

SCRUM ALLIANCE® CERTIFIED SCRUM DEVELOPER (CSD) Learning Objectives

August 2021



PURPOSE

This document describes the Learning Objectives (LOs) that must be covered in a Certified Scrum Developer (CSD) offering in addition to the Scrum Foundational LOs. These Learning Objectives take the following into consideration:

While the application of Scrum is in no way limited to the area of software development, the Scrum Alliance recognizes the special need for additional education in this field.

Scrum is inherently incremental in nature, therefore a software development implementation of Scrum requires an incremental approach to software engineering.

The Learning Objectives for this offering are based on:

- *Scrum Guide*, [scrumguides.org](https://www.scrumguides.org)*
- Manifesto for Agile Software Development, four values and 12 principles, [agilemanifesto.org](https://www.agilemanifesto.org)
- Scrum values, <https://www.scrumalliance.org/about-scrum/values>
- Scrum Alliance [Scrum Foundations Learning Objectives](#)
- Extreme Programming Explained - Embrace Change by Kent Beck
- <https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>

SCOPE

Scrum Alliance adopted the Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game, co-authored and updated (most recently in 2020) by the co-creators of the Scrum framework as the guiding curriculum for this offering. CSD candidates are expected to build a body of knowledge of the Scrum framework, including its accountabilities, 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.

Students attending the CSD course should expect to learn essential concepts and skills to be able to work successfully in a Scrum team that focuses on creating software. They also should expect to be enabled to bring essential collaboration skills into their team.

Thirdly, this course should create an understanding for the need of agile engineering practices and encourage participants to adopt them in their contexts.

This entails covering LOs within the following categories:

- Lean, Agile and Scrum
- Collaboration
- Architecture & Design
- Refactoring
- Test Driven Development
- Integrating Continuously

Students attending a CSD offering should expect that each LO identified in this document will be covered. Students should also expect that the Scrum Foundations LOs are covered either before or during the offering.

Individual educators may choose to include ancillary topics. Ancillary topics presented in a CSD offering must be clearly indicated as such.

A note about Bloom's Taxonomy:

Bloom's-style LOs describe what the learner can do upon completing the offering. Each learning objective should be able to be preceded by, "Upon successful validation of the CSD LOs, the learner will be able to ... "

Bloom's style of Learning Objectives consist of six levels of learning:

-  **Knowledge**
-  **Comprehension**
-  **Application**
-  **Analysis**
-  **Synthesis**
-  **Evaluation**

The levels progress from lower-order to higher-order thinking skills,

The level of each learning objective can be identified using the image designations above.

The examples in italics, listed below each LO, are for illustrative purposes. They are supposed to inspire the student or educator, not limit the learning.

LEARNING OBJECTIVES

1 - Lean, Agile & Scrum

-  1.1 practice utilizing a sprint backlog.
-  1.2 describe the responsibility of the Scrum team for turning PBIs in the sprint backlog into increments of value.
-  1.3 organize and demonstrate the daily Scrum.
-  1.4 list at least three attributes of PBIs.
-  1.5 give at least three examples of how a Scrum team will inspect and adapt PBIs during product backlog refinement.
-  1.6 discuss at least five elements of a definition of done that ensure the increment provides information that enhances transparency and focus against which progress can be measured.

2 - Collaboration & Team Dynamics

-  2.1 explain at least three differences between a working group and a team.
-  2.2 discuss at least three attributes of effective teams.
-  2.3 demonstrate "working together as one team."
-  2.4 describe at least three benefits of developers interacting directly with customers and users.
-  2.5 restate at least three ways a Scrum team may collaborate with stakeholders, customers, and/or users during the sprint.

3 - Architecture & Design

-  3.1 outline at least three benefits of technical excellence.
-  3.2 explain at least one design practice on an agile team.
-  3.3 list at least three principles of architecture in an agile environment.

4 - Refactoring

-  4.1 define refactoring.
-  4.2 describe at least three benefits of refactoring to an agile software development effort.

5 - Test Driven Development (TDD)

-  5.1 explain test-first as a design and development approach and list three benefits.
-  5.2 list at least three differences between "traditional" and "agile" testing.
-  5.3 explain the importance of refactoring in the TDD cycle.
-  5.4 describe at least three qualities of a good agile testing approach.

6 - Integrating Continuously

-  6.1 define continuous integration and list at least three benefits.
-  6.2 describe using three examples of how Scrum supports CI.
-  6.3 describe at least one advantage of an automated build, test, measure pipeline.

PROGRAM TEAM

Certified Scrum Developer Team (2021)

- Axel Wilhelm Berle
- Pascal Gugenberger
- Björn Jensen
- Falk Kühnel
- Paul Moore
- Rob Myers
- Andreas Schliep
- Rickard Jones

*The *Scrum Guide* is © 2020 Ken Schwaber and Jeff Sutherland. Content from The *Scrum Guide* is used by permission of Ken Schwaber and Jeff Sutherland under the terms of the Creative Commons – Attribution – Share-Alike License v. 4.0. which is accessible at <https://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <https://creativecommons.org/licenses/by-sa/4.0/>.