PeopleCert SCRUM Master I

February 2021
Syllabus v1.5

PeopleCert
All talents, certified.

e-mail: info@peoplecert.org, www.peoplecert.org

1. Introduction

Scrum is a framework that helps teams work together. It increases team productivity by encouraging team players to learn through experiences, self-organise, and continuously improve.

IT teams use Scrum to develop, deliver, and maintain complex software products. However, Scrum principles can be applied to all kinds of areas: research, sales, marketing, and customer support - to name a few.

How does Scrum work?

Based on Agile principles, it enables teams to self-organise by encouraging close collaboration between all team members and specialisms involved. Scrum challenges the assumptions of the sequential approach to product development and replaces it with an iterative process. It includes a set of meetings, tools, and roles that work in harmony to help teams structure and manage their work.

The goal of this approach is to timely capture changes in customers' needs, along with any other unpredictable challenges that may occur — for which a sequential approach is not suited. As such, Scrum uses an evidence-based framework that embraces the fact that problems cannot be fully defined upfront. Instead, it focuses on maximising the team's ability to respond to emerging requirements, deliver value quickly, and adapt to new market conditions.

Why is it important to hold a Scrum certification?

Holding a Scrum certification proves the candidate's familiarity with Scrum practices, beyond the mere knowledge of terminology. It shows to employers that the candidate has the potential to be the person they need to cover the Scrum Master role in their teams.

Whether the candidate is a Scrum beginner or a seasoned professional, a certification a significant advantage when aiming to motivate and lead teammates. The PeopleCert Scrum Master credential showcases that the candidate has the skills necessary to lead an agile team successfully.

1.1. PeopleCert SCRUM Qualification structure

The PeopleCert Scrum qualification has been structured as follows:

- **PeopleCert SCRUM Master I** (14-18 training hours) – Candidates get the essential knowledge needed by IT professionals in Agile methodologies and Scrum practices. It focuses on the fundamental principles and concepts of both Agile and Scrum.
- **PeopleCert SCRUM Master II** (14-18 training hours) – Candidates will enhance on all aspects of the Scrum framework through a variety of real-world scenarios and focused practical information related specifically to the Scrum Master.
- **PeopleCert SCRUM Product Owner** (coming soon) – Candidates will cover all aspects of the Scrum framework through a variety of real-world scenarios and focused practical information related specifically to the Scrum Product Owner.

The **PeopleCert SCRUM Master I** certification covers the fundamental **knowledge** required for a candidate to build their knowledge and skills regarding Scrum principles and practices. In addition, the

PeopleCert SCRUM Master II certification (which is the next level of the qualification) covers more advanced skills, practices and knowledge about the Scrum framework.

The body of knowledge underlying these skills are presented in the official courseware provided by PeopleCert to accredited ATOs. The primary purpose of the syllabus is to provide a basis for accreditation of people involved with the Scrum framework. It documents the learning outcomes related to the qualification and describes the requirements a candidate is expected to meet to demonstrate that these learning outcomes have been achieved at the specific qualification level.

2. PeopleCert SCRUM Master I

2.1. Purpose of the SCRUM Master I Qualification

The purpose of this qualification level is to confirm that a candidate has sufficient knowledge, understanding and application of the Scrum framework and be able to work effectively with, or as a member of, a Scrum Team. This qualification is also a pre-requisite for the **PeopleCert Scrum Master II** qualification.

2.2. Target Group/Audience

This certification is the **first level** of the **PeopleCert SCRUM** qualification scheme provided by PeopleCert and is aimed at anyone who wishes to become an efficient member of a Scrum environment and requires candidates to have and demonstrate a solid **knowledge** and **understanding** of the Scrum terms, principles, tools and practices, as well as demonstrate their **application** skills of how to use tools efficiently and effectively. The certification can also cater for candidates seeking personal certification.

This certification will provide all the required level of knowledge to its holders and will certify that they have a solid understanding of Scrum using various tools.

An **advanced** level of skills and knowledge is covered in the next level of the **PeopleCert Scrum** qualification scheme provided by PeopleCert.

3. Learning Objectives

At this qualification level, candidates will be introduced to basic concepts, terms, principles and tools used for Scrum as well as why Scrum is needed in modern enterprises, the Scrum methodology, people and culture implications as well as the practices, processes, and technology used for adapting Scrum within an organization.

Holders of the **PeopleCert SCRUM Master I** certification**,** will be able to demonstrate their knowledge, understanding and practical application of:

- The definition and purpose of Agile
- The three pillars of Scrum: inspection, adaptation, and transparency
- The five Scrum values: commitment, focus, openness, respect, and courage
- The characteristics of effective Scrum Masters, product owners, and development teams
- The purposes of different events, artifacts, and rules in Scrum
- How multilevel planning can help your team develop long-term goals and create more realistic strategies to achieve those goals

- How to identify the attributes of the product backlog and understand how it can evolve as your product and environment changes
- The techniques for managing technical debt
- The purpose and steps involved in Scrum events such as product backlog grooming, sprint planning, execution, daily Scrum, review, and retrospective

3.1. Qualification Scheme Level

Through the above learning objectives, candidates will demonstrate relevant knowledge skills in the following areas:

Main Topics	
Introduction to Agile Project Management	Scrum Planning
Scrum as an Agile Framework	Scrum Events
The Scrum Framework & Team	Information Radiators
Scrum Artifacts	Scale Scrum

4. Examination

The **PeopleCert SCRUM Master I** Certification Exam is designed to validate a candidate's knowledge and understanding of Scrum basics as detailed above as well as how to be able to apply this knowledge through tools and practices in a modern enterprise.

The **PeopleCert SCRUM Master I** exam focuses on the following **two (2)** categories in the cognitive domain of **Bloom's taxonomy**[1] which is a reference for different levels of learning:

- **Knowledge**
- **Comprehension**

4.1. Assessment Approach

The assessment approach used for the **PeopleCert SCRUM Master I** certification focuses on the **two basic** categories of Knowledge and Comprehension.

Knowledge is defined as recalling previously learned material, from facts to theories and represents the lowest level of learning outcomes in the cognitive domain. Such learning outcomes are turned in assessment objectives that include **knowing and recalling** such as:

- Common and/or basic terms, definitions, concepts and principles
- Specific processes
- Processes, procedures and project management methods

Comprehension is the lowest level of understanding and entails the ability to grasp the meaning of the material taught, including some elements of interpretation, translation or estimation during the process. Such learning outcomes and in turn assessment objectives go beyond simply recalling information and may include:

[1] *The Bloom's taxonomy defines **six** (6) levels of learning in the **cognitive** domain (know, comprehend, apply, analyze, evaluate, create), which are both sequential and cumulative and move from the simple to the complex. In order to achieve the 6th level of learning, it must be ensured that the previous five levels have been mastered.*

- Understanding facts, concepts and principles
- Interpreting material (i.e. code, charts, graphs, text, diagrams)
- Justifying a process, procedure and method used

The assessment incorporates the above learning outcomes as it uses assessment objectives that cater for the above cognitive domain categories.

4.2. Entry Criteria/Training Requirements

For this examination, there **are no** formal entry criteria or training requirements.

In order to be eligible for the **PeopleCert Scrum Master I** level examination a candidate must be able to demonstrate knowledge and understanding of basic Scrum terms, principles, processes, practices and tools and it is recommended that the candidate has received **Accredited Training** by a PeopleCert accredited training partner.

4.3. Examination Format

The following table details the examination format for the **PeopleCert SCRUM Master I** exam:

Delivery	Computer (web proctored or classroom)
Type	**40 Multiple Choice Questions (MCQ)** *Each question is awarded one (1) mark*
Duration	**1 hour (60 minutes)** *For non-native speakers or candidates with a disability, an additional 30 minutes of extra time is allowed.*
Pass Mark	**70% (28 marks out of 40)**
Invigilator / Supervisor / Proctor	**Yes** *Physical or Online Proctoring*
Open Book	**No** *No materials are allowed in the examination room*
Prerequisites	**None**
Distinction	**N/A**
Certification validity	**Perpetual**

The tests are derived from a regularly updated question test bank (QTB) based on the test specification detailed below. Questions are used interchangeably among test sets. The overall difficulty level of each test is the same with any other test. A candidate is never assigned the same test in the case of multiple examination attempts.

