

SCRUM3.0

A WHITE PAPER FROM 3BACK, LLC

Authors: Dan Rawsthorne
Douglas Shimp

Abstract

Scrum is Scrum, and it's a beautiful thing. Unfortunately, Scrum has been described/defined in many different, inconsistent and incoherent ways. Scrum 3.0 is a description of Scrum that both unifies and extends older descriptions of Scrum, and is easier to relate to real world scenarios. Scrum 3.0 is the cleanest, most flexible and most agile version of Scrum there is...

Table of Contents

Introduction	4
Scrum 1.0 Overview	6
Scrum 2.0 Overview	7
Scrum 3.0 Overview	9
Purpose of this White Paper	10
Description of Scrum 3.0.....	11
Scrum's Roles.....	11
Scrum's Artifacts.....	13
Scrum's Process.....	15
Conclusion.....	19
Discussion.....	20
References.....	22
Primary References.....	22
Additional Notes and Sources.....	23
About the Authors	24

Required Reading:

[The Well-Formed Team™ white paper³](#)

[Agility white paper⁴](#)

[Leadership white paper⁵](#)

[Done white paper⁶](#)

[Scaling white paper⁷](#)

Only **10%** of organizations are Achieving The Promise of Scrum (3X or more increase in performance). Top Reasons for Failure: partial adoption of Scrum, hybridization with other methods, failing to embrace it at all organizational levels.⁸

Statistics on Scrum Adoption:

82% doing Scrum

35% Clean Code Practice (XP/TDD)

20% Hybrid Methods like Waterfall and SAFe.

Scrum is a beautiful, small-team development process model. Through time, Teams using Scrum found that the Product Owner role needed to evolve and change in order for Scrum to be successful - Scrum had to evolve.

We decided to document a version of this evolved Scrum that is:

- a logical extension of The Scrum Guide™
- maximally agile
- easily understood
- scalable through fractalization

While we were at it, since Agility is 'adapting to, and exploiting chaos', we took advantage of this opportunity and improved the evolved Scrum model and provide it here as **Scrum 3.0**.

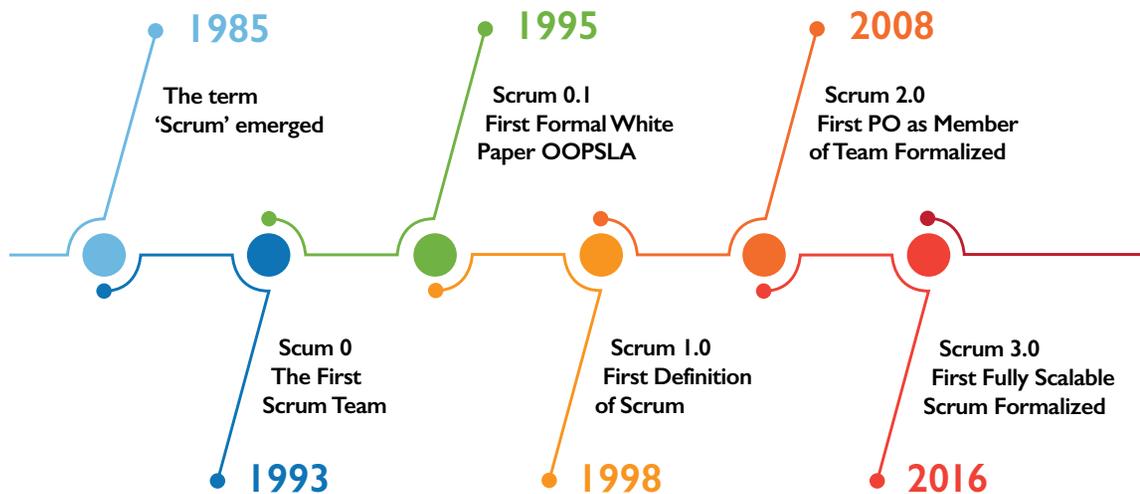
Introduction

We both fell in love with Scrum in the late 1990's. It was clean, elegant, agile, and echoed all the good Teams we'd seen in our lives: in sports, in the military, and on the construction site. At its core, Scrum simply says:

- have a Team that gets its work done all by itself, and is not at the mercy of others;
- have a Product Owner who is the single-source of requirements, so there is no bickering or arguing about what to do;
- have a Scrum Master who improves things and removes Impediments; and
- have multiple, frequent, feedback loops in order to be appropriately agile.

What a beautiful concept!

Unfortunately, most Organizations don't care about concepts. They want to know how to make it work — they want to know how to build Scrum Teams — at their core, they want a recipe or explicit instruction. So we (the Scrum Community) had to develop Frameworks, Guidance, and Rules about Scrum. We had to tailor Scrum to the Organizations where it lived. And we found that Product Ownership, in particular, was very difficult to implement in a consistent way. The concept was clear — be a single-source of requirements — but the implementation was not. We have observed three significant variations in Product Ownership over the years, which we refer to as Scrum 1.0, Scrum 2.0, and Scrum 3.0. Let us give a brief overview of each.

