Scrum for Everyone

CERTIFICATION GUIDE

by Darcy Declute
@scrumtuous

Editors
Prachi Shaw
Amanda Punsammy

XENNIAL INNOVATIONS

Scrum Master Certification Guide

A note from the author

Hi! I'm Darcy DeClute.

As I write this note I'm filled with a mix of conflicting and confusing emotions that range from humility to pride to embarrassment.

It humbles me to know you're reading this book.

If you're reading a book about certification, you're obviously interested in acquiring a Scrum Master accreditation, which means you're a highly-motivated professional intent on learning, resume building and career advancement.

And you've put your faith in *me* to guide you on a learning journey. That's humbling.

As overwhelming as it is to know you've put your faith in me, I'm equally proud of the book I've written here.

I think this book delivers on what it promises. I think this book will play a key role in getting you past the post when it comes time to sit for the Scrum Master certification exam.

Figure 1. At home on native land. Baffin Island, Canada

A number of students have read this book and then passed the Scrum Master exam, with a few even garnering a perfect score. I know this book played a key role in their success and I'm confident it will play a key role in your success as well. That makes me beam with pride.

So many people to thank…

I'm also embarrassed when I see this book with my name on the cover, because I feel like an imposter taking the title credit.

When I mentioned to some of my peers and mentors that I wanted to write a Scrum Master certification guide to fill what I thought was a void in the market, my friends encouraged and inspired me.

Some people I follow on Twitter, people like Walker Boh, graciously donated their own free time to proofread, edit, criticize and add corrections. Other peers and mentors added paragraphs and extended the text when they thought a chapter was light or a topic needed clarification.

Some friends helped to generate images with MidJourney prompts and fixed issues with formatting.

So when I see my name on the front cover of this book, I feel as though I'm stealing credit, because this was completely a team effort.

I can't thank the people who helped me get this book past the finish line enough. I love all of you.

I'm @Scrumtuous everywhere!

This book is self-published.

There's an old saying that asserts the biggest problem with self-published books is the person who publishes them. Having read some really low-quality, self-published tripe, I don't completely disagree with that sentiment.

However, since I do self-publish, I have the benefit of being able to make updates and changes at a moments notice.

If you do find something in this book that is unclear, missspellled, confusing or even incorrect, please find me on Twitter or LinkedIn and let me know.

I'm @Scrumtuous everywhere.

Figure 2. Please come find me on Twitter. I'm @Scrumtuous on all platforms!

Or better yet, just email me:

darcydeclute@gmail.com

As I mentioned before, people have put a lot of faith in me to help them get certified and drive their career forward. I want to do everything I can to make that possible. Any help you can provide would be appreciated not only by me, but by everyone who reads this book in the future.

Oh, and one last thing. If you do enjoy the book, please leave a 5-star review on Amazon. It would not only put a giant smile on my face, but it would also help to get this book in front of other people who need it.

Thanks again for reading this book. I really hope you enjoy it.

Love Darcy.

Foreword

I felt a little nervous when Darcy asked me to write this foreword.

You see, I'm not the biggest Scrum Master cheerleader in the world.

In fact, in many of the Agile transformations I've led, one of our success metrics is how close we come to making the Scrum Master completely unnecessary.

That's not to say I'm against Scrum. I believe Scrum is the right Agile framework to use when starting a new project.

The role of the Scrum Master

Figure 3. Sal Pece, Xennial CEO

I advocate for Scrum, but I also advocate for teams to evolve to such an advanced level of Agile understanding that they don't really need a Scrum Master.

The role of the Scrum Master is to teach, coach, and mentor others in the ways of Scrum, while acting as a leader who serves as the team goes through an Agile transition.

But here's the thing about transitions: they eventually reach completion.

When an organization truly achieves Agility, the need for a Scrum Master to coach them on cross-functional teams, self-management, and the importance of reviews and retrospectives progressively diminishes. These Agile practices eventually become second nature.

Successful Agile teams don't need coaching on these practices because they naturally incorporate them into their work. It becomes odd for them not to.

Measuring Agile Transformation Success

That's when you know your Agile transformation is a success. That's when you know the team has approached Agile enlightenment.

Of course, you can't do Scrum without a Scrum Master. Scrum without a Scrum Master isn't Scrum.

However, over time, a team's dependence on the Scrum Master should diminish.

At the beginning of an Agile transformation, each development team might require its own dedicated Scrum Master.

Well-coached teams will rely less and less on the Scrum Master for leadership. As Agile transformations succeed, a Scrum Master will divide their time across multiple teams because

individual teams become less dependent on their guidance.

Once enlightenment is achieved, a single, underutilized Scrum Master will be shared across multiple teams throughout the organization.

There's a recurring meme in Agile circles that jokes "never ask a Scrum Master what they do, because you'll find out that they don't do much."

I love that meme because it's not only funny but because it should be true if your organization has implemented Scrum correctly. That meme is basically my goal during an Agile transformation.

Keeping Scrum Working

On teams where developers and the Product Owner truly understand how Scrum works, the role of the Scrum Master becomes almost ceremonial. If your teams are building amazing products while the Scrum Master is searching for things to do, you're probably implementing Scrum exceptionally well.

And that's why I was nervous about reviewing this book and contributing a foreword.

I was concerned that this Scrum Master Certification Guide would, like most other books about Scrum, overstate the role of the Scrum Master.

But my nervousness quickly turned into enthusiasm as I turned page after page of this book.

As I read through the text, I felt Darcy was reflecting my thoughts about effective Scrum directly back to me.

Miss DeClute strikes just the right balance between recognizing the importance of the Scrum Master's role within an organization and emphasizing that with highly motivated, self-managed teams, reliance on the Scrum Master accountability should gradually diminish over time.

Darcy gets it right.

Enjoy this certification guide.

It will not only help you pass the Scrum Master certification exam but it will also provide you with a strong understanding of how Scrum *should* work once an enterprise successfully completes an Agile transformation.

Sal Pece
CEO, Xennial Innovations Inc.

When the student is ready, the teacher will appear.
Tao Te Ching

Chapter 0: Getting Started

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

The process is going to go something like this:

- We're going to go through the entire 2020 Scrum Guide together, one line at a time.
- Together we'll extract as much meaning as possible out of the Scrum Guide's words.
- We'll discuss how ideas in the Scrum Guide translate into exam questions.
- We'll 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 exam.

Figure 4. Download and read the official Scrum Guide a few times before reading this book. Read it a few times after reading this book too. The Scrum Guide is the sole source of truth for any Scrum Certification exam.