5. Detailed Syllabus

The syllabus is structured into sections relating to the **major subject headings** and numbered with a single digit section number. A total of **fourteen to eighteen (14-18) hours** of accredited training is **recommended**.

1 Introduction to Agile Project Management	1.1 What Is Agile?	1.1.1	Define the term Agile
		1.1.2	Identify key terms used to describe Agile approaches, like: predictive; iterative; incremental; and adaptive
	1.2 Agile Values and Principles	1.2.1	Define the Agile Manifesto and explain its purpose
		1.2.2	Describe the purpose of the four values of the Agile Manifesto
		1.2.3	Describe the purpose of the 12 principles of the Agile Manifesto
2 Scrum as an Agile Framework	2.1 What Is Scrum?	2.1.1	Define the term Scrum
		2.1.2	Describe the purpose of Scrum
		2.1.3	Describe the benefits of Scrum in relation to the common challenges faced by IT
	2.2 The Three Pillars of Scrum	2.2.1	Know that Scrum is founded on empirical process control
		2.2.2	Define the three pillars of Scrum: inspection; adaptation; and transparency
		2.2.3	Explain the purpose of the three pillars of Scrum
	2.3 Scrum Values	2.3.1	List the five Scrum values: commitment; focus; openness; respect; and courage
		2.3.2	Identify examples for each of the five Scrum values
		2.3.3	Explain the purpose of the five Scrum values and their interaction with the Scrum Team
3 The Scrum Framework & Team	3.1 The Scrum Framework	3.1.1	List the Scrum main practices: roles, events, artifacts and rules

		3.1.2	Identify the purpose of the Scrum Practices
		3.1.3	Define the main Scrum roles: Product Owner, Development Team, Scrum Master
		3.1.4	Define artifact and the main Scrum artifacts: Product Backlog, Sprint Backlog, Product Increment
		3.1.5	Define the terms Timeboxing Sprint and the main Scrum events: Sprint Planning, Sprint Execution, Daily Scrum, Sprint Review, Sprint Retrospective
		3.1.6	List the 13 Scrum rules
	3.2 Roles	3.2.1	Know that the Scrum Team consists of a Product Owner, the development team, and a Scrum Master
		3.2.2	Describe the impacts of people performing multiple Scrum roles
	3.3 Teams	3.3.1	Define the term self-organizing team
		3.3.2	Describe the characteristics of a self-organizing team
		3.3.3	Define the term cross-functional team
		3.3.4	Describe the characteristics of a cross-functional team
	3.4 The Product Owner Role	3.4.1	Describe the characteristics of an effective Product Owner
		3.4.2	Identify the responsibilities and accountabilities of the Product Owner within Scrum
	3.5 The Development Team Role	3.5.1	Describe the characteristics of an effective Development Team
		3.5.2	Identify the responsibilities and accountabilities of the Development Team within Scrum
		3.5.3	Know the ideal Development Team Size
	3.6 The Scrum Master Role	3.6.1	Describe the characteristics of an effective Scrum Master
		3.6.2	Identify the responsibilities and accountabilities of the Scrum Master within Scrum
		3.6.3	Define the term Servant Leadership

		3.6.4	List the characteristics of a successful servant leader
		3.6.5	List the ways that a Scrum Master serves the Product Owner
		3.6.6	List the ways that a Scrum Master serves the Development team
		3.6.7	List the ways that a Scrum Master serves the organization
	3.7 Technical Debt	3.7.1	Define the term technical debt and its impact
		3.7.2	Describe the types, the causes and consequences of Technical Debt
4 Scrum Artifacts	4.1 Product Backlog	4.1.1	Identify the attributes of a product backlog
		4.1.2	Describe the purpose of the product backlog in Scrum and who is responsible for its development and maintenance
		4.1.3	Explain how the product backlog can evolve as the product and environment evolves
		4.1.4	Define the terms product backlog item (PBI), requirements statement, acceptance criteria, increment, and conditions of satisfaction.
		4.1.5	Define the term Product Backlog Grooming
		4.1.6	Define the terms "epic", "feature" "user story", "story points" in relation the PBIs, during product backlog grooming.
		4.1.7	Identify the purpose and process steps involved in the product backlog grooming in Scrum including the roles of the Scrum team
	4.2 Sprint Backlog	4.2.1	Describe the purpose of a sprint backlog in Scrum
		4.2.2	Identify the attributes of a sprint backlog, how it can be altered and which role may perform any alterations
	4.3 Product Increment	4.3.1	Define the term product increment
		4.3.2	Define the potentially shippable product and describe its purpose in Scrum
		4.3.3	Define the terms conditions of satisfaction, definition of done, and definition of ready

		4.3.4	Differentiate between the definition of done (DoD) and acceptance criteria.
		4.3.5	Define the term "velocity" identify estimation techniques used to calculate velocity and its purpose
		4.3.6	List practices that will help Scrum teams deliver a high-quality product increment and reduce technical debt.
	4.4 Artifact Transparency	4.4.1	Explain the importance of artifact transparency
		4.4.2	Describe the impact of artifacts that are not fully transparent
		4.4.3	Identify examples of how transparency must be inspected, adapted and increased at the Scrum events
5 Scrum Planning	5.1 Multilevel Planning	5.1.1	Define multilevel planning and explain how multilevel planning helps teams
		5.1.2	Describe the concept of the planning onion and list the six planning levels (day, iteration, release, product, portfolio, strategy)
	5.2 Release Planning	5.2.1	Define Release backlog, Release Planning and describe its goal
		5.2.2	Identify the timing, participants, and process steps involved in the release backlog
		5.2.3	Define "fixed-scope release" and "fixed-date release".
6 Scrum Events	6.1 Sprint Planning	6.1.1	Describe the purpose of the sprint planning event
		6.1.2	Identify the process steps involved in the sprint planning event including the roles and responsibilities of the Scrum team
		6.1.3	Identify the correct length of time for a sprint.
		6.1.4	Define the term "capacity" in relation to sprint planning including the various factors that influence a team's capacity.
	6.2 Sprint Execution	6.2.1	Describe the purpose of the sprint execution event
		6.2.2	Identify the process steps involved in the sprint execution event including the roles and responsibilities of the Scrum team

	6.3 Daily Scrum	6.3.1	Describe the purpose of the daily scrum event
		6.3.2	Identify the process steps involved in the daily scrum event including the roles and responsibilities of the Scrum team
		6.3.3	Identify the attributes of an effective daily Scrum meeting.
	6.4 Sprint Review	6.4.1	Describe the purpose of the sprint review event
		6.4.2	Identify the process steps involved in the sprint review event including the roles and responsibilities of the Scrum team
		6.4.3	Describe the outcomes of a Sprint Review
	6.5 Sprint Retrospective	6.5.1	Describe the purpose of the sprint retrospective event
		6.5.2	Identify the process steps involved in the sprint retrospective event including the roles and responsibilities of the Scrum team
7 Information Radiators	7.1 Types of Information Radiators	7.1.1	Define the term "information radiator" explain the purpose, and provide examples of information radiators
		7.1.2	List the Types Information Radiators
	7.2 Purposes of Information Radiators	7.2.1	Define sprint backlog chart, Product Backlog chart, Burn-down/Burnup chart and burnup chart and explain the purpose of each.
		7.2.2	Describe the purpose of visual management tools in relation to Scrum and identify the attributes of effective visual management tools, including kanban charts and task boards, and explain the benefits of each
8 Scaled Scrum	8.1 The Purpose of Scaling Scrum	8.1.1	Define the term Scaled Scrum
		8.1.2	Describe the purpose and goal of scaling Scrum
	8.2 Roles in Scaling Scrum	8.2.1	Know the roles in Scaled Scrum
		8.2.2	List the main responsibilities of the Chief Scrum Master in Scaled Scrum
	8.3. Scaled Artifacts and Events	8.3.1	Describe how to scale the product backlog

		8.3.2	Identify the reasons of a consistent definition of done across multiple teams working on the same product log

6. Test Specification

The **PeopleCert SCRUM Master I** examination will consist of **eight** (**8**) sections with the following structure:

Category	Description	Exam (%)
1.0	Introduction to Agile Project Management	5.0%
2.0	Scrum as an Agile Framework	2.5%
3.0	The Scrum Framework & Team	25.0%
4.0	Scrum Artifacts	25.0%
5.0	Scrum Planning	12.5%
6.0	Scrum Events	25.0%
7.0	Information Radiators	2.5%
8.0	Scale Scrum	2.5%
	Total	**100.0%**

7. Bibliography

1. Academic and Student Affairs. "Execution – Deployment Plan." University of Washington. Accessed March 9, 2020 (https://www.washington.edu/asa/project-management-draft/project-management-resources/templates/execution-deployment-plan/).