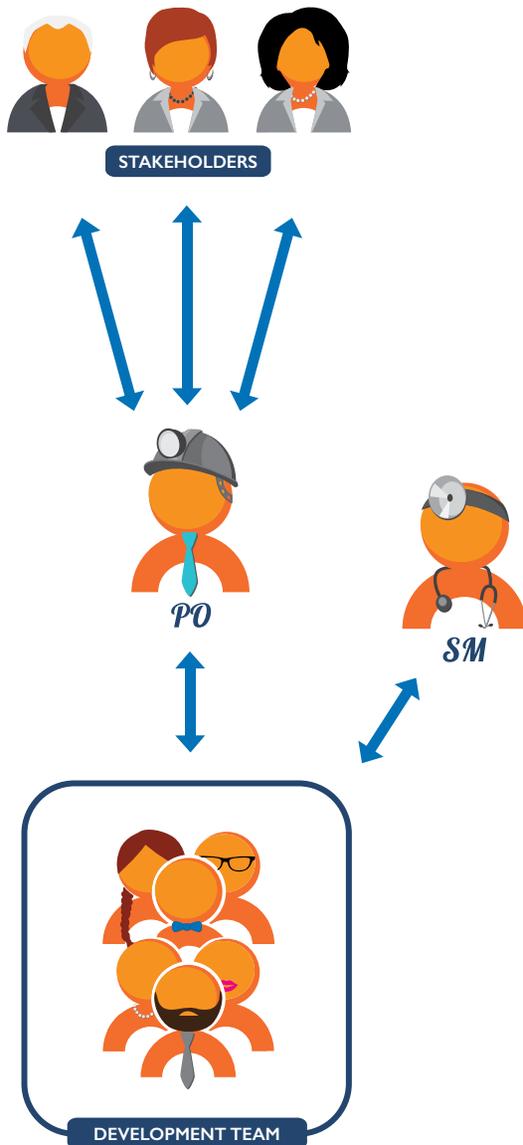


Scrum 1.0

Scrum 2.0

Scrum 3.0 - Most Evolved

This white paper will discuss Scrum 3.0 in further depth.



Scrum I.0 Overview

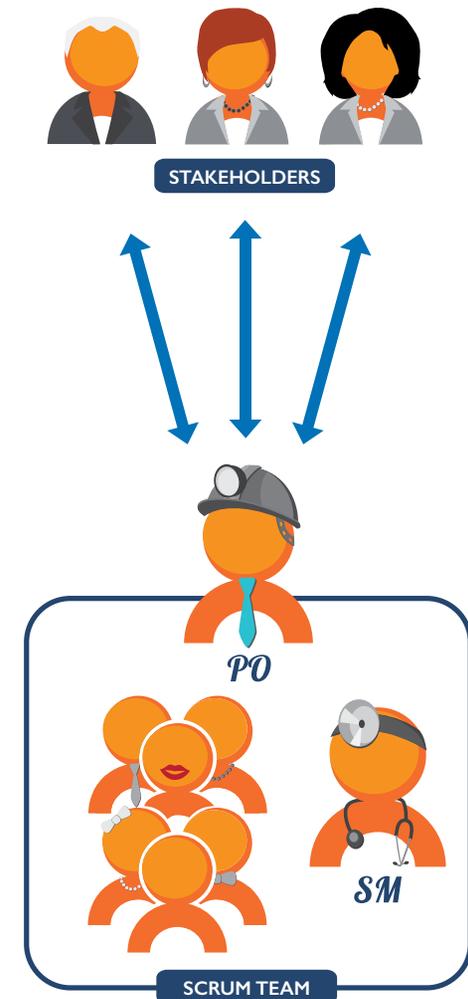
The first implementations of Scrum had a Product Owner who was outside the Team — often on the “Business Side” of the Organization. This Product Owner prioritized requirements at the beginning of a Sprint at a fairly high functional level, and then basically left the Team alone to develop — the Product Owner was only agile at the Sprint boundaries. This style works well for Software Product Development when the functional Requirements are emergent and not overly volatile. This type of Scrum is still around, still being useful. View the diagram to the left.

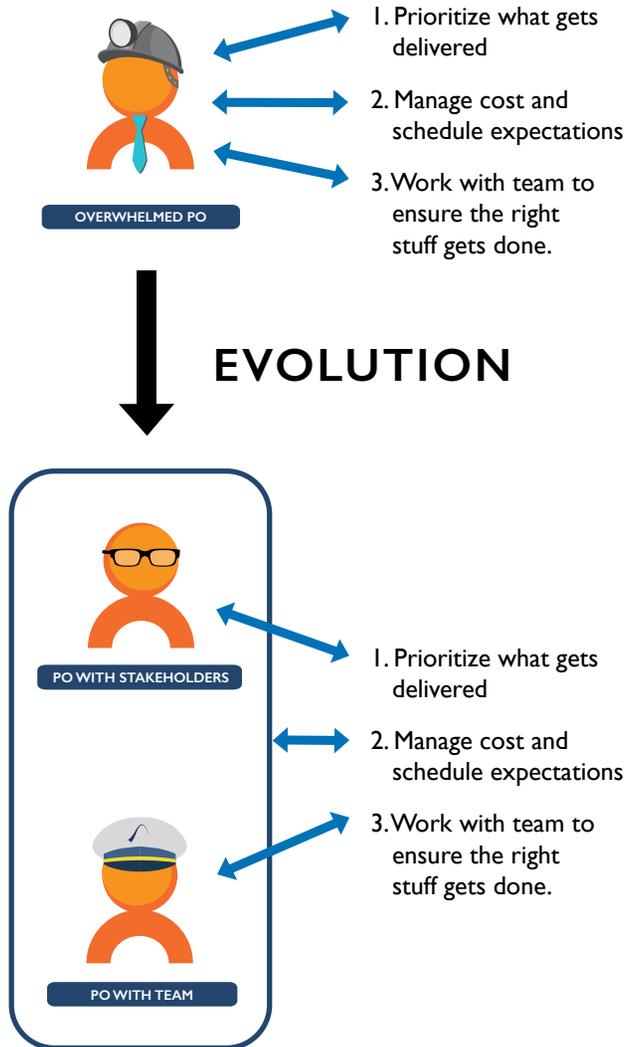
We refer to this type of Scrum as Scrum I.0, and one of its popular variants is described in the Scrum Primer⁹.