Forget Everything You Know

If you've ever worked with Scrum in a professional setting and you want to ace the Scrum Master certification exam, the first thing you need to do is forget everything you think you know about Scrum.

- Forget about story points.
- Forget about use cases.
- Forget about poker planning.

These concepts have nothing to do with Scrum. They are never mentioned in the Scrum Guide, and references to them on the certification exam are only there to distract you and trick unprepared test-takers into selecting incorrect answers.

On the Professional Scrum Master certification exam, you'll be tested almost exclusively on concepts within the Scrum Guide. If it's not in the Scrum Guide, it's not a correct answer on the exam.

The definitive source of truth regarding Scrum comes exclusively from the Scrum Guide. That's what we'll cover in-depth during this guide.

But before we get into the nuanced specifics, I wanted to provide my own, unofficial take on what Scrum is. This is how I describe Scrum to people who ask me about it. I think it's a good overview. It lays the groundwork for future, more technical discussions.

So what is Scrum?

Scrum is a lightweight, purposefully incomplete framework that helps development teams get started and keep going.

- Does someone have a product vision?
- Do you have a team of developers who are anxious to get building?
- Are stakeholders champing at the bit?

If that's the case, Scrum says you should start a development Sprint right now.

- Don't waste a lot of time planning because your plans are going to change.
- Don't waste a lot of time designing because your initial designs might not work.

Instead, start developing. Start a development Sprint now.

A development Sprint is a short period of time in which developers focus on building a few important features that will become part of the final product.

A development Sprint can never last more than a month. Make them shorter if you want to minimize risk.

Do a little planning. Do a lot of building.

As I said before, teams shouldn't waste a lot of time planning. However, Scrum doesn't say teams shouldn't plan at all. The very first thing a Scrum Team does when their development Sprint starts is plan.

Not too much time should be spent planning. Never more than 8 hours, and never plan too far into the future. The further you look into the future, the wider the cone of uncertainty becomes. That's why short Sprints are often better than long ones.

During Sprint Planning, the developers talk to the Product Owner about what to build. The Product Owner is the person on the Scrum Team who has the product vision and knows what stakeholders want.

The Product Owner also has an ordered list of features that need to get developed. That's called the Product Backlog.

During planning, Scrum Developers select Product Backlog items they think they can complete before the end of the Sprint.

At the end of Sprint Planning, the developers walk away with:

- A list of features they are going to work hard to complete.
- A plan to get those features finished.
- An overarching Sprint Goal that keeps the team focused.

Then the work begins!

Every workday, the developers meet for a few minutes in something called a Daily Scrum.

- It's not a standup meeting.
- It's not a status meeting.
- The Product Owner and the Scrum Master aren't even supposed to participate.

It's just a chance for the developers to quickly catch up and get the ball rolling again, nothing more.

Incremental Progress Over Time

Every Sprint has to create something useful.

Every Sprint must create something that will become part of the final product.

In Scrum, we call these useful, additive pieces Increments. Products get built incrementally.

At the end of the Sprint, there's a Sprint Review where the Scrum Team gives stakeholders a chance to play around with the new features.

Stakeholders are only shown features that are done - not partially done or almost done, but done.

Technically speaking, stakeholders are shown features that meet the "Definition of Done," which is a quality standard everyone on the project has agreed upon.

Review and Reflect

Product reviews at the end of every Sprint ensure stakeholders get to see what's being developed at least once every month. It keeps everyone in the loop. Feedback loops are an essential part of Scrum.

After the Sprint Review, the Scrum Team holds a Sprint Retrospective where they discuss what worked and what they need to do better.

When the Sprint Retrospective ends, a new Sprint begins, which means another short planning meeting, Daily Scrums, and finally a Sprint Review and Retrospective.

That's why Scrum is said to be iterative. You just keep doing Sprints until the work is done.

And that's Scrum.

It's a simple framework that says:

- Plan, but don't plan too much.
- Work hard at a sustainable pace to develop something usable and valuable.
- Regularly review your work with stakeholders.
- Regularly take time to talk about team dynamics and how to improve things.

And all of this is accomplished by a team that is made up of:

- One Product Owner who defines the Product Goal and manages the Product Backlog.
- The developers who know how to build the product.
- One Scrum Master, who doesn't do much.

The Scrum Master

As Sal Pece said in the foreword, that's sort of an ongoing joke in Scrum - that the Scrum Master doesn't do anything. But the fact is, on a really well-run team, a Scrum Master doesn't have much to do.

The Scrum Master just makes sure Scrum is applied properly.

- They don't manage the team.
- They don't manage the project.
- They don't schedule Zoom calls.

- They don't book conference rooms.
- They don't manage finances.
- They don't update JIRA tickets.

They just coach teams and organizations on how to properly apply the Scrum framework while behaving as a leader who serves the team. That's the Scrum Master's job.

Scrum describes itself as a simple, lean, incomplete framework, and that's a good description.

Scrum just describes the best practices any team should be doing if they want to work efficiently and effectively.

It may not work for everyone, but everyone should at least give it a try.

Get The Scrum Guide

That's my quick description of Scrum, and I think it's a good one.

Having said that, the Scrum Master certification exam doesn't test you on what I think Scrum is all about, it tests you on what the Scrum Guide says Scrum is all about, which is why I want you to download and print out a couple of copies of the 2020 Scrum Guide.

The Scrum Guide is only 14 pages long, and that includes the cover page, introduction and the table of contents. You won't be destroying a forest by having a couple of printed copies by your side.

Figure 5. Try to hit as many learning modalities as you can whenever you learn something new.

Grab a highlighter as well and markup that printed copy as we go along. It'll help you learn.

Read the Scrum Guide

Along with downloading it, you should also *read* the Scrum Guide before you dig into the first chapter of this book. In fact, read it a couple of times. That way you'll have a better idea of what phrases like *the Sprint Goal* and *a usable Increment* mean.

With that foundation, we'll be able to dig much deeper into the significance of these interesting and important terms.

Now let's get started with the Scrum Guide!

Chapter 1: What is Scrum?

How would you define Scrum in a single sentence?

To be successful on the Scrum Master certification exam you have to commit to the Scrum Guide's definition of Scrum, which means abandoning the biases and misconceptions you may have adopted over years of 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 100% commit to right now if you want to pass the Scrum Master 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 Agile development, how would you rate this definition?

