

Scrum 101: A Pocket Guide

Table of Contents

About 3Back.....	1
Introduction	2
A Bit of HiStory	3
Roles	5
Scrum Roles:Who Does What.....	6
The Scrum Team.....	6
Internal Roles	6
External Roles	9
Time to Reflect.....	10
Events.....	11
Sprint Planning.....	12
The Daily Standup.....	14
Sprint Review	15
Sprint Retrospective	16
The Big Four.....	16
Artifacts.....	17
The Product Backlog.....	18
The Sprint Backlog (aka The Front Burner).....	19
The BuildUp Chart	20
The Right Tool for the Job.....	20
Summary	21
Scrum 101:We've Only Just Begun.....	22
References	23
Primary References.....	24
Additional Notes and Sources	24

About 3Back

We are 3Back; a Well-Formed Team of Experts in Agile Practices. Our expertise lies in several areas. We are Certified Scrum Trainers, Coaches, Consultants, Practitioners, and Authors. We are a Team that is both energized and intrigued by Scrum. We are a Team that is passionate about transforming Scrum fundamentals into what we call Modern Scrum. Put simply, we focus on practical implementation of Scrum principles that make Teams better.

Our approach to Modern Scrum is born out of the real world. Each member of our Team brings a proven blend of knowledge, skills, and tested techniques to the Team environment. Our training, coaching, and consulting is grounded in exploring client-driven experiences and in our company's own practice and process. We don't just talk Scrum, we live Scrum; applying its principles in our own workplace. Our approach, like the foundation of Scrum, is agile and adaptive; reflecting the specific needs of your Team. If your needs change, we adjust our focus and draw on our expertise to tailor a solution for your Team's success. The result is meaningful, hands-on, adult learning focused on improving your Team's efficiency, effectiveness, and empowerment through Scrum.

Scrum 101 draws from the seasoned voices of 3Back Team Members Doug Shimp, Dan Rawsthorne, Marcelo Lopez, and Jan Beaver who, amongst them, have over 50 years of experience with Scrum. Each contributing author adding their own insight to best capture what those new to the world of Scrum need to hit the ground (or their first Sprint) running.


To learn more about the Scrum 101 authors and the 3Back Team, visit 3back.com/team.



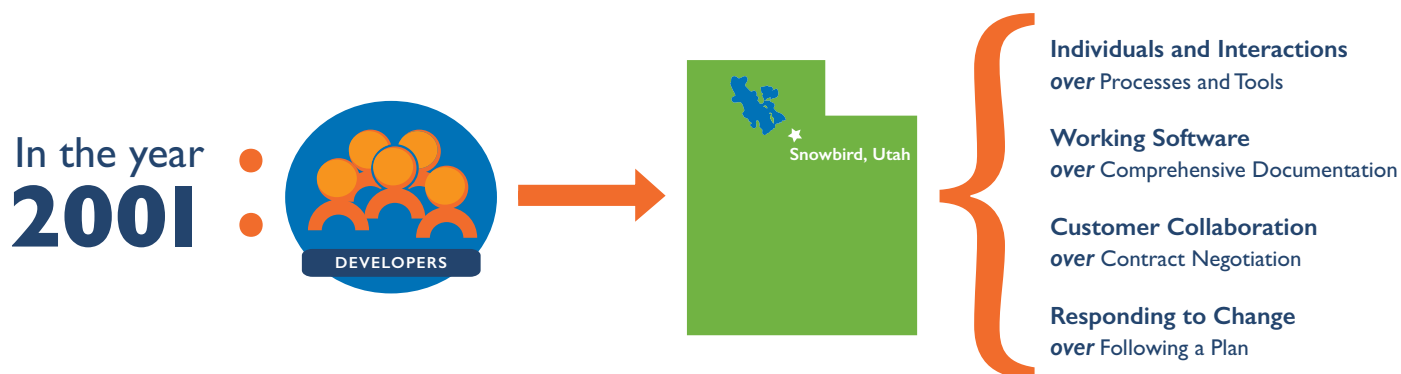
Introduction

For anyone who is new to the world of Scrum – its foundations, its principles, its nuances – there's a lot to digest. That's the inspiration behind this ebook, to cut through some of the high-level, intellectual, book smart aspects of Scrum and, well, tell it like it is. At a high level, Scrum is lightweight and easy to understand, but when you dig in you will find that it is difficult to master¹.

Scrum is a framework or model; it is not a process. Scrum does not tell you how to do things, it tells you *what* needs to be done and lets you figure out how to do it. To make it even more confusing, Scrum is not literal; you must modify what it says to match your circumstances. Scrum is a well-balanced framework, all its parts are needed in order to be effective. The bottom line is that acquiring and using Scrum's benefits is hard work and requires discipline and dedicated practice. You need to work hard to become Scrummish...

To assist you during your journey to Scrummishness, we've taken the liberty of highlighting some Key Scrum Knowledge Nuggets. When you see the skeleton key , you've found a Key Scrum Knowledge Nugget. We encourage you to use Scrum 101 as a back pocket guide to prepare you for Certified ScrumMaster (CSM) or Product Owner (CSPO) Training. If you're a Scrum newbie, it will get you ready to enter your first Sprint, or, if you're a little seasoned in the Scrum world, serve as a gentle refresher of your Scrum knowledge base.

A Bit of HiStory



Scrum was born out of the software development industry as an agile methodology to counter established waterfall-style project management processes. Jeff Sutherland originated the first Scrum project in 1993. Sutherland, working with Ken Schwaber, developed Scrum as a formal process in 1995. In 2001, Sutherland and Schwaber, along with several pioneers of agile thinking converged at a ski resort in Utah to assess commonalities in agile methods. The Agile Manifesto was created out of this group's consensus.