Scrum 2.0 Overview

There is often the need for the Team to not only build software, but maintain the software once it has been deployed. This requires the Team to be interrupted frequently to resolve time-sensitive issues related to bugs, building, testing or deployment that come up in Operations. The Team needs to know: “Which is more important, the new functionality we’re already working on, or the new work that just came up?” — and this is a decision the Product Owner needs to make right now — the Product Owner needs to be agile all the time. The “Business Side” Product Owner is often incapable of making these decisions in a timely manner, and Scrum’s response to this problem was to move the Product Owner onto the Team to be available full-time to the Team in order to prioritize these interrupts real-time. View the diagram to the right.

We refer to this type of Scrum as Scrum 2.0, and its most popular variant is described in the Scrum Guide².





Product Owners often get Overwhelmed

The forces on a single Product Owner often cause the Product Owner to be overwhelmed. According to the Scrum Guide, the Product Owner has three main responsibilities:

1. Prioritize what gets delivered
2. Manage cost and schedule expectations
3. Work with the Team to ensure the right stuff gets done

There are many reasons that the Product Owner may need to be split.

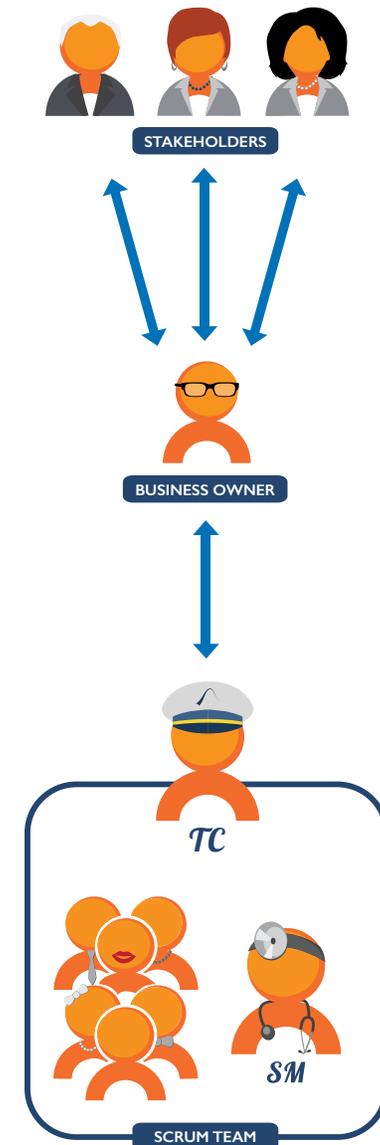
When the Product Owner needs to split, we wind up with two Product Owners: one with the Stakeholders and one with the Team. View the diagram to the left.

Scrum 3.0 Overview

There are many reasons that we may need two Product Owners: one with the Stakeholders to prioritize the new functionality, and one with the Team to prioritize the Work. This kind of Product Ownership is distributed and agile all the time. This is a common variant of Scrum in the real world, and is what many Organizations try to do with Scrum 2.0 and end up actually doing.

We refer to this type of Scrum as Scrum 3.0.

In order to differentiate between the two Product Owners in Scrum 3.0, we call the one with the Stakeholders the Business Owner (*BO*), and the one with the Team the Team Captain (*TC*). View the diagram to the right.



Purpose of this White Paper

The preceding was a very short summary of the evolution of Scrum as we experienced it since the late 1990's. The purpose of this white paper is to describe a commonly-seen Scrum 3.0 variant that extends, and is consistent with, the 2016 Scrum Guide. It is also our intention, in the near future, to produce a more comprehensive and complete definition of Scrum 3.0 that will reify and explain many of the confusing concepts found in most Scrum descriptions.

Description of Scrum 3.0

What follows is a brief description of a commonly seen version of Scrum 3.0.

