performed by the Scrum Team is the job of Product Owner.

Test Yourself

Who first proposes an initial plan for the Sprint at the Sprint Planning meeting?

☐ The Scrum Master

☐ The Product Owner

☐ The Scrum Developers

☐ The Scrum Team as a whole

☐ The stakeholders

Option B is correct.

It is the Product Owner who proposes how the product could increase its value and utility in the current Sprint. This becomes the starting point for negotiations, suggestions and compromises during Sprint Planning.

Sprint Planning and the Sprint Goal

The Sprint Goal must be finalized by the end of the Sprint Planning meeting, and while other things may change during a Sprint, the Sprint Goal is one of the Scrum Artifacts that is not allowed to be edited, adjusted or changed once Sprint Planning has concluded.

> The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders.
>
> The Sprint Goal must be finalized prior to the end of Sprint Planning.
>
> — 2020 Scrum Guide page 8

Each Sprint needs a Sprint Goal. The Sprint Goal keeps the developers focussed throughout the Sprint.

The Sprint Goal also provides another important function - it provides transparency into the Sprint, as it allows stakeholders know what the developers are working on during the Sprint.

Again, the Sprint Goal must be finalized before the Sprint Planning meeting ends. The Sprint Goal

cannot change throughout the Sprint.

The Sprint Plan can change, and the items in the Sprint Backlog can change. In fact, it's expected that those things will change as conditions change throughout the Sprint. But the Sprint Goal must be finalized before the Sprint Planning meeting ends, and it cannot change during the Sprint.

Test Yourself

What happens if it becomes clear towards the end of the Sprint that the team will not achive the Sprint Goal?

☐ The Sprint is cancelled and a new Sprint Planning meeting takes place

☐ The next Sprint adopts the current Sprint's Goal continulally until the goal is achieved

☐ The developers update the Sprint Goal so that it is achievable by the end of the Sprint

☐ The developers talk about the Sprint Goal during the Sprint Retrospective

The last option is correct.

It's not unusual for a Sprint Goal to go unfullfilled. Sometimes things just don't go according to plan.

If the Sprint Goal is not achieved, the Scrum Team talks about what they can do better during the Sprint Retrospective meeting.

Nothing ever gets automatically rolled over from one Sprint into the next.

Each Sprint starts new with an empty Sprint Backlog, as it's assumed that since conditions, expectations and realities are constantly changing, what made sense when the prior Sprint was planned won't necessarily make sense for the current one.

Test Yourself

Who creates the Sprint Goal?

☐ The Scrum Master

☐ The Product Owner

☐ The Scrum Developers

☐ The Scrum Team as a whole

☐ The stakeholders

Option D is correct.

The Scrum Team as a whole creates the Sprint Goal.

Test Yourself

Option E is correct.

The Sprint Goal helps to build transparency into the development process by allowing stakeholders to know what the team is trying to achieve during the current Sprint.

Product Backlog Item Selection

> Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint.
>
> The Scrum Team may refine these items during this process, which increases understanding and confidence.
>
> — 2020 Scrum Guide page 8

During Sprint Planning, the Product Owner proposes what they believe think should be built during the Spring. The Product Owner explains what they believe will provide the most value to stakeholders and customers. However, it's the developers who actually decide which Product Backlog items get added to the Sprint.

Going back to the home construction analogy, the construction workers know more about how to build a house than the home owner.

The developers know what dependencies exist, what order certain Product Backlog items need to be created in, and what's the best path to move forward to create the highest value Product Backlog Items.

For example, the Product Owner might think building a bathroom will provide the most value, but the developers might know that a foundation of the house must be laid down first. In that case, the developers would select laying down the foundation as something to do during the Sprint, despite the foundation not being as important a feature to the stakeholders as a nice bathroom would be.

The developers, not the Scrum Master or the Product Owner, have the final say over what gets added to the Sprint.

Product Backlog Refinement

It's also worth noting that, while Sprint Planning is an opportunity for the developers to talk about the Product Backlog items, clarify them and refine them, this is not the only time developers are allowed to talk with the Product Owner.

The developers can call up the Product Owner at any time during the Sprint to clafify Product Backlog items.

Quite often the Scrum Master resides in the same war room as the developers so they can answer questions about the product any time they arise.

The Scrum Master certification exam will often provide an incorrect option that indicates there are only certain times a developer can talk to a stakeholder, or a developer can talk to a Product Owner.

Discussion between everyone on the Scrum Team and every stakeholder in the organization is never discouraged within the Scrum Guide. The more conversation the better!

Test Yourself

True or false: The Product Owner selects which Product Backlog Items the developers will work on during the Sprint.

This is false.

The Product Owner can prioritize the Product Backlog and inform the developers about which Product Backlog items provide the most value, but the developers have the final say over which items they build during the Sprint.

Test Yourself

True or false: Sprint Planning is the only time Scrum Developers are allowed to talk to the Product Owner to refine Product Backlog items.

This is false.

The Product Owner should always be available to answer questions about the product.

Conversation between the Product Owner, the developers, the Scrum Master and the stakeholders should happen freely and openly. Nothing in the Scrum Guide forbids it.

What can be Done this Sprint?

Selecting how much can be completed within a Sprint may be challenging.

> However, the more the Developers know about:
>
> - their past performance,
> - their upcoming capacity, and;
> - their Definition of Done,
>
> the more confident they will be in their Sprint forecasts.
>
> — 2020 Scrum Guide page 8

According to this paragraph, it is the Developers who are expected to estimate how much can be done in a Sprint.

It is up to the developers to know about their capacity, past performance and ability to follow through on the Definition of Done in order to estimate how much work they can accomplish.

This makes sense. After all, it is the developers who choose the Backlog Items they plan to implment in the Sprint. If they were unable to estimate their own work capacity, they wouldn't know how many Product Backlog items to choose.

Test Yourself

Who estimates how much work can be accomplished in a Sprint?

☐ The Scrum Master

☐ The Product Owner

☐ The Scrum Developers

☐ The Scrum Team as a whole

☐ The Scrum Team and stakeholders

The developers are the one who select how many Product Backlog items to include in the Sprint, so they are the ones who are estimating how much they believe can be accomplished.

Test Yourself

Which of the follwing three are the most empirical measures of how much the Scrum Developers can accomplish in a given Sprint?

☐ Burndown charts

☐ Past performance

☐ Burnup charts

☐ Upcoming capacity

Upcoming capacity, and knowledge of past performance are more empirical measures than

burndown or burnup charts.

How will the chosen work get done?

> For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done.
>
> This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

— 2020 Scrum Guide page 8

A Product Backlog item may take months to complete.

In Scrum, the developers need to compose Product Backlog items down into smaller pieces. This is known as decomposition.

The goal is to break each Product Backlog item down into a series of chunks that can be estimated to take a day or less to complete.

Note that the Scrum Guide never talks about points or stories.