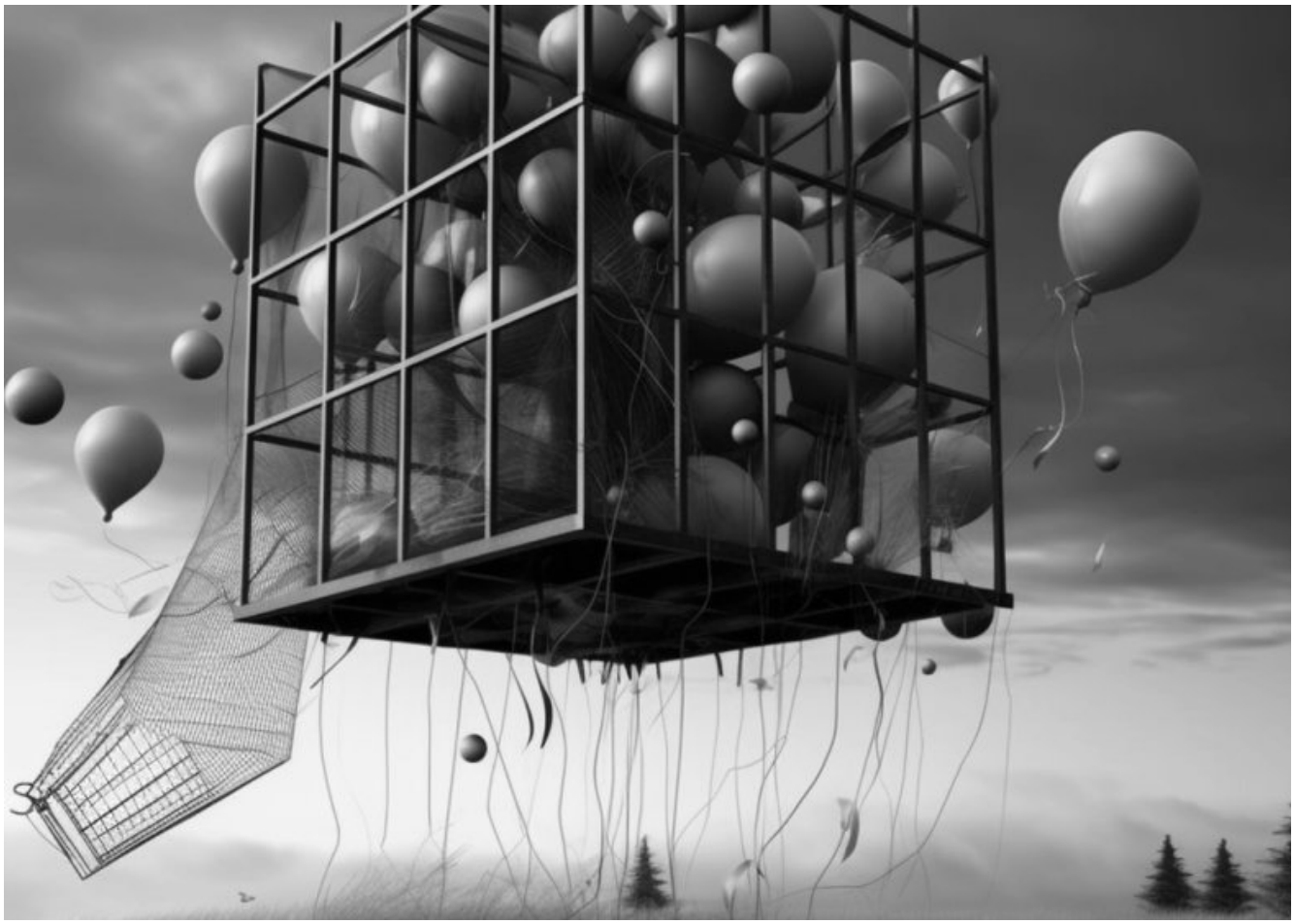


Figure 6. Scrum is a lightweight, incomplete framework that helps teams solve complex problems as they work towards a Product Goal. Scrum advocates downplay the terms 'process' and 'methodology.'

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 insight.

- 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, so commit to it.

The Words Not Spoken

The Scrum Guide say a lot in its brief 14 pages, but sometimes it's more interesting to focus on what the Scrum Guide *doesn't* say. Notice how:

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

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

Test Yourself

Scrum is a proven software development process.

- ▢ True
- ▢ False

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 doesn't try to solve all of your project management problems. It just helps you to get started, to keep going and to minimize risks along the way.

NOTE	While the authors of the Scrum Guide both signed the Agile Manifesto, the word *Agile* never appears once in the 2020 Scrum Guide.

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:

