

3 managing projects with Scrum

The Rules of Scrum



The rules of Scrum are simple. Using it effectively is not so simple.

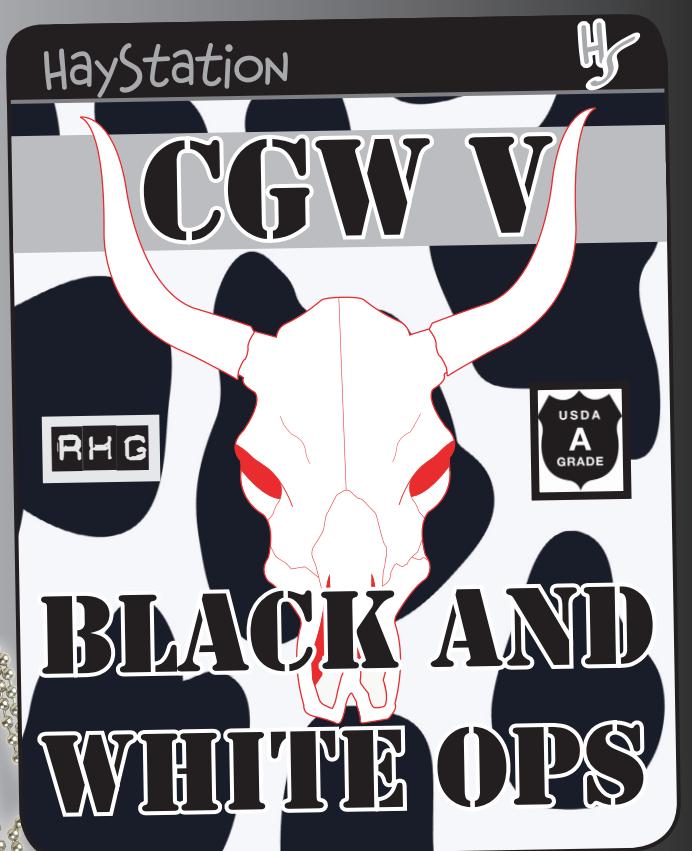
Scrum is the most common approach to agile, and for good reason: the **rules of Scrum** are straightforward and easy to learn. Most teams don't need a lot of time to pick up the **events, roles, and artifacts** that make up the rules of Scrum. But for Scrum to be most effective, they need to really understand the **values of Scrum** and the Agile Manifesto principles, which help them get into an effective mindset. Because while Scrum seems simple, the way a Scrum team constantly **inspects and adapts** is a whole new way of thinking about projects.

COMING SOON FROM RANCH HAND GAMES

THE TEAM THAT
BROUGHT YOU THE
BLOCKBUSTER
INTERNATIONAL HIT
COWS GONE WILD
VIDEO GAMES

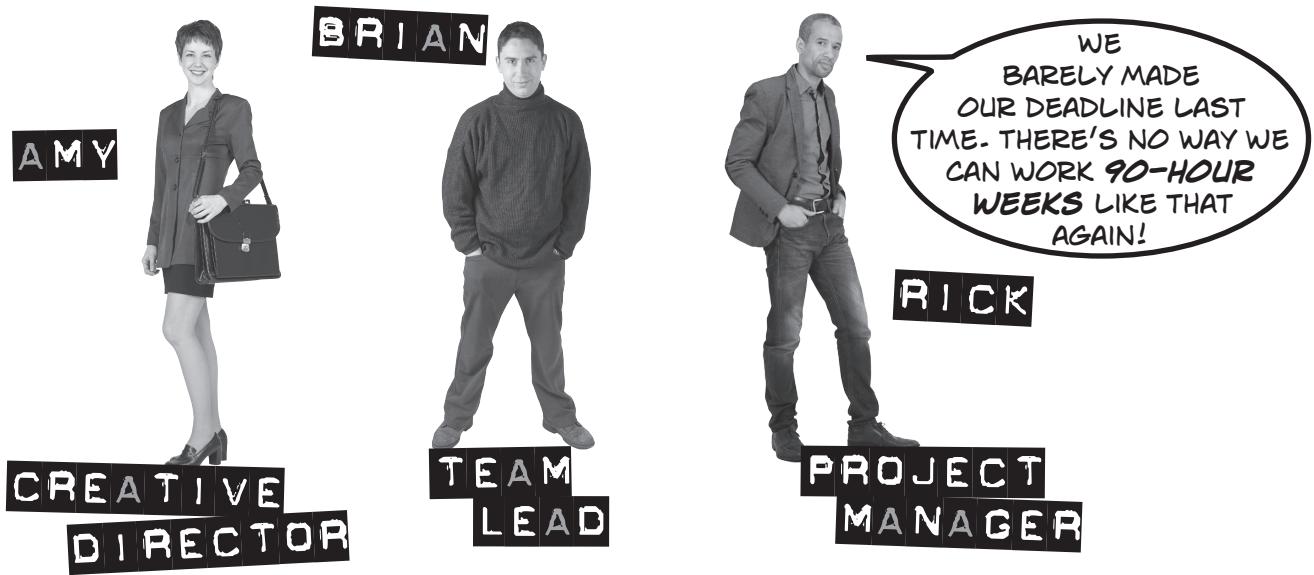


**COWS
GONE
WILD
V**



Meet the Ranch Hand Games team

They're hot off the successful release of *Cows Gone Wild IV: The Milk Man Cometh*, and about to tackle their most ambitious project yet! But while *CGW4* may have sold well, it was far from perfect as a project, and Amy, Brian, and Rick want *CGW5* to be an improvement. Agile to the rescue!



Amy: Wow, I'm so glad you said that. I can't take another project like that. Just staying on top of the last-minute artwork changes was practically impossible.

Brian: Oh man, we can't have that argument again. You know they were necessary because the levels kept changing. I had my whole team working nights and weekends just to keep up.

Amy: I know, I know. We all got overwhelmed trying to tackle too many things at the same time.

Brian: No matter how much planning we did, our schedules never seemed long enough.

Rick: Yeah, that was definitely a problem—and I've been doing a lot of research about how to fix that. What do you guys think about using **Scrum**?

Amy: I've been reading about it, too. I think it could help.

Brian: Look, anything that can help make things less insane around here gets my vote.

Amy: But don't the Scrum rules say that we need a Product Owner and a Scrum Master?

Rick: Well, I'm looking over what a Scrum Master does, and I think I'm up to that task. Amy, you work with the business side a lot, right? What do you think about being the Product Owner?

Amy: I'll try anything once. I'll start letting the business and PR teams know that I'm the Product Owner.

Brian: Let's do this!

The Scrum events help you get your projects done

Scrum is the most popular approach to agile, and for good reason. The rules of Scrum are simple, and teams all around the world have been able to adopt them and improve their ability to deliver projects. Every Scrum project follows the same pattern of behavior, which is defined by a **series of timeboxed events** that always happen in the same order. Here's what the Scrum pattern looks like:

This is the single source for requirements and any changes that will be made to the product throughout the project.

Every Scrum project is organized into timeboxed iterations called Sprints. Many teams use 30-day Sprints, but it's pretty common to see two-week Sprints too.

The Scrum events:

The Sprint

The Sprint planning session

The Daily Scrum

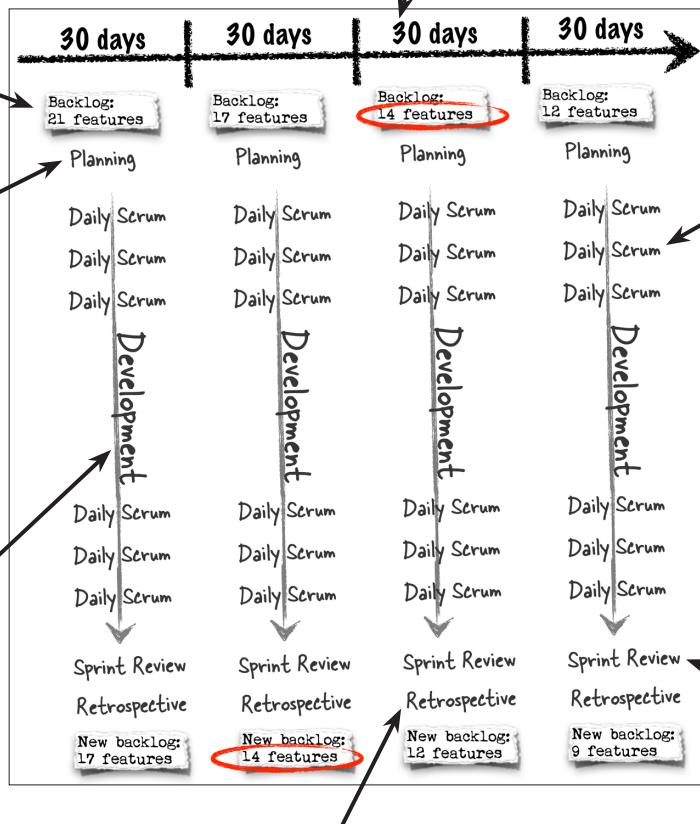
The Sprint Review

The Sprint Retrospective

The team uses the Product Backlog to keep track of the features that they'll build for the whole project.

At the beginning of each Sprint, the Scrum Team gets together for a Sprint Planning session where they choose what items to include in the Sprint.

The items for the Sprint are pulled off of the Product Backlog and added to the Sprint Backlog. During the Sprint, all of the development work is focused on building the items in the Sprint Backlog.

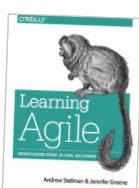


Every day, the team holds a Daily Scrum, a short meeting where each person gives an update on their progress, what they'll work on next, and any roadblocks they've hit.

When Sprint is done, the team holds a meeting called the Sprint Review, where they meet with the users and give a demo of the working software that they built.

The very last thing the team does in the Sprint is meet and hold a Sprint Retrospective, where they talk about what happened during the Sprint so they can reproduce the things that went well and learn from any problems.

If you've read our book *Learning Agile* then you'll recognize this illustration of the basic Scrum pattern!



The Scrum roles help you understand who does what

There are three roles that must be filled on every Scrum team. The first role is the one that's most familiar to us: the **Development Team**. People on the team may have different areas of expertise, and maybe even different job titles in the company, but they all participate in the Scrum events the same way. There are also two very important roles filled by team members: the **Product Owner** and the **Scrum Master**. And when you add them to the Development Team, you get the entire **Scrum Team**.



The Product Owner helps the team understand the users' needs so they can build the most valuable product.

The Product Owner works with the team every day to help them understand the features in the Product Backlog: what items are on it and why the users need them. This is a really important job because it helps the team **build the most valuable software they can**.

The Scrum rules are clear that the Product Owner and Scrum Master roles are each filled by a person and not a committee.