2. Adzic, Gojko. "Effective Root Cause Analysis Techniques." DZone - Agile Zone. Last modified June 11, 2010. Accessed February 29, 2020. https://dzone.com/articles/effective-root-cause-analysis.

3. Agarwal, Shubhranshu. "How to Build a Self-Organizing Team as a Scrum Master." Knowledgehut Blog (blog). Last modified January 11, 2018. Accessed February 29, 2020. https://www.knowledgehut.com/blog/agile/how-to-build-a-self-organizing-team-as-a-scrum-master.

4. Agile Alliance. "Agile Glossary." Last modified 2011. Accessed February 29, 2020. https://www.agilealliance.org/agile101/agile-glossary/.

5. Agile and Scrum Certification Blogs. "The Scrum Master in the Servant-Leader Role." Accessed December 3, 2019. https://www.agilemania.com/blog/the-scrum-master-in-the-servant-leader-role/.

6. Allue, Xavier Quesada. "Visual Management for Agile Teams." Visual Management Blog (blog). Last modified February 11, 2009. Accessed February 29, 2020. http://www.xqa.com.ar/visualmanagement/2009/02/visual-management-for-agile-teams/.

7. Ambler, Scott, Glen Little, et al. "Rolling Wave Release Planning for Agile Delivery Teams." Disciplined Agile (blog). Last modified October 18, 2016. Accessed March 12, 2020. https://www.projectmanagement.com/blog-post/61895/Rolling-Wave-Release-Planning-for-Agile-Delivery-Teams.

8. APMG. *The Effective Change Manager's Handbook*. Edited by Richard Smith, David King, Ranjit Sidhu, and Dan Skelsey. London, UK: Kogan Page, 2014.

9. Badgley, Matthew. "Twelve Awesome Interactive Facilitation Techniques for Agile Teams." CollabNet VersionOne. Last modified September 17, 2013. Accessed February 29, 2020. https://resources.collab.net/blogs/twelve-awesome-interactive-facilitation-techniques-for-agile-teams.

10. Berteig, Mishkin. "The Agile Planning Onion Is Wrong." Agile Advice (blog), Last modified April 24, 2011. Accessed August 12, 2019 http://www.agileadvice.com/2011/04/24/agilemanagement/the-agile-planning-onion-is-wrong/.

11. Bridges, William. *Transitions: Making Sense of Life's Changes*. Reading, Mass: Addison-Wesley, 1980.

12. Busche, Laura. "Up on the Wall: How Working Walls Unlock Creative Insight." *Smashing Magazine*. Last modified January 2, 2014. Accessed March 12, 2020. https://www.smashingmagazine.com/2014/01/how-working-walls-unlock-creative-insight/ .

13. CBTnuggets. "5 Steps for Becoming an Excellent Listener." Last modified November 9, 2017. Accessed February 29, 2020. https://www.cbtnuggets.com/blog/technology/productivity/5-steps-for-becoming-an-excellent-listener.

14. CBTnuggets. "Boost Your Team's Strengths while Eliminating Weaknesses." Last modified November 7, 2017. Accessed February 29, 2020. https://www.cbtnuggets.com/blog/career/management/boost-your-teams-strengths-while-eliminating-weaknesses.

15. Campbell, Sam. "The Ultimate Guide to the Servant Leadership Model." When I Work (blog). Last modified August 1, 2017. Accessed February 29, 2020. https://wheniwork.com/blog/the-ultimate-guide-to-the-servant-leadership-model/.

16. Clarity Visual Management. "Visual Management." Accessed December 4, 2019. https://www.clarityvisualmanagement.com/technique/vm-visual-management/.

17. Cohn, Mike. "Self-Organizing Teams Are Not Put Together Randomly." Mountain Goat Software. Accessed December 4, 2019. https://www.mountaingoatsoftware.com/blog/self-organizing-teams-are-not-put-together-randomly.

18. *Succeeding with Agile*: *Software Development Using Scrum*. Addison-Wesley, 2010

19. "Rolling Look-ahead Planning." Mountain Goat Software. Last modified July 9, 2008. Accessed March 12, 2020. https://www.mountaingoatsoftware.com/articles/rolling-lookahead-planning.

20. Corry, Paddy. "5 Essential Facilitation Skills for Scrum Masters and How to Develop Them." Serious Scrum. Last modified September 30, 2019. Accessed February 29, 2020. https://medium.com/serious-scrum/5-essential-facilitation-skills-for-scrum-masters-and-how-to-develop-them-19088080354c.

21. Crouch, Alan. "Understanding the ScrumMaster's Role in Team Communication." TechWell Insights. Last modified November 8, 2019. Accessed February 29, 2020. https://www.techwell.com/techwell-insights/2019/11/understanding-scrummasters-role-team-communication.

22. Doshi, Punit. "Agile Coach Toolkit #1: 5 Whys." Scrum.org. Last modified March 5, 2018. Accessed February 29, 2020. https://www.scrum.org/resources/blog/agile-coach-toolkit-1-5-whys.

23. Doyle, Alison. "Important Active Listening Skills and Techniques." The Balanced Careers. Last modified December 2, 2019. Accessed February 29, 2020. https://www.thebalancecareers.com/active-listening-skills-with-examples-2059684.

24. Drucker, Peter. "Bill Moyers a World of Ideas." Accessed March 11, 2020. https://billmoyers.com/content/peter-drucker/. 1989.

25. Espy, Leigh. "Scrum Master Responsibilities: What Does a Scrum Master Do?" Project Bliss. Last modified July 10, 2017. Accessed February 29, 2020. https://projectbliss.net/scrum-master-responsibilities/.

26. Ewel, Jim. "Doing Agile vs Being Agile." Agile Marketing. Last modified June 26, 2017. Accessed February 29, 2020. https://www.agilemarketing.net/agile-agile/.

27. Francino, Yvette. "Are You Being Agile—or Just Doing Agile?" Tech Beacon. Last modified July 24, 2019. Accessed February 29, 2020. https://techbeacon.com/app-dev-testing/are-you-being-agile-or-just-doing-agile.

28. Garrido, Marcos. "Facilitation Techniques for Effective ScrumMasters." Last modified May 13, 2014. Accessed February 29, 2020. https://www.slideshare.net/marcosgarrdio/facilitation-techniques-34648407.

29. Ghahrai, Amir. "10 Traits of an Agile Self Organizing Team." Testing Excellence. Last modified December 3, 2018. Accessed February 29, 2020. https://www.testingexcellence.com/10-traits-agile-self-organizing-team/.

30. Goldstein, Ilan. *Scrum Shortcuts without Cutting Corners: Agile Tactics, Tools & Tips*. 1st ed. Boston, MA: Addison-Wesley, 2014.

31. Gonçalves, Luís. "Scrum Master Role: Everything What All Scrum Masters Need to Know." Last modified March 26, 2019. Accessed February 29, 2020. https://luis-goncalves.com/scrum-master-role/.

32. Greenleaf, Robert K. *Servant Leadership: A Journey into the Nature of Legitimate Power and Greatness*. Mahwah, NJ: Paulist Press, 2002.