Quite often you will see a question on the Scrum Master Certification exam where one of the answers talks about breaking user stories down into a certain number of points. Those answers are always wrong.

The closest the Scrum Guide ever gets to talking about stories and points is where it recommends that developers break down, or decompose, Product Backlog Items into a days worth of work or less.

Test Yourself

What is the best way for a developer to approach a complicated Product Backlog item?

☐ Have the development team break the Product Backlog item down into smaller user stories.

☐ Have the Product Owner team break the Product Backlog item down into smaller user stories.

☐ Break the Product Backlog item down into 1 point increments, representing 1 day of work.

☐ Break the Product Backlog items into multiple work items of one day or less

Option D is correct.

There are no user stories in Scrum, and there's no such thing as points, which makes the first three options wrong.

Given a complicated epic, the Scrum Developers should try to decompose those items into multipe

work items of one day or less.

The Outcome of Sprint Planning

> The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.
>
> — 2020 Scrum Guide page 9

Note that the Sprint Backlog is more than just the set of Product Backlog items the team has selected for the Sprint. It also includes a Sprint Goal that cannot change during the Sprint, along with a plan that is expected to change daily.

Test Yourself

The Sprint Backlog is composed of:

☐ The Product Backlog items selected for the Sprint

☐ The Product Backlog items selected for the Sprint and the Sprint Goal

☐ The Product Backlog items selected for the Sprint, the Sprint Goal and the Sprint Plan

☐ The Product Backlog items selected for the Sprint, the Sprint Goal the Sprint Plan and the Product Goal

Option C is correct.

The Sprint Backlog is composed of the Product Backlog items selected for the Sprint, the Sprint Goal and the Sprint Plan.

Chapter 11

The Daily Scrum

Some frameworks and methodologies have a Daily Standup.

We don't have Daily Standups in Scrum.

First of all, the term is abelist, as not everyone can stand. Secondly, the idea of a 'Daily Standup' is that it won't last long because people will eventually get tired of standing.

Scrum practitioners believe there are better motivations to use to keep meetings short other than the infliction of pain.

> The Daily Scrum is a 15-minute event for the Developers of the Scrum Team.
>
> The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.
>
> — 2020 Scrum Guide page 9

Adapting to Change

Things change from day to day, and those changes can mess up your plans.

That's why developers get together everyday for a Daily Scrum so they can talk about anything that is impeding or accelerating their progress towards the Sprint Goal.

If things are going sideways, they adapt. If things are going better than expected, they adapt.

The goal of the Daily Scrum is for developers to talk, discuss progress, and quickly solve any problems that have arisen since they last spoke.

Notice that the Daily Scrum is time boxed to 15 minutes.

Test Yourself

Which of the following are a responsibility of the Scrum Master with regards to the Daily Scrum?

☐ The Scrum Master should coach the developers about keeping the Daily Scrum timeboxed to 15 minutes

☐ The Scrum Master should ensure each developer gets a chance to speak

☐ The Scrum Master starts the Daily Scrum by asking the 3 standard Daily Scrum questions

☐ The Scrum Master starts the Daily Scrum by asking every developer to stand up.

The Daily Scrum is for the Developers

The Daily Scrum is for the developers. It is run by developers and managed by developers.

The only role the Scrum Master has with regards to the Daily Scrum is ensuring it is run in such a way that complies with the rules of Scrum. In this case, that means keeping the Daily Scrum to less than 15 minutes.

Note that past Scrum Guides mentioned the 3 daily scrum questions of:

- What did you do yesterday?
- What do you plan to do today?
- Is anything impeding your progress?

These questions were removed from the Scrum Guide. They can be asked if the developers find them helpful, but the are certainly not a requirement of the Daily Scrum.

Keep the Time and Location Consistent

> To reduce complexity, it is held at the same time and place every working day of the Sprint.
>
> — 2020 Scrum Guide page 9

All Scrum events are supposed to be held at the same time and in the same location. This applies to the Daily Scrum as well.

Test Yourself

The development team has decided to only work one day a week and will only hold the Daily Scrum on Mondays. How do you react as a Scrum Master.

- ☐ Inform the team Scrum development cannot happen only one day a week
- ☐ Have Human Resources talk to the development lead about their proposed work schedule
- ☐ Cancel the Sprint and consult the Product Owner
- ☐ Inform the team this is fine so long as the work takes place on Monday

There's nothing in the Scrum Guide that forbids a team from working once a week.

The only requirement is that the Daily Scrum happens every workday. So if the one day a week the team works is Monday, then the Daily Scrum should take place on the Monday.

Test Yourself

The Scrum team wants to move the Daily Scrum to the atrium on Fridays to help them wind down after a busy workweek. How do you, as a Scrum Master, respond?

- ☐ Explain to the that the Daily Scrum must take place at the same time and place
- ☐ Allow the devlopers to be self-managed and hold their Daily Scrum in the Atrium on Fridays

> ☐ Confirm with the Product Owner that it is okay to change the Daily Scrum location on Fridays
>
> ☐ As the Scrum Master, work to remove any impediments standing in the way of running the Friday Scrum in the atrium

Option A is correct.

The Scrum Guide says the Daily Scrum must take place, each workday, at the same time and place.

When Scrum Masters and POs do Development

Have you ever seen a Scrum Master or Product Owner do some work that becomes part of a usable Increment? If they do, they become developers themselves.

> If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.
>
> — 2020 Scrum Guide page 9

This is an interesting statement that answers a number of question about how Scrum works.

People often wonder if a Scrum Master or Product Owner is allowed to actively do development. Of course they can!

If a Scrum Team of five people are cast away on a desert island and scrambling to build a shelter before a storm comes, everyone on that Scrum Team going to pitch in. If your life is on the line, you're not going to refuse to help building a shelter just because you've accepted the designation of Scrum Master.

On small teams and startups the Scrum Master might also be the Product Owner and they might also be a developer as well. It might not be a best practice, but there's nothing that forbids it. And more to the point, it might actually make a lot of sense in a really small development firm.

So yes, a developer can also be a Scrum Master or a Product Owner or vice versa.

The only requirement is that if a Scrum Master or Product Owner does development, they are expected to attend the Daily Scrum and participate as though they were a developer, not the Scrum Master or PO. They relinquish their Scrum Master or Product Owner accountabilities while the Daily Scrum takes place.

Who participates in the Daily Scrum?

> The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.
>
> — 2020 Scrum Guide page 9

The Daily Scrum is for the developers. It should be run by the developers, organized by developers

and managed by the developers. How they manage or organize it is up to them.

Anyone can attend the daily Scrum. If the developers want to hire a circus clown to create baloon animals while the Daily Scrum proceeds, then all the power to them. But only the developers are supposed to participate.

