

Scrum Alliance[®]

Certified ScrumMaster[®] Level 2

Learning Objectives

February 2017

by the Scrum Alliance CSP[®] Learning Objectives Committee

Introduction

Purpose

This document describes the Learning Objectives (LOs) that must be covered in a Certified ScrumMaster Level 2 course. These Learning Objectives take the following into consideration:

- Every implementation of Scrum is different.
- Teams and organizations apply Scrum within their context, but the fundamental framework always remains the same.

The Learning Objectives for this course are based on:

- *Scrum Guide*, <http://scrumguides.org>
- Agile Manifesto, 4 values and 12 principles, <http://www.agilemanifesto.org>
- Scrum values, <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>

Scope

Scrum Alliance has adopted the *Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game*, coauthored and updated (most recently in 2016) by the co-creators of the Scrum framework, as the guiding curriculum for this course. CSM[®] Level 2 and CSPO[®] Level 2 candidates are expected to build a body of knowledge of the Scrum framework, including its roles, events, and artifacts. Incorporating Scrum principles and practices takes diligence, patience, and a commitment to continuous improvement. Scrum is a framework, not a prescriptive methodology.

Participants in a CSM Level 2 course should expect that each Learning Objective identified in this document will be covered in a CSM Level 2 educational offering. The CSM Learning Objectives fall into the following categories:

1. **Lean, Agile, and Scrum**
2. **Agile Facilitation**
3. **Agile Coaching**
4. **Service to the Development Team**
5. **Service to the Product Owner**
6. **Service to the Organization**
7. **Scrum Mastery**

Individual trainers (CSTs) or coaches (CECs) may choose to teach ancillary topics. Ancillary topics presented in a CSM Level 2 course must be clearly indicated as such.

Learning Objectives

A note about examples used in the following Learning Objectives:

Several Learning Objectives include a list of examples. The examples are used to clarify the intent of the objective. Individual trainers or coaches can use the provided examples, their own examples that still meet the objective, or a mix of both. Examples do not imply that they are the only options, nor that they constitute an exhaustive list.

A note about Bloom's Taxonomy:

*While some Learning Objectives appear to prescribe how to teach, that is not the intent. Bloom's-style Learning Objectives describe what the learner can do upon completing the class. Rather than include that text in each Learning Objective, please mentally append the following phrase to each objective: "After successful completion of the **CSM Level 2 educational offering**, the learner will be able to ..."*

1. Lean, Agile, and Scrum

Agile and Lean Values, Principles, and Worldview

- 1.1. ... demonstrate how the values and principles of the Agile Manifesto are present in Scrum (e.g., frequent inspection and adaptation in review, retrospective, Daily Scrum).
- 1.2. ... outline the historical development of Scrum and Agile (i.e., origins in Lean and OOP/OOD, first Scrum teams in 1980s, first publication from OOPSLA96, Schwaber/Beedle Book 2001).
- 1.3. ... describe at least two other Lean/Agile development frameworks outside of Scrum and explain their value (e.g., LSD, XP, Kanban).
- 1.4. ... discuss a scenario, based upon your personal experience, where there has been violation of Agile principles, and demonstrate how it may be rectified and/or addressed by the Scrum Master.
- 1.5. ... debate at least five personality traits of an excellent ScrumMaster (e.g., proactive, curious, humble, improving, learning, responsible, committed).

Empirical Process Control

- 1.6. ... describe the function of the inspect-and-adapt process in the Daily Scrum, sprint planning, sprint reviews, and retrospectives.
- 1.7. ... evaluate three situations when transparency, inspection, and adaptation are not working effectively (e.g., when the Daily Scrum is just used for status reporting, when retrospectives are skipped, when the results of a sprint review do not influence the product backlog).

2. Agile Facilitation

Basic Facilitation

- 2.1. ... identify at least three indicators when a group is engaged in divergent thinking and at least three indicators when a group is engaged in convergent thinking.
- 2.2. ... identify at least three challenges of integrating multiple frames of reference (i.e., the “Groan Zone”).
- 2.3. ... describe at least three ways a group could reach their final decision (e.g., fist of five, decider protocol, majority vote, etc.).
- 2.4. ... describe at least five facilitative listening techniques (e.g., paraphrasing, mirroring, making space, stacking, etc.) for effective meetings/events and apply at least two of them.
- 2.5. ... describe, using two concrete examples, when the Scrum Master should not act as the facilitator for the Scrum Team.
- 2.6. ... plan the contents and an agenda for at least two collaborative meetings and demonstrate the facilitation of these meetings.

3. Agile Coaching

Coaching Fundamentals

- 3.1. ... demonstrate a coaching stance in an interaction with one or more people (i.e., neutrality, self-awareness, client agenda, etc.) and describe how that coaching stance impacted the interaction.
- 3.2. ... apply at least three coaching techniques (e.g., active listening, powerful questions, reflection, feedback, GROW model, etc.) with team members, Product Owners and/or stakeholders, and describe how the coaching technique impacted each interaction.