Scrum's Roles

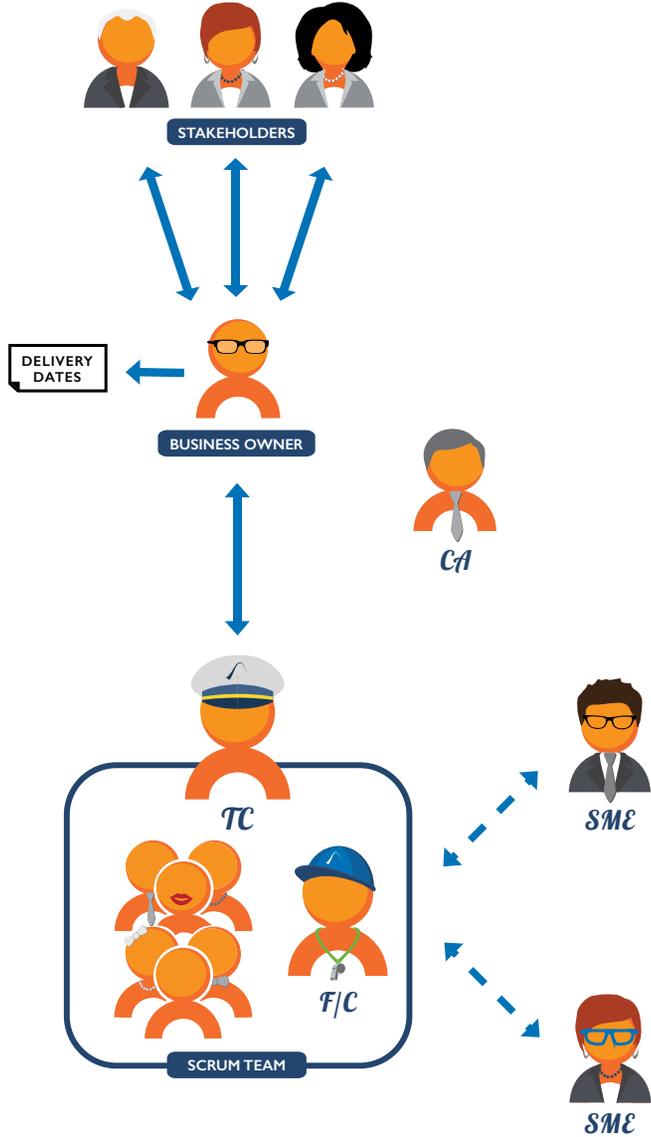
A Scrum Team is a Well-Formed Team that does valuable work coming from a Backlog of Work Items. There are no internally named technical roles on a Scrum Team — they are all called Team Members or Developers. There are six other roles for people involved in Scrum. The two that are on the Scrum Team are: Team Captain (*TC*) and Team Facilitator/Coach (*F/C*). The four outside the Scrum Team are: Business Owner (*BO*), Stakeholder (*SH*), Change Agent (*CA*), and Subject Matter Expert (*SME*). We see this in the diagram to the right.

Scrum breaks the world of Development into three Areas:

- **Production:** doing work in order to produce Results for Stakeholders;
- **Product Ownership:** determining the Results to deliver and when to deliver them; and
- **Scrum Mastering:** improving the ability for the Scrum Team to do its work.

In the Scrum 3.0 model, Production is done by the Team Members, Product Ownership is shared by the *BO* and the *TC*, and Scrum Mastering has been split into the *F/C* and *CA*. Here are brief definitions of these roles:

- **Team Captain (*TC*):** The Team Captain is the formal Leader of the Scrum Team, and is Accountable to the Business Owner for maximizing the value of the Scrum Team's work;



“ Leadership takes many forms and has many styles... **experiencing Leadership on a good Scrum Team is a beautiful thing.**

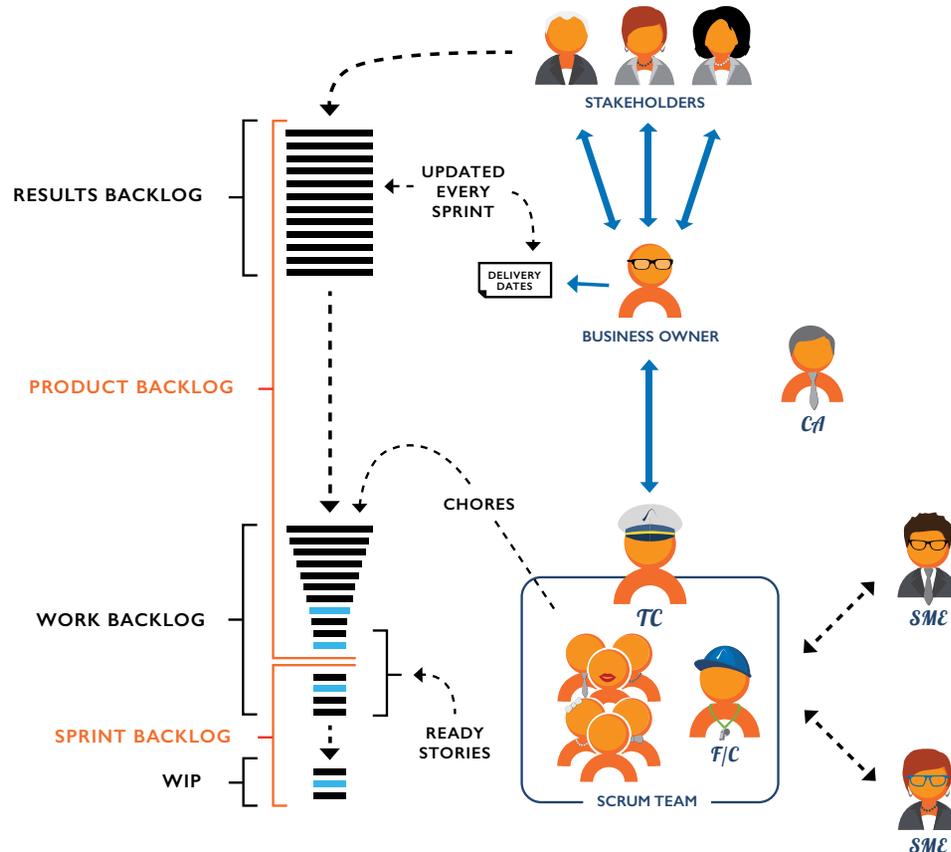
- **Facilitator/Coach (F/C):**The Scrum Team Member who helps the Scrum Team self-organize, remove their impediments, and improve their use of Scrum and agility;
- **Business Owner (BO):**The Business Owner is Accountable to the Stakeholders for maximizing the value of the Deliverables, and may also be accountable for predicting Delivery Dates;
- **Change Agent (CA):**The person who works with the Organization to help it adopt Scrum and understand how best to support and work with the Scrum Team;
- **Stakeholders (SH):**The Stakeholders are the people who want the Deliverables and provide Meaningful Feedback about them; and
- **Subject Matter Experts (SME):**The Subject Matter Experts have knowledge and expertise the Scrum Team needs but doesn't have on the Team.