Flip back to the last chapter. Can you find an agile principle that talks about delivering valuable software?



The Scrum Master helps the team understand and execute Scrum.

Scrum may be simple to describe, but it's not always easy to get right. That's why there's one person on the team, the Scrum Master, whose whole job is to help the Development Team, the Product Owner, and the rest of the company to do exactly that—get Scrum right.

The Scrum Master is a leader (which is why the word “master” is right there in the name). But he or she demonstrates a very *particular* kind of leadership: the Scrum Master is a **servant leader**. This means that the person in this role spends all of his or her time helping (or “serving”) the Product Owner, the Development Team, and people throughout the organization:

- ★ Helping the product owner find effective ways to manage the backlog
- ★ Helping the Development Team understand the Scrum events, and facilitating them if needed
- ★ Helping the rest of the organization to understand Scrum and work with the team
- ★ Helping everyone do the best job they can to deliver the most valuable software possible

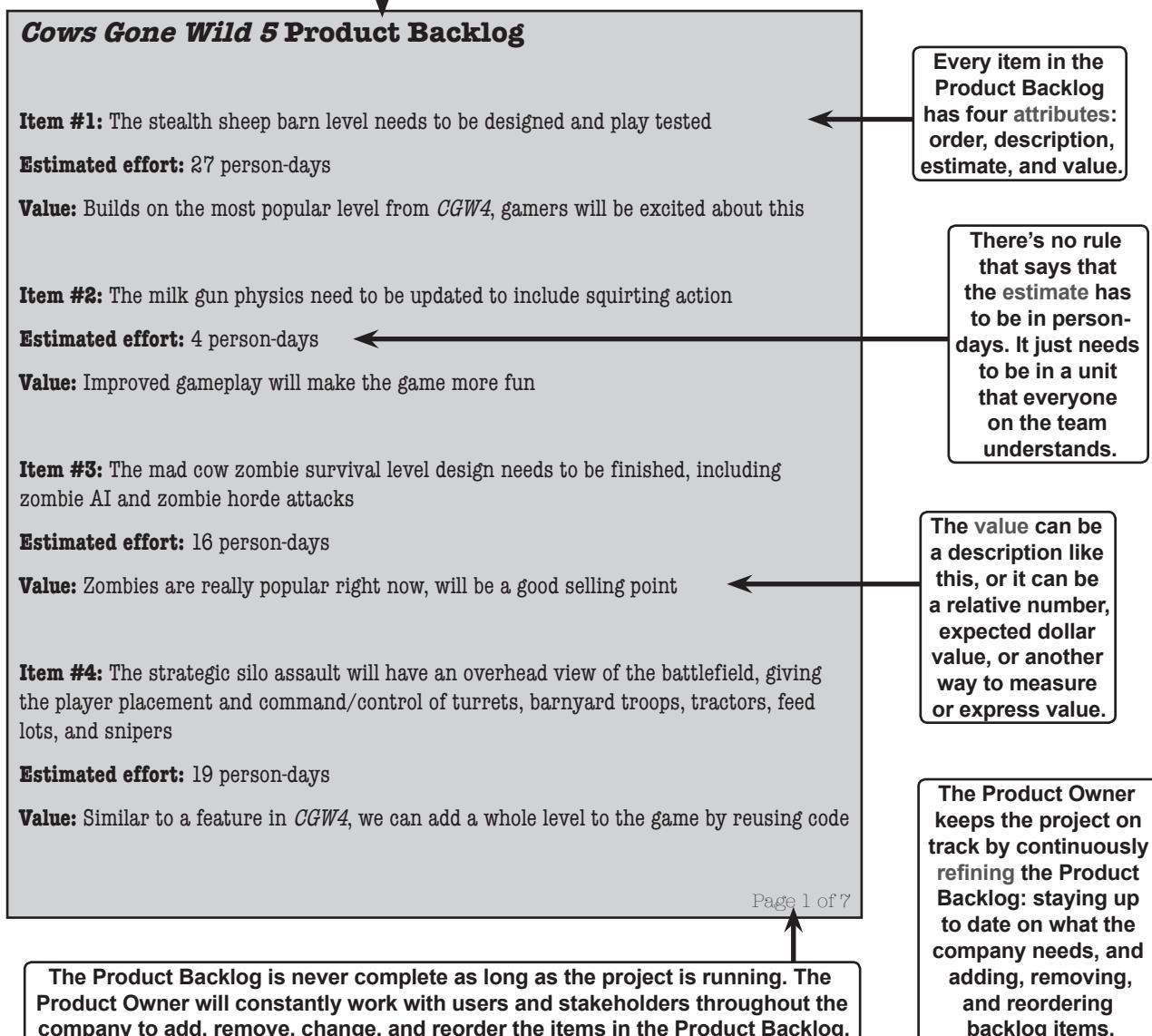
The Scrum Guide lays out the rules for how teams use the Scrum framework.

Before you move on to the next page, go to <https://www.scrum.org> and download a copy of the Scrum Guide by Ken Schwaber and Jeff Sutherland, the originators of Scrum. It contains the definition of Scrum, and is updated regularly with the latest ideas about how teams use Scrum: when new thinking is incorporated into Scrum, that's where you'll find it. Also, did you notice how Scrum, Daily Scrum, Sprint, Sprint Planning, Sprint Review, Product Backlog, and other terms are **capitalized** in this chapter? That's done to follow the standard in the Scrum Guide.

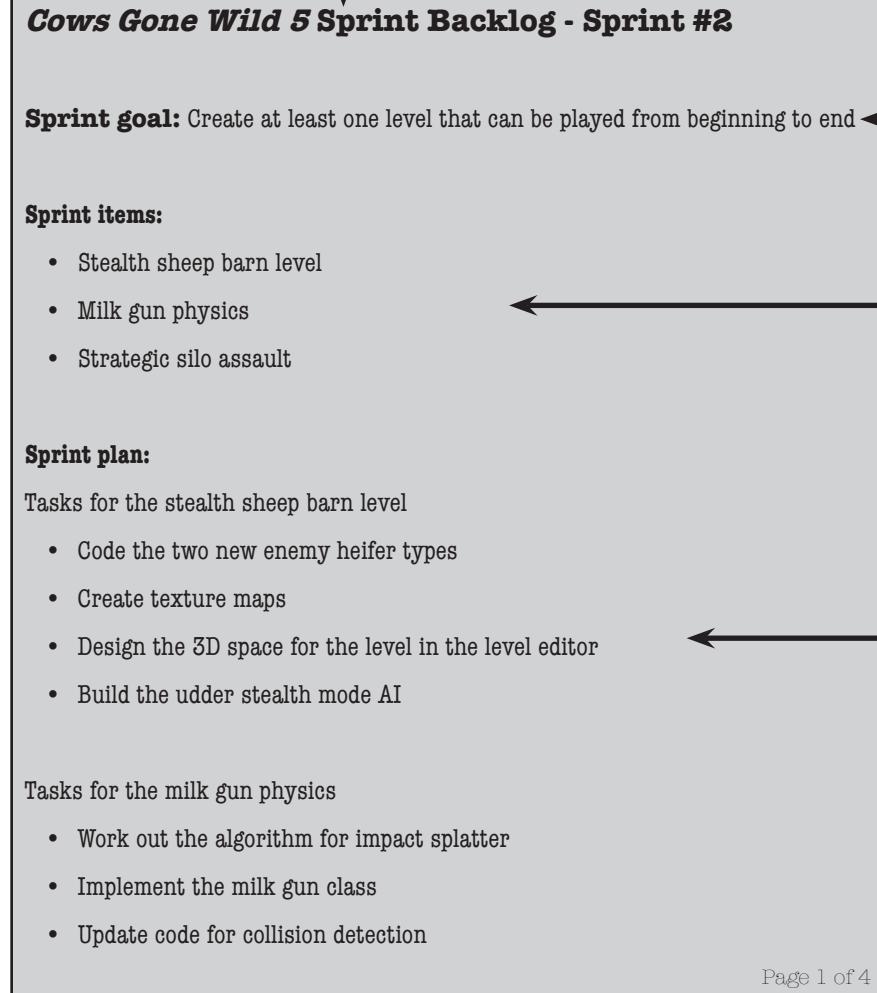
The Scrum artifacts keep the team informed

Software projects run on information. The team needs to know about the product that they're working on, what they're building in the current Sprint, and how they'll get it built. Scrum teams use three **artifacts** to manage all of this information: the **Product Backlog**, the **Sprint Backlog**, and the **Increment**.

Here's an example of a **Product Backlog**—but there's no rule that says that it has to look like this. Many teams keep spreadsheets, or add entries to a database, or use software tools to manage their backlogs.



Here's an example of a Sprint Backlog (again, there's no rule that says that it has to be formatted exactly like this). The team creates it during Sprint Planning, but it can evolve based on what the team learns as the Sprint unfolds.



The team crafts the Sprint Goal, or the objective that will be met by the team by completing the Sprint items.

In the first part of the Sprint Planning session, the team determines what can be done in the Sprint by choosing which items to include in it.

In the second part of the Sprint Planning session, the team determines how the chosen work will get done by decomposing the Sprint items into tasks.

The Sprint Planning session's timebox often expires before the team is done decomposing all of the items into tasks, so they typically start with the items that will get built first and finish the plan along the way.

Page 1 of 4

The Increment is the sum of all backlog items that are actually completed and delivered at the end of the Sprint

Scrum is incremental, which means the project is broken into “chunks” that are delivered one after another. Each of those “chunks” is called an **Increment**, and each Increment represents the result of one complete Sprint: the working software that the team demonstrates to the users in the Sprint Review. That working software typically includes all of the features that they previously delivered—which makes sense, because they’re not going to delete them!—so the *product Increment* is the sum of all of the backlog items completed during this Sprint and all of the previous Sprints.



The Scrum Sprint Up Close

30 days

The **Sprint** is a **timeboxed** iteration. Most teams use a two-week Sprint, but it's common to see 30-day Sprints as well.

Planning

Daily Scrum

Daily Scrum

Daily Scrum

Development

Daily Scrum

Daily Scrum

Daily Scrum

Sprint Review

Retrospective

The **Sprint Planning** session is a meeting with the whole team, including the Scrum Master and Product Owner. For a 30-day Sprint it's timeboxed to 8 hours, for 2-week Sprints it's 4 hours, and other Sprint lengths have proportionally sized timeboxes. It's divided into parts, each timeboxed to half of the meeting length:

- ★ In the first half, the team figures out **what** can be done in the Sprint. First the team writes down the **Sprint Goal**, a one- or two-sentence statement that says what they'll accomplish in the Sprint. Then they work together to pull items from the Product Backlog to create the **Sprint Backlog**, which has everything they'll build during the Sprint.
- ★ In the second half, they figure out **how** the work will get done. They break down (or **decompose**) each item on the Sprint Backlog into **tasks** that will take one day or less. This is how they create a **plan** for the Sprint.