The [Agile Manifesto](#) details some fundamental agile philosophies, one of which is a preference for Empirical Process control – which maintains that knowledge is derived from experience and decision making is based on what is known. Scrum is an Empirical Process based on inspection, adaptation, and transparency. gave the name “Agile” to the movement.



Scrum is a lightweight framework for project management that is used to complete complex work.

The Agile Manifesto also discusses the importance of people, Product, and feedback. Scrum is consistent with the Manifesto, as its core elements are: deliver working Products every sprint, inspect and adapt every day, and trust the Team. Although Agile methods had been in practice for some time, and its philosophical roots date back 200 years, the Agile Manifesto successfully gave the name “Agile” to the movement. Scrum has changed a lot since 1993. Modern Scrum, as we know it today, takes the underpinnings of the Agile Manifesto and puts it into a usable reality.

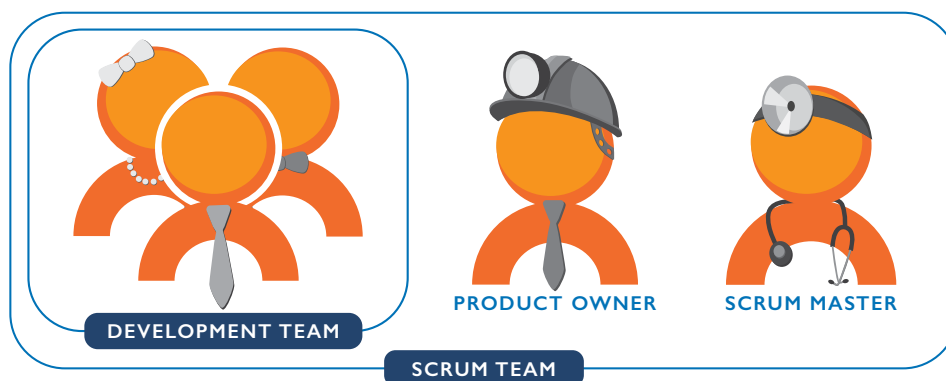
Let's cut to the chase... Scrum is a lightweight framework describing how a single Scrum Team does complex work by incrementally developing a Product, Sprint-to-Sprint. Every Sprint, Stakeholders provide feedback on these Product Increments so that the Scrum Team can improve, adapt, and extend them in order to converge on, and release, the Product the Stakeholders want and need.

Typically, Scrum is practiced in the software development realm, but its effectiveness in tackling innovative work is catching on, spreading to non-IT industries worldwide. Scrum offers Teams real-world ways to apply the Agile philosophy to, simply put, get the job done right. So what does that look like to the Scrum freshmen class? It all starts with getting a grasp on who does what, aka Scrum Roles.

Roles

Scrum Roles: Who Does What

There are two sets of roles involved in Scrum: roles that are internal to the Scrum Team and roles that are external to the Scrum Team. Our journey begins with a general description of the Scrum Team itself, moves to a discussion of the three roles inside the Team (the Product Owner, the ScrumMaster and the Development Team), and finishes with a description of roles outside the Team (the Business Owner, Stakeholders, and Subject Matter Experts).



The Scrum Team

A Scrum Team (commonly called the “Team”) is small (3 to 9 not including the ScrumMaster and Product Owner), co-located (at least virtually), self-organized, self-contained, value-driven, full-time group of people called, simply, Team Members. Some of these terms need defining:

Self-Organized: A self-organized Team is one that chooses how best to accomplish its work, rather than being directed (micro-managed) by others outside the Team. Since the Team Members work together, this facilitates learning and motivates the Team to take ownership of its process.

Self-Contained: A self-contained (or cross-functional) Team is one that contains all the knowledge and skills necessary to accomplish its objectives and goals, which allows it to finish its work without outside help.

Value-Driven: The Team Members value working together; they are constantly improving themselves, their Team, their environment, and their tools; and they strive to have personal values such as Openness, Focus, Commitment, Respect, and Courage.

Internal Roles

The Product Owner



Every member of the Scrum Team plays the role of Team Member; but only one Team Member is held accountable to the Business for the Scrum Team's success and the value of the Scrum Team's results. That's the Product Owner or PO, for short. And that accountability is a big thing – it defines the PO as the formal leader of the Team as far as the outside world is concerned.



The Product Owner is accountable for the value of the Scrum Team's results.

The PO is the Scrum Team's eyes and ears to the outside world (to the Stakeholders). He or she is the Scrum Team's one point of formal contact, the conduit of information. Add to this that the PO has the Scrum Team's back. Meaning, in being held accountable for the Scrum Team's results, the PO is consumed with making sure the Scrum Team is getting the right feedback to make the right Product at the right pace. The PO spends a lot of time scoping the Product, clarifying murky expectations, negotiating delivery dates, and making it all fit for the Scrum Team.

We should also note that even though the word 'Owner' is in the title, the PO may not be the Product expert. Granted, the PO has a whole lot of knowledge and skills, but the role is defined by the accountability, not by Product-specific skills. A good PO realizes there may be a wealth of smarts both inside and outside the Scrum Team and knows how to leverage this for both the Scrum Team's good and the Product's good.