Now that's not to say the developers can't ask the Scrum Master or Product Owner a question during the Daily Scrum. That may be necessary to properly adapt their plan towards the sprint goal. But the Scrum Master, Product Owners and stakeholders shouldn't be active participants driving the meeting. The meeting should be driven by the developers.

Purpose of the Daily Scrum

> Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.
>
> — 2020 Scrum Guide page 9

Things change from day to day. The Daily Scrum is a time for developers to deal with issues that have arisen that may delay their progress and put the Sprint Goal in jeopardy.

Hopefully having the whole team of developers together in Scrum will help to bring about quick solutions to problems the team might face.

Also note that the goal of the Daily Scrum is to reduce the need for other meetings.

One complaint I often hear about Scrum is that there are too many meetings. That shouldn't be the case. The Daily Scrum should eliminate the need for other meetings.

Meetings Don't Replace Pragmatic Communication

> The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.
>
> — 2020 Scrum Guide page 9

Don't ever think that the Daily Scrum is the only time developers are allowed to talk about their work, or that it's the only time to change the Sprint plan.

If a team of construction workers were putting up a roof, and a wind gust blew all their shingles away, would they wait until the next day's Scrum to form a new plan, or would they reformulate thier plans right away?

Developers can meet with each other any time they like. They can schedule additional meetings as a group. They can meet one on one at each other's desks. They can have dinner together after work.

There's nothing in the Scrum Guide that restricts communication beteen developers, stakeholders, product owners, scrum masters or anyone else. The only thing the Scrum Guide provides is a few timeboxed events that guarantee opportunities for communication, transparency and adaptation take place.

Test Yourself

A critical bug has appeared in your code that may put the Sprint Goal at risk. What should you, as a developer, do?

☐ Speak to your fellow developers as soon as possible to find a way to adapt the Sprint plan

☐ Bring the issue up at tommorow's Daily Scrum

☐ Inform the Scrum Master and have the Scrum Master remove the impediment

☐ Pass the issue to the Quality Assurance team and continue working on Product Backlog items

Any time an issue comes up it should be addressed as soon as possible.

If a critical bug appears in your code, and you think it will impact the Sprint Goal, then meet with your fellow developers and see if you can adapt by adjusting the Sprint plan.

Don't ever let the Scrum Guide and the various Scrum events and artifacts impede pragmatic thinking. If a problem arises that needs to be taken care of immediately, take care of it immediately. Don't wait for the next Scrum event to adapt.

Chapter 12

Sprint Review

On any project, not just in the domain of Scrum, stakeholders should continually get updates on the state of the product they're invested in having built.

In fact, that's one of the ideas behind the continuous delivery of software - to constantly deliver software to your clients and stakeholders every time a new feature or Increment of value has been completed.

The more often the stakeholders see new features added to the product, the more in tuned they will be with the pace at which the project is progressing.

Of course, not every organization can continuously delivery updates on a daily basis. But at the very least, a development team should give the stakeholders an opportunity to inspect what has been completed and provide feedback so the team can adapt accordingly.

The Sprint Review exists to make sure that stakeholders get at least one chance to inspect the work completed during every Sprint, but it certainly doesn't have to be the only time stakeholder inspection takes place.

Review, Inspect and Adapt

> The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.
>
> — 2020 Scrum Guide page 9

The Sprint Review is the one Scrum event where everyone is invited and asked to participate. That not only includes everyone on the Scrum Team, but stakeholders as well.

The purpose of the Sprint Review is pretty self-explanatory. It's for everyone to review what was done and talk about what they like, what they don't like and where they think things should go next.

Everyone gets to speak at the Sprint Review. Developers can speak to stakeholders, stakeholders can speak to the Scrum Master, and the Product Owner can speak to developers.

Note that there is never any restriction in Scrum about developers talking to stakeholders, or the Scrum Master talking to stakeholders or anything like that.

The Spring Review intentionally and concertedly brings everyone together, but there's no rule that says developers can't talk to stakeholders about development when they pass each other in the hallways at work. The more openness and transparency the better.

Test Yourself

Which of the following is true about the Sprint Review?

- ☐ Only the Scrum Master is allowed to talk directly to stakeholders
- ☐ Only the Product Owner is allowed to talk directly to stakeholders
- ☐ Developers are not allowed to talk directly to stakeholders
- ☐ The Sprint Review is the only Scrum event where both Stakeholders and the Scrum Team are invited

Option D is correct.

Anyone is allowed to talk during the Sprint Review. There's no rule anywhere in Scrum that restricts developers from talkign directly to stakeholders.

The Sprint Review is a planned event that brings everyone together to discuss progress, but stakeholders and members of the Scrum Team are allowed to talk directly to each other any time during the Sprint

What to show in the Review?

There's a small detail left out of this discussion of the Sprint Review which is somewhat important. For example, the following sentence doesn't exactly specify what the stakeholders get to see during the Sprint Review.

> During the event, the Scrum Team and stakeholders review **what was accomplished** in the Sprint and what has changed in their environment.
>
> — 2020 Scrum Guide page 9

The term **what was accomplished** is somewhat vague.

We discover in the Scrum Guide's section on artifacts that only completed Increments, that is work items that have met the Definition of Done, are to be presented at the Spring Review.

> Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism.
>
> Work cannot be considered part of an Increment unless it meets the Definition of Done.
>
> — 2020 Scrum Guide page 12

Any work that is partially done is not to be presented at the Sprint Review. Only Increments, or the sum of all the Increments created during the Sprint, are to be discussed at the Sprint Review.

A Discussion on What to do Next

> Based on this information, attendees collaborate on what to do next.
>
> The Product Backlog may also be adjusted to meet new opportunities.
>
> — 2020 Scrum Guide page 9

The Sprint Review gives everyone on the team and the stakeholders a chance to talk about what they like, what they don't like and what they think should change in terms of how the product is being developed.

If the stakeholders think some features don't work and should be removed, the Product Owner may remove those features from the Product Backlog. If there are features the stakeholders decide they want, then the Product Owner can either add them to the Product Backlog or prioritize them if they already exist.

The Sprint Review helps create transparency, is an opportunity for inspection and allows for adaptation.

Sprint Review != a Presentation

> The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.
>
> — 2020 Scrum Guide page 9

The Scrum Master Certification exam will likely include a question or two where one of the incorrect options implies that the purpose of the Sprint Review is to simply do a demo or deliver a presentation. Don't fall for that.

The purpose of the Sprint Review is to inspect the work that was performed during the Sprint and adapt according to feedback recieved from the stakeholders. At the end of a Sprint Review, the stakeholders should know what was done during the previous Sprint, and the Scrum Team should have a better idea of what they should be working on next.

> The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint.
>
> For shorter Sprints, the event is usually shorter.
>
> — 2020 Scrum Guide page 9

You are guaranteed to get questions about the timebox length of every event. Remember that the Sprint Review is timeboxed to a maximum of 4 hours.

Which of the following Scrum events are timeboxed to 4 hours?