4. Service to the Development Team

Self-Organization

- 4.1. ... apply at least two coaching techniques to foster greater self-organization within teams (e.g., powerful questions, autonomy/mastery/purpose, active listening, etc.).
- 4.2. ... apply a countermeasure to reduce the impact of at least three different challenges facing a self-organizing team (e.g., bad forecast, technical debt, someone is leaving the team).
- 4.3. ... describe how a self-organizing team approaches at least three challenges that may occur during a retrospective.

Team Dynamics

- 4.4. ... explain the difference between a working group and a team (e.g., teams demonstrate on-demand leadership, ability to deal with conflicts,

- equal voice, well-known and practiced norms, shared goals, mutual accountability, long-term composition, full dedication).
- 4.5. ... identify at least three key attributes of effective Agile Teams (e.g., ground rules in place, awareness of capabilities and capacities, effective and efficient collaboration).
 - 4.6. ... apply at least two methods for improving team performance (e.g., common goals/purpose, shared accountability, working agreement, psychological safety, etc.).
 - 4.7. ... identify at least two pitfalls of a homogenous team (i.e., lack of different perspectives, experiences, and viewpoints).
 - 4.8. ... describe a multi-staged model for team formation and development (e.g. the Tuckman model).

Definition of Done

- 4.9. ... organize and facilitate the creation of a strong Definition of Done with the Product Owner and Development Team.
- 4.10. ... apply at least two techniques that could be employed to encourage the Scrum Team to improve how they describe “Done.”
- 4.11. ... describe how a Definition of Done could be formulated for a non-software product (e.g., insurance tariff, hardware, event).

Value of Engineering Practices

- 4.12. ... describe at least five technical practices (e.g., from Extreme Programming: test-driven development, pair programming, continuous integration, collective code ownership, refactoring) that will help Scrum Teams deliver a high-quality product increment and reduce technical debt each sprint.
- 4.13. ... describe at least three ways technical practices may impact the Development Team’s ability to deliver a potentially releasable Increment each sprint (e.g., continuous integration helps to detect integration errors earlier and speed up releasing, refactoring improves product quality and thus minimizes adjustments for new features, collective code ownership reduces island knowledge and bottlenecks due to unnecessary specialization).
- 4.14. ... identify at least three engineering practices that are essential when using Scrum at scale (e.g., simple design, continuous integration, test-driven development).

5. Service to the Product Owner

Coaching the Product Owner

- 5.1. ... practice facilitating the creation (or refinement) of the product vision between the Product Owner and the Development Team.
- 5.2. ... explain at least two techniques for moving from product vision to product backlog (e.g., product vision board, business model or Lean canvas, customer journey, impact mapping, user story mapping).
- 5.3. ... list three benefits that arise if a Product Owner participates in the retrospective.

5.4. ... organize and facilitate a product backlog refinement session with stakeholders and/or team members and explain two techniques that could be used to create product backlog items that are ready to be taken into the next sprints (e.g., PBI splitting, BDD, SbE, estimating).

5.5. ... explain Scrum to a business stakeholder (e.g., as in “Agile product ownership in a nutshell” by Henrik Kniberg).

6. Service to the Organization

Resolving Impediments

6.1. ... identify at least three typical impediments for a Scrum Team and describe at least one way to address them (e.g., late attendance in meetings, blocked work, supplier issues).

6.2. ... list at least three techniques to evaluate impediments in depth (e.g., root-cause analysis, fishbone, 5 whys) and describe when they might not be working.

6.3. ... analyze an impediment and identify a root cause(s) and/or underlying issue(s).

Scaling Scrum

6.4. ... illustrate, with at least two reasons, why scaling might not be such a great idea (e.g., products created by small teams, communication overhead, TCO).

6.5. ... identify at least three techniques for visualizing, managing, or reducing dependencies between teams.

6.6. ... differentiate the impact of feature teams versus component teams on the delivery of value.

6.7. ... recognize at least different three scaling frameworks or approaches (LeSS, DAD, Scrum at Scale, Enterprise Scrum, etc.).

6.8. ... experiment with at least one large-scale, participatory meeting format (Open Space, World Cafe, etc.) to scale Scrum meetings.

Organizational Change

6.9. ... apply at least two techniques to effect change in an organization in order to help Scrum Teams be more productive.

7. Scrum Mastery

Personal Development

7.1. ... evaluate your personal fulfillment of the five Scrum Values and identify how you could improve upon at least two of them (e.g., using a radar chart or other scale).

7.2. ... analyze your own fundamental driving factors (e.g., respect, wealth, relationships).

7.3. ... describe three characteristics of a destructive conflict (e.g., emotionality, tone of voice, low interest in solution).

7.4. ... compare at least three different ways to respond to conflict (e.g., denial, consensus, giving in, overpowering, withdrawal), and reflect on your default pattern(s) for responding to conflict.

Servant Leadership

7.5. ... describe at least two goals of a servant leader and express at least three attributes of an effective servant leader (e.g., putting people first, communicating skillfully, being a systems thinker).

7.6. ... appraise, through two specific examples, how the Scrum Master attempted to resolve an organizational impediment while showing the attributes of a servant leader (e.g., how did they put people first? How did they show that they are skilled communicators? How did they demonstrate that they are systems thinkers?).