The ScrumMaster



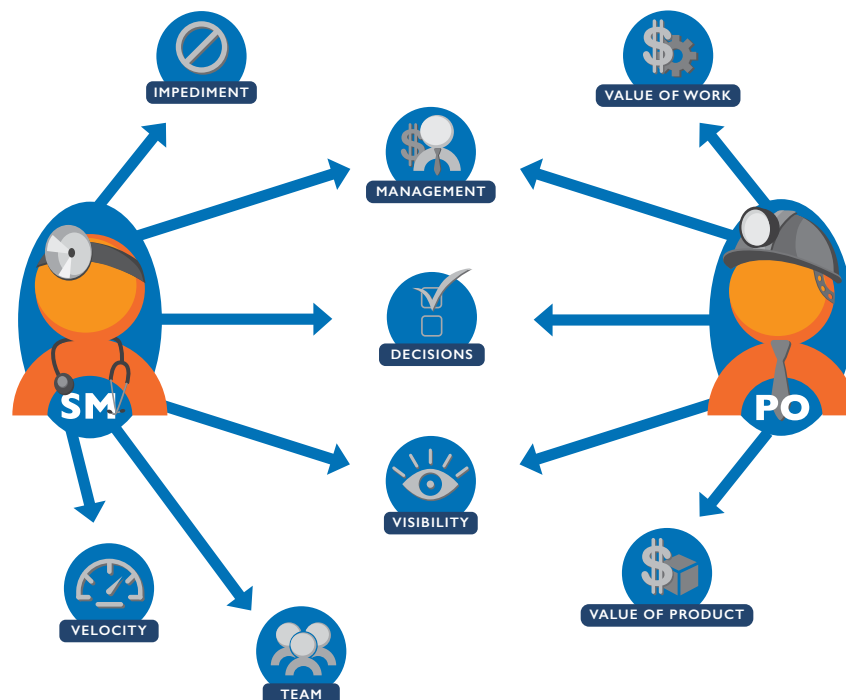
While the PO is the eyes and ears toward the outside world, in many ways, the ScrumMaster's eyes and ears are pointed decidedly inward. The ScrumMaster (SM) is an informal leader worried about what's going on internally within the Scrum Team and making sure that Scrum is being used correctly. Part facilitator, part influencer, part guru, the ScrumMaster is a leader without managerial responsibilities. Rather, he or she is laser focused on the health of the Scrum Team and the Scrum Team's continuous improvement, especially when it comes to the Scrum Team's use of Scrum.

The ScrumMaster does this by helping improve the Scrum Team behaviors and working relationships, removing stumbling blocks (in Scrum Speak called 'impediments') and dealing with constraints. Many people new to Scrum quickly understand getting to know their ScrumMaster is essential to feeling grounded and Productive.

Because the ScrumMaster is inward-looking, and the Product Owner is outward-facing, it is inappropriate for the same Team Member to play both the Product Owner and ScrumMaster roles. In fact, most Scrum experts believe it would be harmful to the Scrum Team if this were to happen.



The ScrumMaster is responsible for the health of the Team.



The Scrummaster and Product Owner are accountable for many things.

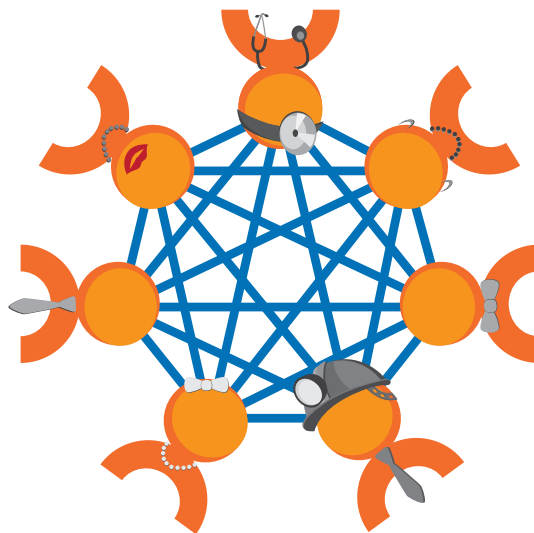
The Development Team



The term “Development Team” is used to represent the portion of the Scrum Team that is currently developing or creating the Product – and this may, or may not, include the PO and SM. It is entirely proper, and often useful, for the PO and SM to be on the Development Team, but they must always realize that their leadership roles come first.

You can think of the Development Team a little like the Three Musketeers, operating by the motto, “All for one and one for all.” When Scrum is really working well, and Teams are performing at a high level, the Development Team acts in the Development Team’s best interest, with everyone doing their part to achieve a collective goal, with each member being held responsible for the Production of the Development Team’s Product.

Unlike many work Teams, Development Teams function on a level playing field. There are no specialized roles. It’s as straightforward as it sounds; every Team Member is just that, a Team Member. Development Teams tend to be small, ideally around five people. However, at times a project may dictate a slightly larger group. Every Team Member shares the same primary goal of partnering with the other Team Members to deliver the right Product to the best of their abilities. Every Team Member is also invested in the same secondary goal to help avoid spinning its wheels and improve as a self-organizing, cross-functional group.



All Team Members are individually accountable to each other for doing good work.



The Scrum Team is comprised of the Product Owner, ScrumMaster and Development Team. The Development Team is the portion of the Team that creates the product.

External Roles

The three most important external roles are a sort of triumvirate of Scrum. The external roles of Business Owner (BO), Stakeholder, and Subject Matter Expert (SME) maintain an important level of influence in the Scrum process.

Business Owner



The role of the Business Owner (BO) sets up the framework for the Scrum Team's relationship with the organization. The BO is the person the PO is accountable to for the work results of the Scrum Team. To this end, the BO frequently plays the part of "making things happen" by providing assistance and necessary resources to the Team through the

PO. The BO also works with the PO to weigh in on the priority of future work in the Backlog, assist in altering the Release Plan if needed and help the PO better understand the needs of the Stakeholders. On an organizational level, the BO works closely with the ScrumMaster to help alleviate organizational impediments that get in the way of the Scrum Team's success.



The Business Owner is usually the 'lead' Stakeholder, the Team's Sponsor, or the Product Owner's Product Owner.

The Stakeholder



We can think about a Stakeholder as anyone who is invested in the project or the Product. To this point, there are two types of Stakeholders: internal and external. Every Team member is an Internal Stakeholder but, for the sake of today's journey, we will focus our attention on the external Stakeholders, a.k.a. "The Big Kahunas."