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

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

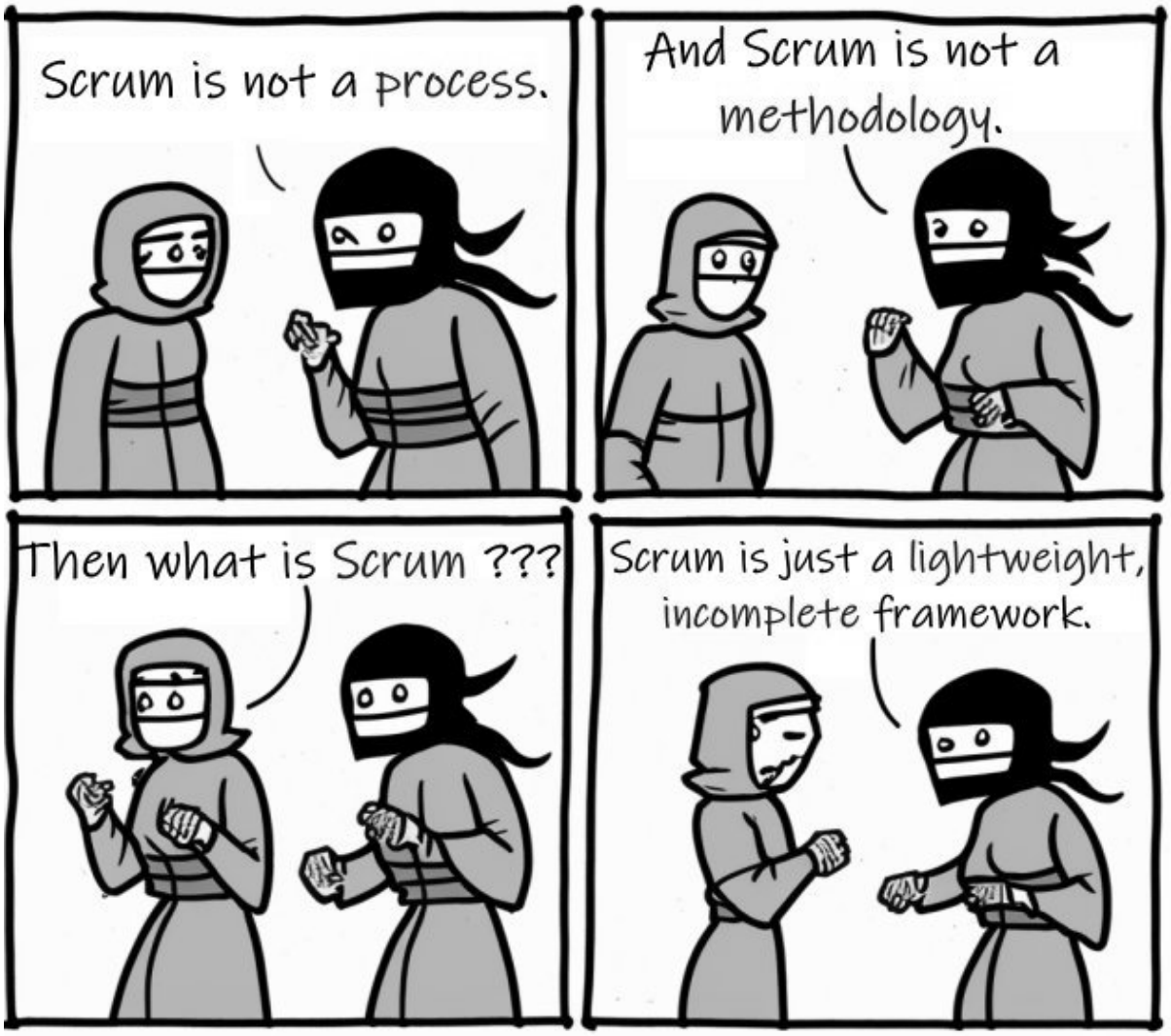
There is a big push in the Scrum community to gain acceptance outside of software development. Any certification questions that pigeonhole Scrum into a software development box will be wrong.

Test Yourself

> Which one of the following statements most accurately reflects the definition of Scrum?
>
> - ▢ A) Scrum is a software development methodology
> - ▢ B) Scrum is an Agile process for teams and organizations to follow
> - ▢ C) Scrum is a lightweight framework to help teams tackle complex problems
> - ▢ D) Scrum is a lightweight framework to help teams and organizations build software

Option C is correct.

Scrum is a "lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems." References to Scrum being a *methodology* or a *process* will always be a wrong option on the Scrum Certification exam. It's not just about software either.



Scrum and the Scrum Master

Here's the Scrum Guide's high-level overview of how Scrum is supposed to work, along with a quick note about the role the Scrum Master plays in keeping Scrum working.

> 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

Don’t Overstate the Role of the Scrum Master

The name 'Scrum Master' sounds intimidating and authoritative.

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

They don’t. The Scrum Master controls very little. In fact, the word 'control' is antithetical to what the Scrum Master is all about.

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

Scrum is Simple, Pragmatic and Lean

People tend to overthink 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

Test Yourself

Which of the following statements are true about the Scrum framework?
(Choose 2)

- ▢ A) Scrum describes an iterative process
- ▢ B) Scrum is an iterative framework
- ▢ C) Scrum generates value by repeatedly delivering usable increments to the stakeholders
- ▢ D) Scrum only allows stakeholders to inspect progress when the final product is delivered

Options B and C are correct.

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

Scrum is also 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 a long development cycle. So Option C is correct while Option D is wrong.

It's a Guide. It's 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, saying that it promises *not* to 'provide people with detailed instructions.'

There are very few actual rules in the 14-page Scrum Guide. Outside of the few rules Scrum does have, the framework encourages people to discover strategies that work best for them.

TIP It often helps to think about certification exam questions outside of the domain of software development. Reframe an exam question in terms of an isolated team building a shelter on a deserted island and answers may become more clear.

Test Yourself

Scrum is a complete and proven framework that helps teams achieve goals and create value.

- ▢ True
- ▢ False

This is false. Scrum self-identifies as a *purposefully 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 for the Scrum Master certification exam, all you need to know is that it's an alternate development strategy.

Test Yourself

Scrum can be used alongside various other methodologies including Kanban and Lean.

- ▢ True
- ▢ False

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.

TIP	Waterfall gets its name from the fact that isolated development phases like planning and design flow into each other, in only one direction, just like water in a waterfall.

Test Yourself

When implemented properly, Scrum has the capacity to expose ineffective management.

- ▢ True
- ▢ False

This is true.

The iterative and incremental nature of Scrum, where constant inspection and adaptation is encouraged, will shine a light on ineffective practices that happen external to the Scrum Team.

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."

And with that question answered, we’re done with the definition of Scrum.

Now on to a little overview of what Scrum theory is and what it’s based on.

Chapter 2: 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

Rationalists vs Empiricists

Rationalists say knowledge is best derived through reason, logic and deduction. Mathematics is a very *rational* pursuit.

Empiricists say evidence acquired through experience and experimentation is a better way to solve a puzzling problem.

Pragmatists sit somewhere in the middle and can't make up their mind as to which approach is better.

I actually consider myself a bit of a pragmatist, but when it comes to solving complex problems, Scrum wants nothing to do with wish-washy pragmatism. :)

Scrum is founded upon the concepts of *empiricism* and *lean thinking*.

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

Empiricism also ties in tightly with the Scrum Pillars of transparency, inspection, and adaptation. That is, if you can see exactly what is happening (transparency), you can honestly assess your progress (inspection) and based on this real knowledge, you can adapt.

Empirical analysis of your current situation is always a more effective way to approach a complex problem than following a plan written up six months ago, or making decisions based on trendlines on a historical chart.

Empirical Example

Here's a simple example of empiricism to drive the concept home for you.

Imagine you wanted to know how long it would take for a hockey puck to fall to the ground when dropped from the top of Toronto's CN Tower.

One approach would be to grab a pen and paper, look up the height of the CN Tower, grab Newton's equation for gravity, factor in wind resistance and use the power of math to determine a result.

That's a good approach, but it's very rational. It's not at all empirical.

Scrum Teams place the greatest amount of value on empirical results.

To answer this question, a Scrum Team would have their developers each channel their inner *Spiderman* and climb to the top of the CN Tower with a bag of pucks.

They'd drop the pucks from the tippy top and use a timer to see how long it takes for the black, vulcanized rubber to hit the ground.