33. Wade, Lori. "How to Use Cause and Effect Analysis to Solve Business Problems." Creately (blog). Last modified January 16, 2018. Accessed February 29, 2020. https://creately.com/blog/diagrams/how-to-use-cause-and-effect-analysis/.

34. Harned, Brett. "How to Clear Project Confusion with a RACI Chart." TeamGantt. Last modified: September 16, 2019. Accessed: March 12, 2020. https://www.teamgantt.com/blog/raci-chart-definition-tips-and-example.

35. Heiliger, Max. "Is My Scrum Team Really Self-Organizing? Some Hard Questions to Ask Yourself as a Scrum Master." Serious Scrum. Last modified March 9, 2019. Accessed February 29, 2020. https://medium.com/serious-scrum/is-my-scrum-team-really-self-organizing-a76d5c177f46.

36. Hughes, Charlotte. "Leading and Communicating Change in Organizations." Health Inclusion. Last modified July 5, 2018. Accessed March 11, 2020. https://healthinclusion.com/back-up-leading-and-communicating-change/.

37. Scrum Institute™. "If I Had 5 Minutes to Explain Scrum Master as a Facilitator." Last modified October 3, 2017. Accessed February 29, 2020. https://www.scrum-institute.org/blog/If-I-had-5-Minutes-to-explain-Scrum-Master-As-a-Facilitator.

38. Intersog. "How Being Agile Is Different from Doing Agile." Intersog. Last modified May 31, 2017. Accessed February 29, 2020. https://medium.com/@Intersog/how-being-agile-is-different-from-doing-agile-9098e8b679f1.

39. In-Tuition. "the Impact on Productivity." In-Tuition - Practical Management Skills. Accessed January 24, 2020. http://www.practical-management-skills.com/trainer-skills.html.

40. J, Neha. "The 7 Soft Skills of Successful Scrum Masters." LinkedIn. Last modified January 22, 2017. Accessed February 29, 2020. https://www.linkedin.com/pulse/7-soft-skills-successful-scrum-masters-neha-jain.

41. Jarrell, Jeremy. "The 3 Levels of a Scrum Master Removing Impediments." JeremyJarrell.com. Last modified May 9, 2016. Accessed February 29, 2020. https://www.jeremyjarrell.com/scrum-master-removing-impediments/.

42. Karasek, Nathalie. "How to Do an Ishikawa Diagram in Software Development." Work@Life. Last modified May 6, 2018. Accessed February 29, 2020. http://www.nathaliekarasek.com/ishikawa-diagram-software-development/.

43. Kasperowski, Richard. "Self-Management and Self-Organization: Agile Games with Motion." Kasperowski.com. Last modified October 29, 2012. Accessed February 29, 2020. https://kasperowski.com/self-management-and-self-organization-agile-games-with-motion/.

44. Khan, Rash. "Being Agile vs Doing Agile, Why Is It Important to Differentiate!?" Last modified December 5, 2015. Accessed February 29, 2020. https://www.linkedin.com/pulse/being-agile-vs-doing-why-important-differentiate-rash-khan.

45. Knight, Jason. "W00t! Well Written." Comment on "The Scrum Master as a Servant-Leader." Scrum.Org. Last modified 2015. Accessed February 29, 2020. https://www.scrum.org/resources/blog/scrum-master-servant-leader.

46. Larman, Craig, and Bas Vodde. "Top Ten Organizational Impediments to Large-Scale Agile Adoption." informIT. Last modified August 13, 2009. Accessed March 12, 2020. https://www.informit.com/articles/article.aspx?p=1380615.

47. Lean Agile Training - Think Further. "The Impediment List." Last modified April 19, 2018. Accessed February 29, 2020. https://www.leanagiletraining.com/better-agile/the-impediment-list/.

48. Lucidchart Content Team. "Mastering the Key Stages of Group Development." Lucidchart. Accessed February 29, 2020. https://www.lucidchart.com/blog/stages-of-group-development.

49. Lynn, Rachaelle. "What Is a Self-Organizing Team?" Planview. Accessed December 4, 2019. https://www.planview.com/resources/articles/what-is-self-organizing-team/.

50. Maretzke, Michael. "When to Use Waterfall, when Agile?" Agile Minds. Last modified March 12, 2019. Accessed January 28, 2020. https://www.agile-minds.com/when-to-use-waterfall-when-agile/.

51. McConnell, Eric. "Virtual Teams – Definition, Management, Benefits, and Improvement." MyManagementGuide.com. Last modified July 31, 2019. Accessed March 12, 2020. https://mymanagementguide.com/managing-virtual-teams-understanding-definition-management-and-benefits/.

52. McIntosh, Philippa. "What Is Visual Management?" 100% Effective Ltd. Last modified February 13, 2017. Accessed February 29, 2020. https://www.100pceffective.com/blog/what-is-visual-management/.

53. Metz, Terrence. “Scrum Master Facilitator Techniques Improve Meetings.” MG Rush Facilitation Best Practices. Accessed December 3, 2019. https://mgrush.com/blog/scrum-master-facilitator/.

54. DZone - Agile Zone. “Scrum Master: Servant Leadership in Practice.” Last modified April 26, 2013. Accessed February 29, 2020. https://dzone.com/articles/scrum-master-servant.

55. Morris, David. *Scrum in Easy Steps – An Ideal Framework for Agile Projects*. UK: In Easy Steps Limited, 2017.

56. Murphy, Mark. “Which of These 4 Communication Styles Are You?” *Forbes*. Last modified August 6, 2015. Accessed February 29, 2020. https://www.forbes.com/sites/markmurphy/2015/08/06/which-of-these-4-communication-styles-are-you/#6b9534fc3adb.

57. NutCache. “6 Must Have Skills to Become a Notable Scrum Master.” Accessed December 3, 2019. https://www.nutcache.com/blog/6-must-have-skills-to-become-a-notable-scrum-master/.

58. Ockerman, Stephanie. “6 Coaching Tips for Scrum Masters.” Agile Socks. Last modified April 2, 2017. Accessed February 29, 2020. https://www.agilesocks.com/6-coaching-tips-scrum-masters/.

59. “Getting to Done: Tackling Impediments.” Scrum.org. Last modified January 3, 2017. Accessed February 29, 2020. https://www.scrum.org/resources/blog/getting-done-tackling-impediments.

60. Overeem, Barry. “Characteristics of a Great Scrum Team.” InfoQ. Last modified April 15, 2016. Accessed February 29, 2020. https://www.infoq.com/articles/great-scrum-team/.

61. “The Scrum Master as a Coach.” BarryOvereen.com. Last modified April 20, 2015. Accessed February 29, 2020. https://www.barryovereem.com/the-scrum-master-as-a-coach/

62. “The Scrum Master as a Coach.” Scrum.org. Last modified August 14, 2015. Accessed February 29, 2020. https://www.scrum.org/resources/blog/scrum-master-coach-0?gclid=CjwKCAiArJjvBRACEiwA-WiqqzxL107mFDOJOsS0cOYLdEFJ1zPRDYJk0HdPyBiQtwfPPk6YT03CwxoCIvcQAvD_BwE.

63. “The Scrum Master as a Servant-Leader.” Scrum.org. Last modified July 20, 2015. Accessed February 29, 2020. https://www.scrum.org/resources/blog/scrum-master-servant-leader.

64. “The Scrum Master as an Impediment Remover.” BarryOvereen.com. Last modified March 26, 2016. Accessed February 29, 2020. http://www.barryovereem.com/the-scrum-master-as-an-impediment-remover/

65. “The Scrum Master as a Teacher.” Scrum.org, October 25, 2015. https://www.scrum.org/resources/blog/scrum-master-teacher?gclid=Cj0KCQjw3qzzBRDnARIsAECmrypFKzfEbUoUF3FK9pMAyTVGAzm4bz3rxrmVoLr-pYnr0lJOuDyLs6AaAr9fEALw_wcB.