If it weren't for the external Stakeholders, there wouldn't be a Scrum Team producing a Product. There will always be a lot of people interested in the Scrum Team's Product; however, it's only the Stakeholders who truly have a vested interest in the Product. It's important for the Scrum Team to recognize the Scrum "food chain" here and give the Stakeholders their props. It's important for the Team to try their hardest to satisfy the Stakeholders' needs and work with the PO and ScrumMaster when things get in the way of realizing these needs.



While stakeholders can be internal members of the Scrum Team, the term Stakeholder usually refers to external stakeholders.

The Subject Matter Expert



As we've noted above, Scrum Teams are cross-functional groups who are knowledgeable about a lot of things, but there may be times when the Scrum Team doesn't know everything about everything it needs to know. And that's why Scrum utilizes Subject Matter Experts (SMEs). SMEs bring something special to the Team that it doesn't possess on its own. Perhaps it's technical

knowledge, business knowledge, database expertise, or a host of other things that are needed to make the right Product the right way. A SME is a consultant in every ounce of the word. He or she imparts know-how to the Scrum Team and walks away. No accountability or responsibility. With that comes a bit of risk. Despite the SME being chock full of critical expertise, he or she does not share the Scrum Team's priorities and goals. Scrum Teams should follow some sage advice when using SMEs: try to treat them as part of your Team, but remember and respect that they may have priorities that pull them in a different direction.



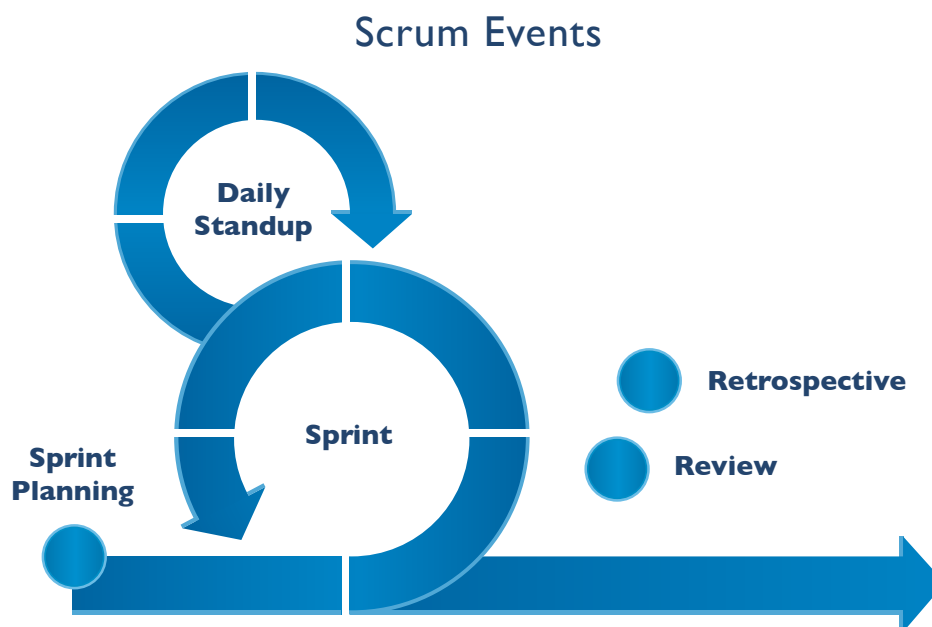
The term Subject Matter Expert (or SME) typically refers to those outside the Team.

Time to Reflect

As you continue down the road of mastering Scrum, take the time to reflect on the Scrum Team and the internal and external roles. It's one thing to read about it. It's completely another to experience it in action. Pause, observe and reflect on what you see with the Scrum roles in your world. Before you know it, you'll be honing your Scrum skills with a keen eye and taking big steps toward reaching the ultimate goal of becoming a Well-Formed Team.

Events

Now that we've given you a big picture view of Scrum and touched on some nitty gritty details of Scrum roles, we take on the topic of Scrum Events, sometimes referred to as Ceremonies – where Agility, collaboration, and feedback intersect to form a pathway for improvement. We will focus on the four Events that make Scrum distinctively Scrum: Sprint Planning, the Daily Standup (sometimes referred to as the Daily Scrum or Daily SyncUp), Sprint Review, and Sprint Retrospective.



Sprint Planning

Before we start, let's discuss what a Sprint is, just so that we're all on the same page. The fundamental process flow of Scrum is a Sprint. A Sprint, by definition, is a timebox of one month or less during which a Product is created. (In general, a timebox is a defined period of time during which tasks must be accomplished; they are commonly used to manage risk, and all Scrum events are timeboxed.) Two weeks is a common Sprint duration which we will use in our examples.

If we think of a Sprint as a run of a Broadway performance and the Scrum Team as the stars of the show, then Sprint Planning is the culmination of all the stuff that happens behind the scenes before the curtain goes up on opening night – the casting, the set design, the stage blocking, etc. In Scrum Speak, this “behind the scenes” work is referred to as “Refinement and Planning.” Sprint Planning, as the last act of Refinement and Planning, is a formal discussion where the Scrum Team commits to a Sprint Goal and agrees to take on certain Stories within the upcoming Sprint.

Sounds pretty simple. But, what does that mean in action? It means that at the get-go, Sprint Planning is timeboxed. For a two week sprint, it should last no more than two hours. We've all been in meetings that have lasted too long, draining everyone's energy and creative output. Sprint Planning is no different. If done well, as with all things Scrum, Sprint Planning will achieve a rhythm, and that rhythm will demand to wrap it up upon nearing two hours.



A Sprint is a fixed period of time (or timebox) during which development on a product occurs.



Backlog Refinement is critical for successful Planning. For greatest success, the backlog should be well-refined before beginning Sprint Planning.

