ScrumAlliance®
Transforming the World of Work®

Scrum Alliance®
Certified ScrumMaster® Level 3
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 3 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 3 and CSPO® Level 3 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 3 course should expect that each Learning Objective identified in this document will be covered in a CSM Level 3 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 3 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 3 educational offering, the learner will be able to … "

ScrumAlliance®
Transforming the World of Work®

1. Lean, Agile, and Scrum

Lean Thinking

1.1. … describe the roots of Lean Thinking (e.g., TQM, Toyota Production System).
1.2. … define a kaizen mindset (e.g., focus on people, optimize the whole, continuous improvement).
1.3. … list at least five of the seven wastes in product development (e.g., extra features, partially done work, extra processes, handoffs, defects, delays, task switching).
1.4. … relate the seven wastes in product development to the seven wastes in Lean manufacturing.
1.5. … explain the five core concepts of Lean Thinking (e.g., one-piece flow, pull, limit WIP, small batches, kaizen, reduce variability, teamwork) and how they can be applied to Scrum.
1.6. … relate at least three Agile engineering practices to Lean practices (e.g., continuous integration, test-driven development, collective code ownership).

2. Agile Facilitation

Advanced Facilitation

2.1. … describe three alternatives to open discussion (e.g., structured go-arounds, individual writing, listing ideas, dialogue in pairs or small groups, etc.) and explain when they may be effective.
2.2. … practice at least two alternatives to open discussion.
2.3. … identify at least one action the facilitator can perform to support meeting participants during divergent thinking, integration, convergent thinking, and closure that will support the development of an inclusive solution (e.g., powerful questions).
2.4. … apply five visual facilitation techniques for a collaborative session (e.g., card question, clustering, dot voting, visual note taking).
2.5. … design and facilitate a retrospective with senior leaders and executives to foster continuous improvement at the organizational level.

2.6. … identify at least three practices for facilitating remote meetings (e.g., turn-taking between those face-to-face with remote participants, establishing communication protocol, shared note taking).

3. Agile Coaching

3.1. … describe at least five elements of a fundamental coaching agreement (e.g., role of the coach, duration, expectations, feedback, responsibilities).
3.2. … discuss the importance of at least two fundamental coaching assumptions (i.e., coaching is at the request of the coachee, coachee knows the solution to their problem).
3.3. … list at least three fundamental psychological concepts that help understand and transform individual behavior (e.g., EQ/emotional intelligence, MBTI, ORSC, DISC, transactional analysis, mindset, empathy, DRIVE).

4. Service to the Development Team

Team Dynamics

4.1. … apply at least two different models for team development (e.g., Tuckman model, team performance curve, etc.).
4.2. … appraise the effectiveness of at least two different development frameworks for supporting an Agile team's growth.

Team Effectiveness

4.3. … apply at least three techniques for addressing team dysfunctions (e.g., building trust, encouraging healthy conflict, fostering mutual accountability, etc.).
4.4. … apply at least three techniques or activities (e.g. physical team building exercises like abseiling, empathy walks or trust fall, trust-building conversation modes like the McCarthy protocols, NVC, appreciative inquiry, emphasis on consistent behavior of the Product Owner or stakeholders) for building trust in a team.

Starting New Agile Teams

4.5. … explain at least three reasons why the start of a new Agile Team should be handled differently from a traditional project kickoff/charter (e.g., level of collaboration, lack of experience in Agile environments, importance of shared understanding).

4.6. … explain how purpose, alignment, and context are set and used during the start-up of a new Scrum Team to accelerate teamwork.

4.7. … explain how seeing the whole system, emphasizing collaborative work, focusing on a good start, continuous learning, and "good enough for now" support the launch of a new Agile team.

4.8. … describe at least three responsibilities each for the sponsor (e.g., clarify constraints, context, and stakeholder expectations), Product Owner (communicate vision, purpose, and customer needs), and Development Team members (get to know each other, create transparency about capabilities, create ground rules and working agreements) when starting a new Agile team.

4.9. … organize and facilitate the launch of a new Agile team that defines purpose, alignment, and context.

Software Craftsmanship

4.10. … define software craftsmanship.

4.11. … demonstrate at least two training or coaching methods to introduce more advanced engineering practices.

4.12. … describe the acceptance criteria for a Product Backlog item using a format suitable for automated testing (e.g., Fitnesse, Gherkin).

Coaching the Development Team

4.13. … create a coaching agreement with the Development Team.

4.14. … design learning and capability goals for one individual (or team) and analyze the execution of these goals.

4.15. … demonstrate at least two techniques for raising team accountability (eg. The Five Steps from Responsibility to Accountability, facilitation of a Community of Practice).

5. Service to the Product Owner

Coaching the Product Owner

5.1. … organize and facilitate the creation (or refinement) of the product vision between the Product Owner and stakeholders.

5.2. … apply at least two techniques for moving from product vision to product backlog (e.g., innovation games, user story mapping, user story workshop, brainstorming, etc.).

5.3. … appraise at least three criteria that can be used for structuring a complex or multi-team product backlog (e.g., feature area, team).

6. Service to the Organization

Organizational Development

6.1. … describe the nature of complex systems (eg. cause-and-effect only visible after the event, high level of uncertainty and disagreement, emerging systems, products and practices).

6.2. … explain the importance of taking a systemic view (i.e. convince a stakeholder that the system as a whole needs to be optimized, regard the bigger picture, understand causal loops and delayed effects).

6.3. … describe at least two systematic methods for helping organizations improve their Scrum adoption (eg. causal loop analysis, value stream mapping).

6.4. … apply one systematic development approach (e.g., systems thinking).

6.5. … describe at least two frameworks for catalyzing organizational change (e.g., Kotter's 8-Step model, the Grief Cycle, 4D Model /Appreciative Inquiry).

6.6. … describe your approach to a complex intervention that addresses the root cause(s) of an organizational dysfunction and analyze the long-term impact (i.e. identify the situation, the underlying root cause(s), list measures/experiments and results).

6.7. … demonstrate at least two tangible examples of how you developed and changed the culture of your team (or organization) from a command-and-control to an Agile mindset.

6.8. … identify at least three ways the cultural change from a command-and-control to an Agile mindset added value to the Development Team, Product Owner, and eventual product.

Scaling

6.9. … describe an organizational design that enables multiple-team Scrum (e.g., feature teams, shared product backlog, Spotify, Matrix).

6.10. … explain the pitfalls of too much or too little prescription (e.g., lack of autonomy, lack of alignment, no slack, integration mess, overly detailed planning, not meeting the Definition of Done, overly slow pace, death by meetings, etc.).

6.11. … contrast two patterns for scaling the Product Owner role (e.g., shifting clarification responsibility to the Development Team, defining feature areas or different sub-products, PO team, Chief Product Owner).

6.12. … describe at least five techniques to improve inter-team collaboration (e.g., Scrum of Scrums meeting, Open Space, shared planning sessions, colocation, cross-visits during Daily Scrum meetings, etc.) and experiment with at least two of them.

6.13. … explain at least three benefits of supporting strong engineering practices when working with multiple teams.

6.14. … organize and facilitate at least one large-scale, participatory meeting format (Open Space, World Cafe, etc.) to scale Scrum meetings.

7. Scrum Mastery

7.1. … analyze your fulfillment of the five Scrum values.

7.2. … illustrate at least two concrete examples of how they actively applied a Scrum value(s) in their work.