This document describes the Learning Objectives (LOs) that must be covered in an Advanced Certified Scrum Developer℠ (A-CSD℠) offering.

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 offering are based on:

• Scrum Guide, scrumguides.org*
• Manifesto for Agile Software Development, four values and 12 principles, agilemanifesto.org
• Scrum values, https://www.scrumalliance.org/about-scrum/values
• Scrum Alliance Scrum Foundations Learning Objectives
• Scrum Alliance Guide Level Feedback

Specific guiding resources are mentioned in the examples below.
Students attending an A-CSD offering should expect that each Learning Objective identified in this document will be covered. A link to Scrum and how its benefit should always be transparent.

The A-CSD Learning Objectives fall into the following categories:

1. Lean, Agile and Scrum
2. Collaboration & Team Dynamics
3. Architecture & Design
4. Refactoring
5. Test Driven Development
6. Integrating Continuously
7. Learning by Delivering Continuously

Individual Educators may choose to include ancillary topics. Ancillary topics presented in an A-CSD offering must be clearly indicated as such.

A note about Bloom’s Taxonomy:

Bloom’s-style Learning Objectives describe what the learner can do upon completing the offering. Please mentally start each Learning Objective with the following phrase: “Upon successful validation of the A-CSD Learning Objectives, 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, Knowledge (📍) through Evaluation (✔). The level of each learning objective can be identified using the image designations above.
LEARNING OBJECTIVES

1 - Lean, Agile & Scrum

1.1 apply a modelling technique to visualize the flow of work.
1.2 describe at least three concepts that help identify improvements to a work system.
1.3 discuss at least three different types of wastes in product development environments and how they could be addressed in a Scrum Team's Definition of Done.
1.4 practice formulating and iteratively evolving a Definition of Done (DoD) and identify at least three reasons why and how the DoD should evolve.
1.5 discuss at least three methods Developers could use to address challenges arising when working with multiple teams on one product.
1.6 evaluate at least one improvement you or your team introduced into your way of working as a result of a Retrospective.
1.7 discuss at least one business perspective on development work.

2 - Collaboration & Team Dynamics

2.1 compare and contrast at least three different approaches of working together.
2.2 apply at least one technique to improve listening and understanding others.
2.3 practice giving and receiving feedback.
2.4 apply a collaborative development practice.
2.5 describe the differences between utilization, efficiency, and effectiveness.
2.6 practice at least one way to size Product Backlog Items so they fit into a Sprint.

3 - Architecture & Design

3.1 explain at least three differences between up-front and emergent architecture.
3.2 explain at least three design principles that inform agile architecture considerations.
3.3 explain at least three approaches how to design for and verify system constraints, and practice one of them.
3.4 compare and contrast at least three code and product quality metrics.

4 - Refactoring

4.1 demonstrate at least one approach to refactor a system for maintainability.
4.2 explain at least three possible code and product smells and demonstrate how to approach one of them during refactoring.
4.3 explain refactoring to a non-technical stakeholder.
4.4 explain technical debt, outline at least three causes that lead to technical debt, and discuss how to address one of the causes.
LEARNING OBJECTIVES

5 - Test Driven Development (TDD)

5.1 restate at least three guiding principles of TDD and explain why they are necessary.
5.2 demonstrate designing a software or product entity using TDD as a design approach.
5.3 apply at least five unit-testing principles and practices.
5.4 identify at least five measures to improve the quality and effectiveness of tests and apply at least three test refactoring approaches.
5.5 outline at least one concept to categorize testing and assign different methods for testing to the different categories.
5.6 list at least three attributes of a test first business facing collaborative approach.
5.7 apply at least one approach to implement a test driven feedback loop with stakeholders and users.
5.8 apply at least one technique to deal with missing or resource inefficient components or subsystems.
5.9 discuss at least three different ways to approach technical excellence by validating and improving the inner quality of a system; and practice at least one of them.

6 - Integrating Continuously

6.1 discuss at least five areas of concern that need to be dealt with when integrating continuously.
6.2 practice creating a build that is automated, self-testing, and fast.
6.3 apply at least one Continuous Integration (CI) approach with a team.

7 - Learning by Delivering Continuously

7.1 define Continuous Delivery (CD) and discuss at least three benefits.
7.2 describe at least three technical practices for Continuous Delivery.
7.3 discuss at least one approach to incorporate feedback about the expected outcome of a delivery.
7.4 outline a continuous deployment approach.
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/.*