So what happens during the magical two-hour time slot? It begins with the Team discussing the upcoming Sprint's goals and priorities. Should the Sprint Goal be determined at the beginning of Sprint Planning? Not exactly. It's not uncommon for Teams to waste time arguing on the specifics of the Sprint Goal at the beginning of Sprint Planning. So we've found it useful to simply discuss it and then agree to let the actual goal setting sit for a bit. The Sprint Goal, like fresh cream, will naturally rise to the surface by the end of Sprint Planning.

After goals are discussed but not determined, one-by-one the Product Owner selects a single Story from the Back Burner (otherwise known in overly simplified, non-Scrum terms as picking an item from the work to-do list). The PO discusses the Story and works with the Team to negotiate and finalize the Definition of Done for the Story. Definition of Done, you ask? That's a very critical slice of the Sprint Planning pie. In a nutshell, or rather a pie shell, the Definition of Done, or simply DoD for short, is the shared agreement about what it means for this Story to be finished; and is stated in as objective and clear terms as possible.

The Definition of Done has two parts: the Acceptance Criteria and the Standard of Care. The Acceptance Criteria defines what the Stakeholders will "see" (or "get") from the Story, and the Standard of Care is a description of how the Team will provide the necessary technical quality for the work. The Standard of Care is usually common across multiple Stories, and evolves through time as the Team learns better how to do its work. Having a Standard of Care is very important, as meeting the Standard of Care prevents the Team from sacrificing quality when put under time pressures – which is a definite 'no-no.'

This is way more than a quick nod that says, "Yah. No worries. We got it." Rather, in Scrum, the Team owns its effort. The Team decides and agrees to how much work it can do in each Sprint. The Team may not know how long a Story will take, but they negotiate with the PO what "Done" means for each Story so that they can fully understand the effort involved and when to stop working on it. Through the Definition of Done, the Team can feel confident in their ability to get the Story to Done while still doing quality work.

Here's a quick, dramatic reenactment of how the Sprint Planning process might go.

PO: "Ok, Team, here's the first Story - it's heavy on the database work."

(PO plus Team negotiates DoD, Team agrees to do the Story.)

Team: "Ok, What's next?"

PO: "Here's another database heavy one..."

(PO plus Team negotiates DoD, Team agrees to do the Story.)

Team: "Ok, what's next?... but no more database heavy Stories; we're full up..."

PO: "Well, the next Story is another database one, so I'll skip to the Story below that... This one is a new business rule."

(PO plus Team negotiates DoD, Team agrees to do the Story.)

Team: "Got it. We have room for one more Story, but it's got to be a small one. And remember, no database heavy ones..."

PO: "Alright, how about this 'clean up the interface' one?"

(PO plus Team negotiates DoD, Team starts a discussion...)

Team: "Sorry, that won't fit. We're done."

PO: "Okay."



The Doneness Agreement is an agreement between the Product Owner and the rest of the Team that defines when a Story will be complete (or Done).

After several of these types of conversations, the Stories for the Sprint are established, Definitions of Done are complete, and the Sprint Goal has pretty much written itself. Badda Bing, Badda Bang. You've completed Sprint Planning and are ready to start the Sprint.

The Daily Standup



- 1 What have you done since the last Daily Standup?
- 2 What are you going to do until the next Daily Standup?
- 3 What impediments are standing in your way?

The Daily Standup

To the non-Scrummified world, the Daily Standup might appear to be a simple Team status update. This is not the case. The reality of this meeting gets to the heart of practicing Scrum. In a word, the Daily Standup is for one thing: coordination.

In Scrum terms, the Daily Standup gives a Scrum Team an allotted and preserved time for face time and planning with each other. In real terms, this looks like a 15-minute Team meeting every morning with the added bonus of increasing communication and visibility.

There are a couple of critical things that need to happen to make the Daily Standup as effective as possible.

1. The entire Team should be present. That includes the PO and ScrumMaster in their Team Member roles.
2. Everyone should stand. Why? It keeps the meeting short. Plus, standing naturally engages the Team, keeping them more on their toes, so to speak. Standing also has the added benefit of forcing people to think before they talk, which lends itself to more focused dialogue rather than long-winded, self-absorbed soliloquies.
3. The Daily Standup typically follows a standard format to help keep the Team on track. A common format for the Daily Standup includes each Team Member taking a turn answering these questions:
 - What have you done since the last Daily Standup?
 - What are you going to do until the next Daily Standup?
 - What impediments are standing in your way?

At times it may be helpful to add a question to address items that may need further discussion, perhaps in a sidebar, such as:

- Is there anything else we need to talk about?



The Daily Standup gives a Scrum Team an allotted and preserved time to make sure everyone stays on the same page.



A sidebar is a conversation which is taken 'offline'. It often occurs following the Daily Standup and only includes those Team Members to which the topic is pertinent.

4. The PO or ScrumMaster keeps a list of sidebar discussions that emerge from the above questions. If something needs to be done with the information that is gathered, separate discussions often occur after the Daily Standup. Only Team Members that need to be involved in the follow-up discussion should attend. The Daily Standup is not the time for the entire Team to be involved, or rather be trapped, in someone else's discussion.

The Daily Standup is a fundamental conversation that discovers the Team's current realities and takes immediate and appropriate steps to do what needs to be done to move the work in the right direction.



Sprint Review

The Scrum Team has been cohesively and diligently working, and the Sprint ends. Now what? It's time for the Sprint Review.

The Sprint Review is a natural result of the Sprint and a uniquely Scrummy opportunity to gather feedback from Stakeholders. At this timeboxed 2-hour meeting, the Scrum Team shows what they have accomplished during the Sprint. Meaning, the project is assessed against the Sprint Goal that was determined during the Sprint Planning meeting.