Because the Scrum Team is Self-Organized (one facet of being a Well-Formed Team™), the Scrum Team works together (with no outside Manager) to do their part of these responsibilities. The Scrum Team works together with their SMEs to Produce Results, they work together with the Business Owner to do Product Ownership and they work together with the Change Agent to do Scrum Mastering. In Scrum 3.0, everybody on the Scrum Team is in it together – and it would be nice if the Business Owner, Change Agent, and SMEs acted as if they were on the Team, as well. Each Team Member is Accountable to the other Team Members to live the Team Values and help each other do the Scrum Team's work.

While they are doing that work, there is Leadership taking place constantly. Leadership takes many forms and has many styles. On the Scrum Team, we have Formal Leadership (the Team Captain), Servant Leadership (the Facilitator/Coach), and Emergent Leadership (any Team Member). The combination of these different kinds of Leadership leads to better teamwork and an increase in productivity, and experiencing Leadership on a good Scrum Team is a beautiful thing.

Scrum's Artifacts

The work of a Scrum Team is driven by a Backlog, which is “a list of Work Items that represents everything that anyone interested in the Scrum Team’s Results or Process has thought is needed or would be a good idea.” There are several different pieces of the Backlog in Scrum 3.0, and I will discuss them below the picture.



“ [The] Backlog, which is “a list of Work Items that represents everything that anyone interested in the Scrum Team’s Results or Process has thought is needed or would be a good idea.”

“ [The] Backlog is a list of work we might do. The work often changes along the way and becomes something other than what we expected it to be from the start.

Note: You may be asking: “what about the Product Backlog and Sprint Backlog? That’s what I’m used to seeing!” Well, they are there too. In Scrum 3.0 we emphasize the continuous nature of Development, and think of the Sprint as being an interrupt of that Continuous Flow; we don’t think of the Sprint as a Container of Stories. Unfortunately, your tools often treat the Sprint as if it were a Container of Stories... If you want to think of the Sprint as a Container of Stories, you can do that, with the following definitions (as you can see in the picture):

Sprint Backlog: The Sprint Backlog consists of the Work In Progress along with the Ready Stories that have been brought into the Sprint, whether they are committed to, forecasted to be worked on, or whatever.

Product Backlog: The Product Backlog consists of the Deliverable Results Backlog along with the portion of the Work Backlog that is not in the Sprint Backlog.

Here are the pieces of the Backlog:

Deliverable Results Backlog: The Deliverable Results Backlog is a prioritized list of Results (Deliverables) that the Business Owner hopes to deliver to the Stakeholders.

- These Deliverables are often Epics — big, chunky, Work Items with ill-defined Acceptance Criteria.
- The Business Owner and Team Captain collaborate to determine which Stories (small, well-defined, Work Items) should be extracted from these Epics and ‘passed down’ to the Scrum Team to work on.
- (If necessary) The Business Owner predicts delivery dates for the Stakeholders. These will normally be updated every Sprint as a result of the Progress Review.

Work Backlog: The Work Backlog consists of the Stories that the Scrum Team expects to be asked to work on ‘soon.’ These Stories are of two types:

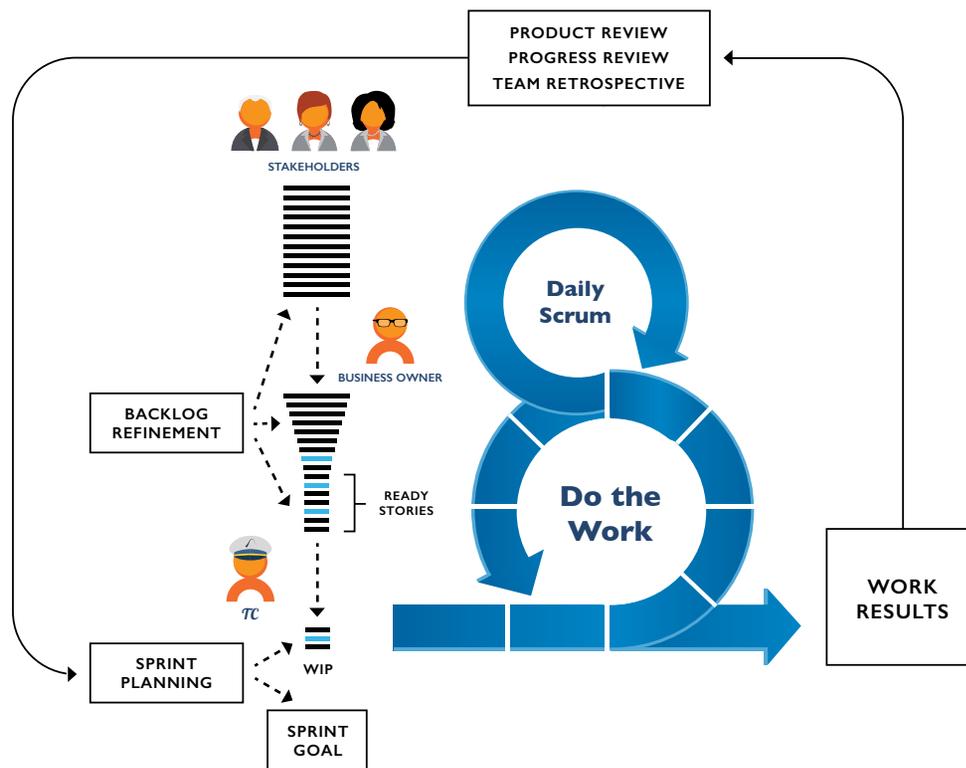
1. Capabilities that the Business Owner wants the Scrum Team to produce in order to deliver Results (Deliverables) to Stakeholders; and
2. Chores added by the Scrum Team; work the Scrum Team must do but does not directly provide deliverable Results (e.g., ‘doing the dishes.’)