☐ The Sprint Review only

☐ The Sprint Retrospective only

☐ Sprint Planning only

☐ The Sprint Review and the Sprint Retrospective

☐ The Sprint Review and Sprint Planning

Both the Sprint Review is timebosed to 4 hours.

The Sprint Retrospective is 3 hours.

The Sprint Planning event is capped at 8 hours, and the Daily Scrum is 15 minutes.

When the Sprint Review is complete, it's time for the team to do a Sprint Retrospective, after which the Sprint comes to an end.

Chapter 13

Sprint Retrospective

When the work is done, and the Sprint Review is complete, it's time for the Scrum Team to circle the wagons and chat a bit about what went right, what went wrong, and what they want to change when the next Spring begins.

> The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.
>
> The Scrum Team inspects how the last Sprint went with regards to:
>
> - individuals
> - interactions
> - processes
> - tools
> - the Definition of Done.
>
> Inspected elements often vary with the domain of work.
>
> — 2020 Scrum Guide page 10

One question that comes up regularly on the Scrum Master certification exam is when should the Definition of Done be discussed or updated. This paragraph makes it clear that the Sprint Retrospective is the correct time in the Sprint to do so.

Also notice how the Scrum Guide indicates that discussions about how tools are working, how people are getting along and how well processes implemented within the Scrum Framework are best discussed during the Sprint Retrospective.

If the certification exam ever asks when developers should talk about how well their tools are working, or how well they have applied the Scrum framework, the correct answer will be the Sprint Retrospective, not the Daily Scrum. This is a bit counter-intuitive, because you would think developers would be encouraged to talk about their tools and the process when they were all together in the Daily Scrum. But if it's a meta-discussion about how things are working, that type of discussion belongs in the retrospective.

If a given tool is causing a problem and creating a blocker or impediment that puts the Sprint Goal at risk, then the discussion would be more apt to take place during the Daily Scrum.

Test Yourself

The developers are not happy with how the Sprint framework has being implemented. When is the best time to discuss this?

☐ During the Sprint Review

☐ During the Daily Scrum

☐ During the Sprint Planning

☐ During the Sprint Retrospective

Option D is correct.

Discussions about how things are working, how various processes are being implemented, and even discussion about how to do Scrum better should happen during the Sprint Retrospective. '''

The 80/20 Principle

> Assumptions that led them astray are identified and their origins explored.
>
> The Scrum Team discusses: - what went well during the Sprint, - what problems it encountered, and - how those problems were or were not solved.
>
> The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.
>
> — 2020 Scrum Guide page 10

This is a weird section. It's the most wishy-washy section of the entire Scrum Guide.

Terms like 'most helpful', 'most impactful' and 'may even be' don't provide much clarity. It's strange, because the Scrum Guide is usually very precise with regards to the terms and phrases that is uses.

I think the guide is just reflecting the 80/20 pricinple that asserts a few of your best ideas will end up having the largest impact, so don't feel the need to include every single suggestion made during the Sprint Retrospective into the next Sprint.

The idea is to act upon the best suggestions for improvement right away. That might mean writing a note on the whiteboard in the war room, or even going so far as adding them to the Sprint Backlog for the next Sprint.

Adding to the Next Sprint's Backlog?

> They may even be added to the Sprint Backlog for the next Sprint.
>
> — 2020 Scrum Guide page 10

That's actually an interesting line.

In Scrum, we're never really supposed to look beyond the current Sprint.

We don't plan for future Sprints before they start because we never really know how conditions are going to change. One of the Scrum values is focus.

Here is one of the rare instances in the Scrum Guide where something generated in one Sprint is added directly to the next Sprint.

In fact, it was stated earlier in this book that in the Sprint Planning meeting, the Sprint Backlog is empty, as all unfinished Sprint Backlog items are returned to the Product Backlog when the Sprint ends.

I guess this is one instance where a Sprint Backlog item can remain where it is when the teams moves from one Sprint to the other.

Test Yourself

Focus in an important part of the Scrum framework. Scrum Teams are expected to stay focused on the current Sprint. When during the Sprint are developers expected to be thinking about the next Spring?

☐ During Sprint planning before the Sprint begins

☐ During the Sprint Review as they collaborate on what to do next

☐ During the Sprint Retrospective as the Scrum Team identifies ways to improve its effectiveness

☐ Scrum Teams should never look beyond the timeline of the current Sprint

Options B and C are correct.

During the Sprint Review, the Scrum Team collaborates with stakeholders and discuss what to do next, and in the Sprint Retrospective, the Scrum Team may add strategies for improvement to the next Sprint's backlog.

Test Yourself

The Scrum Team has come up with two great ideas on how to improve their processes during the Sprint Retrospective. What should they do with these ideas?

☐ Add them to the current Sprint's Sprint Backlog

☐ Add them to the current Product Backlog

☐ Add them to the next Product's Product Backlog

☐ Add them to the next Sprint's Sprint Backlog

According to the Scrum Guide, great ideas that come out in the Sprint Retrospective can be placed in the Sprint Backlog of the next Sprint.

The Conclusion of the Sprint

> The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.
>
> — 2020 Scrum Guide page 10

The Sprint Retrospective is the last timeboxed event mentioned in the Scrum Guide. If you're keeping track, the length of the events in Scrum are:

• Sprint - up to 1 month

- Planning - up to 8 hours
- Review - up to 4 hours
- Retrospective - up to 3 hours

While a Sprint is given a fixed time that cannot be extended while a Sprint is in progress, the Scrum Guide never refers to a Sprint as a 'timeboxed' event.

And yes, I do realize this is probably the fourth or fifth time I've mentioned the duration of each of the Scrum Events in the book. That's because you're likely going to get five or six questions on it, so it's not a trivial matter.

After the Sprint

Once the Sprint Retrospective ends, the Sprint is over and the next Sprint begins.

Remember that there is nothing that happens after one Sprint but before the next Sprint begins.

Everything happens within the scope of the Sprint. And when one Sprint ends, the next one starts immediately.

Test Yourself

Which Scrum event marks the end of the Sprint?

- ☐ Sprint Planning
- ☐ Sprint Review
- ☐ Sprint Retrospective
- ☐ Sprint Deployment

The final event in the Sprint is the Sprint Retrospective.

Test Yourself

A new Sprint starts:

- ☐ Immediately after Sprint Planning
- ☐ Immediately after the Sprint Review
- ☐ Immediately after the Sprint Retrospective
- ☐ Immediately after deployment of the Increment

Test Yourself

What is the order, from longest to shortest?

- ☐ Sprint, Review, Retrospective, Planning
- ☐ Sprint, Planning, Review, Retrospective
- ☐ Sprint, Planning, Retrospective, Review
- ☐ Planning, Sprint, Review, Retrospective
- ☐ Planning, Review, Retrospective, Sprint

Option B is correct.