Who do they show their work to? They show it to the people that matter the most, the Stakeholders. True, it can initially feel daunting to present to the Big Kahunas. But who better to get that feedback from in order to move the Team forward in performance and make sure the right Product is being built?

Who else is present at the Sprint Review? The PO, the ScrumMaster, the Development Team, and (if possible) any Subject Matter Experts who are helping the Team build the Product. The PO plays a big role during this meeting, acting as a true owner of the Team's work Product and presenting the results. The ScrumMaster facilitates any dialogue. And the Scrum Team has a rare and valuable opportunity to be face to face with the Stakeholders, learning how to receive feedback and, with every Sprint, gaining a solid understanding of the people for whom they are developing the Product.

The outcome of the Sprint Review for the Scrum Team? A sense of pride in their work, feeling invigorated for the next Sprint, and most importantly... owning and incorporating feedback that will impact their work in future Sprints.



During the Sprint Review, the Scrum Team demonstrates what they have accomplished during the Sprint.

Sprint Retrospective



Sprint Retrospective

There's a famous quote from philosopher George Santayana that hits the point of the Sprint Retrospective on the head: "Those who fail to learn from hiStory are doomed to repeat it."[†] Granted, this quote is a bit negative, focusing on the doom of repeating mistakes. But whisk off some of that negativity and focus on the learning piece and you have the importance of the Sprint Retrospective, to learn from the good things so we can repeat them and learn from the bad so we can improve immediately. In Scrum Speak, this is often referred to as "failing fast."

So what happens during the Sprint Retrospective? The entire Team, including the ScrumMaster and the PO, gather after the close of a Sprint for a 2-hour time boxed meeting to discuss and agree upon what went well and ways they could improve their practices, Teamwork, environment or organization for the next Sprint. It is important to note that the Sprint Retrospective is not a whine session. It is essential the Team spend adequate time identifying what worked well so that the Team doesn't start sliding backward. Only after the Team knows what it doesn't want to change can the Team start thinking about what it should change. This is the time for the ScrumMaster to facilitate the conversation. Although we strongly recommend using more robust Retrospective activities, a simple tactic to gather and process input is to ask each Team Member to write down two to three things that went well and two to three things that didn't go well. The ScrumMaster then guides a timeboxed discussion of the items.

Once issues have surfaced to improve upon for the next Sprint, the ScrumMaster works with the Team to move forward with all types of changes. Changes may be within the Team's control. For example, changing how the Team room is set up. Sometimes changes are outside of the Team's control and in these cases, the ScrumMaster and the PO step in to work with the Business Owner and other Stakeholders for assistance.

The Sprint Retrospective, if facilitated well, should foster a feeling of Team unity, empowerment and rejuvenation, gearing up the group for an even better next Sprint.

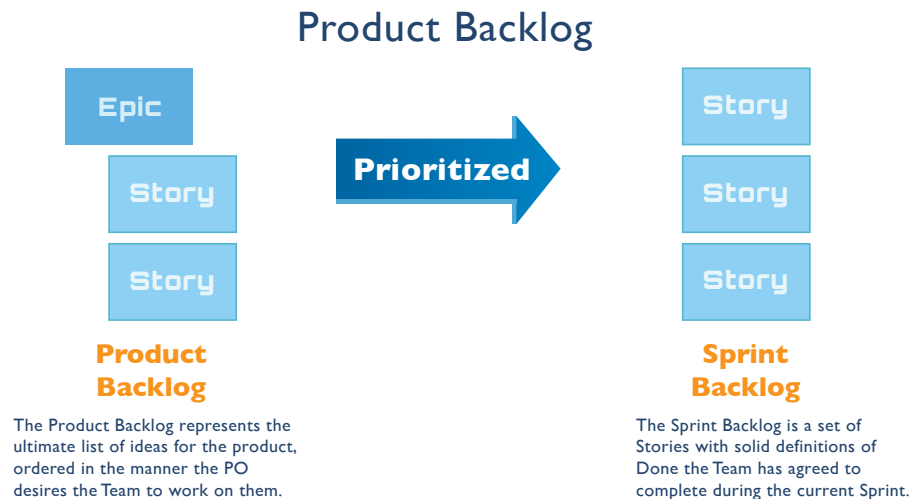
The Big Four

These four Events: Sprint Planning, the Daily Standup, Sprint Review and Sprint Retrospective lay the groundwork for Scrum. Successful Scrum Teams actively engage in each step of the process, fostering greater agility, performance, and satisfaction.

[†]George Santayana, *The Life of Reason* (1905-1906), Vol I. *Reason in Common Sense*

Artifacts

We've talked Scrum Roles. We've talked Scrum Events. Both of these topics give you a solid context of what Scrum looks like in action. But what about having the right tools to get the job done? That's where Scrum Artifacts come in. The main artifact is, of course, the Product Increment being produced each Sprint. In order to develop and manage the Product Increment, we will be exploring three other Scrum Artifacts: the Product Backlog, the Sprint Backlog and the BuildUp Chart. Each tool, or Artifact, helps the Scrum Team get their work done in a way that makes sense to the whole Team by being both transparent and organized in an agreed upon manner. We'll start by examining the Product Backlog.



The Product Backlog

In a sense, the Product Backlog is the Granddaddy of Scrum Laundry Lists. It represents the ultimate list of ideas for the Product, ordered in the manner the PO desires the Team to work on them. The Product Backlog captures all work, from bug fixes to documentation to whatever the Team thinks may be needed for the Product. Logistically, Teams are encouraged to maintain one Backlog, so they are not confused by what list to work from. Practically, it serves to help manage the work the Scrum Team is doing or will be doing. You can think of it this way: if it isn't in the Backlog, it doesn't exist.