A subset of the Work Backlog should always be Ready to be moved to the Work In Progress (WIP) to work on. A Story is ready when the Scrum Team knows what it ‘needs to know’ to start work on the Story. Typically, this means that they know the Story’s Acceptance Criteria, what Standard Of Care to use for the Story, and which Subject Matter Experts they will be working with.

The Work In Progress (WIP): The Work In Progress consists of the Stories the Scrum Team is actively working on. The Scrum Team often uses a WIP Limit to help manage the complexity of the swarming the Scrum Team will do to get the Stories to Done.

Scrum's Process

Scrum's Process has a fairly small collection of ceremonies, discussions, activities, or meetings. The purpose of these ceremonies is to have conversations, do work, sync-up, and make decisions. See the following diagram.



At its most basic, Scrum's Process can be thought of as a Continuous Flow, with an interrupt every Sprint to allow for Feedback, Improvement, and Re-Planning.

Note: So, while Scrum is widely adopted, people are still failing dramatically in their ability to adopt it and receive its benefit. Instead, they choose hybrid approaches because it seems more familiar and creates a false sense of doing Scrum.

“ The purpose of these ceremonies is to have conversations, do work, sync-up, and make decisions.

“ *The Scrum Team works with its Subject Matter Experts, its Business Owner, and its Stakeholders to do Backlog Refinement that makes sure there are sufficient Ready Stories in the Work Backlog.* ”

Continuous Flow

Let me first describe the Continuous Flow, which is actually done day-by-day. Every day:

- The Scrum Team has a Daily Scrum, when it self-organizes and plans its day. Typically, the Scrum Team discusses its current state, its progress towards its Sprint Goal, and its Impediments to figure out what it hopes to do today. As we like to say, the Team Members leave the Daily Scrum ‘walking with a purpose’ — they know what they’re doing next — even though it could change in just a few minutes.
- The Scrum Team Does Work, which means:
 - The Scrum Team swarms with its Subject Matter Experts to get Stories that are In Progress to Done;
 - The Team Captain determines which Ready Stories should be moved from the Work Backlog to become Work In Progress;
 - The Scrum Team works on its chosen Kaizen (a small change to improve itself); and
 - The Scrum Team discusses and removes Impediments.
- (As necessary) The Stakeholders and Business Owner maintain the Deliverable Results Backlog, by
 - Adding new Deliverables to the Deliverable Results Backlog; and
 - Prioritizing/Re-Prioritizing the Deliverable Results Backlog.
- (As necessary) The Scrum Team works with its Subject Matter Experts, its Business Owner, and its Stakeholders to do Backlog Refinement that makes sure there are sufficient Ready Stories in the Work Backlog. This includes:

-
- Extracting Capabilities from the Deliverable Results Backlog to the Work Backlog;
 - Adding appropriate Chores to the Work Backlog to support developing the Capabilities;
 - Prioritizing/Re-prioritizing the Stories (both Capabilities and Chores) in the Work Backlog;
 - Refining the Stories near the top/front of the Work Backlog to make them Ready to be worked on. This typically includes:
 - decomposing/Combining Stories to make them the 'right-size' to be worked on;
 - determining what Done means for these Stories; and
 - determining which SMEs will help the Scrum Team with these Stories.

Sprintly Interrupt

When the Sprint End comes around, the Continuous Flow is interrupted, and the Scrum Team has four Ceremonies/Discussions/Meetings in order to allow for Feedback, Improvement, and Re-Planning. They are:

- **Product Review:** The Scrum Team has a Product Review with its (product) Stakeholders at the end of the Sprint in which the Stakeholders provide Meaningful Feedback to the Team about the Team's Work. This discussion is 'owned' by the Team Captain, and consists of discussions driven by questions like: "What do you like about what we did?", "What don't you like?", "What would you like to change?", and "What would you like to see next?" the whole Scrum Team is after answers to these questions, so that they can do 'better' Work and produce 'better' Deliverable Results.

“When the Sprint End comes around, the Continuous Flow is interrupted, and the Scrum Team has four Ceremonies/Discussions/Meetings in order to allow for Feedback, Improvement, and Re-Planning.”