The **Daily Scrum** is a 15-minute timeboxed meeting that's held at the same time every day. In it, the Development Team and Scrum Master meet and the Product Owner is strongly encouraged to participate. Each person answers three questions:

- ★ What have I done since the last Daily Scrum to meet the Sprint Goal?
- ★ What will I do between now and the next Daily Scrum?
- ★ What roadblocks are in my way?

All of the work is planned, but not all of it is decomposed. The meeting timebox can expire before the team's done decomposing every Sprint Backlog item, so they concentrate on decomposing work for the first days of the Sprint.

In the **Sprint Review** the whole team meets with key users and stakeholders who have been invited by the Product Owner. The team demonstrates what they built during the Sprint, and gets feedback from the stakeholders. They'll also discuss the Product Backlog, so that everyone knows what will *probably* be on it for the next Sprint. For 30-day Sprints, this meeting is timeboxed to four hours.

The **Sprint Retrospective** is a meeting that the team uses to figure out what went well and what can be improved. Everyone on the team participates, including the Scrum Master and Product Owner. By the end of the meeting they'll have written down specific improvements that they can make. It's timeboxed to three hours for a 30-day Sprint.

The Sprint is over **when its timebox expires**.

there are no Dumb Questions

Q: Wait, is that all there is to Scrum?

A: That's all of the Scrum rules, but that's definitely not all there is to Scrum. Scrum was designed to be lightweight and simple to understand. But mastering Scrum takes a lot more than just following the rules. You learned in the last chapter how the values of the Agile Manifesto can make a big difference in how a team uses a practice. That same idea applies to Scrum: **mindset and experience** make the difference between a team with an empty "just following the rules" approach and an effective Scrum team that really "gets" it.

Q: I already break my projects into phases. Aren't Sprints the same thing?

A: Not at all. Traditional waterfall projects are often broken into phases, with a complete deliverable at the end of each phase. But those phases are still planned at the beginning of the project. And if there's a change discovered that will affect the next phase, the project needs to be replanned, which usually involves a separate change control process. In other words, the team basically assumes the project plan is mostly correct, and that it's the project manager's job to deal with the relatively few changes that happen along the way.

Scrum is different because it's **iterative**, which is a lot more than just breaking a project down into phases. The team doesn't even plan the next iteration until the current one is complete. The team might discover changes partway through the Sprint that affect the next one, or which may even affect the current one. That's why the Product Owner is such an important member of the team: he or she has the authority to make decisions on behalf of the business or customers about what features the team will build, which gives them the power to make those changes immediately.

Q: What's the difference between the Product Backlog and the Sprint Backlog?

A: The Product Backlog contains a list of everything that might be needed in the product. Scrum teams are usually working on ongoing releases of a single product, so the Product Backlog is never complete. The first version they release typically contains the requirements that are best understood by everyone. As the project goes on, the Product Owner will add and remove items based on what's valuable to the company.

The Sprint Backlog contains the specific items that the team will build during the Sprint, which were moved from the Product Backlog during Sprint Planning. The software that the team finishes during the Sprint is the **Increment**: the team delivers and reviews a complete Increment in each Sprint. The Sprint Backlog *also* contains a **plan for delivering the Increment**. The team builds that plan during Sprint Planning by decomposing backlog items into tasks.

Q: What exactly *is* a backlog item?

A: Every item in the Product Backlog consists of a brief description, a (usually rough) estimate of how long it will take, the business value, and an order. The Product Owner will routinely **refine the Product Backlog**. This means going through the items in the backlog, removing ones that are no longer useful, re-evaluating the value of each item, and updating the order so the most valuable ones are first.

Q: Wait a minute—in the story, Brian is a team lead. But there are only three roles in Scrum, and "team lead" isn't one of them. What's going on there?

A: Roles and jobs aren't the same thing. As far as Scrum is concerned, Brian is just another member of the Development Team,

even though his job in the company is team lead, giving him more authority than the other the developers, and he has his own skills and expertise. Scrum may not have a distinct role for him to play, but *he still plays an important and unique role on the team*. It's just that there are no Scrum-related events or artifacts that are specific to him.

Q: What do you mean by "artifacts"?

A: An artifact is just a by-product specific to a process or methodology. Scrum has **three artifacts**: the Product Backlog, the Sprint Backlog, and the Increment.

Q: What happens if we get partway through a Sprint and discover there's some sort of emergency situation? Do we still have to wait for the timebox to expire before the Sprint can end?

A: On *very rare occasions* the Product Owner has the authority to cancel the Sprint before the timebox is over. When he or she does this, a Sprint Review is held for any Sprint Backlog items that are done, and all other items are put back on the Product Backlog and left for the next Sprint Planning session. *Be really, really careful about cancelling a Sprint.* It's an in-case-of-emergency "break glass" action because it can waste a lot of the team's energy—and more importantly, it causes people in the company to distrust the team, and to lose confidence in the effectiveness of Scrum.

Scrum was designed to be lightweight and simple to understand, but mastering Scrum takes a lot more than just following the rules.

they followed the rules but something went wrong



Sharpen your pencil

Write down the name of each of the Scrum events, when the event happens, and the length of the event's timebox. We filled in the first event. Then write down the three Scrum roles and three Scrum artifacts.

Event names in the order they occur

Sprint

When the events happen

Length of the event's timebox

Assume that the Sprint is timeboxed to 30 days

Write down the Scrum roles

Write down the Scrum artifacts

→ Answers on page 112

FOUR MONTHS LATER AT A SPRINT RETROSPECTIVE...



I DON'T THINK THIS SPRINT WENT SO WELL.

WHAT TIPPED YOU OFF, SHERLOCK? WAS IT THE FACT THAT WE BARELY GOT ANYTHING DONE?



Rick: Hey, there's no need for insults. We're all doing our best here!

Amy: Sorry. Things have just been really tense with the business lately, and it's got me on edge. I spend so much time with Scrum that I don't have time to do my job.

Rick: What do you mean?

Amy: I mean, you guys constantly ask me to make decisions. Like when we were planning this Sprint, and I had to decide whether Brian's team should start the strategic silo assault level or improve the milk grenade mechanics.

Rick: Yeah. You had us get started on the strategic silo assault. Was that wrong?

Amy: Oh man, you have no idea! After the demo I got hauled into the CEO's office and screamed at for an hour. Gamers hated the strategic levels in *CGW4*, and the last thing our stakeholders want is to see bad reviews for them in *CGW5*.

Rick: Wait, what? We worked super hard on those. I thought they came out well!

Amy: Me too. But they said that I had no business making the decision to include any strategic levels in the Sprint—or any other decision like that.

Rick: But you're the Product Owner! Making those decisions is part of your job now.

Amy: Exactly. So I have no idea what to do. Plus, I spend so much time answering the team's questions about features that I barely have time for my real job.

Rick: Well, it's no easier for me. It's getting harder and harder to drag Brian's team members into my Daily Scrums—you know how programmers hate meetings.

Amy: You know what? We're following the Scrum rules, it's kind of helping, I think. Right? Maybe? Um... are we still sure this Scrum stuff is actually worth the trouble?



The team followed the Scrum rules to the letter, but the project still ran into trouble. What went wrong?

The Scrum values make the team more effective

We already saw how agile teams are much more effective when everyone on the team has a mindset that's driven by the values in the Agile Manifesto. Scrum comes with its own set of five **values** that do exactly the same thing for Scrum teams.



Each of the five Scrum values is represented by one word. In this case, the value is "openness".

You always know what your team members are working on, and you're comfortable that they know what you're working on. If you run into a problem or an obstacle, you can bring it up with the team.

It's not always easy to talk to your teammates when you run into problems. None of us like making mistakes, especially at work. That's why everyone on the team needs to share this value: it's a lot easier to be open with teammates about problems and roadblocks that you run into if they're just as open with you about theirs—and that helps the whole team.



I'M REALLY SORRY, I KNOW
I PROMISED I'D BE DONE WITH THE CODE FOR
THE SILO SNIPER LEVEL BY NOW, BUT I JUST CAN'T
GET THE LOGIC RIGHT.

It was uncomfortable and a little embarrassing for Brian to be open about this roadblock, but it's what was best for the project.

I WON'T LIE, THAT'S
GOING TO SET US BACK. BUT
WE'LL FIND A WAY TO GET YOU THE
TIME YOU NEED.



Respect



You and your teammates have mutual respect for each other, and every person on your team trusts everyone else to do their jobs.

Trust and respect go hand in hand. People on Scrum teams listen to each other, and when they disagree they take the time to understand each other's ideas. It's natural for people on teams to disagree on an approach. On an effective Scrum team, you'll listen when your teammates disagree with the approach you're taking, but in the end they'll respect your decisions, and give you the benefit of the doubt if you take an approach they disagree with.

Scrum teams need at least three members, and they usually max out at nine people. A Scrum team needs at least three people in order to get a significant amount of work done in a Sprint. But once the team size gets up to ten or more people, it becomes really hard for them to coordinate with each other—the Daily Scrum gets too chaotic, and it gets difficult to plan effectively.

Trust doesn't always come easy to a traditional waterfall team where the project manager does the planning by demanding estimates from the team members. If things go wrong, the project manager can blame the team for underestimating, and the team can blame the project manager for a bad plan.



Courage

Scrum teams have the courage to take on challenges. Individual people on the team have the courage to stand up for their project.

What do you do when the boss demands that you and your team meet an impossible goal? What if he demands that they meet a two-week deadline for a project that can't possibly be done in less than two months? Effective Scrum teams have the courage to stand up and say "no" to impossible goals because they have the planning tools to show what is possible, and users and stakeholders who trust them to deliver the most valuable software that they can.



Focus

This is why the team writes a one- or two-sentence Sprint Goal at the beginning of the Sprint Planning meeting. It helps keep them focused during the Sprint.

Every team member is focused on the Sprint Goal, and everything they do in the Sprint helps move them toward it. During the Sprint, everyone works only on Sprint tasks, and does one task at a time until the Sprint is done.

During a Scrum Sprint, every single person on a Scrum team is focused exclusively on items in the Sprint Backlog and tasks that they decomposed them into during the planning meeting. Each person works on one item in the Sprint Backlog at a time, focuses exclusively on one task in the plan, and moves on to the next task only when the current one is done.



WHAT ABOUT MULTITASKING?
AREN'T TEAMS MORE EFFECTIVE
WHEN PEOPLE DO MULTIPLE TASKS AT
THE SAME TIME?

People on Scrum teams know that focusing on one task at a time is more effective than trying to multitask.