66. Paffrath, Holger. “The Fishbone Diagram.” DZone - Agile Zone. Last modified February 6, 2018. Accessed February 29, 2020. https://dzone.com/articles/the-fish-bone-diagram.

67. Partogi, Joshua. “Scrum Master: The Master of the Art of Facilitation.” Scrum.org. Last modified August 19, 2018. Accessed February 29, 2020. https://www.scrum.org/resources/blog/scrum-master-master-art-facilitation.

68. Pink Elephant Inc. “Agile Scrum Essentials.” Burlington, ON. 2019.

69. “Root Cause Analysis Specialist.” Burlington, ON. 2019.

70. Ravlani, Kamlesh. “The Scrum Master as a Coach.” Agile for Growth. Last modified April 17, 2019. Accessed February 29, 2020. https://agileforgrowth.com/blog/scrummaster-coach/.

71. Ravlani, Kamlesh. “The Scrum Master as a Servant-Leader.” Agile for Growth. Last modified April 7, 2019. Accessed February 29, 2020. https://agileforgrowth.com/blog/scrum-master-servant-leader/.

72. Product Owner of the Year. “The Scrum Master as a Teacher.” Accessed March 13, 2020. https://www.productowneroftheyear.com/the-scrum-master-as-a-teacher/.

73. Render, Joshua. "Information Radiators and Virtual Teams." Agile-Mercurial. Last modified March 6, 2019. Accessed February 29, 2020. https://agile-mercurial.com/2019/03/06/information-radiators-and-virtual-teams/.

74. "What Is Osmotic Communication?" Agile-Mercurial. Last modified January 26, 2019. Accessed February 29, 2020. https://agile-mercurial.com/2019/01/26/what-is-osmotic-communication/.

75. Reyes, Gauri. "They Say, 'Be Agile'. What Does that Really Mean?" Mindflash. Accessed November 29, 2019. https://www.mindflash.com/blog/they-say-be-agile-what-does-that-really-mean.

76. Rubin, Kenneth. *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Boston, MA: Pearson Education, 2013.

77. Sarandeska, Ivana. "Identify and Eliminate Problems and Obstacles with Root Cause Analysis (RCA)." Kanban Zone. Last modified March 23, 2019. Accessed February 29, 2020. https://kanbanzone.com/2019/root-cause-analysis-rca/.

78. Schwaber, Ken and Jeff Sitherland. *The Scrum Guide™. The Definitive Guide to Scrum: The Rules of the Game.* Accessed March 11, 2020. (https://www.scrumguides.org/downloads)

79. Schwaber, Ken. "Scrum as a Framework." Ken Schwaber's Blog: Telling It Like It Is. Accessed March 11, 2020 (https://kenschwaber.wordpress.com/2010/09/08/scrum-as-a-framework/).

80. Service Futures ISS. "The Difference between Change and Transition Management in Service Outsourcing." Accessed February 24, 2020. https://www.servicefutures.com/difference-change-transition-management-service-outsourcing.

81. Sharma, Nagesh. "Key Coaching Technique for Scrum Masters to Enter the ZOUD!" DZone - Agile Zone. Last modified June 2, 2017. Accessed February 29, 2020. https://dzone.com/articles/key-coaching-technique-for-scrum-masters-to-enter.

82. Shift314. "Doing Agile vs. Being Agile." Agilitrix. Last modified April 26, 2016. Accessed February 29, 2020. https://agilitrix.com/2016/04/doing-agile-vs-being-agile/.

83. Shiralige, Avienaash. "5 Whys: Sprint Failed – Team Did Not Deliver Committed Work." Malonus. Last modified 2012. Accessed February 29, 2020. http://www.agilebuddha.com/agile/5-whys-sprint-failed-team-did-not-deliver-committed-work/.

84. Stein, Judith. "Using the Stages of Team Development." Human Resources - Massachusetts Institute of Technology. Accessed December 4, 2019. https://hr.mit.edu/learning-topics/teams/articles/stages-development#site-header.

85. Sundaram, Sathish. "Phases of Self-Organized Team." DZone - Agile Zone. Last modified October 17, 2018. Accessed February 29, 2020. https://dzone.com/articles/phases-of-self-organised-team.

86. Surdek, Steffan. "Three Common Misunderstandings of Self-Organized Teams." *Forbes*. Last modified December 20, 2016. Accessed February 29, 2020. https://www.forbes.com/sites/forbescoachescouncil/2016/12/20/three-common-misunderstandings-of-self-organized-teams/#62aedd4b195e.

87. Sutherland, Jeff. *Jeff Sutherland's Scrum Handbook.* Scrum Training Institute, 2010.

88. *Scrum: The Art of Doing Twice the Work in Half the Time*. New York, USA: Crown Publishing Group, 2014.

89. Team Nuggets. "How Being a Facilitator Makes You an Elite Scrum Master." CBTnuggets. Last modified November 6, 2017. Accessed February 29, 2020. https://www.cbtnuggets.com/blog/career/management/how-being-a-facilitator-makes-you-an-elite-scrum-master.

90. The App Business. "Servant Leadership and the Role of a Scrum Master." TAB Kin+Carta. Last modified June 29, 2016. Accessed February 29, 2020. https://www.theappbusiness.com/insights/servant-leadership-and-the-role-of-a-scrum-master.

91. toggl. "5 Stages of Team Development: Understanding Team Development Process." Accessed December 4, 2019. https://toggl.com/stages-of-team-development/.

92. TXM Lean Solutions. "The Definitive Guide to Visual Management." Accessed December 4, 2019. https://txm.com/visual-management/.

93. Tomić, Smilja. "Active Listening as a Tool for Scrum Masters." LinkedIn. Last modified December 18, 2017. Accessed February 29, 2020. https://www.linkedin.com/pulse/active-listening-tool-scrum-masters-smilja-tomi%C4%87.

94. Veerman, Elise. "Building a Self-Organizing Team." GAIKU Beta. Last modified November 9, 2017. Accessed February 29, 2020. https://gaiku.io/blog/self-organizing-team.

95. Verheyen, Gunther. *Scrum: A Pocket Guide: A Smart Travel Companion*. Netherlands: Van Haren Publishing, 2013.

96. Verwijs, Christiaan. "How to Kickstart a Great Scrum Team (10 Practical Things to Do)." The Liberators. Last modified August 11, 2017. Accessed February 29, 2020. https://medium.com/the-liberators/how-to-kickstart-a-great-scrum-team-10-practical-things-to-do-2143bdde1a8d.

97. "Making Communication in (Scrum and Other) Teams More Effective; a Few Skills to Learn." The Liberators. Last modified January 18, 2012. Accessed February 29, 2020. https://medium.com/the-liberators/making-communication-in-scrum-and-other-teams-more-effective-a-few-skills-to-learn-f447b457dd50.

98. Viscardi, Stacia. *The Professional ScrumMaster's Handbook – Break the Chains of Traditional Organization and Management (Professional Expertise Distilled)*. Birmingham, UK: Packt Publishing, 2013

99. Visual Paradigm. "What Is a Self-Organizing Team in Scrum?" Accessed December 4, 2019. https://www.visual-paradigm.com/scrum/what-is-self-organizing-team-in-scrum/.

100. Watts, Geoff. *Scrum Mastery - From Good to Great Servant Leadership*. Cheltenham, UK: Inspect and Adapt Ltd., 2013.

101. Wijetunge, Rumesh. "The 5 Whys Approach to Root Cause Analysis in Agile Teams." Knowledgehut Blog (blog). Last modified August 28, 2019. Accessed February 29, 2020. https://www.knowledgehut.com/blog/agile/root-cause-analysis-agile-teams.

102. Wrike Team. "7 Easy Steps to Encourage Self-Organization in Your Team." Wrike (blog). Last modified July 26, 2012. Accessed February 29, 2020. https://www.wrike.com/blog/7-easy-steps-to-encourage-self-organization-in-your-team/.

103. Your Dictionary. "Effective Communication." Accessed February 29, 2020. https://www.yourdictionary.com/effective-communication.

