

CONTENTS

- About Scrum
- Scrum Values
- Scrum Roles
- Scrum Events
- Scrum Artifacts... and more!

Scrum

UPDATED BY STEVE PORTER AND DAVE WEST. ORIGINAL BY MICHAEL JAMES

ABOUT SCRUM

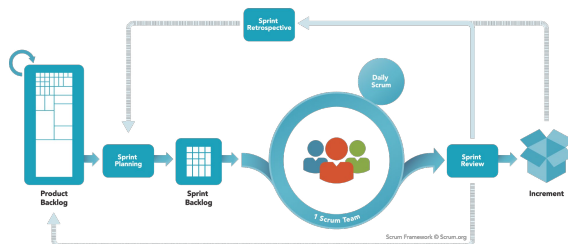
Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum is:

- Lightweight
- Simple to understand
- Difficult to master

Scrum is a process framework that has been used to manage complex product development since the early 1990s. It was created by Ken Schwaber and Jeff Sutherland working with others and originally presented at OOPSLA 1995. Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques that fit your organization's needs. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve.

Scrum is the most popular Agile process framework, with over 70% of all Agile teams using Scrum or a Scrum hybrid.¹

 FIGURE 1: SCRUM FRAMEWORK²


SCRUM THEORY

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

AN ALTERNATIVE TO WATERFALL

Scrum's incremental, iterative approach trades the traditional phases of "waterfall" development for the ability to develop a subset of high-business value features first, incorporating user feedback sooner.

Scrum has been used for a variety of work, but has initially been most popular for delivering software based products. Because Scrum employs an empirical approach, a team would incrementally deliver work to mitigate risks and increase understanding.

FIGURE 2: VALUE DELIVERY IN WATERFALL DEVELOPMENT

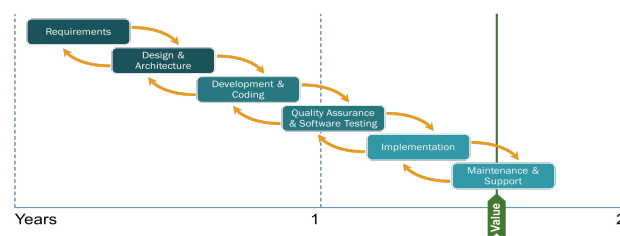
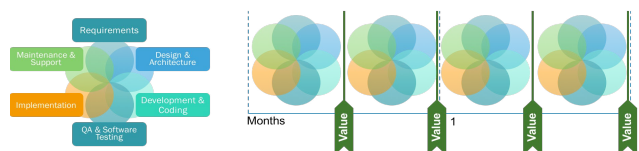


FIGURE 3: VALUE DELIVERY IN SCRUM DEVELOPMENT



A DISRUPTIVE FRAMEWORK TO TRANSFORM ORGANIZATIONS

The reality checks forced by the short feedback loops in Scrum are intended to expose challenges at the individual, team, and organizational level. Rather than modify Scrum to mask these challenges, organizations are encouraged to address these constraints and transform themselves.

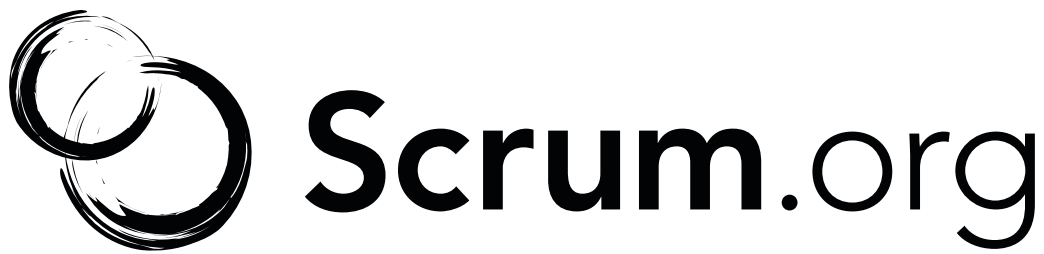


TRAINING & CERTIFICATION

Professional Scrum Teams Get It DONE.