There's a myth that switching back and forth between multiple tasks many times a day is more effective than concentrating on one at a time. Think about it this way: let's say you have two tasks that will each take one week. If you try to multitask and do them at the same time, the best you'll do is deliver them both at the end of two weeks. But if you don't start the second task until the first one is done, you'll finish the first task a week earlier.

There's one more Scrum value. Flip the page... ➔

Commitment



Every person on the team is committed to delivering the most valuable product that they can—and the rest of the company is committed, too.

When the team is committed to the project, it means that the tasks that they planned out to meet the Sprint Goal are the most important tasks they have to work on. Each team member feels that his or her personal success at the company is tied to the success of the project. Not only that, but every single person on the team then *feels committed to each item in the Increment*, not just the items that they're working on. This is called **collective commitment**.

But what happens if something comes up that's important to the company, but not part of the project? For Scrum to be effective, the boss needs to keep that from happening. In other words, **the company needs to be fully committed to the project**, to respecting the team's collective commitment to the Sprint Goal, and to following the rules of Scrum.

So how does a company express that kind of commitment?

By **giving the team the authority** to determine what features are going to be developed during each Sprint, and trusting that's how the team will deliver the most valuable software possible. The company does that by assigning a full-time Product Owner to the team with the authority (and willingness!) to decide what features will be built and accept them as done.

Collective commitment means that every team member feels committed to delivering the complete Increment, and not just the items he or she is working on.





Story Time



Once upon a time there lived a pig and a chicken who were the very best of friends.

One day the chicken said to the pig, "I have a great idea. Let's open a restaurant together!"

The pig said, "That is a great idea! What should we call it?"



The chicken replied, "How about Bacon 'n Eggs?"

The pig thought about it for a minute. Then he said, "You know what? Never mind. You're just involved, but I'm committed."



Q: Uh... why are you talking about farm animals?

A: Because the fable of the pig and the chicken is a good way to understand commitment (because the chicken just lays eggs, but the pig has to let himself get eaten—a much greater commitment). In fact, some teams even use these terms in practice this way, calling people who are committed to the project “pigs,” and those who have an interest but not a true commitment “chickens.” (Occasionally, they’ll even call themselves pigs in cultures where “pig” is used as an insult!)

The Scrum value of commitment means that when you’re on a Scrum team, you genuinely believe that your professional success or failure rests on your ability to deliver valuable software, and that makes you a committed “pig.” Your users and stakeholders may have a big interest in the project getting done, which makes them “chickens”—they’re very important to the

there are no Dumb Questions

project, but they don’t feel the same sense of strong personal commitment as “pigs” do.

Q: What if the team doesn’t really “get” some of the Scrum values?

A: An important part of the Scrum Master’s job is to help everyone on the team to understand the Scrum values, and to eventually internalize them so that each person has the right mindset for Scrum. Very few teams start out Scrum genuinely believing in all of these values at first. They grow and evolve together over time. The Scrum Master helps by using the Scrum values to help the team understand and deal with project obstacles that come up.

Q: What if my team can’t find a Product Owner with enough authority?

A: If the Product Owner doesn’t have the authority to decide what the team will build,

or to accept each item in the Sprint Backlog as done, then the company hasn’t given the team the authority to do their jobs, and they haven’t truly committed to the project—or to Scrum. That can be trouble.

A lot of projects fail because the team did a great job building the wrong software. The Product Owner keeps that from happening because his or her full-time job is to work with the rest of the company and understand what the business needs, decide what features will be built, and help the team to understand those features.

**If the Product Owner
doesn’t have the authority
to decide what features
the team builds, then
the company isn’t really
committed to delivering
the project with Scrum.**

BULLET POINTS

- Scrum is the most popular approach to agile, and is most successful when used by software teams of at least three and up to nine people.
- Scrum includes five timeboxed **events**: the Sprint, Sprint Planning, the Daily Scrum, the Sprint Review, and the Sprint Retrospective.
- Scrum has three **roles**: the Product Owner, the Scrum Master, and the Development Team.
- Scrum uses three **artifacts**: the Product Backlog, the Sprint Backlog, and the Increment.
- Projects are divided into **Sprints**, iterations that are typically timeboxed to 30 days (but can be shorter).
- Each Sprint starts with **Sprint Planning**, a timeboxed meeting in which the team decides which items (such as features) to include in the Sprint Backlog, and decomposes them into tasks for at least the first week.
- The **Daily Scrum** is a meeting where each team member talks about what they’ve done, what they’ll work on next, and if they see any roadblocks ahead.
- The Product Owner invites key stakeholders to the **Sprint Review** where the team demonstrates working software and discusses the next Sprint Backlog.
- At the **Sprint Retrospective** the team talks about what went well and identifies specific things that can be improved.
- Scrum teams have five **values** that help them get into a more effective mindset: openness, respect, courage, focus, and commitment.



I'VE GOT GREAT NEWS! I MET WITH A LOT OF PEOPLE ON THE BUSINESS SIDE ABOUT OUR COMMITMENT PROBLEMS, AND THEY AGREED TO ASSIGN ALEX TO OUR TEAM AS A FULL-TIME PRODUCT OWNER.

Alex is a senior manager who's been in the gaming industry for a long time. He meets with users, advertisers, and game reviewers. He knows all about what makes games sell.

GOOD TO MEET YOU GUYS. RICK EXPLAINED HOW SCRUM WORKS, AND THE BOSSSES ALL THINK SCRUM IS **SO IMPORTANT** THAT THEY ASSIGNED ME TO WORK FULL TIME ON IT.



THERE'S **NO QUESTION ABOUT AUTHORITY**. IF I SAY IT GOES INTO THE GAME, IT GOES INTO THE GAME. THE OTHER SENIOR MANAGERS TRUST MY JUDGMENT, AND I TRUST YOU GUYS TO GET THE PROJECT DONE.

PRODUCT OWNER

THAT'S A HUGE RELIEF! NOW I CAN GO BACK TO "JUST" BEING CREATIVE DIRECTOR.

THE DEVELOPERS HAVE A LOT OF QUESTIONS FOR YOU! I'LL INTRODUCE YOU TO THEM.



Sprint 5 Product Backlog

~~strategic silo assault~~

CAN'T DO THAT,
GAMERS HATED IT!
so what do we do instead?

With a new Product Owner, the team should be able to figure out the most valuable features to include in the next Sprint.



Here are some things we overheard Amy, Rick, and Brian saying. Draw a line from each speech bubble to either **COMPATIBLE** or **INCOMPATIBLE**, and then to the Scrum value that it's either compatible or incompatible with.



COMPATIBLE

IT'S MY TURN TO TALK? OK. SINCE THE LAST DAILY SCRUM I KEPT WORKING ON THE SAME FEATURE, AND I'LL DO THE SAME UNTIL THE NEXT SPRINT. NO ROADBLOCKS. WHO'S NEXT?

INCOMPATIBLE

Courage



COMPATIBLE

LOOK, ALEX, I KNOW REDOING ALL OF THE GRAPHICS WILL IMPRESS REVIEWERS, BUT THERE'S ABSOLUTELY NO WAY THE TEAM CAN DO THAT WITHOUT MISSING THE RELEASE DATE.

INCOMPATIBLE

Focus



COMPATIBLE

THE ONLY THING I'M IMPRESSED WITH IS TECHNICAL ABILITY. IF YOU CAN'T CODE, I HAVE NO USE FOR YOU.

INCOMPATIBLE

Openness



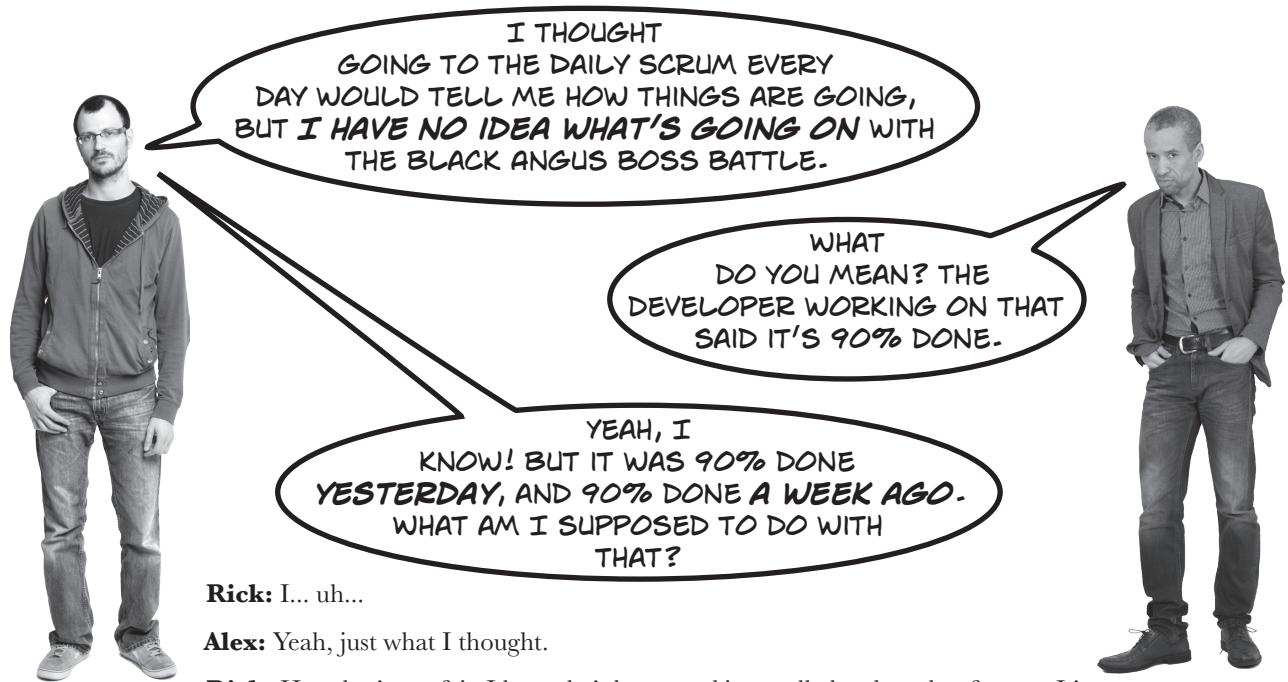
COMPATIBLE

I CAN'T WORK ON ANY SPRINT TASKS TODAY. ANOTHER TEAM HAS A REALLY BIG DEADLINE AND THEY NEED TO BORROW ME.

INCOMPATIBLE

Respect





Rick: I... uh...

Alex: Yeah, just what I thought.

Rick: Hey, that's not fair. I know he's been working really hard on that feature. It's just taking a lot longer than any of us thought it would.

Alex: So what are you going to do about it?