Every item from the Product Backlog has a description and, eventually, a Definition of Done. If the item is small enough that the Team could actually do it all at once, it is referred to in Scrum Speak as a "Story." If an item is too big to be done all at once, then it is called an "Epic," meaning it is in fact made up of several Stories. The Product Backlog truly is a "living artifact," in that it is changed, added to, reworked and revised constantly. Frequently, the Product Backlog may start out as a fairly vague list, but under the guidance of the PO, it will get whittled down into small, well-defined chunks.

In Scrum Speak, this is called "Product Backlog Refinement." This process is also sometimes referred to as "Grooming." Whether you call it Refinement or Grooming, both refer to an ongoing process of reviewing Product Backlog items and ensuring they are accurately prioritized and defined in a clear and deliverable manner for the Scrum Team, ensuring a more streamlined Sprint Planning. When all is said and done, this process leads to stories that are "Ready" to be worked on.



The Product Backlog represents the ultimate list of ideas for the product, ordered in the manner the PO desires the Team to work on them.



Product Backlog Refinement/ Grooming is the ongoing process of reviewing product backlog items and ensuring they are accurately prioritized and clearly defined.

Typically, Refinement takes up to 10% of the Scrum Team's total effort, and involves the Product Owner, the Development Team, and appropriate Subject Matter Experts. Everyone is there to bring clarity to the Stories being discussed, with the goal being a well-stated and understood Definition of Done for the Stories that will be done "soon." The Development Team asks for clarifications that will help them identify dependencies, the Product Owner provides prioritization advice, and the SMEs give clarifications based on their experience. Because everybody is involved, the timing and format of Backlog Refinement is a self-organization issue decided by the Scrum Team and the SMEs.

Remember back to our discussion of Scrum Roles? The responsibility and accountability for maintaining the Product Backlog fall solidly on the PO. The Scrum Team, Business Owner, Stakeholders, and Subject Matter Experts have a lot of say in developing and refining the Product Backlog, but the PO is the one who dots the I's and crosses the T's on it. Most important to the PO is using the Product Backlog to best prioritize the Scrum Team's workflow.

There are many ways to prioritize the Product Backlog. At 3Back, we use a cooking metaphor to describe the types of prioritization buckets where Stories are placed.

- Front Burner: Stories that the Team has agreed to do in the current Sprint.
- Back Burner: High priority, well-defined Stories that are being made Ready so that they will be ready for the next Sprint Planning.
- Fridge: Items (Stories or Epics) that are in scope for the Release, but are not yet ready to be taken to Sprint Planning either due to prioritization or readiness.
- Freezer: Items that are out of the scope of the Release.
- Inbox: Items that have not yet been prioritized into a bucket (or, in this case, an appliance).
- Done: Stories that have been completed.

The Sprint Backlog (aka The Front Burner)

The Sprint Backlog is the culled-over list of Product Backlog items for development in the current Sprint. In cut and dry terms, the Sprint Backlog is a set of Stories the Team has agreed to work on during the current Sprint. Each Story in the Sprint Backlog is written with a solid Definition of Done based on a shared Team Agreement, meaning there should be no ambiguity regarding what "Done" looks like for each Story. Remember the importance of transparency as a foundational Scrum aspect that we mentioned in the earlier chapters? Knowing what "Done" looks like is all about the Scrum emphasis on transparency.



The Sprint Backlog is a set of Stories the Team has agreed to complete during the current Sprint.

The BuildUp Chart

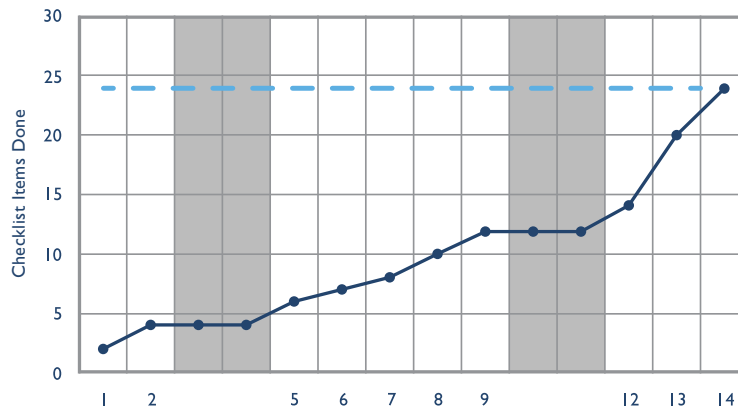
Scrum Teams often need a way to visualize progress throughout a Sprint or Release. We, at 3Back, recommend using a BuildUp Chart for this purpose. The BuildUp Chart measures how much stuff has been "Done" so far, and focuses on the Definitions of Done that were made during Sprint Planning. After all, that's what the Team actually agreed to do. And the BuildUp accurately reflects the development of the Product (not just busy work the Team may have been doing), which is something of value that the Stakeholders want to see.



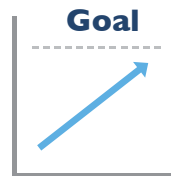
The BuildUp Chart accurately reflects the development of the product by focusing on the Doneness Agreements made during Sprint Planning.

So, how do we make a BuildUp Chart? It's simple. We convert the Definition of Done for each Story into a checklist. All items on the checklist must be checked off in order for the Story to be completed. The Team's progress can easily be measured by counting up the number of checked off items and graphing it. Chart the dates (Day 1, Day 2, Day 3 of the Sprint) on the horizontal axis against the cumulative number of completed checklist items on the vertical axis. If you want to, add the total number of checklist items for the Sprint as a line across the top, to show what the graph is climbing toward.

Doneness Checklist Item BuildUp Chart

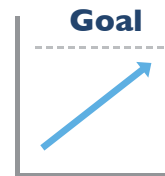


BuildUp Chart



Sprint BuildUp Chart

The Sprint BuildUp Chart's focus is on the Doneness Agreements that were made during Sprint Planning. After all, that's what the Team agreed to actually do.