The developers would then analyze the results of their experiment and come up with a result that was based on what they observed. Making decisions based on what you observe through your six senses is what empiricism is all about.

In Scrum, we want to base our decisions on empirical observations. That's not to say Scrum developers aren't rational or pragmatic.

In fact, taking a pragmatic approach to many question on the Scrum Master exam will help you get closer to achieving a perfect score on the test. But when given the choice to base an important decision based on rational, pragmatic or empirical analysis, Scrum wants you to prioritize empiricism every time.

Lean Thinking

Lean thinking is also a pivotally important *concept* to master if you want to pass any Scrum exam. You don't have to be an expert on *lean manufacturing* practices, but you do have to master the art of *lean thinking*.

The idea of lean thinking comes from productivity enhancements Toyota discovered in the 1950s and 60s. Obviously Toyota was building cars, but the lessons of lean thinking are universal to all product development domains.

Key concepts of lean manufacturing that manifest themselves in the Scrum framework include:

- The idea that all unnecessary steps in a process should be eliminated
- The belief that all work should be tied tightly to a specific outcome or goal
- The insistence that products should not be over-engineered with unnecessary features
- The need for a direct connection with the customer or stakeholder
- The insistence that teams have strong leaders whose priority is to serve their team
- The desire for workers not to be left idle during the production process
- The fact that improvements should be made in accordance with empirical observations

Empiricism and Lean Thinking

Just looking at this list of lean concepts you can see how:

- The Scrum Master role maps to the lean demand that teams have strong leaders
- The Sprint and Product Goals map to the idea that work should be tied to a specific outcome
- Scrum's Sprint Review embraces the ideas of maintaining a tight connection with the stakeholder

Avoid waste and embrace minimalist thinking when you're working on a Scrum project. That's *lean thinking.*

You are guaranteed to get five or six questions about empiricism and lean thinking on the Scrum Master certification exam. They are easy marks if you just understand what 'empiricism' and 'lean thinking' means.

Test Yourself

Scrum theory emphasizes empiricism. That means a decision made by Scrum teams:
(Choose 3)

- ▢ A) Should be based on factual evidence
- ▢ B) Should be based on intelligent assumptions
- ▢ C) Should be based on verifiable observations
- ▢ D) Should be driven by pragmatic speculation
- ▢ E) Should be based on experience

Options A, C, and E 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."
>
> — TechTarget WhatIs Definition, Empiricism

Test Yourself

Which of the following are characteristics of lean thinking?
(Choose 2)

- ▢ A) A focus on waste reduction and efficiency
- ▢ B) A command and control approach to problem-solving
- ▢ C) A minimalist approach that removes unnecessary steps
- ▢ D) A methodology 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.

The 'Command and Control' approach that the military takes is the opposite of the collaborative, lean approach Scrum takes to decision making, so option B is incorrect.

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 Master 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 that get repeated over and over again. The iterative sequence of steps as outlined in the first section of the Scrum Guide are:

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

You just keep repeating this sequence of steps ad infinitum until you either run out of Product Backlog items to work on or the project comes to an end.

The Incremental Nature of Scrum

Scrum is incremental.

> "The Scrum Team turns a selection of the work into an Increment of value during a Sprint."
>
> — 2020 Scrum Guide page 3

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 finished.

Piece by piece, through the delivery of value added upon value, the product gets built. That's the incremental process.

By getting these small increments into the hands of stakeholders, and getting immediate feedback from which the Scrum Team can adapt, Scrum reduces risk and allows developers to better service the needs of their clients.

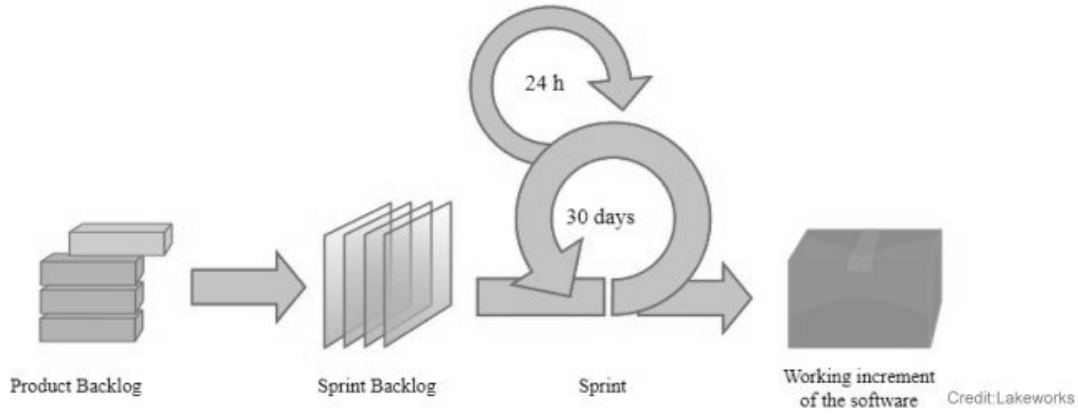


Figure 7. Scrum builds products incrementally through a set of steps that are iteratively repeated throughout the product development lifecycle.(Credit Lakeworks)

Cross-Functional Teams

The idea that teams should be cross-functional and self-managed is a key concept in Scrum, and it's one that you'll 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 towards 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 someone to document the product? That person is on the Scrum team.
- Does your project need an architect? That person is on the Scrum team.
- Does your project need people to do quality assurance(QA)? Then those QA people are developers 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 your team acquires them.