Rick: Hey, we're all on the same team here. And that includes you, Alex. So I think you meant to ask what are we going to do about it?

Alex: OK, fine. Well, since I'm on the team, let me give you an answer. Other teams build contingency and padding into their schedules so *they never have to tell a senior manager like me* that they're running late.

Rick: Sure. Yeah. I've done that before on other projects I've managed. I'd add extra tasks to my schedule to account for things taking a lot longer than we anticipated.

Alex: But you're not doing that now?

Rick: Well, no. The Scrum rules don't really give me a way to add contingency, padding, or extra tasks. I don't really see any way to pad the schedule without breaking the Scrum rules.

Alex: Then maybe there's something wrong with Scrum.

Is padding the schedule really any different than lying about the schedule or hiding information about it from the boss, except that everyone's decided that it's OK?



What do you do when one of the tasks in the Sprint takes a lot longer than the team planned for it to take?

Question Clinic: The “which-comes-next” question



A LOT OF PRACTICES OR EVENTS HAPPEN IN A SPECIFIC ORDER, AND YOU'LL BE ASKED ABOUT THAT ORDER IN A "WHICH-COMES-NEXT" QUESTION. THESE ARE QUESTIONS THAT QUIZ YOU ON HOW PRACTICES FIT TOGETHER ON A REAL PROJECT. THESE QUESTIONS AREN'T USUALLY VERY DIFFICULT, BUT THEY CAN BE A LITTLE MISLEADING.

Most which-comes-next questions describe a situation and ask you what you'd do next. Sometimes it's not immediately obvious that it's asking about the order of events. Watch for any question that lays out a scenario and then asks what comes next, what happens afterward, or how the team should proceed.

Don't be thrown if the question asks about an industry you don't know much about.

The key to a "which-comes-next" question is figuring out what the team is currently doing. So what's another word for "objective that will be met"? That's the definition of the Sprint Goal! So this must be happening at the beginning of the Sprint Planning session.

27. You're the Scrum Master for an automotive industry software team working on firmware for an antilock brake system. Your team just finished writing down the objective that will be met when the team completes the Sprint items. What is the next thing your team should do?

- A. Decompose the Sprint Backlog items into tasks
- B. Review the working software with the users
- C. Meet with the business users
- D. Decide what items to include in the Sprint Backlog

Neither of these answers are things that happen during the Sprint Planning session.

This also happens during the Sprint Planning session. But when you compare this with the other answers, there's another one that happens before it.

Aha! If the team is holding a Sprint Planning session and just finished writing the Sprint Goal, the next thing they'll do is decide what items from the Product Backlog will be included in the Sprint Backlog. This is what comes next!

HEAD LIBS



Fill in the blanks to come up with your own “which-comes-next” question! Start by thinking of a Scrum event or activity to be the correct answer, and then figure out exactly what the team just finished doing before it—that’s what you’ll describe in the question!

You’re a Product Owner on a _____ project. Your team just finished _____
(an industry)

_____ . A _____ just informed you that
(description of a Scrum activity) (a type of user)

your project. _____.
(a problem that came up on the project)

Which of the following does the team do next?

The last part of this question
doesn’t change the answer at all.
A lot of questions are like that.

- A. _____
(the correct answer—description of the Scrum activity, tool, or practice that comes next)
- B. _____
(description of a different Scrum activity, tool, or practice)
- C. _____
(the name of activity, tool, or practice that’s part of a different methodology)
- D. _____
(description of one of the Scrum values or roles)

THE WHICH-COMES-NEXT QUESTION
DOESN’T ALWAYS LOOK LIKE IT’S ASKING ABOUT
THE ORDER OF EVENTS, TOOLS, OR PRACTICES!
KEEP AN EYE OUT FOR QUESTIONS THAT DESCRIBE
SPECIFIC ARTIFACTS THAT GET CREATED OR ACTIONS
YOU’LL PERFORM AND THEN ASK YOU WHAT YOU’RE
SUPPOSED TO DO NEXT.



If you look in the Scrum guide, you'll actually see it refer to a "Done" product increment.

A task isn't done until it's "Done" done

When you're on a Scrum team, you work on one backlog item or task at a time—that's what the value of Focus is all about—and you work on it until it's done. But when, exactly, is it done? Is there still some testing left to do? It's easy to think that you're done... except for this one teeny little thing. That's why Scrum teams have a **definition of "Done"** for every item or feature that they add to the backlog. Before an item can go into the Sprint Backlog, everyone on the team needs to understand and agree about exactly what it means for it to be not just done, but "Done" done. And since every item in the backlog has a definition of "Done," the **entire Increment has a definition of "Done"** and the team is committed to delivering a "Done" Increment at the end of the Sprint.



Under the Hood: Sprint Planning

Sprint Planning relies on the definition of "Done"

During the first part of the Sprint Planning meeting, the team decides which items to include in the Sprint Backlog. But what happens if there's an item that doesn't *really* have a definition of "Done" that everyone on the team understands? Say one person thinks that it means all the code is done, but someone else assumes that it also includes documentation or testing. Even if they don't realize that there's a disagreement about what it means for that item to be done, they'll discover that they can't come to a real agreement on how many other items they can deliver in the Sprint. That's why Sprint Planning only works when every item has a clear definition of "Done" that the whole team can agree on.

Scrum teams adapt to changes throughout the Sprint

Project teams have to make decisions every day: Which features will we build in this Sprint? What order will we build them in? How will users interact with this feature? What technical approach will we take? Traditional waterfall teams have one answer: all of the planning is done at the beginning of the project. The problem is that at the time the plan is being built, most of those questions don't have answers yet. So the project manager works with the team to make assumptions, and relies on a change control process to change the plan when they guessed incorrectly.

Scrum teams *reject the idea* that every project question can be answered at the beginning of the project, or even at the beginning of a Sprint. Instead, they make decisions based on real information as soon as it's discovered. They do this using the **three pillars** of Scrum: a cycle of **transparency**, **inspection**, and **adaptation**:

- ★ The cycle starts with **transparency**, where the team decides together what items to include in the Sprint, and what it means for each of them to be done. All work done by everyone is visible to the team all the time.
- ★ The whole team meets every day in the Daily Scrum to **inspect** each item that's being worked on.
- ★ If they discover changes, they **adapt** to them (like adding or removing items from the Sprint Backlog when roadblocks happen).
- ★ The next day **the cycle begins again**, with team members giving complete transparency into what they're doing at the Daily Scrum. This cycle continues every day until the timebox expires and the Sprint is complete.
- ★ The team **also inspects and adapts at the other Scrum events**—Sprint Planning, the Sprint Review, and the Sprint Retrospective—by reviewing and modifying the Sprint Goal, items, tasks, and the way they work.

Q: This is starting to sound *really theoretical*. Can you tie it back to the real world?

A: Sure. "Transparency" just means that every single person understands all of the features they're building in the Sprint, and are open about the work they're doing, what they plan to do next, and what problems they've run into. "Inspection" means that they constantly make sure that knowledge is up to date using the Scrum events (*especially* the Daily Scrum). And "adaptation" means that they're constantly finding ways to change what they plan to do next based on this new information.

Q: If the team plans some of the tasks but not all of them, won't that just cause chaos later on in the project?

A: No. Making all of the decisions at

there are no Dumb Questions

the beginning of the project lets you feel like everything's under control. But very often they aren't, leaving you surprised that somehow your project—which seemed fine yesterday—is suddenly late, and now everyone's in crunch mode. That's why so many teams turn to agile: because their traditionally planned projects keep blowing their schedules, and *something* needs to change. Scrum avoids those pitfalls by recognizing that many important decisions depend on information that **won't be known** until partway through the project.

Q: Can you give me an example of how that would work in real life?

A: Here's one that our *Cows Gone Wild* team might face. Brian needs to program the behavior for a new tractor vehicle for the player to use, but he can't start until

Amy finishes the basic behavior for it. If Rick used traditional project management, he would either have to assume that Amy will finish first (and add padding to the schedule in case she takes too long), or he'd have Brian work on something else first, even if the other task is less important.

Now that they're using Scrum, the team can give themselves more options. They know Brian can't start the coding for the tractor until Amy finishes designing its behavior. But they also know they'll **constantly inspect their progress and adapt the plan along the way**. So instead of deciding *today* whether Brian work on the tractor or on a less important task, they can add **both** of them to the Sprint Backlog—and they'll **put off the decision** until Brian is ready to start the code. If Amy is done with her work, he can start coding. If she's not, he'll pull the other task off of the Sprint Backlog and come back to the tractor coding when he's done (but only when he's "Done" done!).



O O

I STILL DON'T SEE
HOW SPRINT PLANNING CAN POSSIBLY
WORK. HOW CAN YOU HAVE A TIMEBOXED
PLANNING SESSION? YOU SAID THAT THE TIMEBOX
OFTEN EXPIRES BEFORE ALL OF THE WORK IS
PLANNED. DOESN'T A HALF-BAKED PLAN LEAD TO
HALF-FINISHED PROJECTS?

No—because agile teams make decisions as late as possible.

When a lot of people learn about Scrum, they're surprised that the Sprint Planning session is timeboxed to four hours for a two-week Sprint (sized proportionally for different Sprint lengths), because they're accustomed to having every single task in the project completely planned out before any of the work can begin. But we've already seen that teams following a traditional waterfall process often have trouble when their plans change partway through the project. In fact, changes that would deliver more value often get rejected during the change control process simply because replanning a months-long schedule is too much work.

Scrum teams rarely (if ever!) run into this problem because they don't plan every single task at the very beginning of the project. In fact, usually they don't even plan out all of the tasks at the beginning of the Sprint. Instead of planning everything up front, they make decisions at the **last responsible moment**. What that means is that they only need to do the planning that's absolutely necessary to get started with the Sprint. If more planning needs to be done, it can be done later in the Sprint.



This is a new way of thinking about planning for a lot of teams. Luckily, we have the Agile Manifesto to help our teams get into a mindset that's really effective.

MINI



Sharpen your pencil

The Agile Manifesto is really useful because it helps get into a mindset where concepts like the last responsible moment *really make sense*. One of the twelve Agile Manifesto principles is especially helpful for understanding the last responsible moment. Write down which one you think it is—you can see our answer on page 98.



The Daily Scrum Way Up Close

The “ceremony”

Even though a lot of teams refer to answering the Daily Scrum questions as a “ceremony,” every single person is engaged and paying attention.

The whole team gets together at the same time every day, and most teams start with a different person each time. Everyone (including the Product Owner and Scrum Master) answers three questions:

- ★ What did I do **yesterday** to move us toward the Sprint Goal?
- ★ What will I do **today** to help meet the Sprint Goal?
- ★ Are there any **impediments** that prevent me from meeting the Sprint Goal?