8. Glossary

Term	Definition
acceptance criteria	Acceptance criteria are a formal list of external quality characteristics specified by the Product Owner that give guidance about whether a product backlog item has been successfully developed and will therefore be accepted by the customer.
adaptation	One of the three pillars of empirical process control; and with the other two – transparency and inspection – provide the feedback necessary to make continual improvements to the product being developed.
affinity estimation	A quick and easy technique used by the development team to quickly estimate a large number of user stories using categories that indicate relative size. Example: t-shirt sizes (xs, s, m, l, xl)
affinity mapping	It's a quick-start, low-fidelity method for organizing a magnitude of low-level information points into key themes that can help you see the hundred-foot view rather than the one-centimeter view. Team members simply group items that are like-sized together, in a visual way, order them to form small to large groups, and assign estimation numbers to these different groups. It works best with a small group of people and a relatively small number of items.
Agile	A project management style focused on early delivery of business value, continuous improvement, scope flexibility, team input, and delivering well-tested products that reflect customer needs.
Agile Manifesto	The Agile Manifesto was a document written in 2001 by seventeen independent-minded software practitioners in Snowbird, Utah that is comprised of four foundational values and 12 supporting principles, which guide the development and delivery of high-quality, working software.
artifact	Scrum artifacts provide key information that the Scrum team and the stakeholders need to be aware of for understanding the product under development, the activities being planned, and the activities done in the project. Artifacts defined by Scrum are specifically designed to maximize the transparency of key information so that everybody has the same understanding of the artifact. The following artifacts are defined in Scrum: the product backlog, the sprint backlog, and the product increment.
board	A physical board to visualize information for and by the Scrum Team, often used to manage Sprint Backlog. Scrum boards are an optional implementation within Scrum to make information visible.
capacity	1. The quantity of resources available to perform useful work. 2. A concept used to help establish a Work in Progress (WIP) limit by ensuring that we only start work to match the available capacity to complete work.
Chronological Analysis	Chronological Analysis is a simple, commonsense approach towards tracking down the root cause of a problem. Chronological Analysis works by piecing together a timeline of activities working back from the problem being raised or otherwise bought to the attention of the relevant staff.
colocation	Colocation is having all Scrum core team members located in the same work place leveraging the advantages of better coordination, problem-solving, knowledge sharing, and learning.
Conditions of Satisfactions (COS)	The conditions under which a product owner would be satisfied that a product backlog item is done. Conditions of satisfaction are acceptance criteria that clarify the desired behavior.
daily Scrum	The daily Scrum is a 15-minute time-boxed event for the development team that is held at the same time and place every day of the sprint. During this time the development team plans work for the next 24 hours. The daily Scrum is a synchronization, inspection, and adaptive planning activity that optimizes team collaboration and performance.
daily stand-up	See daily Scrum.

Term	Definition
decomposition	Decomposition is a technique whereby high-level tasks are broken down into lower level, more detailed tasks. The user stories are decomposed into tasks by members of the Scrum team. Prioritized product backlog user stories should be sufficiently decomposed to a level that provides the Scrum team adequate information to create deliverables from the tasks mentioned in the task list.
DEEP	An acronym coined by Roman Pichler and Mike Cohn for remembering a set of criteria used to evaluate the quality of a product backlog. The criteria are detailed appropriately, emergent, estimated, and prioritized.
definition of done	An agreed-upon list of the activities deemed necessary to get a product increment, usually represented by a user story, to a done state by the end of a sprint. The definition of done ensures everyone on the team knows exactly what is expected of everything the team delivers. It supports transparency and quality fit for the purpose of the product and organization.
definition of ready	Definition of ready involves creating clear criteria that a user story must meet before being accepted into an upcoming iteration. This is typically based on the INVEST set of criteria.
development team	A development team is structured as a self-organizing and cross-functional team of three to nine people who are guided by the Scrum Master and are responsible for delivering potentially shippable product increments every sprint.
development work	Development work refers to creating and refining, estimating, and prioritizing product backlog items into increments of potentially shippable functionality.
empirical process control	An empirical process is a process based on empiricism, which asserts that knowledge comes from experience and decisions are made based on what is known.
epic	Large, unrefined, user story in the product backlog that is written in the initial stages of the project and can span an entire release.
Fault Tree Analysis (FTA)	A fault tree analysis is realized using a top-down diagram to analyze an unwanted state or failure of a system. Events are broken down to lower-level events that are combined using Boolean logic operators (AND, OR, and their variations). The goal of the analysis is to identify and minimize the risks and failures of systems.
Fault Isolation	The function of fault isolation is to exactly locate the reason and/or the origin of fault. The step of fault isolation ensures that we are able to retrieve some information about the fault such as fault type and/or location. Is one of the three main steps of FDD (Fault detection and diagnosis) procedure.
Fibonacci Sequence	The Fibonacci scale consists of a sequence of numbers used for estimating the relative size of user stories in points, resulting in reduced complexity, effort, and doubt when determining the time required for product backlog items. Each number is the sum of the two preceding ones (1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144…). It was named after the Italian mathematician Leonardo of Pisa.
Five-Why's	Five whys (or 5 whys) is an iterative interrogative technique used to explore the cause-and-effect relationships underlying a particular problem. The primary goal of the technique is to determine the root cause of a defect or problem by repeating the question "Why?". Each answer forms the basis of the next question. The "five" in the name derives from an anecdotal observation on the number of iterations needed to resolve the problem. Not all problems have a single root cause.
fixed-date release	A release that must be delivered on a known future date. The scope of the release, and possibly the cost, needs to be flexible. Contrast with fixed-scope release.
fixed-scope release	A release that must have a specific set of features. The date on which the features are delivered and/or the costs are flexible. Contrast with fixed-date release.
Hypothesis Testing	Hypothesis Testing is the statistical validation of the Root Causes identified in the 5 Why's and Root Causes Analysis. Once the potential causes are functionally identified,

Term	Definition
	Hypothesis Testing helps to statistically reject or fail to reject these assumptions with a pre-defined level of confidence in making the right decision.
impediment	Impediments can be any type barricades, hurdles or obstacles that are considered "blockers" that prevent a Scrum Team from completing work, which in return impacts velocity. Anything that prohibits the team from doing work is considered an impediment. Impediments can range from a sick team member, a missing resource, faulty equipment, cultural or waterfall issues lack of management support, unresolved dependencies or even a cold team room.
impediment log	An impediment log is a file where all impediments are recorded. The Scrum Master is responsible for tracking, monitoring and ensuring that impediments are adressed. Organizational impediments added to the Impediments Log are prioritized and addressed on an ongoing basis. There should only be a single Impediment log for a Scrum Master to manage.
incremental development	In an Agile context, incremental development means that each successive version of the product is usable, and each builds upon the previous version by adding user-visible functionality.
information radiator	Information radiator is a popular term invented by Alistair Cockburn that is used to describe any artifact that conveys project information and is publicly displayed in the workspace or surroundings. Information radiators are very popular in the Agile world, and they are an essential component of visual management. Information radiators can be digital or physical.
inspection	Scrum users must frequently inspect Scrum artifacts and progress toward a sprint goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.
INVEST	The acronym INVEST stands for a set of criteria used to assess the quality of a user story. If the story fails to meet one of these criteria, the team may want to reword it. The criteria are: • Independent • Negotiable • Valuable • Estimable • Small • Testable
Ishikawa Diagrams	Ishikawa diagram, also called fishbone diagram or cause-and-effect diagram, is an effective and quick way to identify the root cause of an issue or defect and decide on corrective actions. It helps you identify all possible causes of a problem and use the outcome of process improvements. The resultant diagram resembles a fishbone. It also can be used in conjunction with the 5 Whys tool.
iterative development	Agile projects are iterative insofar as they intentionally allow for "repeating" software development activities, and for potentially "revisiting" the same work products (the phrase "planned rework" is sometimes used; refactoring is a good example). They are iterative in a third, less-essential sense, in being most often structured around a series of iterations of fixed calendar length.
iteration	An iteration, in the context of an Agile project, is a timebox during which development takes place, the duration of which: 1. may vary from project to project, usually between 1 and 4 weeks, 2.is in most cases fixed for the duration of a given project. Nowadays the terms "iteration" and "sprint" are used interchangeably without any particular connotation attached.