[LEARN MORE](#)

PSF PSM PSPO PSD SPS



GLOBAL TRAINING & CERTIFICATION

Professional Scrum Teams Get It DONE.



LEARN MORE

Scrum.org provides comprehensive training, assessments and certifications to improve the profession of software delivery. Throughout the world, our solutions and community of Professional Scrum Trainers empower people and organizations to achieve agility through

Scrum. Ken Schwaber, the co-creator of Scrum, founded Scrum.org in 2009 as a global organization, dedicating himself to improving the profession of software delivery by reducing the gaps so the work and work products are dependable.

© 2017 Scrum.org All Rights Reserved.

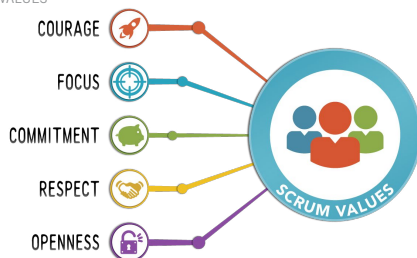
www.scrum.org | **PSF** **PSM** **PSPO** **PSD** **SPS**

Scrum is a framework, not a defined process or methodology. Scrum provides a simple structure of roles, events, artifacts, and rules. Scrum teams are responsible for creating and adapting their processes within this framework. Scrum's management practices are similar to those of eXtreme Programming (XP)³, but, unlike XP, Scrum does not prescribe specific engineering practices.

SCRUM VALUES

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The Scrum Team members learn and explore those values as they work with the Scrum events, roles and artifacts. Successful use of Scrum depends on people becoming more proficient in living these five values.

FIGURE 4: SCRUM VALUES⁴



SCRUM ROLES

PRODUCT OWNER

The Product Owner is a leadership role that is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

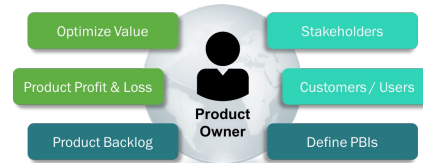
The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Ensuring the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have members of the Development Team do it. However, the

Product Owner remains accountable for the value being delivered by the team.

FIGURE 5: ACCOUNTABILITY OF THE PRODUCT OWNER



THE DEVELOPMENT TEAM

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness.

Development Teams have the following characteristics:

- They are self-organizing. No one (not even the Scrum Master or Product Owner) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole; and,
- Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis.

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work. Smaller Development Teams may encounter skill constraints during the Sprint, causing the Development Team to be unable to deliver a potentially releasable Increment. Large Development Teams may generate too much complexity for an empirical process to manage.

SCRUM MASTER

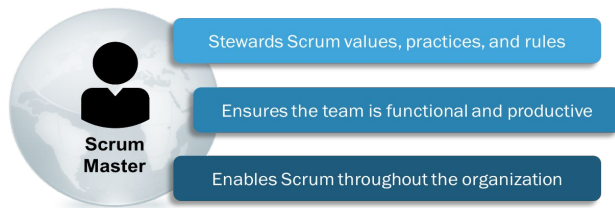
The Scrum Master is responsible for ensuring Scrum is understood and enacted. Scrum Masters do this by ensuring that the Scrum Team adheres to Scrum theory,

practices, and rules. The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master serves both the Product Owner and the Development Team in several ways, including:

- Teaching the Scrum Team to understand the need for clear and concise Product Backlog items;
- Facilitating Scrum events as requested or needed;
- Coaching the Development Team in self-organization and cross-functionality;
- Teaching and helping the Development Team to create high-value products;
- Removing impediments to the Development Team's progress.

The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren't. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

FIGURE 6: ACCOUNTABILITY OF THE SCRUM MASTER



SCRUM EVENTS

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed events, such that every event has a maximum duration. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved as long as they don't go over their time box, ensuring an appropriate amount of time is spent without allowing waste in the process.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

THE SPRINT

The heart of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Sprints best have

consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a definition of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product.