Each person’s answers are brief and to the point because the meeting is timeboxed to 15 minutes.

Inspect and adapt

The reason that each person answers those three questions is to give complete **transparency** into what he or she is doing. But this only works if every other person on the team is listening carefully (which is why it’s important to keep the updates brief!). One of the most common things that happens during a Daily Scrum is that one team member realizes his or her teammate is going to do something that doesn’t quite make sense: maybe they’re starting one Sprint Backlog item when a different one is more valuable, or taking one approach when there’s a better alternative, or has hit a roadblock that other team members can help with.

When this happens, they’ll set up a meeting later in the day to talk about it. More often than not, they’ll have a discussion that causes them to change the plan: they may take a different approach, or choose another item from the Sprint Backlog, or have to add more work (and possibly remove a different item from the Sprint Backlog) to get around the roadblock. This is how the team **adapts** to the change.

THIS IS THE THREE PILLARS AGAIN, RIGHT? ANSWERING THE QUESTIONS IS THE INSPECTION PART, AND WHEN THEY MAKE CHANGES THAT'S THE ADAPTATION PART.



That's right. And it all works because of transparency.

Each team member answers the questions every day, so everyone has current, accurate information about the project. Not to get too theoretical, but this is actually a really good example of **empirical process control theory**, which tells us that when a process (in this case, an agile approach) is based on empiricism, it lets the team optimize the way they work to reduce risk and give them sustainable, predictable results.

DICTIONARY DEFINITION

em-pir-i-cism, noun

the theory that knowledge comes from experience, and that decisions are to be made based on information that is known

*The team was driven by **empiricism**, and rejected the project plan based on guesswork.*

The Agile Manifesto helps you really “get” Scrum

We learned back in Chapter 2 that the most effective way to approach any agile methodology, framework, method, or approach—like Scrum, for instance—is with a mindset that’s driven by the values and principles of the Agile Manifesto. Let’s take a closer look at **three of those principles** that are especially helpful for Scrum teams.



THERE'S NO WAY I CAN
ACCEPT THE HAY THRESHER BATTLE AS
DONE UNTIL YOU GUYS GET THAT MILK GUN
WORKING.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

A really important word in this principle is “valuable.” Everyone on the team takes it really seriously, and does their best to deliver the most value that they can.

The Product Owner makes sure the team delivers value

That’s the whole reason that the team needs a Product Owner with the authority to accept items in the Sprint Backlog as Done—or, if they’re not “Done” done, refuse to accept them. If everyone on the *Cows Gone Wild 5* team really “gets” this principle, then they won’t be mad at Alex for not accepting this feature as done yet, because they genuinely want to deliver the most **value** that they can. They want to deliver it **early** and **continuously**, and the best way to do that is to finish the current item (which means it’s “Done” and ready to demo to the users in the Sprint Review) so that they can move on to the next one and get as many backlog items done in the Sprint as they can.



Under the Hood: The Sprint Review

The Sprint Review is all about maximizing value

During the Sprint Review, the team works with key users and stakeholders invited by the Product Owner to inspect the Increment and the Product Backlog, and everyone understands the goal is to maximize value by reviewing the value delivered during the Sprint, and maximizing the value of the items for the next Sprint. Here’s how they do it:

- The Product Owner goes over what was “Done” in the Sprint and the team demonstrates the working software.
- The team talks about what went well and what could be improved, and answers user and stakeholder questions.
- The Product Owner walks everyone through the current Product Backlog, and everyone collaborates on what they think should probably go into the next Sprint, so the team learns directly from the users and stakeholders **what they think is going to be most valuable**. This is really important for planning the next Sprint.
- They’ll have an open and honest discussion of anything that’s changed in the marketplace (in case it changes the most valuable thing to do next), the company’s timeline and budget, and anything else that’s relevant.

Here's Amy's update at the Daily Scrum.

I FINISHED THE CHICKEN COOP AIR ASSAULT ARTWORK, SO THAT'S ANOTHER SPRINT BACKLOG ITEM DONE. THERE ARE A FEW ITEMS I CAN CHOOSE TO DO NEXT. I WAS THINKING I'D WORK ON THE HORSE CORRAL BATTLE LEVEL DESIGN.

I'M NOT SURE THAT'S THE RIGHT THING TO WORK ON NEXT. CAN WE MEET RIGHT AFTER THIS DAILY SCRUM IS OVER AND TALK IT THROUGH?

Sounds like Amy may be going down the wrong track. If Brian hadn't been paying attention, he might not have caught this potential problem. Now they can figure it out together.

The best architectures, requirements, and designs emerge from self-organizing teams.

Self-organizing means deciding as a team what to work on next

If you've worked on a traditional waterfall team with a project manager, then you've probably worked on tasks that were part of a project plan created by a project manager and assigned to you by either the project manager or your boss. But that's not how Scrum teams work. Scrum teams are **self-organizing**, which is a new way to work for a lot of people.

The whole Scrum team works together to plan the project. There isn't a single person building a plan and telling them what to do. The Development Team members decide what will be delivered, adding new work to the Sprint Backlog as needed. The whole team decides together how they'll meet those goals.

But self-organization doesn't just happen during Sprint Planning. They **constantly adapt their plan** by checking in with each other at the Daily Scrum: each person tells the whole team what they're planning on doing next to meet the goals they decided on during Sprint Planning. If a teammate sees a problem with the approach, they'll work together that day to fix it.

The Daily Scrum is timeboxed, so when two team members discover an issue like this, they'll meet afterward to work it out (inviting other team members to join if they have input). Brian and Amy will meet and talk about how to handle this situation. Once they've got a new plan, they'll review it at the next Daily Scrum to make sure everyone is on board, and to see if anyone has anything to add.

What did you get for your



MINI Sharpen your pencil

answer on page 94? Flip the page to find out ours... ➔



Did you come up with a different answer? This is the agile principle we think is most helpful for understanding the last responsible moment, but other principles can shed light on it too!

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Making decisions at the last responsible moment probably sounds really weird if you've never done it before. But to someone who *genuinely welcomes changing requirements*, it really does feel as normal as riding a bike.

This is a new way of thinking about planning for a lot of teams.

That's part of the mindset shift that we talked about in the last chapter. Here's how it works on a Scrum project in the real world:

- ★ There's **just enough** written down about each item in the Product Backlog to be able to start Sprint Planning.
- ★ During Sprint Planning, the team decomposes enough of the Sprint Backlog to get everyone started, but they **don't feel like they need to decompose everything**. (Have a look at the Sprint Planning section in the Scrum Guide, which says the team decomposes *work planned for the first days of the Sprint* by the end of the meeting.)
- ★ Self-organizing teams don't need to decide the exact order of every single task in painstaking detail when they're planning a Sprint. They **trust themselves** to make good decisions when the time comes.
- ★ As the team works through the Sprint Backlog over the course of the Sprint, they discover new tasks and changes and bring them up in the Daily Scrum, and they use that information to work together to **create a plan for the next 24 hours**.
- ★ They're **constantly inspecting and adapting**, so they trust themselves to make good decisions in the future.

Scrum teams make decisions at the last responsible moment.

They only make project decisions that need to be made now, and leave the rest for later.



THE MAD COW ZOMBIE
LEVEL WE BUILT JUST ISN'T AS FUN AS WE
THOUGHT IT WOULD BE, AND GAMERS WON'T LIKE IT. LET'S
PUSH IT BACK TO THE PRODUCT BACKLOG. MAYBE WE'LL PUT
IT OUT AS DOWNLOADABLE CONTENT AFTER THE
GAME'S RELEASED.

It's a good thing Alex is comfortable making decisions at the last responsible moment! In this case, the only way the team could find out the feature wouldn't be fun (and deliver the value the project needs from it) was to build it. Notice how he's staying positive, not blaming anyone for wasting effort, and keeping the option open to release it later.



Watch it!

The Daily Scrum is timeboxed to 15 minutes, so make sure everyone keeps their updates focused and to the point.

That's easier said than done. Once you and your team start holding a Daily Scrum every day, you'll discover that some people are really uncomfortable talking about their work in front of everyone else on the team, while others just can't seem to stop talking, and will take up the entire 15 minute timebox if you let them.

This is why it's really important that the Scrum Master takes his or her role seriously—especially the responsibility of making sure that the team understands and adheres to the Scrum rules:

- *If someone is reluctant to speak up, the Scrum Master can help that team member to accept the Scrum value of openness and understand how transparency is critical for making Scrum work.*
- *When team members give updates that take up too much of the Daily Scrum, the Scrum Master can show them which facts are relevant and help them manage their Daily Scrum time better.*
- *If a team member gives an update that goes beyond just answering the three questions and raises issues for discussion, the Scrum Master can remind the team to set up a separate meeting for some of the team members to discuss the issue and report back to the whole group.*

BULLET POINTS

- The team agrees on the **definition of “Done”** for every item in the Sprint Backlog and for the entire increment.
- The Product Owner doesn't accept an item to include in the Sprint Review until it's **“Done”** (i.e., it meets the definition of “Done” that the team decided on for it).
- Scrum uses **empirical process control**, based on the **three pillars** of *transparency, inspection, and adaptation* to make decisions based on real facts.
- **Transparency** (or visibility) means everyone on the team understands what their teammates are doing.
- **Inspection** means they frequently check with each other on what they're building and how they're building it at the Daily Scrum and the other Sprint events.
- The team constantly **adapts** their plan based on what they learn from that inspection.
- Agile teams make decisions at the **last responsible moment**, and plan only those tasks that absolutely need to be planned right now.
- The team delivers value **early** (by working on each backlog item until it's done) and **continuously** (by delivering a complete Increment that's “Done” at each Sprint Review).
- Scrum teams are **self-organizing**: they decide as a team how to meet their goals and who will do the work.
- Scrum teams **welcome changing requirements** more easily than traditional waterfall teams because they're self-organizing and make decisions as late as possible.

A bunch of Scrum artifacts, events, and roles are playing a party game, “Who am I?” They’ll give you a clue, and you try to guess who they are based on what they say. Write down its name, and what kind of thing it is (like whether it’s an event, role, etc.).

And watch out—another Scrum concept that’s not an event, artifact, or role might just show up and crash the party!

I’m a servant leader who guides the team in understanding and implementing Scrum, and helps people outside of the team grasp it.

I’m held at the end of the Sprint to do an inspection of each item that the team built with users and stakeholders who were invited.

I’m how the team inspects itself, where they look for the things that went well, and make a plan to improve things that didn’t.

I’m the sum of all items that the team delivers to the users at the end of the Sprint, and I can only be delivered if every item in me is “Done.”

I’m the group of professionals who actually do all of the work that’s needed to deliver the software to the users and stakeholders.