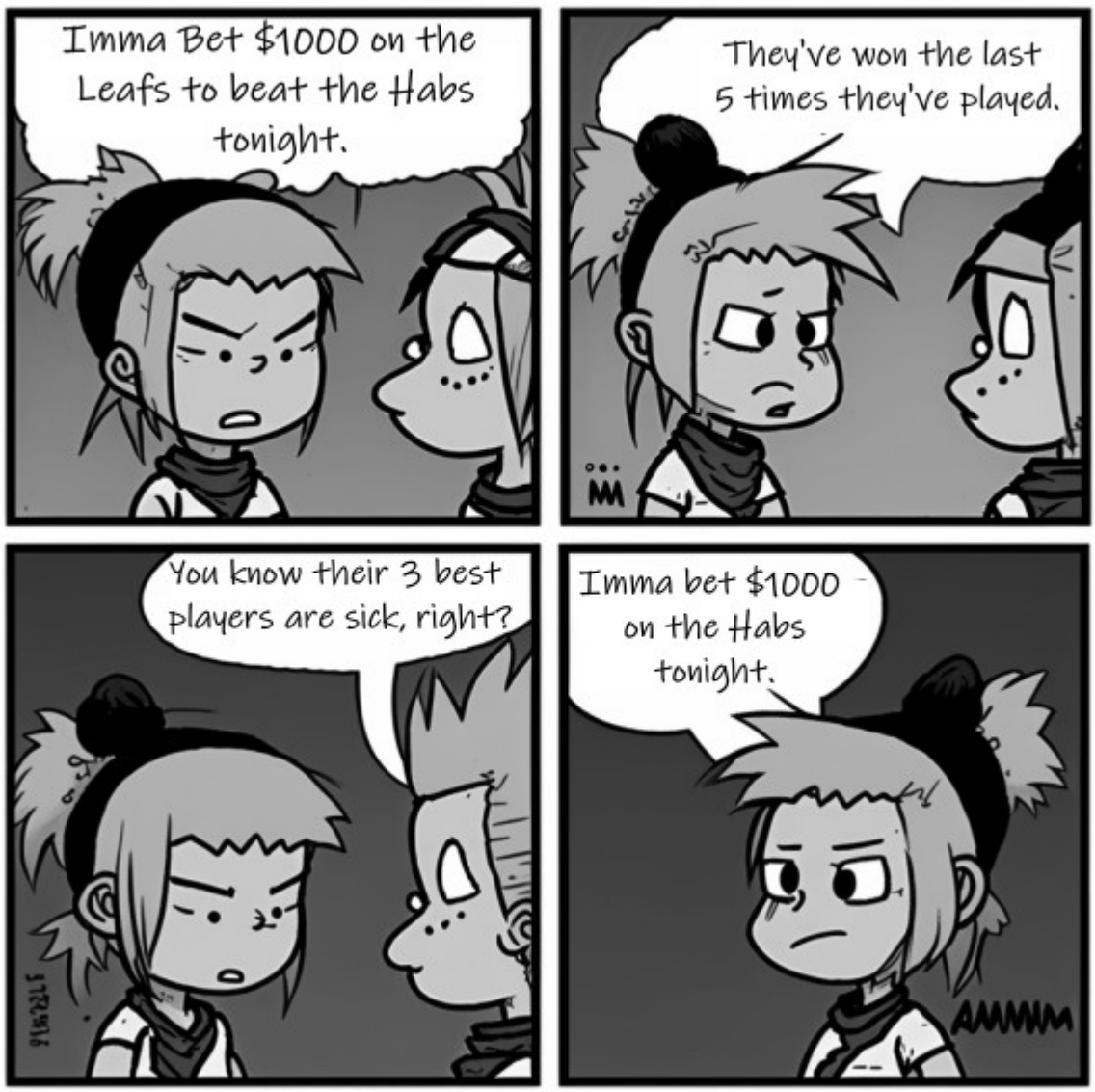


Figure 8. Predictive tools are useful, but they don't replace empirical knowledge acquired through experience and experimentation. Knowledge of pertinent facts should always take priority over charts and trendlines.

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.

Which one of the following options is the best way for the team to move forward?

- ▢ A) Outsource the development of a spaceship to a third party
- ▢ B) Remove the development of a spaceship from the project's requirements
- ▢ C) Explain to the Product Owner that you don't have the skills to build a Mars spaceship
- ▢ D) Get the team to start learning about how to build a spaceship to Mars

Option D is correct, and yes, this question is silly to the extreme, but it makes an important 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's 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?
- When a Sprint ends, when does the next Sprint begin?

Scrum has four timeboxed 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.

Moving from Sprint to 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 things have to happen during the Sprint.

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

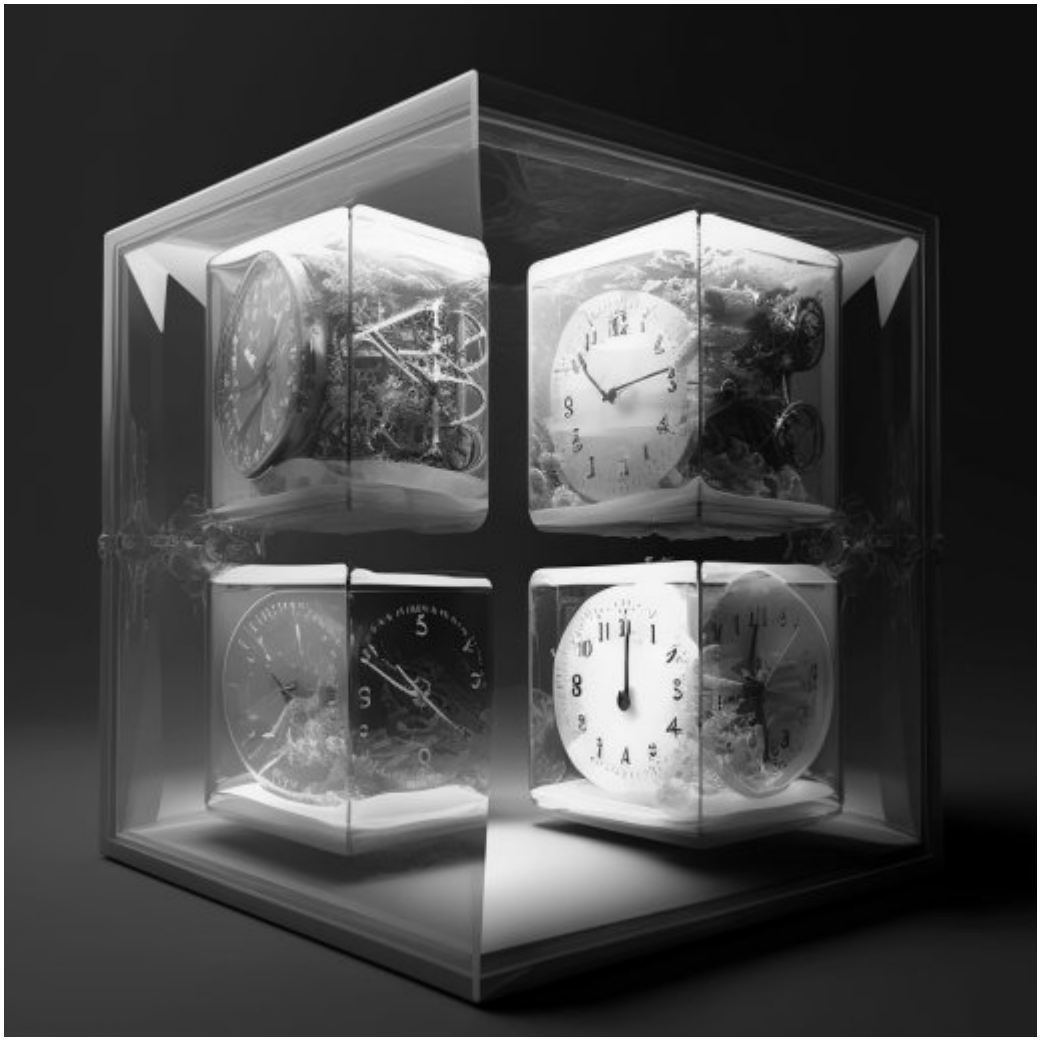


Figure 9. The four timeboxed events in Scrum all occur inside a fifth event known as the Sprint.