Sprint - up to 1 month Planning - up to 8 hours Review - up to 4 hours Retrospective - up to 3 hours

Test Yourself

In what order to the Scrum events run?

- ☐ Sprint, Review, Retrospective, Planning
- ☐ Sprint, Planning, Review, Retrospective
- ☐ Sprint, Planning, Retrospective, Review
- ☐ Planning, Sprint, Review, Retrospective
- ☐ Planning, Review, Retrospective, Sprint

Option B is correct.

Everything happens within a Sprint.

Once the Sprint starts, planning occurs. Then work is done and a review is performed. Then a retrospective allows the team to discuss ways to improve for the next Sprint, which starts when the Sprint Retrospective ends.

Chapter 14

The Scrum Artifacts

There are three Scrum artifacts:

- The Product Backlog
- The Sprint Backlog
- The Increment of work the developers create

> Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.
>
> — 2020 Scrum Guide page 10

The existence of these artifacts help provide transparency by answering the three questions stakeholders want to know.

- What is currently being done? That's answered by the Sprint Backlog
- What needs to be done? That's answered by the Product Backlog
- What's been done? That's answered by the Increments of work that have been produced

Stakeholders get angry and upset when they don't have answers to these questions. The existence of transparent and inspectable Scrum Artifacts has a tendency to calm the nerves of nervous stakeholders.

Test Yourself

The stakeholders are waiting on a new feature to be developed and they want to know if it's being actively developed in this Sprint. Where would they look to find this information?

- ☐ The Product Backlog
- ☐ The Sprint Backlog
- ☐ The Definition of Done
- ☐ The Spring Goal

The Sprint Backlog provides transparency into what is being worked on during the current Sprint.

Scrum Commitments

The Scrum commitment is something new that was added to the 2020 Scrum Guide.

Each artifact now maps to a Scrum commitment.

> Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured: - For the Product Backlog it is the Product Goal. - For the

> Sprint Backlog it is the Sprint Goal. - For the Increment it is the Definition of Done. These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

— 2020 Scrum Guide page 10

Every arrow needs a target. The Scrum comitments act as a target for their corresponding artefact.

A Scrum commitment is the evaluation criteria against which a Scrum artifact is measured. Scrum commitments improve transparency and enhance an Agile Scrum developer's focus as the project progresses.

Test Yourself

The goal of every Sprint is to create an Increment that lives up to what ultimate measure?

☐ Standard performance metrics

☐ Quality Assurance review

☐ The Definition of Done

☐ A conflict free merge into the master branch

C is correct.

The Incrment makes a commitment to measuring up to the Definition of Done.

Test Yourself

How do the Scrum Artifacts reinforce empiricism for stakeholders?

☐ They help the stakeholders chart velocity

☐ They help Quality Assurance teams confirm quality metrics are achieved

☐ They provide insight into what has been done and what is being done

☐ They help stakeholders project the overall cost of the project

Empiricism is all about observation and experience.

The Scrum Artifacts allow stakeholders to know exactly what has been done and what is currently being worked on.

This provides more empirical evidence of the state of the project than burndown charts or cumulative flow diagrams.

Test Yourself

> True or False: The Scrum Artifacts exist to reinforce empiricism only for the Scrum Team.

This is false. The Scrum Artifacts are accessible by the Stakeholders as well as the Scrum Team. They help reinforce empiricism for everyone involved in the project.

Chapter 15

The Product Backlog

As we learned earlier, the Product Owner is the one person on the Scrum Team who is tasked with the job of optimizing the value of the work done by the Scrum Team.

The way the Product Owner does that is by keeping a list of all of the features and facets the product under development requires.

This list is known as the Product Backlog, and by prioritizing entries in the Product Backlog and encouraging developers to work on the most important ones, stakeholders and customers get delivered the features they want and need the most.

> The Product Backlog is an emergent, ordered list of what is needed to improve the product.
>
> — 2020 Scrum Guide page 10

This definition makes it clear that the Product Backlog is a list of the features that need to be built to complete the project.

This definition also demands that the list be ordered, in a proprity sequence. The most important items are at the top.

Test Yourself

True or False: The Product Backlog is a prioritized lists the things that are needed to improve the product.

This is true. This precicely describes the purpose of the Product Backlog.

Single Source of Work

> The Product Backlog is the single source of work undertaken by the Scrum Team.
>
> — 2020 Scrum Guide page 10

The fact that this is the single source or work to be undertaken by the Scrum Team indicates that it is also the only source of work undertaken by the Scrum team.

If the Scrum Team is going to work on something, then it has to first be part of the Product Backlog.

True or False: The Scrum Master can assign work to the Scrum Developers that is not a part of the Product Backlog.

This is false.

The Product Backlog is the only source of work for the Scrum Team.

Product Backlog Refinement

> Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event.
>
> They usually acquire this degree of transparency after refining activities.
>
> Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items.
>
> This is an ongoing activity to add details, such as a description, order, and size.
>
> Attributes often vary with the domain of work.
>
> — 2020 Scrum Guide page 10

Developers may see a high priority Product Backlog item that is far too complex to complete in a 3 or 4 week Sprint. When this happens, the developers work with the Product Owner to break the item down into something more manageable.

The goal is always to break a Product Backlog item into something that can be completed within a Sprint. And remember that druing Spritn planning, the developers are expected to break all of the selected Product Backlog items down into units of work that can be completed in a day or less.

> For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less.
>
> — Sprint Planning page 8

Test Yourself

When picking Product Backlog items for the current Spring, Scrum Developers must:

- ☐ Only choose Product Backlog items they believe the can complete within Sprint
- ☐ Refine large Product Backlog items down until they are large enough to complete in a Sprint
- ☐ Not be afraid to choose Product Backlog items that may span multiple Sprints
- ☐ Decompose large Product Backlog items down until they are large enough to complete in a day

Option B is correct.

There will often be Product Backlog items that are too large to complete in a single Sprint.

In those instances, the development team refines those large Product Backlog Items until a piece is uncovered that can be completed in the current Sprint.

Test Yourself

Breaking Product Backlog items selected for the Sprint down into smaller work items of one day or less is called:

- ☐ Grooming
- ☐ Refinement
- ☐ Decomposition
- ☐ Planning

Option C is correct.

The Scrum Guide desribes decomposition as breaking Product Backlog items into smaller work items of one day or less.

Test Yourself

Breaking large Product Backlog items down into smaller, more precise items is known as:

- ☐ Grooming
- ☐ Refinement
- ☐ Decomposition
- ☐ Planning

Option B is correct.

The Scrum Guide defines Product Backlog refinement as: the act of breaking down and further defining Product Backlog items into smaller more precise items.

Test Yourself

When does Product Backlog refinement occur:

- ☐ Product Backlog refinement happens during Sprint Planning
- ☐ Product Backlog refinement happens during the Sprint Retrospective
- ☐ Product Backlog refinement happens during the Daily Scrum
- ☐ Product Backlog refinement ongoing activity that happens throughout the Sprint

Option D is correct.