I’m responsible for deciding what items will go into the product, and I have the authority to accept them as “Done” on behalf of the company.

I’m a 15-minute timeboxed meeting held every day, where team members create a plan for the next 24 hours.

I’m what the Product Owner helps the team optimize and maximize, and the team tries to prioritize the items that have the most of me.

I’m the set of items the team will build during the Sprint, along with a plan to build them (usually a set of tasks the items are decomposed into).

I’m a timeboxed meeting where the team comes up with a Sprint Goal, decides which items they’ll deliver, and decomposes them into tasks.

I’m an ordered list of all of the items (with descriptions, estimates, and values) that might be needed in the product at some time in the future.

Who am I?



Name _____

Kind of thing _____

there are no Dumb Questions

Q: Is the *Cows Gone Wild* team's story realistic? Can you really use Scrum for something like a video game, which requires a lot of creativity and on-the-fly—and sometimes last-minute—changes with heavy deadline pressure?

A: Not only can Scrum be used for a complex and dynamic project that constantly changes, it's actually better suited to an environment like that than a traditional waterfall process. Scrum teams constantly look for changes and find ways to adapt to them, which makes them much better at dealing with complexity and even chaos. And the timeboxed nature of Sprints helps the team meet their deadlines. Alex just showed us a good example of the kind of decision that video game teams make in real life. The team built a feature but it turned out not to be fun enough in its current form, so they shelved it for now and might revisit it to sell later as downloadable content. Scrum gave the team the flexibility to handle this on the fly, while a traditional waterfall team would probably have to go through a lengthy change control process. More importantly, a Scrum team **sees this change as a victory** because they'll welcome any change that delivers more value to the product. A traditional waterfall team is likely to see it as a defeat because it "wasted" effort and required a change to the plan.

Q: Does "self-organizing team" mean that there's no boss?

A: Of course there's a boss. If you work in a company and you're not the CEO, then you have a boss. But an effective self-organizing team typically has a manager who doesn't micromanage, and who trusts all of his or her employees to deliver the most valuable software they can. Self-organizing teams are given the authority to decide which features to include in the software, usually by having a Product

Owner assigned to the team who's senior enough to make those decisions. They're given the freedom to plan the work so that they can build those features in the way they feel is most effective. And they're given the flexibility to make decisions at the last responsible moment, because that's the most effective time to make important project decisions.

Q: What exactly happens during the Sprint Retrospective?

A: The Sprint Retrospective is how the team inspects the Sprint that just ended and tries to find ways that they can improve. They look at all sorts of things: the people on the team can improve the processes and tools that they use to do the job, find ways to improve the quality of the software they're building, work on their relationships with others in the organization, and do anything else that might have an impact on the work—especially anything that can make their work more enjoyable or effective. By the time the Sprint Retrospective ends, the team has put together a plan for improvement. This plan typically consists of a small number of discrete and specific tasks that individual people on the team will carry out. Before the meeting, the Scrum Master helps everyone understand how it works, and makes sure that they all respect the timebox. This happens *before* the meeting because the Scrum Master and Product Owner have to participate in the retrospective as team members, offering their own opinions and ideas.

Q: Hold on—the Product Owner goes to the retrospective too? Does the Product Owner really need to be at *all* of the Scrum events?

A: Absolutely. The Product Owner is a real member of the team, and participates in the Scrum events just like everyone else. In fact, on a lot of Scrum teams the

Product Owner does his or her share of the development work. But even in that case, that person still has the authority to make decisions about what items will go into the backlog and how to maximize the value of what the team delivers, and the company still respects his or her decisions.

Q: I'm having trouble finding anyone who has the clout to be the Product Owner but also has enough time to attend all of the Scrum events. Can the Product Owner be a committee instead?

A: Absolutely not. The Product Owner definitely **must** be a person. That person **must** have the authority to make the decisions about what goes into the software and what doesn't, and *the Product Owner role must be his or her top priority*.

Modifying Scrum like this almost always renders it *much less effective*, usually by removing the parts that make its empirical process control work. Teams often try to "customize" Scrum by bending or breaking any rules that highlight a serious flaw in the way their team is structured. In this case, Scrum makes it really obvious that the team doesn't have the authority to decide what features go into the software. It's **scary** for a manager to say, "The team should build this feature, but not that one." The wrong decision will cost the company a lot of money, and **there will be a lot of blame** if it goes wrong. That's why Scrum requires the team to have a Product Owner who has the authority to make that call.

Teams often try to "customize" Scrum when they don't have a true Product Owner who can make tough decisions about what to build.

demo went great we're halfway there

Things are looking good for the team

Cows Gone Wild 5 is this year's most anticipated video game release!
Now they just have to get it out the door. (Easier said than done?)



GREAT NEWS! I DEMOED THE
HAY THRESHER LEVEL AT THE WISCONSIN
GAME CONFERENCE LAST WEEK AND WE
TOTALLY KILLED IT!



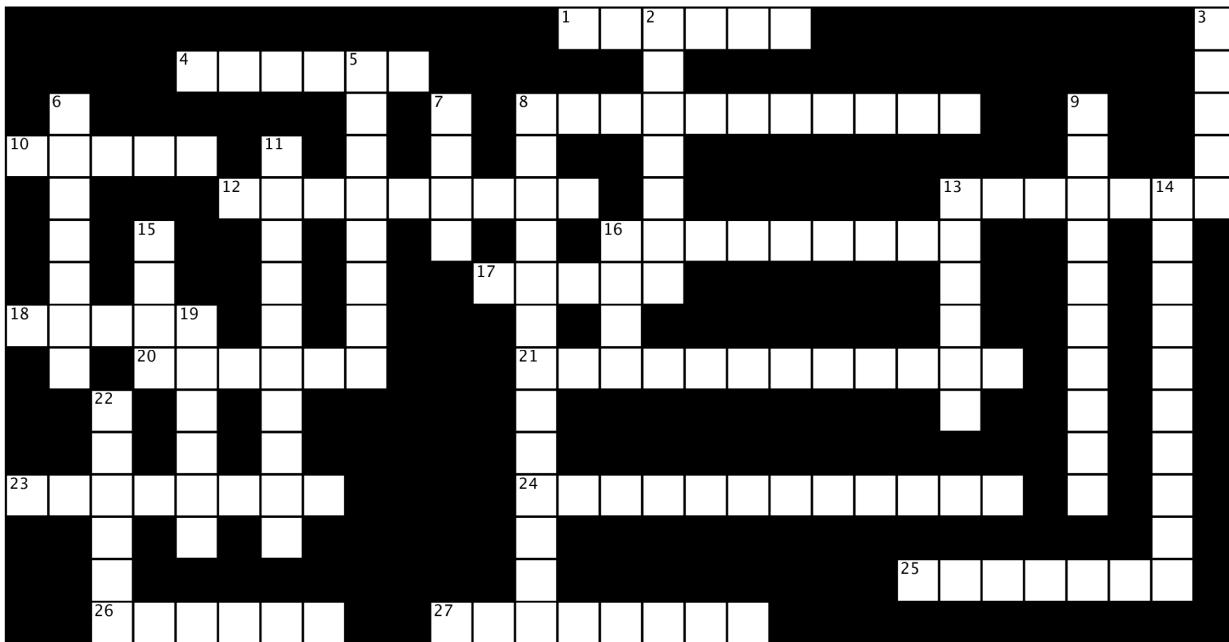
YEAH! I WAS JUST READING SOME
GREAT REVIEWS ON THE TOP TWO GAMER
BLOGS FROM PEOPLE WHO ATTENDED. LOOKS LIKE
CGW5 IS THE MOST ANTICIPATED GAME OF
THE YEAR!





Scrumcross

Here's a great opportunity to seal the Scrum concepts, values, and ideas into your brain. See how many answers you can get without flipping back to the rest of the chapter.



Across

1. The Sprint, Sprint Planning, the Daily Scrum, the Sprint Review, and the Sprint Retrospective
4. The Product Owner is allowed to _____ the Sprint, but it wastes the team's energy and damages the company's trust in the team
8. The most effective time to make a decision is at the last _____ moment
10. The Product Owner makes sure the team maximizes this
12. The Product Backlog, the Sprint Backlog, and the Increment
13. When this value isn't part of your mindset, you don't trust your teammates, and tend to blame them when things go wrong
16. What the team does to turn the Sprint Backlog items into tasks
17. Product Owner, Scrum Master, and Development Team
18. When this is part of your mindset, you don't even consider trying to do two things at once
20. What Product Owners routinely do to keep the Product Backlog current
21. Responsible for maximizing the value of the product and managing the Product Backlog
23. During Sprint _____ the team chooses what items to include in the Sprint and builds a plan
24. The three Daily Scrum questions give the team complete _____ into what each team member is doing
25. The _____ Backlog is the single source of requirements and changes for the product
26. The timeboxed iteration used in Scrum

Down

2. The Sprint is over when the timebox _____
3. The team decides what can be done in the Sprint during the _____ part of the Sprint Planning session
5. Each item in the backlog has a description, the business value, an order, and a rough _____
6. The Sprint _____ contains items the team will build during the Sprint
7. The Sprint _____ is an objective crafted by the team when they plan the Sprint
8. The Sprint _____ is how the team inspects itself and creates a plan to improve
9. The theory that knowledge comes from experience
11. A self _____ team decides as a team how they'll meet their goals
13. The Sprint _____ is how the team inspects what they built and adapts the Product Backlog
14. What pigs have, chickens don't, and everyone on a Scrum team feels collectively toward the whole project
15. The Sprint Review is timeboxed to _____ hours for a 30-day sprint
16. What teams do with working software at the Sprint Review
19. The team decides how the work will be done during the _____ part of the Sprint Planning session

Exam Questions

These practice exam questions will help you review the material in this chapter. You should still try answering them even if you're not using this book to prepare for the PMI-ACP certification. It's a great way to figure out what you do and don't know, which helps get the material into your brain more quickly.

1. The Scrum Master is responsible for all of the following except:

- A. Helping the team understand what goes on during the Daily Scrum
- B. Giving the Product Owner guidance in effectively managing the Product Backlog
- C. Helping the team understand customer requirements
- D. Giving the rest of the organization guidance on understanding Scrum and working with the team

2. Which of the following is NOT an attribute of a Product Backlog item?

- A. Status
- B. Value
- C. Estimate
- D. Order

3. Juliette is a Product Owner on a Scrum project in a healthcare organization. She was called into a meeting with a steering committee made up of her company's senior managers because she decided to include a planned health privacy feature in the most recent Sprint. At the meeting, the senior managers told her in the future that she must consult with the whole committee before making business decisions like that.

Which of the following BEST describes Juliette's role?