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 listed options include the terms 'inspect' or 'adapt', those will likely be the correct answers.

Test Yourself

Which one of the following options best describes when a new Sprint begins?

- ☐ A) A new Sprint begins when Sprint Planning is completed
- ☐ B) A new Sprint begins when the Sprint Review is completed
- ☐ C) A new Sprint begins when the Product Owner begins the Sprint in JIRA
- ☐ D) A new Sprint begins after Sprint Planning when the Scrum Master declares the start of the Sprint
- ☑ E) A new Sprint begins as soon as the previous Sprint ends

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

Technically speaking, the last event in the Sprint is the Sprint Retrospective, and the end of this event officially marks the end of the Sprint. A new Sprint could be said to start immediately after this event concludes, depending upon how pedantic you want to be about the answer.

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

Test Yourself

What is the purpose of the Daily Scrum?
(Choose 2)

- ☐ A) For the Scrum Master to get daily status updates from the developers
- ☐ B) To allow the developers to inspect their progress toward the Sprint Goal
- ☐ C) For the Product Owner to track the development team’s progress on Product Backlog items
- ☐ D) 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.

Test Yourself

Which of the following concepts are consistent with a lean approach to product development?
(Choose 3)

- ▢ A) The elimination of unnecessary steps in a process
- ▢ B) Isolated development phases that flow into each other
- ▢ C) The removal of unnecessary features in a product
- ▢ D) Significant up-front planning and design
- ▢ E) The elimination of idle time where developers are not working
- ▢ F) Waiting for a manager to give you permission to proceed with a task
- ▢ G) Waiting for a planned event to discuss an important matter

Options A, C and are E are correct.

These three options are core concepts learned from *lean manufacturing*.

Options B and D represent the waterfall model, which is the antithesis of Scrum and lean.

Forget Everything You Think You Know

Part of passing the Scrum Master certification exam is to first forget everything you *think* you know about Scrum, and then focus on the core concepts contained within the Scrum Guide.

That's why you won't hear me talking about the *Agile Manifesto* or discussing *story points* in this book. Other Scrum Master certification books invest a lot of time in those topics, but I think that does you a disservice.

The Scrum Guide never once mentions the word *Agile* and it never talks about *user stories* or *story points*.

Furthermore, the Scrum Master exam never tests you on these concepts, and whenever a potential answer references a *user story* or a *story point*, those answers will be red herrings trying to trip you up and get you to select an incorrect option.

That's also why you'll often find me repeating myself ad nauseam about certain topics throughout the book. For example:

- I will annoyingly repeat myself about the time allotted to the various Scrum events.
- I constantly emphasize that you don't have to wait for a Scrum event to change your plan.
- Scrum is a framework, it's not a process or a methodology.
- Scrum is not just for software development but for the development of *any product*.

My goal is to get you Scrum Master certified. A *lean approach* to doing so is to not waste your time discussing peripheral topics that will distract and confuse you.

Keep the Focus on the Goal

One of the Scrum values is focus.

Imma try my very best to keep you focused, not waste your time on topics that aren't heavily tested on the exam, and at the same time, constantly re-inforce topics that I know will be tested heavily.

That's how we're going to get you past the Scrum Master certification finish line!

Figure 10. With lean thinking and focus, we're going to get you past the Scrum Master certification finish line. (Image: Joshbdork, GFDL1.2)

Chapter 3: Scrum Pillars and Values

'Scrum values' is one of the 13 categories upon which the Scrum Master certification exam is graded. The 'Scrum Pillars' don't constitute a specific category, but they frequently weave their way into answers about why a given event is taking place, or why a given Scrum artifact is important.

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

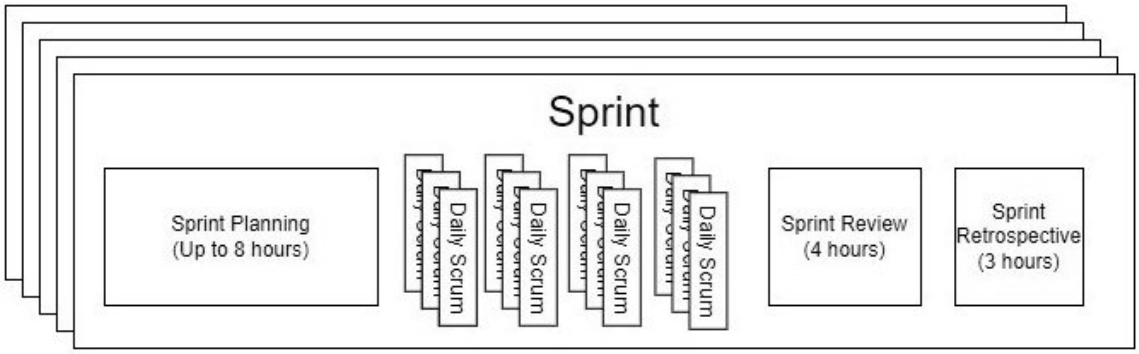


Figure 11. The Scrum pillars and values justify the need for the various Scrum events.

The Scrum Pillars

We've already seen references to two of the Scrum Pillars in our discussion of the Scrum Guide:

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

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

> 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 enough. The three pillars of Scrum are:

- Transparency
- Inspection
- Adaptation

Everything that happens in Scrum harks back to the values and pillars. The question as to why a particular event happens or why a particular artifact is required typically refers 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:** The idea that a Scrum Team is made up of conscientious and motivated developers who are dedicated to achieving the Product Goal.
- **Focus:** The idea that the Scrum Team concentrates on the work at hand and doesn't plan too far into the future.
- **Openness:** The idea that developers work in a psychologically safe space where people can voice their opinion and share ideas without fear of reprisal.
- **Respect:** The idea that you will treat people with dignity and compassion, with others doing the same in turn.
- **Courage:** The idea that a member of the Scrum Team will speak out against unethical practices or behaviors that put the project at risk.