The Scrum Guide has very few rules about what should happen when.

If a Product Backlog item needs more details, be refined, be decomposed or anything else, then just do it!

When work needs to be done, don't wait for an official Scrum event to do it. Just get the work done!

Test Yourself

The developers are not happy with how the Sprint framework has being implemented. When is the best time to discuss this?

☐ During the Sprint Review

☐ During the Daily Scrum

☐ During the Sprint Planning

☐ During the Sprint Retrospective

Option D is correct.

Discussions about how things are working, how various processes are being implemented, and even discussion about how to do Scrum better should happen during the Sprint Retrospective.

Estimation and Sizing

> The Developers who will be doing the work are responsible for the sizing.
>
> The Product Owner may influence the Developers by helping them understand and select trade-offs.
>
> — 2020 Scrum Guide page 10

This point keeps coming up and up again in the Scrum Guide, and you can expect it to come up again and again on the certification exam.

Only the developers know what it takes to accomplish a given piece of work. Only the developer can size up Product Backlog Items and estimate how much time a given feature will take.

Test Yourself

Who is responsible for estimating how many Product Backlog Items can be completed in a Sprint?

☐ The Product Owner

☐ The Scrum Master

☐ The Scrum developers

Option C is correct.

The developers are the experts. Only the developers know how long it will take to complete a Product Backlog item.

Test Yourself

True or false: Taking into account upcoming vacation time is more empirical than estimating productivity based on burndown charts?

This is true.

Burndown charts and velocity calculations are great, but they are not a replacement for actualy knowledge and real-world experience.

Commitment: Product Goal

> The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against.
>
> — 2020 Scrum Guide page 11

Every arrow needs a target.

As the Scrum Team works to build the product, they need to know what the product is that they're building. The Product Goal serves this purpose.

True or false: The Product Goal represents the current state of the project?

This is false.

The Product Goal represents the future state of the project.

Product Goal and the Product Backlog

> The Product Goal is in the Product Backlog.
>
> The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal.
>
> — 2020 Scrum Guide page 11

This line, "the Product Goal is in the Product Backlog", actually confuses me.

The Product Backlog is an ordered list of what is needed to improve the product. That's what it says on page 10 of the Scrum Guide.

> The Product Backlog is an emergent, ordered list of what is needed to improve the product.
>
> — 2020 Product Backlog Definition page 10

So does this statement mean the Product Goal is one of the items in the list?

The Product Backlog has many items that have never been assessed and likely won't be worked on within the next year. What should the Product Owner do to clean up the Product Backlog?

- ☐ Create a second Product Backlog and moved these items there
- ☐ Delete the old items items from the Product Backlog
- ☐ Prioritize these Product Backlog items so the developers complete them
- ☐ Leave the Product Backlog items in the Product Backlog as they are

Options D is correct.

The Product Backlog contains everything needed to create the product and achieve the Product Goal.

If a Product Backlog item contributes to the completion of the Product Goal, then it must be in the Product Backlog.

Value Delivery

> A product is a vehicle to deliver value.
>
> A product has:
>
> - a clear boundary,
> - known stakeholders,
> - well-defined users or customers.
>
> A product could be a service, a physical product, or something more abstract.
>
> — 2020 Product Backlog Definition page 11

The people behind the Scrum Guide have worked hard to get people out of the mindset that Scrum is just for software development.

This section further emphases the point that Scrum can be used to develop more than just software.

Test Yourself!

During the Sprint, the product being built has:

☐ An approved budget

☐ Known stakeholders

☐ Well-defined users

☐ Has unclear boundaries

Options B and C are correct.

The Scrum Guide does not say anything about the topic of budgets, so option A is correct.

Option D is incorrect because the Scrum Guide insists that the product being built has clear boundaries.

One and Only Objective

> The Product Goal is the long-term objective for the Scrum Team.
>
> They must fulfill (or abandon) one objective before taking on the next.
>
> — 2020 Product Backlog Definition page 11

According to the Scrum Guide, a Scrum Team can only work on one project at a time. They can't have their efforts split between two separate projects.

It's noteworthy that that doesn't mean an individual Developer or Scrum Master can't be on multiple teams at the same time.

I've quite often seen UI designers who have only a small part to play on a set of active projects be part of multiple Scrum Teams at the same time, and it's also very common for Scrum Masters to be on three or four different teams at a time. There is no rule against an individual being on multiple teams with separate objectives, but a single Scrum Team can only have one objective in mind.

Test Yourself!

The Product Owner beleves the Scrum Team has enough capacity to work on two separate projects concurrently. How do you, as the Scrum Master, advise the Product Owner to go forward with this plan?

☐ Double the timebox for all Scrum events

☐ Create a second Product Backlog for the new project

☐ Add features for the new Project into the current Product Backlog

☐ Inform the Product Owner that a Scrum team can only work on one product at time

Option D is correct.

A Scrum Team must fullfill or abandone one objective before taking on the next.

Chapter 16

The Sprint Backlog

> The Sprint Backlog is composed of:
>
> - the Sprint Goal (why)
> - the set of Product Backlog items selected for the Sprint (what)
> - an actionable plan for delivering the Increment (how)
>
> The Sprint Backlog is a plan by and for the Developers.
>
> — 2020 Scrum Guide page 11

The Developers on the Scrum Team completely control the Sprint Backlog.

The Product Owner cannot add to it and the Scrum Master can't delete anything from it.

The Developers completely control the Sprint Backlog.

Contrast that against the Product Backlog which is entirely controled by the Product Owner.

What does that leave the Scrum Master to control? Very little, other than a responsibility to guide the team and organization on how to properly apply the Scrum framework.

Test Yourself

The Sprint Backlog answers which three questions?

- ☐ When, where and why
- ☐ Who, when and where
- ☐ Why, what and how
- ☐ How, what and where

Option C is correct.

The Sprint Backlog answers the questions why, what and how.

Test Yourself

The CEO talks to you, the Scrum Master, and tells the project will be cancelled if a given feature is not implemented before the end of the Sprint. What should you as the Scrum Master do?

- ☐ Add the feature as a Sprint Backlog item in the current Sprint
- ☐ Cancel the Sprint and have the team work on this important feature

Nobody can add anything to the Sprint Backlog other than the developers.

In this case, the Scrum Master should have the Product Owner speak to the CEO.

The Product Owner could then speak with the developers and see if they could include the new feature in the Sprint Backlog without putting the Sprint Goal at risk.

> The Sprint Backlog is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.
>
> — 2020 Scrum Guide page 11

One of the pillars of Scrum is transparency.

The Sprint Backlog helps to create transparency by allowing everyone on the Scrum Team, along with all of the stakeholders, know exactly what's going on during a Sprint.