Note: *if you are used to Sprint Backlogs, then you learned Sprint Planning is largely about bringing Stories into the Sprint. You may have learned to ‘Fill the Sprint’ with Stories that were Committed to or Forecasted. This is not unreasonable, but it is unnecessary once you start thinking of the Sprint as a Continuous Flow with a Sprintly interrupt — as we do in Scrum 3.0.*

- **Progress Review:** The Team Captain has a Progress Review with the Business Owner and other Project/Progress Stakeholders about how the work is progressing — information to help determine expected Delivery Dates, Team Velocity, and so on. This discussion is owned by the Business Owner, and is largely a discussion about Dates and Dollars. Discussions involve questions like: “How fast are we going?”, “When will we be done?”, and “How do we adapt Cost, Scope, and Schedule to match the realities we see?”
- **Team Retrospective:** The Scrum Team has a Team Retrospective (run by the Facilitator/Coach) to discuss and agree on ways they could improve their Practices, teamwork, environment, or Organization for the next Sprint. This discussion is owned by the Facilitator/Coach, and is driven by the questions: “What did we do well?” and “What could we improve?” One major goal of the Sprint Retrospective is to discover/discuss at least one small process/teamwork improvement (called a Kaizen) that the Team will implement in the next Sprint.

The last of these discussions ‘between the Sprints’ is actually the beginning of the next Sprint.

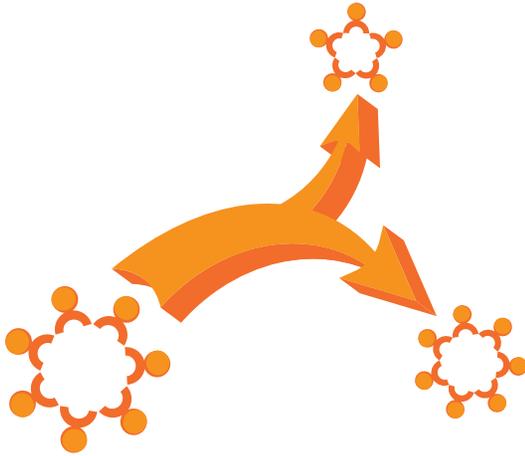
- **Sprint Planning:** Sprint Planning is a discussion the Scrum Team has at the beginning of the Sprint in order to:
 1. determine a Sprint Goal;
 2. determine the Sprint End;
 3. decide on which Kaizen the Team will implement in the next Sprint; and
 4. make sure there are enough Stories either In Progress or Ready to be able to ‘get back to work’.

Conclusion

What you have just read is a description of a particular variant of Scrum 3.0. This description is neither complete nor comprehensive, but allows someone who already knows Scrum to understand what Scrum 3.0 is – and how it differs from what he or she already knows. It is our intention to provide a complete definition of Scrum 3.0 in the near future.

“ *Scrum 3.0 is a type of Scrum that both unifies and extends other, older, descriptions of Scrum. **Scrum 3.0 is the cleanest, most flexible, and most agile version of Scrum there is.*** ”

Scrum 3.0 is scale-ready through fractalization in either the dimensions of Product Ownership and/or Scrum Mastering¹⁰.



To learn how Scrum 3.0 scales, review the patterns of Scaling Scrum with Scrum¹¹.

Discussion

Scrum 3.0 is a type of Scrum that both unifies and extends other, older, descriptions of Scrum. Scrum 3.0 is the cleanest, most flexible, and most agile version of Scrum there is. Scrum 3.0 is an aspirational kind of Scrum – rather than a recipe — as most teams don't really need that much agility. In fact, most good Scrum Teams that I see are actually doing what we call Scrum 2.75, which means:

- Splitting Product Ownership into a Business Owner and Team Captain, and
- Forecasting Sprint Backlogs, rather than going all the way to continuous planning by using Kanban's WIP, which is what Scrum 3.0 does.

Could these Teams be better if they moved all the way to Scrum 3.0? Maybe, but they are doing just fine where they are, so why should they change?

We documented Scrum 3.0 because we see the need for two PO-types so often. Many, if not most, Organizations need one PO with the Stakeholders and another PO with the Scrum Team. One of our favorite definitions of agility is “adapting to, and exploiting, chaos,” so we took advantage of the need to improve Product Ownership to also simplify Scrum 3.0's Sprint Planning method. The combination of these two ‘design decisions’ makes Scrum 3.0 the most versatile and agile version of Scrum that there is.

Scrum 3.0 is an extension of both Scrum 1.0 and Scrum 2.0, even though they, themselves, are incompatible with each other. Scrum 3.0 can be collapsed to either Scrum 1.0 or Scrum 2.0, or even Scrum 2.75, if necessary:

To collapse Scrum 3.0 to Scrum 1.0, you do three things:

- I. Remove the Team Captain, move the Team Captain's responsibilities to the Business Owner;

-
2. Call the Business Owner a Product Owner; and
 3. Modify Sprint Planning to use a ‘Fill the Sprint’ strategy and then Commit to the Stories in the Sprint Backlog.

To collapse Scrum 3.0 to Scrum 2.0, you do three things:

1. Remove the Business Owner, move the Business Owner’s responsibilities to the Team Captain;
2. Call the Team Captain a Product Owner; and
3. Modify Sprint Planning to use a ‘Fill the Sprint’ strategy to Forecast the Stories in the Sprint Backlog.

