Course Outline

1: Agile Overview

We begin by discussing why an organization would want to switch from traditional development techniques to Agile ones.

1. Making the Case for Change – Organizational change (which an Agile transformation is) is difficult to achieve unless there is a clear understanding among everyone involved about why it is necessary.

Exercise: Make a list of software project problems that you would like to correct.

2: The Agile Paradigm Shift

Agile techniques are based on a completely different mental model and set of paradigms about projects. In this section, we begin to explore the mindset that that the Agile methods are built upon.

1. A Paradigm for Complexity – The inherent complexity of developing software is the source of most software project failures, so the Agile methods embrace paradigms that are designed for complexity
2. Predictive vs. Adaptive – A key difference between traditional and Agile paradigms concerns our ability to predict how the project will unfold. The Agile approach is to expect that our predictions may be less-than-accurate, and to structure the project so we can adapt to these surprises as they unfold

3: The Agile Foundation

We continue exploring the Agile mindset by examining the foundations upon which the Agile methods were built.

1. Agile Manifesto – The value system upon which Agile is built
2. Agile Principles – The necessary elements for making Agile work
3. Agile Benefits – Actual benefits as reported in the State of Agile Report

4: Agile Methodologies

With the reason for Agile in mind, and an understanding of the Agile mindset, we are ready to explore the variety of Agile practices that your teams can embrace. Since there is not a single “correct” way to be Agile, we explore the practices associated with four of the most commonly-cited Agile methods.

1. Lean – All of the Agile methods are based on the principles first formalized in Lean Manufacturing, so we start with Lean Software Development
2. Scrum – The most widely-used of the Agile methods, Scrum is a good method to use to understand the basic iterative practices employed by most Agile teams
3. Kanban – Originally created by the Lean Movement and more recently embraced by the Agile Community, Kanban provides an alternative process structure that (unlike the other Agile methods), is not based on iterations.
4. XP – Extreme Programming (XP) is one of the few Agile methods that goes into detail about technical programming practices, so we will take some time to explore them.
5. Custom Hybrid – We complete this section by observing that many teams create their own custom Agile method by drawing practices from several Agile methods and combining them in unique ways.

Exercise: See for yourself how common practices that Lean counsels against can make a team less efficient.

5: Building the Agile Team

The Agile practices only work as promised when used in the context of an Agile self-organizing team. In this section we delve into what self-organization means, and the attributes of an Agile team, including roles, responsibilities, and the team dynamics that make Agile practices successful.

1. Systems Thinking – Seeing the Agile team as more than a collection of people
2. What Makes a Great Team – The secret of an effective team goes way beyond the process they are using

3. Agile Team Roles – Agile teams have some unique and important roles
4. Team Best Practices – Guidance from Agile coaches on how to lead an Agile team to be the best they can be

Exercise: Make a list of the attributes of great teams you have been a part of.

6: Inspect and Adapt

Continuous Improvement (the heart of Lean principles) is the heart of Agility as well. We will focus on the practices then enable Agile teams to ensure not only that they are building the right product, but also that they are constantly improving their capacity to do it. They do these things at the end of every iteration (every few weeks)!

1. Iteration Review – A status check helps the team to stay on track and know if they need to take corrective action
2. Demo – A show-and-tell with the customer ensures that what they just built is indeed what the customer expected and needs
3. Retrospective – A mini-lessons-learned gives them the opportunity to improve how they work—beginning the very next day

7: Agile Adoption

We wrap up with a quick look at how to move forward with an Agile transformation.

1. Leading Change – A look and organizational change management

Exercise: What will you do in your organization with what you just learned?