Management shouldn't have to pester the Scrum Team with status meeting and email updates because the Sprint Backlog should answer any questions about what is being developed and how things are moving along.

Test Yourself

When should the Sprint Backlog be updated?

☐ During Sprint planning when more details emerge

☐ During the Daily Scrum as the Scrum developers adapt

☐ During the Sprint review as stakeholders give feedback

☐ During the Sprint any time more is learned

Option D is correct.

The Sprint Backlog should be an up-to-date reflection of what is being done and what has been done.

It should be updated regularly.

Sprint Goal

> The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the

The Sprint Goal is the commitment the Sprint Backlog makes.

The Sprint Goal describes what the developers are trying to achieve in the current Sprint. It gives the team something to aim at.

Now let's make one thing clear - the development team doesn't always achieve the Sprint Goal, and that's okay.

In fact, some teams might not achieve the Sprint Goal the majority of the time. If that's a problem, them maybe it's something the team can talk about in the Sprint Retrospective. But at the same time, it might not actually be a problem.

What matters is that the Sprint Goal drives the team forward. So long an a usable increment of work is produced at the end of the Sprint and progress is made towards building the product, a missed goal isn't the end of the world.

Test Yourself

Issues beyond the Scrum Team's control have made almost impossible to achieve the Sprint Goal, and there's still a week left in the Sprint. What should the Scrum Master do?

☐ Change the Sprint Goal

☐ Cancel the Sprint

☐ Schedule a mid-Sprint planning session

☐ Have the team continue to work towards the goal

The Sprint can't be cancelled unless the Sprint Goal is obsolete, and that's rare.

And you're also not allowed to change the Sprint Goal once it's finalized.

The correct answer here is to just keep working towards the Sprint Goal.

It's not the end of the world if the Sprint Goal is not achieved. Sprints are short. You can create a new Sprint Goal when the current Sprint is finished.

It's not unusual for teams to overestiate how much work they can accomplish during a Sprint.

If developers need to decompose backlog items or even remove items from the Sprint Backlog, that's fine, so long as all of this happens without risking the Sprint Goal.

Test Yourself

If Developers find the scope of work scheduled for a Sprint is too much, with whom should they negotiate the Sprint Backlog's scope?

☐ The Product Onwer

☐ The Scrum Master

☐ Their fellow developers

☐ The stakeholders.

Any time the developers need to scale back on their work, or breakdown Product Backlog items into smaller pieces, it's always wise to speak with the Product Owner for clarification. The Product Owner can also help developers understand which product features should be included in order not to put the Sprint Goal at risk.

Chapter 17

The Increment

Scrum is an interative and incremental framework.

We see how Scrum is iterative. Scrum just does Sprint after Sprint after Sprint. The same events repeat over and over again.

But how is Scrum incremental?

Iterative and Incremental

In Scrum, we add to the product every Sprint. Every Sprint is supposed to deliver an Increment that is useful and valuable.

You never do a Sprint that doesn't try to create something.

There are no 'infrastructure Sprints' in Scrum where you just make sure everyone has the right software installed and access to the JIRA board.

In Scrum, every Sprint has to create something of value that adds to the existing product in a useful and meaningful way.

That's how Scrum is incremental. Every Sprint adds to what was created in the Sprints prior. At least that's the objective.

> An Increment is a concrete stepping stone toward the Product Goal.
>
> Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together.
>
> In order to provide value, the Increment must be usable.
>
> — 2020 Scrum Guide page 11

Integrated Increments

An increment must be add to what's already been done, and it must be able to fully integrate and work with all previous increments.

If an increment isn't compatible with past increments, then it's not a valid increment. If an increment stands alone and doesn't increase the value of the product built to date, then it's not a valid increment.

On the certification exam, there will be many questions about integration, which is confusing, because there is no talk in the Scrum Guide about integration.

- The are no integration Sprints in Scrum
- There are no integration teams in Scrum
- There are events where integration happens

However, whatever you create today must be able to 'integrate with' or 'work with' whatever you created yesterday for it to be a valid increment.

Understanding Integration

Some questions on the Scrum Master certification exam will ask you about integration. Software developers get tripped up on the word 'integration' because they get visions of merging code together or fixing Git conflicts or doing GitHub pull requests. Don't confuse yourself with those things. The idea of 'integration' is much simpler.

Just imagine you're a carpenter and you built a door for the house your team is building. The door is currently tied to the roof of your car. Is it integrated?

No, a door sitting on the roof of your car that supposed to go on the house your building is not integrated with the house if it's strapped to the roof of your car.

When the door is put on its hinges and attached to the house, it is now integrated. And, it becomes a useful, valuable increment. By putting the door on the house, you have integrated your work into the product, and you have added a valueble increment of work that stakeholders can see, touch, feel and even criticize.

A Poorly Integrated Increment

Now imagine you took the door off the roof of your car, tried to attach it to the house and:

- The door is too large
- The hinges are incompatible
- You realize the door is the wrong color

You've done work on the door, there's no arguing about that. But has that work been integrated?

Did you just create a usable increment of work that adds to the existing project? No, you didn't, because your door is useless.

When to Integrate

Let's go back to the super, amazing door you created. When should you attach it to the house?

The Scrum Master exam will often ask you when you should integrate your work into the existing project. People get all flustered with these questions, wonring when the best time is to integrate your work.

SO, when should you integrate your work into the existing product?

- During the Daily Scrum?
- During the Sprint Review?
- During the Sprint Retrospective?
- Right now?

Of course 'right now' is the right answer. If you've got the perfect door ready to be put on your new house's hinges, you don't sit two weeks waiting for the Sprint Review to do it. You integrate it right away. It would be nuts to imagine waiting two weeks to hand the front door on the house, or as we'd say in Scrum, 'integrate your work into the product.'

As soon as your work is done you integrate your work into the product. Then you move onto the next peice of work that needs to be done.

And you keep doing that over and over and over again until it's time for the Sprint Review. All of which reinforces the next part of the Scrum Guide.

Test Yourself

Which of the following is correct

☐ Scrum is incremental but not iterative

☐ Scrum is iterative but not incremental

☐ Scrum is both incremental and iterative

☐ Scrum is neither incremental nor iterative

Option C is correct.

Scrum is iterative in that a set of steps get repeated each sprint, and it is incremental in that each Sprint produces an increment of work that gets added to all past increments.

Multiple Increments

The goal of each Sprint is to get at least one valuable and usable Increment added to the project. But that shouldn't stop you from integrating as many Increments as you can.

If a carpented adds a front door to a house, she doesn't wait until the next Sprint to start working on the back door. She gets working on the back door and integrates that into the house as soon she's done with that. Increments should keep being added to the product constantly throughout the Sprint.