- A. She is in a servant-leadership role
- B. She is not committed to the project
- C. She needs to concentrate on focus and courage
- D. She does not have the authority to adequately fill her Product Owner role

4. When is the Increment considered done?

- A. When the timebox expires
- B. When every item to be delivered meets its definition of "Done" and the Product Owner accepts it
- C. When the team holds the Sprint Review and demonstrates it to the users and stakeholders
- D. When the team holds the Sprint Retrospective

Exam Questions

5. Which of the following is an example of collective commitment?

- A. Everyone on the team feels personally responsible for delivering the entire Increment, not just their individual parts of it
- B. Everyone on the team always stays late and often works weekends
- C. Everyone on the team is responsible for delivering an important part of the project
- D. Everyone on the team participates in the Sprint Planning and retrospective meetings

6. Which of the following is NOT a Scrum event?

- A. Sprint Review
- B. Product Backlog
- C. Retrospective
- D. Daily Scrum

7. Amina is a Scrum Master on a team that is working on adopting Scrum. She wants to make a change to help her team get better at self-organizing. Which of the following is the best area to focus their improvement effort?

- A. Daily Scrum
- B. Sprint Planning
- C. Sprint Retrospective
- D. Product Backlog

8. When is a Scrum Sprint over?

- A. When the team finishes the work
- B. When the team completes the Sprint Retrospective
- C. When the timebox expires
- D. When the team completes the Sprint Review

Exam Questions

9. Each person on the team answers all of the following questions during the Daily Scrum except:

- A. What roadblocks are in my way?
- B. What planned work did I fail to accomplish?
- C. What will I do between now and the next Daily Scrum to meet the Sprint Goal?
- D. What have I done since the last Daily Scrum?

10. Barry is a developer at an online retailer. His project manager told him the deadline for the current feature that he's working on is three weeks from now, even though Barry made it clear that he would need four weeks, and there were no specific deadlines or external pressures that require it to be done earlier than that. Barry's team is starting to adopt Scrum. Which of the Scrum values will make the team's Scrum adoption difficult or less effective?

- A. Openness
- B. Respect
- C. Courage
- D. Focus

11. Sandeep is a product owner on a Scrum team working on a telecommunications project. The business users let him know about a major regulatory change in one of his regular meetings with them. Handling this regulatory change is now a very high priority for the team, and will need to be the main objective of the next Sprint.

Which of the following is used to describe the main objective of the next Sprint?

- A. The Increment
- B. The Sprint Backlog
- C. The Sprint Goal
- D. The Sprint plan

12. What aspect of empirical process control theory involves frequently examining the different Scrum artifacts and making sure the team is still on track to meet the current goal?

- A. Examination
- B. Adaptation
- C. Transparency
- D. Inspection

Exam Questions

13. What is an Increment in Scrum?

- A. The items from the Sprint Backlog that the team actually completes during the Sprint
- B. The items from the Product Backlog that the team plans to complete during the Sprint
- C. The result of decomposing the Sprint Backlog items
- D. A statement that describes the objective of the Sprint

14. Which of the following helps Scrum teams focus?

- A. Multitasking
- B. Holding a Daily Scrum
- C. Writing a Sprint Goal
- D. Holding a retrospective

15. Danielle is a Product Owner on a Scrum team. She's talking to one of her business users, who gives her a new requirement. Which of the following should Danielle do next?

- A. Update the Product Backlog
- B. Hold a Sprint Planning session
- C. Update the Sprint Backlog
- D. Bring up the new requirement at the next Daily Scrum

16. Which of the following BEST describes how the team determines what specific work will be needed to complete Sprint Backlog items?

- A. The Product Owner works with the business users to determine which items go into the Product Backlog
- B. The team decomposes Sprint Backlog items into tasks
- C. The team chooses which Product Backlog items to include in the Sprint Backlog
- D. The team decides on each Sprint Backlog item's definition of "Done"

17. Which of the following does NOT take place during a Sprint Review?

- A. The Product Backlog is updated to reflect what will probably be in the next Sprint
- B. The team collaborates with business users on what they will work on next
- C. The working software the team built during the Sprint is demonstrated
- D. The team looks back at the Sprint and creates a plan to improve

~~Answers~~~~Exam Questions~~

Here are the answers to this chapter's practice exam questions. How many did you get right? If you got one wrong, that's OK—it's worth taking the time to flip back and re-read the relevant part of the chapter so that you understand what's going on.

1. Answer: C

It's the Product Owner's job to help the team understand customer requirements, not the Scrum Master's. The other three answers are good examples of the Scrum Master's servant leadership role.

2. Answer: A

The Product Backlog does not contain any information about the status of a task. This makes sense—none of the items on in the Product Backlog are currently being worked on, so they all have the same status of not having been started yet.

3. Answer: D

 Take a minute and read through the description of the Product Backlog in the Scrum Guide. It says that Product Backlog items have the attributes of a description, order, estimate, and value.

One of the most common problems that Scrum teams run into is that the person in the Product Owner role does not have the authority to decide on behalf of the company what features the team will build during the Sprint, or accept them as done on behalf of the company.

4. Answer: B

The Increment is done when every item to be delivered by the team meets its definition of "Done" and the Product Owner accepts it. Every item in the Sprint Backlog has a definition of "Done" that the team uses to determine when it's ready to release to the users. The Product Owner can only accept an item on behalf of the company if it meets its definition of "Done"—any item that isn't "Done" done when the timebox expires must be pushed to the next Sprint.

5. Answer: A

Collective commitment means that everyone on the team feels a personal sense of responsibility to deliver not just the piece that he or she is working on, but to do what it takes to help the team deliver the whole Increment at each Sprint.

 Just because everyone on a team works long hours, that doesn't mean they feel a genuine commitment to it. In fact, they might resent the project and the organization for interfering with their lives, and only work the extra hours out of pressure to keep their jobs.

~~Answers~~~~Exam Questions~~

6. Answer: B

The Product Backlog is a Scrum artifact, not an event.

7. Answer: A

Self-organizing teams take responsibility for their own planning to meet their objectives, assign work themselves (rather than depending on a single manager or project manager to make those assignments), and fix problems with the plan as they come up. Of all of the practices listed as answers, the Daily Scrum is the only one that impacts the way the team plans their work and executes that plan.

8. Answer: C

One reason the Daily Scrum is so important is that it's part of the transparency-inspection-adaptation cycle. The team inspects the plan every day, and adjusts it as they uncover new information about the project.

The Sprint is over when the timebox expires. The same is true of any timeboxed event. Answer D seems correct because the Sprint Retrospective is typically the last thing that the team does during the Sprint. But if the timebox expires before the team has a chance to hold their retrospective, the Sprint still ends. (And that's a good opportunity for the Scrum Master to help them understand how to plan better next time.)

9. Answer: B

The Product Owner is allowed to cancel the Sprint before the timebox is over. But this can waste a lot of the team's energy, and cause people in the company to lose trust in the team, so it should be extremely rare.

The purpose of the Daily Scrum questions is to give everyone a good idea of how each person on the team is progressing, so they can help identify problems with the current plan that need to be fixed. But none of the questions are about failures—that could create a negative and possibly embarrassing environment, and detract from the environment of openness.

10. Answer: B

The project manager is having trouble with the Scrum value of respect. Barry gave an honest assessment of the work that needed to be done, but the project manager ignored it and demanded a shorter deadline even though there was no business need to apply the extra pressure. That's disrespectful.

It's also extremely demotivating!

11. Answer: C

When the team holds their Sprint Planning meeting at the start of the Sprint, the first thing they do is to decide on the Sprint Goal, a brief description of the objective of the Sprint that will be met by completing backlog items.

Answers

~~Exam Questions~~

12. Answer: D

The core of empirical process control theory—the theoretical underpinning for Scrum—is the “three pillars” cycle of transparency, inspection, and adaptation. In the inspection step, the Scrum team members frequently examine the Scrum artifacts, as well as their current progress toward the Sprint Goal. They try to detect any differences between where they are versus where they expected to be, so that they can take action (which is what adaptation is all about).

13. Answer: A

The Increment is what the team actually delivered during the Sprint. The items that the team intended to complete at the beginning of the Sprint often don’t exactly match up with the work they actually did. That’s a good thing—it means the team used the information they learned along the way to change direction. The Increment is the product of what actually happened, and the team doesn’t know exactly what the current Sprint’s Increment will contain until they’ve delivered it.

14. Answer: B

The Sprint Goal helps the team focus on the specific objective that they planned on accomplishing during the Sprint.

We learned earlier that Scrum takes an incremental approach. It's the delivery of successive Increments that makes it incremental.

Retrospectives and Daily Scrums can be very useful, but holding meetings is not typically a tool that teams use to help focus.

15. Answer: A

The Product Backlog is the single source for product requirements, and it's maintained by the Product Owner. When the Product Owner discovers a new requirement, she adds it to the Product Backlog.

Answers~~Exam Questions~~

16. Answer: B

Scrum teams determine what work will be done during the Sprint to complete Sprint Backlog items by decomposing them into tasks. The other answers are also things that the team does during Sprint Planning, but it's not how the team determines what work they're going to do.

17. Answer: D

During the Sprint Review, the team meets with the business users and customers to review what they've done and collaborate on what the next Sprint will accomplish. They'll review the Increment, which typically involves demonstrating the working software that they built. They'll also discuss the backlog and update it to show the items that they'll probably work on in the next Sprint. The Sprint Review isn't for looking back at what happened and making improvements—that's what the Sprint Retrospective is for.

The updated backlog only reflects the probable items that will be worked on during the next Sprint. That's not the same thing as committing to build certain items—the team will come up with the Sprint Backlog during Sprint Planning, and the Product Owner can make changes to it during the Sprint.



DID YOU GET SOME OF THE QUESTIONS WRONG? THAT'S ABSOLUTELY OK! JUST KEEP TRACK OF THEM, AND MAKE SURE TO GO BACK AND RE-READ THE PARTS OF THE CHAPTER THAT COVERED THEM.

When you get a question wrong now, that actually makes it more likely that you'll get a question on the same topic right when you take the exam!



Sharpen your pencil Solution

Here are the five Scrum events, three Scrum roles, and three Scrum artifacts. Remember, each event is timeboxed, but the length of the timebox changes proportionally if the team uses a shorter Sprint.

Event names in the order they occur

When the events happen

Length of the event's timebox

Sprint

throughout the project

Assume that the Sprint is timeboxed to 30 days

Sprint Planning

beginning of the Sprint

8 hours

Daily Scrum

every day

15 minutes

Sprint Review

end of the Sprint

4 hours

retrospective

after the Sprint Review

3 hours

Write down the Scrum roles

Write down the Scrum artifacts

Scrum Master

Sprint Backlog

Product Owner

Product Backlog

Development Team

Increment