SPRINT GOAL

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog items. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. The selected Product Backlog items deliver one coherent function, which can be the Sprint Goal. The Sprint Goal can be any other coherence that causes the Development Team to work together rather than on separate initiatives.

As the Development Team works, it keeps the Sprint Goal in mind. In order to satisfy the Sprint Goal, it implements the functionality and technology. If the work turns out to be different than the Development Team expected, they collaborate with the Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

SPRINT PLANNING MEETING

The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team.

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.

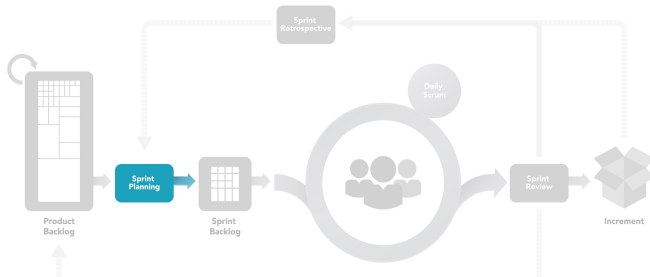
Sprint Planning answers the following:

- What can be delivered in the Increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

The Development Team works to forecast the functionality that will be developed during the Sprint. The Product Owner discusses the objective that the Sprint should achieve and the Product Backlog items that, if completed in the Sprint, would achieve the Sprint Goal. The entire Scrum Team collaborates on understanding the work of the Sprint.

By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

FIGURE 7: SPRINT PLANNING MEETING



DAILY SCRUM

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. This is done by inspecting the work since the last Daily Scrum and forecasting the work that could be done before the next one. The Daily Scrum is held at the same time and place each day to reduce complexity.

The Development Team uses the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint. The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or replan, the rest of the Sprint's work.

The Scrum Master ensures that the Development Team has the meeting, but the Development Team is responsible for conducting the Daily Scrum.

Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge. This is a key inspect and adapt meeting.

SPRINT REVIEW MEETING

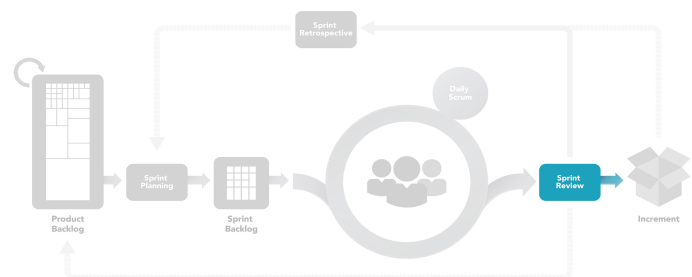
Development Teams deliver an Increment of product functionality at least every Sprint, but not limited to only once per Sprint. The Development Team may deliver an Increment more than once a Sprint. A Sprint Review is held at the end of the Sprint to inspect the Increment(s) and adapt the Product Backlog if needed. During the Sprint Review, the Scrum Team and stakeholders collaborate about what was done in the Sprint. Based on that, and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value. This is

an informal meeting; not a status meeting, and not a phase gate before release. The presentation of the Increment is intended to elicit feedback and foster collaboration.

This is a four-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box.

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint. The Product Backlog may also be adjusted overall to meet new opportunities.

FIGURE 8: SPRINT REVIEW MEETING



SPRINT RETROSPECTIVE

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is a three-hour time-boxed meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches all to keep it within the time-box. The Scrum Master participates as a peer team member in the meeting and has accountability over the Scrum process.

The purpose of the Sprint Retrospective is to:

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.