> Multiple Increments may be created within a Sprint.
>
> The sum of the Increments is presented at the Sprint Review thus supporting empiricism.
>
> However, an Increment may be delivered to stakeholders prior to the end of the Sprint.
>
> The Sprint Review should never be considered a gate to releasing value.
>
> Work cannot be considered part of an Increment unless it meets the Definition of Done.
>
> — 2020 Scrum Guide page 12

Continuous Delivery of Valuable Increments

Some people get the impression that the only time stakeholders should be allowed to see what developers have done is at the Sprint Review.

If the developers create an increment they want the stakeholders to see, there is nothing stopping them from presenting it to them at any time throughout the Sprint.

And why would you wait? If the developers have created something they are proud of, and the stakeholders are anxious to see it, why would the development team wait until the Sprint Review to show it to them?

The Sprint Review makes sure the stakeholders get to see progress at least one time throughout a Sprint, but that just ensures a minimum. Developers are always encouraged to be transparent and share their progress with interested stakeholders.

Commitment: Definition of Done

> The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.
>
> The moment a Product Backlog item meets the Definition of Done, an Increment is born.
>
> — 2020 Scrum Guide page 12

The definition of done must be clear and understood by everyone on the Scrum Team and by the stakeholders and the organization. It creates a common understanding about what has to be done in order for a feature to be complete.

Test Yourself

True or false: The Definition of Done is an informal understanding of what needs to be done to turn a Product Backlog item into an Increment.

This is false

The Definition of Done is a formal description, not an informal understanding.

Test Yourself

The application has failed to scale beyond a single processing core, and this has causes various production issues. What is the best way for the Scrum Team to proceed with this issue.

☐ Assign multi-core processing support to the DevOps team

☐ Create a testing team to implement multi-processor support

☐ Add multi-processor support to the Definition of Done

Option C is correct.

If there is a quality metric that the product must support, it is the development team's responsiblitity to support it. If the requirement is added to the Definition of Done, the developers will not be allowed to release or integrate any increments that don't meet the multi-processor requirement.

Remember that there are no sub-teams in Scrum, and there is no talk anywhere about DevOps, UAT or QA teams anywhere in the Scrum Guide.

Scrum Pillars, Transparency and 'Done'

> The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment.
>
> If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review.
>
> Instead, it returns to the Product Backlog for future consideration.
>
> — 2020 Scrum Guide page 12

Test takers often killed on this point, so pay attention.

When a Product Backlog item is not completed during a Sprint, it is put back ito the Product Backlog.

- It's not presented at the Sprint Review
- It is not 'partially presented' at the Sprint Review
- Partial points are not assigned to it (There are no 'points' in Scrum!)
- The feature is not automatically added to the next Sprint's Sprint Backlog
- The feature is not partially released

If a Product Backlog item selected for the Sprint does not meet the Definition of Done by the end of the Sprint, it is thrown back into the Product Backlog as though nobody had ever even been working on it. Even if it's 90% complete, it's thrown back into the Product Backlog for the Product Owner to reprioritize.

An important, critical feature is 99% done at the time of the Sprint Review. What should the team do?

☐ Present the work completed for stakeholders to see at the Sprint Review

☐ Assign 99% of the points to the current Sprint and assign 1% of the points to the next Sprint

    ☐ Automatically add the Product Backlog item to the next Sprint's Sprint Backlog

    ☐ Extend the Sprint by an extra day to allow the developer to get the feature to 100%

    ☐ Return the item to the Product Backlog and do not present it at the Sprint Review

Option E is correct.

The Scrum Guide is clear. If a feature does not meet the definition of done, it is not presented at the Sprint Review, and it is returned to the Product Backlog.

Organizational Standards

> If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum.
>
> If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.
>
> — 2020 Scrum Guide page 12

Who creates the Definition of Done?

It's created by the Scrum Team, unless there is already an existing orginzational standard.

If the Scrum Team wants to create a more rigorous Definition of Done, they are certainly allowed to do that, but under no circumstances

Done Over Time

One of the interesting things that happens with Scrum Teams as they become more familiar with the framework and the product they are building is they extend and enhance their Definition of Done. Over time, the Definition will likely become more rigorous, with more quality gates to pass than when the team first adopted Scrum.

Who creates the Definition of Done?

    ☐ The Product Owner if there is no organizational standard

    ☐ The Scrum Master if there is no organizational standard

    ☐ The Scrum Developers if there is no organizational standard

    ☐ The stakeholders if there is no organizational standard

Option C is correct.

The Scrum Team creates a Definition of Done if an organizational standard does not exist.

> The Developers are required to conform to the Definition of Done.
>
> If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

— 2020 Scrum Guide page 12

There are only a few instances in the Scrum Guide where it discusses multiple teams working on the same project.

The rules the Scrum Guide requires for multiple teams working on the same project are few, but you will be tested on all of them. Those rules demand that multiple Scrum Teams that work on the same project must:

- Each share the same Product Owner
- Each share the same Product Backlog
- Each share the same Product Goal
- Each share the same Definition of Done

Also, teams may invite members from other teams to their Sprint Planning meeting.

What Scrum Doesn't Say About Multiple Teams

There is nothing in the Scrum Guide that asserts multiple teams working on the same project must:

- Start their Sprints at the same time
- Have Sprints that are the same length
- Have the same number of team members
- Have the same Scrum Master

There will be a number of questions about multiple teams working on the same project on the certification exam, and those questions will provide options that seem reasonable but are incorrect.

When answering questions in the the certification exam, focus on answers that map as closely as possible to what is in the Scrum Guide. Don't hunt for pragmatic answers. Hunt for correct answers.

What must shared between multiple teams working on the same product?

☐ The Product Owner
☐ The Scrum Master
☐ The Sprint Backlog
☐ The Product Backlog

Options A and D are correct.

Each Scrum team has its own Sprint Backlog.

A Scrum Master can be shared between multiple teams, or a Scrum master can dedicate 100% of their time to one team. the Scrum Guide guide doesn't advise against either scenario.

When multiple teams work on the same project together:

- Each team shares the same Product Owner
- Each team share the same Product Backlog
- Each team shares the same Product Goal
- Each team shares the same Definition of Done

//// include::focused/fq-artifacts.adoc[]

//// include::focused/fq-coaching.adoc[]

//// include::focused/fq-definition-of-done.adoc[]

//// include::focused/fq-empiricism.adoc[]

//// include::focused/fq-events.adoc[]

//// include::focused/fq-facilitation.adoc[]

//// include::focused/fq-forecasting-and-releases.adoc[]

//// include::focused/fq-product-backlog.adoc[]

//// include::focused/fq-product-value.adoc[]

//// include::focused/fq-scrum-team.adoc[]

//// include::focused/fq-self-management.adoc[]

//// include::focused/fq-stakeholders.adoc[]

//// include::focused/fq-stakeholders.adoc[]

//// include::focused/fq-stakeholders.adoc[]

//// include::final-exam-1.adoc[]

//// include::final-exam-2.adoc[]