Release BuildUp Chart

The Release BuildUp Chart accurately reflects development of the product which is something of value that the Stakeholders want to see.

Many Teams have found value in using a version of the BuildUp Chart made out of Stories and include the Sprint Goal. Regardless of the specifics on the BuildUp Chart, it should summarize the Sprint based on accumulated value. A good BuildUp Chart will be able to answer the Team's questions, "How much will we finish?" and "How are we doing?" Both answers will nicely help the Team and the Stakeholders inspect and adapt.

The Right Tool for the Job

The Product Backlog, the Sprint Backlog and the BuildUp Chart give the Scrum Team the right tools at the right time for the right project, enhancing the Team's collaboration and innovation through Scrum.

Summary

Scrum 101: We've Only Just Begun

So there you have it. As we said at the start, we encourage you to use this Scrum 101 ebook as your back pocket guide to prepare you for Certified ScrumMaster Training or get you ready to enter your first Sprint.

But, we've really just scratched the surface of Scrum. As you wet your feet in the Scrum pool and put your new knowledge into practice, you may find you want and need even more.

▶ **Never fear. We've got training for that.** Our Scrum knowledge runneth over. We offer public courses and customized, on-site training and consultation.

▶ **Jump on the 3Back Bandwagon** and join over 3000 people who receive [a weekly email](#) with the latest posts from our blog. Our 3Back team regularly blogs on all things Scrum, from introductory to expert and covering a wide range of topics, such as [Appropriate Sprint Goals](#), [What is a Team Swarm?](#) and [When a Team Can't Reach Their Sprint Goal](#).

▶ **Talk to us.** If you have questions, are seeking clarification or simply want to talk Scrum, [contact 3Back today](#). We have a whole Team ready to help.

If you're looking for a daily dose of Scrum find us on [Facebook](#) and [Twitter](#).

And, as always, Stay Agile.

References

Primary References

1. Ken Schwaber and Jeff Sutherland. (2016). The Scrum Guide. <http://scrumguides.org/scrum-guide.html>
2. Dan Rawsthorne and Doug Shimp. (2010). Exploring Scrum. Retrieved from <https://exploringscrum.com>
3. Scrum Dictionary. (2017). Retrieved from <https://scrumdictionary.com>

Additional Notes and Sources

- Agile Atlas. (2016). Retrieved from <https://www.scrumalliance.org/why-scrum/agile-atlas>
- David J. Anderson. (2003). Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Retrieved from <https://www.amazon.com/Agile-Management-Software-Engineering-Constraints/dp/0131424602>
- Esther Derby and Diana Larsen. (2006). Agile Retrospectives: Making Good Teams Great. Retrieved from <https://www.amazon.com/Agile-Retrospectives-Making-Teams-Great/dp/0977616649>
- Dan Rawsthorne and Doug Shimp. (2013). Exploring Scrum: the Fundamentals, 2nd Edition. Retrieved from <http://www.amazon.com/dp/1461160286>
- Dan Rawsthorne. (2013). Exploring Scrum: Patterns that Make Scrum Work. Retrieved from <https://leanpub.com/PPSAD>
- Mihaly Csikszentmihalyi. (1991). Flow: The Psychology of Optimal Experience. Retrieved from <https://www.amazon.com/Flow-Psychology-Experience-Mihaly-Csikszentmihalyi/dp/0060920432>
- 37Signals. (2006). Getting Real. Retrieved from <http://gettingreal.37signals.com>
- Luke Hohmann. (2006). Innovation Games®: Creating Breakthrough Products Through Collaborative Play, 1st Edition. Retrieved from <https://www.amazon.com/Innovation-Games-Creating-Breakthrough-Collaborative/dp/0321437292>
- Alan Mark Davis. (2005). Just Enough Requirements Management: Where Software Development Meets Marketing. Retrieved from <https://www.amazon.com/Just-Enough-Requirements-Management-Development/dp/0932633641>
- Daniel Goleman. (2006). Social Intelligence: The New Science of Human Relationships. Retrieved from <https://www.amazon.com/Social-Intelligence-Science-Human-Relationships/dp/0553803522>
- Larry Constantine and Lucy Lockwood. (1999). Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, 1st Edition. Retrieved from <https://www.amazon.com/Software-Use-Practical-Methods-Usage-Centered/dp/0201924781>
- Ilan Goldstein. (2013). Scrum Shortcuts Without Cutting Corners: Agile Tactics, Tools & Tips, 1st Edition. Retrieved from <https://www.amazon.com/Scrum-Shortcuts-without-Cutting-Corners/dp/0321822366>
- Kent Beck. (2002). Test Driven Development: By Example, 1st Edition. Retrieved from <https://www.amazon.com/Test-Driven-Development-Addison-Wesley-Signature/dp/0321146530>
- Larry Gonick. (1993). The Cartoon Guide to (Non) Communication : The Use and Misuse of Information in the Modern World. Retrieved from <https://www.amazon.com/Cartoon-Guide-Non-Communication-Information/dp/006273217X>
- Hirotaka Takeuchi and Ikujiro Nonaka. (1986). The New New Product Development Game. Retrieved from <https://hbr.org/1986/01/the-new-new-product-development-game>
- Stephen Covey. (2004). The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change. Retrieved from <https://www.amazon.com/Habits-Highly-Effective-People/dp/0743269519>
- Eliyahu Goldratt. (1999). Theory of Constraints. Retrieved from <https://www.amazon.com/Theory-Constraints-Eliyahu-M-Goldratt/dp/0884271668>
- Mike Cohn. (2004). User Stories Applied: For Agile Software Development. Retrieved from <https://www.amazon.com/User-Stories-Applied-Development-Addison-Wesley/dp/0321205685>