Term	Definition
ITIL	ITIL (Information Technology Infrastructure Library) is a set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business.
kanban	The *kanban* method is a product and systems thinking approach to improving delivery of services or products to customers and the environment in which those delivering the service or product operate and interact with one another.
minimally viable product (MVP)	A minimum viable product is, as Eric Ries said, the "version of a new product that allows a team to collect the maximum amount of validated learning about customers with the least effort."
MoSCoW Method	The MoSCoW method is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement; it is also known as the MoSCoW prioritization or MoSCoW analysis. The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories (must have, should have, could have, and won't have), with the interstitial Os added to make the word pronounceable. While the Os are usually in lower-case to indicate that they do not stand for anything, the all-capitals MOSCOW is also used.
paired programming	Agile software development practice popularized by Extreme Programming in which two team members jointly create new functionality.
personas	Personas are highly detailed fictional characters, representative of the majority of users and of other stakeholders who may not directly use the end product. Personas are created to identify the needs of the target user base.
planning onion, Cohn's	Mike Cohn refers to multi-layer planning as the planning onion. At different layers of this planning onion, the project team focuses on varying level of details at different stages of project. The planning onion helps teams choose the right level of planning for each timeframe for which they are planning, and has six levels - strategy, portfolio, release, iteration, daily, and continuous.
planning poker/estimation poker	An approach to estimation used by Agile teams used to estimate relative sizes of product backlog items that balances group thinking and individual thinking.
potentially shippable product	Results that are completed to a high degree of confidence and represent work of good quality that is potentially shippable to end customers at the end of a sprint. Being potentially shippable does not mean the results will actually be delivered to customers. Shipping is a business decision; potentially shippable is a state of confidence.
potentially shippable product Increment	The output of every Sprint is called a Potentially Shippable Product Increment. The work of all the teams must be integrated before the end of every Sprint—the integration must be done during the Sprint.
product backlog	A prioritized list of new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome of a product. The Product Owner is the sole person responsible for managing the product backlog.
product backlog grooming	Backlog grooming is when the Product Owner and some, or all, of the rest of the team refine the backlog on a regular basis to ensure the backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery.
product backlog item	1. An item such as a feature, defect, or occasionally technical work that is valuable from the Product Owner's perspective. 2. An item in the product backlog.
product increment	Product Increment is the integration of all the completed list of Product Backlog items during the sprint. The Product Increment goes on getting incremented in the subsequent sprints. So, in a particular sprint, the Product increment is the integration of all the completed list of Product Backlog Items where as in a Project, Product Increment is the integration of all the completed list of Sprint backlog items. With each sprint, the product increment increases in terms of delivered functionality.

Term	Definition
Product Owner	The Product Owner is responsible for maximizing the value of the product resulting from work of the development team. He or she is responsible for making sure the team delivers the desired outcome. The Product Owner is the sole person responsible for managing the product backlog. The Product Owner is a part of the Scrum Team.
product roadmap	A description of the incremental nature of how a product will be built and delivered over time, along with the important factors that drive each individual release. Useful when developing a product that will have more than one release.
product vision	A brief statement of the desired future state that would be achieved by developing and deploying a product. A good vision should be simple to state and provide a coherent direction to the people who are asked to realize it.
release	1. A combination of features that when packaged together make for a coherent deliverable to customers or users. 2. A version of a product that is promoted for use or deployment. Releases represent the rhythm of business-value delivery and should align with defined business cycles.
release backlog	The goal of a given release is to deliver a subset of the product backlog, known as the release backlog. After identifying which user stories will go into a particular release, the user stories become part of a release backlog, which are then prioritized by the development team, who estimate the amount of time needed to complete each item.
release plan	1. The output of release planning. On a fixed-date release, the release plan will specify the range of features available on the fixed future date. On a fixed scope release, the release plan will specify the range of sprints and costs required to deliver the fixed scope. 2. A plan that communicates, to the level of accuracy that is reasonably possible, when the release will be available, what features will be in the release, and how much will it cost.
release planning	Release planning is about making the scope, date, and budget trade-offs for incremental deliveries. It is all about 'high-level planning' of multiple sprints (three to twelve iterations). Most of the times, it is sensible and important to carry out Initial Release Planning after product planning and before beginning the first Sprint related to the Release.
return on investment (ROI)	Return on investment (ROI), when used for project justification, assesses the expected net income to be gained from a project. It is calculated by deducting the expected costs or investment in a project from its expected revenue and then dividing this (net profit) by the expected costs in order to get a return rate.
risk	Risk is defined as an uncertain event or set of events that can affect the objectives of a project and may contribute to its success or failure.
root-cause analysis	Root cause analysis (RCA) is a systematic process for identifying "root causes" of problems or events and an approach for responding to them. RCA is based on the basic idea that effective management requires more than merely "putting out fires" for problems that develop, but finding a way to prevent them.
Scrum	A framework, in use since the early 1990s, within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.
Scrum event	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. These include, but are not limited to, the sprint, sprint planning, sprint review, and sprint retrospective. Each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection.
Scrum Master	The Scrum Master works with the Product Owner, development team, and other involved parties to ensure artifacts maintain transparency. They ensure that the Scrum events take place and that the attendants understand their purpose. They also teach the development team to keep the events within the time-box.
Scrum of Scrums (SoS)	A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of five to 10. Each daily Scrum within a sub-team

Term	Definition
	ends by designating one member as the "ambassador" to participate in a daily meeting with the ambassadors from other teams, called the Scrum of Scrums.
Scrum team	A self-organizing team consisting of the Scrum Master, the Product Owner and the Development Team.
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.
Scrum board	A Scrum board is a tool used by the Scrum team to plan and track progress during each sprint. The Scrum board contains four columns to indicate the progress of the estimated tasks for the sprint: a “to do” column for tasks not yet started, an “in progress” column for the tasks started but not yet completed, a “testing” column for tasks completed but in the process of being tested, and a “done” column for the tasks that have been completed and successfully tested.
self-organization	1. A bottom-up emergent property of a complex adaptive system whereby the organization of the system emerges over time as a response to its environment. 2. A property of a development team that organizes itself over time, without an external dominating force applying traditional top-down, command-and-control management. 3. Reflects the management philosophy whereby operational decisions are delegated as much as possible to those who have the most detailed knowledge of the consequences and practicalities associated with those decisions.
servant leader	Servant leaders employ listening, empathy, commitment, and insight while sharing power and authority with team members. Servant leaders are stewards who achieve results by focusing on the needs of the team. This style is the embodiment of the Scrum Master role.
SMART	SMART is a mnemonic/acronym, giving criteria to guide in the setting of objectives, for example in project management, employee-performance management, and personal development. The letters S and M generally mean specific and measurable. Possibly the most common version has the remaining letters referring to attainable (or achievable), relevant/ realistic, and timely.
Sologic Method	Sologic Root Cause Analysis (RCA) is a universal, scalable problem-solving methodology and is built on the principle that causal relationships exist for all events and can be graphically modeled by using evidence-based inputs along with conditional logic. Minimizes personal bias and maximizes analytical thinking.
sprint	The heart of Scrum is a sprint, a time-box of one month or less during which a “done”, usable, and potentially releasable product increment is created. Sprints have consistent durations throughout a development effort and 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. During the sprint: No changes are made that would endanger the sprint goal.
sprint backlog	The sprint backlog is the set of product backlog items selected for the sprint, plus a plan for delivering the product increment and realizing 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.
sprint burndown and burnup charts	Burndown charts and burnup charts track the amount of output (in terms of hours, story points, or backlog items) a team has completed across an iteration or a project.
sprint demo	1. An activity of a sprint review where the completed (done) product backlog items are demonstrated with the goal of promoting an information-rich discussion between the Scrum team and other sprint review participants. 2. A term that is frequently used synonymously to refer to the entire sprint review.
sprint execution	Sprint Execution is performed during each Sprint by the Scrum team (the Scrum Master, development team, and Product Owner) to decide the most ideal way to meet the Sprint goal. It starts after Sprint Planning and ends before Sprint Review.