The Scrum Master encourages the Scrum Team to improve, within the Scrum process framework, its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by adapting the definition of "Done" as appropriate.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

SCRUM ARTIFACTS

Scrum's artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

PRODUCT BACKLOG

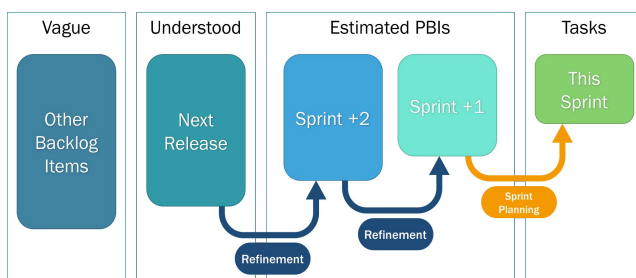
The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, its Product Backlog also exists.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value. Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail.

As a product is used and gains value, and the marketplace provides feedback, the Product Backlog becomes a larger and more exhaustive list. Requirements never stop changing, so a Product Backlog is a living artifact. Changes in business requirements, market conditions, or technology may cause changes in the Product Backlog.

FIGURE 9: PRODUCT BACKLOG LEVELS OF DETAIL



BACKLOG REFINEMENT

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised. The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team. However, Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Product Backlog items that will occupy the Development Team for the upcoming Sprint are refined so that any one item can reasonably be "Done" within the Sprint time-box. Product Backlog items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in a Sprint Planning. Product Backlog items usually acquire this degree of transparency through the above described refining activities.

The Development Team is responsible for all estimates. The Product Owner may influence the Development Team by helping it understand and select trade-offs, but the people who will perform the work make the final estimate.

SPRINT BACKLOG

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, in support of the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a "Done" Increment.

The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal.

The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

INCREMENT

The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the Increments

of all previous Sprints. By the end of a Sprint, the new Increment must be “Done,” which means it must be in useable condition and meet the Scrum Team’s definition of “Done.” It must be in useable condition regardless of whether the Product Owner decides to actually release it.

DEFINITION OF “DONE”

When a Product Backlog item or an Increment is described as “Done,” everyone must understand what “Done” means. Although this varies significantly per Scrum Team, members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of “Done” for the Scrum Team and is used to assess when work is complete on the product Increment. The purpose of each Sprint is to deliver Increments of potentially releasable functionality that adhere to the Scrum Team’s current definition of “Done.”

Development Teams deliver an Increment of product functionality at least every Sprint. The Development Team may deliver an Increment more than once a Sprint. This Increment is useable, so the Product Owner may choose to immediately release it. If the definition of “Done” for an Increment is part of the conventions, standards or guidelines of the development organization, all Scrum Teams must follow it as a minimum. If “Done” for an Increment is not a convention of the development organization, the Development Team of the Scrum Team must define a definition of “Done” appropriate for the product. If there are multiple Scrum Teams working on the system or product release, the Development Teams on all of the Scrum Teams must mutually define the definition of “Done.”

ARTIFACT TRANSPARENCY

Scrum relies on transparency. Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis. To the extent that the artifacts are incompletely transparent, these decisions can be flawed, value may diminish and risk may increase. The Scrum Master must work with the Product Owner, Development Team, and other involved parties to understand if the artifacts are completely transparent.

MONITORING PROGRESS

At any point in time, the total work remaining to reach a goal can be summed.

For the Product Backlog, the Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint Reviews to assess progress toward completing projected work by the desired time for the goal. This information is made transparent to all stakeholders.

For the Sprint Backlog, the Development Team tracks this total work remaining at least for every Daily Scrum

to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Development Team can manage its progress.

SCALING

Although the Scrum Guide does not include information on how Scrum should be practiced when you have more than one Scrum Team working together on the same product, many teams have used the Scrum framework to successfully deliver Integrated Increments in a scaled environment.

The dependencies between multiple teams can add complexity to an already complex environment, and to help address that complexity, several approaches have been suggested. Here is a sample of the more popular approaches.