To collapse Scrum 3.0 to Scrum 2.75, you do one thing:

1. Modify Sprint Planning to use a ‘Fill the Sprint’ strategy to Forecast the Stories in the Sprint Backlog.

Scrum 3.0 is also a very clean and lean process. Sprint Planning has (essentially) been reduced to selecting a Sprint Goal, a Sprint End, and a Kaizen to work on, and there will never be Stories in the Sprint that are not being worked on — there is no inventory of selected Stories; every Story in the Sprint is In Progress – actually being worked on. There is a focus on getting to Done, not Velocity or Dates. The Stories get Done because of the Team Members’ commitment to the Team Values and the Story’s definition of Done (Acceptance Criteria and Standard Of Care) — there is no coercion to speed up and compromise the definition of Done from either the Organization or from time pressure (either external or internal).

In short, this expression of Scrum 3.0 is the most lean, clean, agile, and flexible version of Scrum to be found. We hope you enjoy it.

“*The Stories get Done because of the Team Members’ **commitment to the Team Values and the Story’s definition of Done** (Acceptance Criteria and Standard Of Care).*”

Primary References

1. Dan Rawsthorne and Doug Shimp. (2013). *Exploring Scrum: the Fundamentals*, 2nd Edition. <http://www.amazon.com/dp/1461160286>
2. Ken Schwaber and Jeff Sutherland. (2016). *The Scrum Guide*. <http://scrumguides.org/scrum-guide.html>

Additional Notes and Sources

3. Dan Rawsthorne and Doug Shimp. (2016). *The Well-Formed Team™* [White paper]. Retrieved from 3Back.com: <https://3back.com/well-formed-team-scrum-white-paper/>
4. Dan Rawsthorne and Doug Shimp. (2016). *Agility* [White paper]. Retrieved from 3Back.com: <https://3back.com/agility-scrum-white-paper/>
5. Dan Rawsthorne and Doug Shimp. (2016). *Leadership* [White paper]. Retrieved from 3Back.com: <https://3back.com/scrum-leadership-white-paper/>
6. Dan Rawsthorne and Doug Shimp. (2016). *Done* [White paper]. Retrieved from 3Back.com: <https://3back.com/done-a-scrum-white-paper/>
7. Dan Rawsthorne and Doug Shimp. (2016). *Scaling* [White paper]. Retrieved from 3Back.com: <https://3back.com/scaling-scrum-white-paper/>
8. Jeff Sutherland. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. New York, USA: Crown Business.
9. Pete Deemer, Gabrielle Benefield, Craig Larman, and Bas Vodde. (2012). *The Scrum Primer, A Lightweight Guide to the Theory and Practice of Scrum, Version 2.0*. Retrieved from scrumprimer.org.
10. 3Back, LLC. (2016). *The Scrum Dictionary*. Retrieved from 3Back.com: <https://3back.com/the-scrum-dictionary/>.
11. 3Back, LLC. (2016). *Scaling Scrum with Scrum™*. Retrieved from 3Back.com: <https://3back.com/agile-scaling-professional>.

For More Information

To learn more about 3Back, LLC and our Scrum related services, please contact us at info@3back.com. Follow [@Scrum_Coach](https://twitter.com/Scrum_Coach) on Twitter, and to subscribe to our newsletter, visit: <https://blog.3back.com/>.

We Make Teams Better.

We don't just talk Agile, we live Agile. Our 3Back Team is a well-formed, Agile team; applying Scrum 3.0 in our own workplace. From our hands-on support staff to our seasoned consultants and trainers, every member of the 3Back Team is, at minimum, a CSM (Certified Scrum Master). Within every level of the 3Back team, we bring a real world appreciation and understanding of your team's needs.

Managing work

At 3Back we are a fully distributed team. We actively build and manage [Get To Done \(gettodone.com\)](http://gettodone.com), an online Scrum software development tool. [Get To Done](http://gettodone.com) helps us train as a pedagogical tool, explore new ways of collaborating and focus our most precious resource, attention, on work being done.

About the Authors



Dan Rawsthorne has developed software in an agile way since 1983. He has worked in many different domains, from e-commerce to military avionics. He has a PhD in Mathematics (number theory), is a retired Army Officer, and a Professional Bowler and Coach. Dan is very active in the Agile/Scrum community and speaks quite often at conferences and seminars. He is a transformation agent, coaching Organizations to become more successful through agility.

His non-software background has helped him immeasurably in his coaching: his formal training in mathematics guides him to look for underlying problems rather than focus on surface symptoms; his military background helps him understand the importance of teamwork and empowerment; and his work with bowlers has helped him understand that coaching is a two-way street.



Doug Shimp has worked in the technology field since 1992 and has played many key roles on software teams, including Coder, Tester, Analyst, Team Leader, Manager, Coach, and Consultant. Doug's passion is for team learning to improve product development, and he is a leader in the area of Agile/Scrum transitions and applied practices. He believes that the core basis for applied agility is that 'You must see the result for it to be real;

otherwise it is all just theory...' Much of his experience with teamwork and agility comes from outside the software field, including an earlier career as an owner/manager of a painting company – which enabled him to learn about small-team dynamics in a very hands-on way.



SAVE THE DATE
SCRUM3.0
CONFERENCE

CHICAGO • JUNE 2017

REGISTER
3BACK.COM/SCRUM30