Term	Definition
sprint goal	A high-level summary of the goal the Product Owner would like to accomplish during the sprint. Frequently elaborated through a specific set of product backlog items.
sprint planning	The sprint planning meeting is conducted at the beginning of a sprint as part of the create-a-sprint-backlog process. It is time-boxed to eight hours for a one-month sprint and is divided into two parts – objective definition and task estimation.
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 at most a three-hour 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.
sprint review	A sprint review is held at the end of the sprint to inspect the increment 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 the presentation of the increment is intended to elicit feedback and foster collaboration.
stakeholders	Stakeholders are people who affect or are affected by your project. Internal stakeholders are within your company or organization; they could be from the legal, sales, marketing, management, procurement, or any other division of your company. External stakeholders could be investors or users.
strategic technical debt	A form of technical debt that is used as a tool to help organizations better quantify and leverage the economics of important, often time-sensitive, decisions. Sometimes taking on technical debt for strategic reasons is a sensible business choice. Contrast with naive technical debt, which is unavoidable technical debt.
story-mapping	Story mapping consists of ordering user stories along two independent dimensions. The "map" arranges user activities along the horizontal axis in rough order of priority (or "the order in which you would describe activities to explain the behaviour of the system"). Down the vertical axis, it represents increasing sophistication of the implementation.
story point	Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work. Product backlog items such as stories, are usually sized/estimated with story points. This is usually based on the Fibonacci sequence or t-shirt sizes.
sustainable pace	The team aims for a work pace that they would be able to sustain indefinitely. This entails a firm refusal of what is often considered a "necessary evil" in the software industry – long work hours, overtime, or even working nights or weekends. As such this "practice" is really more of a contract negotiated between the team and their management.
task board	An information radiator used during the sprint execution to communicate the progress and flow of task-level work within a sprint, and ensures efficient dissemination of relevant information to the entire team.
technical debt	Technical debt (also referred to as design debt or code debt) refers to the work that teams prioritize lower, omit, or do not complete as they work towards creating the primary deliverables associated with the project's product. Technical debt accrues and must be paid in the future.
Technical Observation Post (TOP)	A prearranged gathering of specialist technical support staff from within the IT support organization brought together to focus on specific aspects of IT Availability. Its purpose being to monitor events, real-time as they occur, with the specific aim of identifying improvement opportunities or bottlenecks which exist within the current IT Infrastructure.
technical team	See development team.
three pillars	Scrum uses the empirical process control that relies on the three pillars of transparency, inspection, and adaptation.

Term	Definition
timebox/timeboxing	A timebox is a previously agreed period of time during which a person or a team works steadily towards the completion of some goal. Rather than allow work to continue until the goal is reached, and evaluating the time taken, the timebox approach consists of stopping work when the time limit is reached and evaluating what was accomplished.
transparency	Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen. For example, those performing the work and those inspecting the resulting increment must share a common definition of "done".
triangulation	Estimation is done using triangulation: choose several user stories as the reference for the different sizes (S, M, L, XL, XXL) and the other user stories will be estimated with comparison to the chosen one.
unavoidable technical debt	A form of technical debt that is usually unpredictable and unpreventable and accrues through no fault of the team building the product. Contrast with the naive technical debt, or strategic technical debt.
user stories	In consultation with the customer or Product Owner, the team divides up the work to be done into functional increments called "user stories." Each user story is expected to yield, once implemented, a contribution to the value of the overall product, irrespective of the order of implementation; these and other assumptions as to the nature of user stories are captured by the INVEST set of criteria. A user story is similar to a PBI, but it goes beyond a specific change or requirement.
velocity	At the end of each iteration, the team adds up effort estimates associated with user stories that were completed during that iteration. This total is called velocity. Knowing velocity, the team can compute (or revise) an estimate of how long the project will take to complete, based on the estimates associated with remaining user stories and assuming that the velocity over the remaining iterations will remain approximately the same. This is generally an accurate prediction, even though rarely a precise one.
visual management	The practice of using information visualization techniques to manage work. A simple example is using sticky notes on a wall to manage a list of tasks, a better (and more complex) example is *kanban*.
voice of customer	The voice of the customer (VoC) can be referred to as the explicit and implicit requirements of the customer, which must be understood prior to the designing of a product or service. The Product Owner represents the voice of the customer.
Waterfall methodology	Waterfall is a linear approach to software development process. Projects are broken down into linear and sequential stages, where every piece of the project relies on the completion of preceding deliverables.

GLOSSARY RESOURCES

Agile Alliance. "Agile Glossary." Last modified: 2011. Accessed: October 30, 2019. https://www.agilealliance.org/agile101/agile-glossary/

Ken Schwaber and Jeff Sutherland. *The Scrum Guide™, The Definitive Guide to Scrum: The Rules of the Game*. ScrumAlliance, 2017. Accessed August 12, 2019. https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide

Rubin, Kenneth. E*ssential Scrum: A Practical Guide to the Most Popular Agile Process*. Boston, MA: Pearson Education, 2013.

Volkerdon.com. "Agile Glossary." Accessed October 30, 2019. https://www.volkerdon.com/pages/agile-glossary

California Department of Technology. "The Agile Project Charter." Agile. Accessed February 27, 2020. https://projectresources.cdt.ca.gov/agile/the-agile-project-charter/

Canty, Denise. "Impediment Logs in Scrum". Access February 1st 2021. https://www.projectmanagement.com/blog-post/46050/Impediment-Logs-in-Scrum

Layton, Marc C. *Agile Project Management for Dummies*. New Jersey: John Wiley & Sons, 2017.

Lowe, David. "Acceptance Criteria and Conditions of Satisfaction." Scrum & Kanban, October 26, 2015. https://scrumandkanban.co.uk/acceptance-criteria-and-conditions-of-satisfaction/

Morris, David. *Scrum in Easy Steps: - An Ideal Framework for Agile Projects*. In Easy Steps Limited, 2017.

"'Open Campus' Multiple Levels of Planning." Agile Certified Practitioner (PMI-ACP). Accessed August 13, 2019. https://www.greycampus.com/opencampus/agile-certified-practitioner/multiple-levels-of-planning

Rubin, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Boston, MA: Pearson Education, 2013, 14.

MoSCoW method. Accessed February 26, 2021. https://en.wikipedia.org/wiki/MoSCoW_method

Root Cause Analysis. Accessed February 26, 2021. https://en.wikipedia.org/wiki/Root_cause_analysis

Five whys. Accessed February 26, 2021. https://en.wikipedia.org/wiki/Five_whys

Scrum.org. "Scrum Glossary". Accessed February 26, 2021. https://www.scrum.org/resources/scrum-glossary

KnowledgeHut. "Scrum Tutorial." Accessed February 26, 2021. https://www.knowledgehut.com/tutorials/scrum-tutorial

Software Ideas. "Fault Tree Analysis Diagram (FTA)". Accessed February 26, 2021. https://www.softwareideas.net/fta-fault-tree-analysis

Agile Alliance. "Agile Glossary". Last modified: 2011. Accessed: October 30, 2019. https://www.agilealliance.org/agile101/agile-glossary/

Rubin, Kenneth. E*ssential Scrum: A Practical Guide to the Most Popular Agile Process*. Boston, MA: Pearson Education, 2013.

Volkerdon.com. "Agile Glossary". Accessed October 30, 2019. https://www.volkerdon.com/pages/agile-glossary

Sologic. "Root Cause Analysis." Accessed February 26, 2021. https://www.sologic.com/en-us/about/root-cause-analysis

Simplilearn. "Problem Detection and Resolution." Accessed February 26, 2021. https://www.simplilearn.com/problem-detection-and-resolution-part-1-tutorial

This page is left intentionally blank -

This page is left intentionally blank --

This page is left intentionally blank -

This page is left intentionally blank -


PeopleCert
Scrum.