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

Test Yourself

Which three of the following are Scrum pillars?

- ▢ A) Commitment
- ▢ B) Inspection
- ▢ C) Openness
- ▢ D) Transparency
- ▢ E) Adaptation

Options B, D and E are correct.

The Scrum pillars are transparency, inspection, and adaptation.

The Endless Feedback Loop

The Scrum Pillars feed into each other. It's a feedback loop that drives the entire Scrum framework.

- Transparency allows all participants to honestly inspect progress.
- Inspection can reveal flaws in the plan. This leads to adaptation.
- As teams adapt, they must be open about how well the adaptation has worked. This requires the team to be transparent, which then leads to inspection and adaptation. It's a constant loop.

Inspection

> 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.
>
> An 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 are 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 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 well the team worked together during the Sprint.
- Why do we have Scrum Artifacts? So people can *inspect* them and adapt accordingly.

The need for inspection throughout a Sprint is a common justification for Scrum events and artifacts. Events and artifacts enable the feedback loop that is built upon the Scrum Pillars.

Test Yourself

Inspection in Scrum is: (Choose 2)

- ▢ A) Done frequently to detect potential problems
- ▢ B) Done infrequently to allow teams to concentrate on development
- ▢ C) Done to identify undesirable variances
- ▢ D) Done to ensure undesirable variances do not occur

Options A and C are correct.

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.

Transparency

To know exactly how product development is progressing, everyone 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 the team's goal is during this Sprint, and the team's plan for achieving that goal. If stakeholders want 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

Which one of the following problems will occur if product development lacks transparency?

- ▢ A) Technical debt will embed itself in deliverables
- ▢ B) Product delivery dates will be difficult to estimate
- ▢ C) Stakeholders will begin asking for status updates
- ▢ D) Honest inspection of the Scrum Team's progress becomes impossible.
- ▢ E) All of the above

Option E is correct.

Without transparency, it is impossible to know exactly what is happening throughout the development process. This destroys trust, makes honest inspection impossible and prompts management to want to micro-manage developers.

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

Adaptation

Things never go according to plan. That's why 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.

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, which is why Scrum has become popular in those domains as well.

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

The Scrum Guide describes several artifacts and timeboxed 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 problem 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 are requirements 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:

- ▢ A) End the Sprint early and begin a new round of Sprint Planning
- ▢ B) End the Sprint early and do a Sprint Review with all stakeholders
- ▢ C) End the Sprint early and do an internal Sprint Retrospective to see what went wrong
- ▢ D) Adapt during the Sprint and continue to push towards the Sprint goal.

Option D is correct.

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 Scrum Developers cannot end a Sprint early in Scrum.
- The Scrum Master cannot end a Sprint early in Scrum.

Only the Product Owner can do that, and only under the very special condition in 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 toward 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 the problem to fix it. When should this issue be discussed with your fellow developer?

- ▢ A) When the Scrum Master can coordinate a meeting between the two of you
- ▢ B) After the Quality Assurance (QA) team has time to investigate
- ▢ C) During the next scheduled Daily Scrum
- ▢ D) You should go over to your fellow 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 to 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.

Commitment, Focus, Openness, Respect and Courage

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 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, and independent people are respected as such by the people with whom they work.
>
> The Scrum Team members dare to do the right thing to work on tough problems.
>
> These values give direction to the Scrum Team about 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 help to justify why we perform the Scrum events and why we create the Scrum artifacts.

Quite often when a Scrum Master, Product Owner, or Scrum Developer is faced with a difficult challenge, the answer to the problem lies in how to conjure 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 what they mean.

Test Yourself

According to the Scrum Guide, which of the following is **not** a Scrum Value? (Choose 5)

- ▢ A) Agreeableness
- ▢ B) Commitment
- ▢ C) Conscientiousness
- ▢ E) Openness
- ▢ E) Respect
- ▢ F) Extroversion
- ▢ G) Focus
- ▢ H) Emotional stability
- ▢ I) Courage
- ▢ J) Honesty

Options A, C, F, H, and J are correct.

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

Honesty 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.

The big 5 personality traits are worth looking into if you're into psychology and human behavior, but you won't be tested on them when you take the Scrum Master certification exam.

A few people I need to thank…

Everyone says that social media is so negative and divisive, but the fact is, I don't know where I'd be if it wasn't for the ongoing support and encouragement from the friends I've made on Twitter. (And yes, I know it's called X now.)

Xennial Innovations	@xennialinc
Chuck LaPress	charleslapress.com
Michael Lloyd	#DysfunctionMapping
Walker Boh	@nakedMCSE
Bojan	@bojbojbojbojboj
Kris Hultner	@KrisHultner
The Pocket Programmer	@PocketProgram
Gareth	@ggwhiting
Andrew Brown	@andrewbrown
Josh Long	@starbuxman
Ted Neward	@tedneward
Jack Forge	@thejackforge
Open Dev Log	@OpenDevLog
Stoic Michael McGill	@mcgillmd921
Uncle Bob Martin	@unclebobmartin
Cameron McKenzie	@cameronmcnz
Allen Holub	@allenholub
Krishna S	@Firestarter7676
Brian Dordevic	@briandordevic
Kevin McAleer	@kevsmac
Sean Kaye	@SeanDoesLife
Kaje	@Quasicodo
Ben Weddle	@ben__weddle
Eddie Jaoude	@eddiejaoude
Minimalish	@Minimalish_YT
Rami M. Amin	@rmtux
Liz Rowe	@lizroweyt

I know I've forgotten a bunch of people. If you're a long-time supporter and follower, message me on Twitter (@Scrumtuous) so I can correct the mistake. Again, thank you all so much!

More to come!

This Scrum Master Certification Guide is just the start!

Follow me on Twitter, head over to my website or subscribe to my newsletter to learn more about:

- My upcoming Product Owner Practice Exams book
- My online Scrum Master Practice Tests with over 350 unique sample questions
- My Discord channel where we talk about all things Scrum
- My online Scrum Master Certification Course

Much of this is all in the development stage, so keep in touch and join me as I build in public!

Go to www.scrumtuous.com to learn more!

Figure 12. Please come find me on Twitter. I'm @Scrumtuous on all platforms!

When the student is *truly* ready, the teacher will disappear.
Tao Te Ching