NEXUS™ FRAMEWORK

Nexus is a process framework for developing and sustaining scaled product and software development initiatives. It uses Scrum as its building block to drive to the heart of scaling: cross-team dependencies and integration issues. It is an exoskeleton that rests on top of multiple Scrum Teams who work together to create an Integrated Increment. This framework consists of Nexus roles, events, artifacts, and the rules that bind them together.⁵

Nexus was developed by Ken Schwaber and Scrum.org.

scrum.org/nexus

LARGE SCALE SCRUM (LESS)

LeSS is a product development framework with a collection of principles, rules, guides and experiments that can be used in a scaled software development environment. LeSS provides a framework like Scrum which assumes multiple teams working on the same product. Within the LeSS Framework, product groups can apply the experiments and discover what works best for them at a certain moment.⁶

LeSS was created by Craig Larman and Bas Vodde.

less.works

SCALED AGILE FRAMEWORK® (SAFE®)

The Scaled Agile Framework is a methodology with a knowledge base of proven, integrated patterns for enterprise-scale Lean-Agile development. It is scalable and modular, allowing each organization to apply it in a way that provides better business outcomes and happier, more engaged employees. It is designed to help enterprises deliver value continuously and more efficiently on a regular and predictable schedule, making them more Agile.⁷

SAFe was created by Dean Leffingwell.

scaledagileframework.com

RELATED PRACTICES

Scrum is a framework coinciding with the agile movement in software development. Scrum has been popularized by its creators Ken Schwaber and Jeff Sutherland, organizations like the Scrum Alliance, Scrum.org, and many others.

Because Scrum is a framework, it does not prescribe specific practices. However, through experience, practitioners have found many ancillary practices that complement the core purpose of Scrum, which is to create potentially releasable software in 30 days or less.

Some of the more common practices are:

- Velocity-based planning;
- Automated testing;
- Continuous integration;
- Continuous delivery;
- Story point estimation;
- Extreme Programming.

REFERENCES

1. VersionOne 2016 Stat of Agile Report stateofagile.versionone.com
2. The Scrum Framework © Scrum.org
3. Extreme Programming extremeprogramming.org
4. Scrum Values image © Scrum.org
5. Scrum.org scrum.org/nexus
6. Bas Vodde infoq.com/articles/less-framework
7. Dean Leffingwell, SAFe 4.0 Introduction scaledagileframework.com/introduction-to-safe

All non-referenced material is based on the Scrum Guide, ©2016 Scrum. Org and Scrum Inc.

The Scrum Guide by Ken Schwaber & Jeff Sutherland scrumguides.org/scrum-guide.html

The original Scrum Guide is offered for license under the Attribution Share-Alike license of Creative Commons.

ABOUT THE AUTHORS



DAVE WEST is the CEO and Product Owner at Scrum.org. He is a frequent keynote at major industry conferences and is a widely published author of articles and research reports, along with his acclaimed book: *Head First Object-Oriented Analysis and Design*, that helped define new software modeling and application development processes. He led the development of the Rational Unified Process (RUP) for IBM/Rational. After IBM/Rational, West returned to consulting and managed Ivar Jacobson Consulting for North America. Then as VP, research director Forrester research where he ran the software development and delivery practice. Prior to joining Scrum.org he was Chief Product Officer at Tasktop where he was responsible for product management, engineering and architecture.



STEVE PORTER works with the Scrum.org team of experts and its wider trainer community to create and maintain its Professional Series of courses and assessments. Steve guides this collaborative process to ensure training is current and meets the needs of the professional Scrum community. Before joining Scrum.org, Steve was the product owner for TeamPulse, Telerik's agile project management tool. He has also provided application lifecycle management (ALM) consulting services for organizations around the globe. At his core, Steve is still a developer. He taught himself how code at an early age and has spent years turning ephemeral ideas into functioning software. It's the joy of this craft that Steve brings to all his interactions at Scrum.org.



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheat sheets, research guides, feature articles, source code and more.

"DZone is a developer's dream," says PC Magazine.

Copyright © 2017 DZone, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

DZONE, INC.

150 PRESTON EXECUTIVE DR.
 CARY, NC 27513
 888.678.0399
 919.678.0300

REFCARDZ FEEDBACK WELCOME
refcardz@dzone.com

SPONSORSHIP OPPORTUNITIES
sales@dzone.com

BROUGHT TO YOU IN PARTNERSHIP WITH

