TECHNOLOGY TRANSFER PRESENTS

CRAIG LARMAN

Certified ScrumMaster Course PLUS

OCTOBER 24-26, 2016

Agile TDD and Refactoring

OCTOBER 27-28, 2016

RESIDENZA DI RIPETTA - VIA DI RIPETTA, 231
ROME (ITALY)

info@technologytransfer.it
www.technologytransfer.it

About this seminar

The standard, basic CSM is two days; this CSM PLUS is a **three**-day in-depth seminar on the art of serving as an effective ScrumMaster or Agile Leader, on the Scrum framework, and the organizational changes involved in adopting Scrum. In addition to all standard CSM content, you will learn - through exercises and discussion - more key concepts and skills in Scrum and Agile methods, including Large-Scale Scrum, Agile organizational design, enterprise-level adoption and transition, Scrum in the context of offshore and/or multisite development, specification by example and its relationship to Acceptance TDD, how to do Agile estimation, splitting large requirements into smaller customer-centric requirements, Product Backlog prioritization, how to facilitate a Scrum Retrospective, how to coach a Sprint Review, and more.
Surveys show that Scrum is by far the most popular Agile method worldwide - and with good reason.
There are three roles in Scrum: Team(s), Product Owner, and ScrumMaster. This latter role is NOT a Project Manager in any traditional sense. The ScrumMaster is a process coach - someone who is a master of the Scrum method and helps the Teams and product owner apply it and maximize value with it.
Agile Management is as radically different from traditional Project Management as Agile processes are different from traditional methodologies. Rather than plan, instruct and direct, the Agile Coach facilitates, coaches and leads. This person is called a ScrumMaster in the Scrum Agile process to denote the difference and remind the person filling this role of the new responsibilities. Accepted participants learn how to be a ScrumMaster and how to make a development team, a product group, or an organization Agile. Exercises, case studies, and examples used to bring home the realization of how to be a ScrumMaster instead of a Project Manager.

PREREQUISITES: PRE-READINGS

Participants must read (available online), before the course:

- The Scrum Guide
- Scrum Primer

OBJECTIVES

Through education and practice, a ScrumMaster should be able to:

- Remove the barriers between development and the customer so the customer directly drives development
- Teach the customer how to maximize ROI and meet their objectives through Scrum
- Improve the lives of the development team by facilitating creativity and empowerment
- Improve the productivity of the development team in any way possible
- Improve the engineering practices and tools so each increment of functionality is potentially shippable

In addition to the above, PLUS objectives include ability to coach or advise more in-depth Agile and Scrum practices, including retrospectives, Agile engineering practices, and estimation sessions.

CERTIFICATION

There is an online exam to take, at the ScrumAlliance, to complete the certification process. The registration fee for the exam is included.

Outline

1. Agile Overview

Agile product development frameworks aim to increase flexibility or agility in the development organization, and thereby enable business agility. They emphasize increasing transparency, inspection, and adaption, and flexible workers. By far, Scrum is the most popular Agile framework, with good reason. We examine some of the big ideas.

2. Scrum Background

The roots of Scrum start in the 1986 Harvard Business Review paper, *The New New Product Development Game*. We examine the background behind this research, and the key ideas in the paper, which were the impetus for Scrum.

3. Scrum Overview

In this section, we summarize the key elements of Scrum, including roles, events, artifacts, and values. And especially, the need for transparency, inspection, and adaptation.

4. Scrum Roles and Responsibilities

Team, Product Owner, and ScrumMaster are the three key roles of Scrum. We explore the behaviors and responsibilities of each role, and how that may differ from your current organization.

5. Scrum Artifacts

The Product Backlog of items, and Sprint Backlog of tasks, and the Product Increment are the three key artifacts of Scrum. We examine their contents, and how to create, modify, and use the backlogs.

6. Sprint

The Scrum Sprint is the short iteration within which the Team creates a Potentially Shippable Product Increment (PSPI). We explore the implications of working in Sprints that deliver PSPI.

7. Sprint Planning Meeting

This meeting kicks off the Sprint, in which the Product Owner offers a list of Product Backlog Items to the Team, and the Team creates a Sprint Backlog and forecast. We look at the outcomes and techniques for this meeting.

8. Daily Scrum

This is a daily short meeting in which the self-managing Team members update each other on their progress, to foster coordination and collaboration. We examine the rules and techniques.

9. Product Backlog Refinement

During the Sprint, the Team and Product Owner need to refine (or groom) the Product Backlog for future Sprints (much urgently, for the upcoming next Sprint). This should normally take between 5-10% of the overall Sprint duration. In this section we summarize the motivation, details, and various methods for estimation, requirements splitting, and detailed analysis.

10. Sprint Review Meeting

The goal of the Sprint is to product a PSPI, and in this meeting involving the Team, Product Owner, and other stakeholders, the group closely inspect the PSPI so that the Product Owner can decide to accept or reject the items and decide to actually release the product. In addition, the Product Owner seeks input for how to proceed in subsequent Sprints.

11. Sprint Retrospective Meeting

The Sprint ends with a workshop in which the Team, ScrumMaster (and perhaps Product Owner) reflect on their ways of working, relationships, and environment, and generate an improvement experiment to try in the next Sprint.

12. Release Planning

Before the first Sprint, the group needs to create and prepare a Product Backlog so that the Team knows what to work on. In addition, there may be a larger release goal that spans several Sprints. We examine the goals, activities, and techniques of release planning before Sprint 1.

13. The Art of the ScrumMaster

The ScrumMaster is a teacher and coach of Scrum, who focuses on getting the Team, Product Owner, and organization to work well and to get Scrum to work well in the organization. We explore the myriad bodies of knowledge, issues, and techniques that ScrumMasters need to be aware of.

14. Tools for Fostering Self-Organizing Teams

Teams in Scrum are Self-Organizing or Self-Managing, without a team lead or project manager. In the transition to Scrum, people at first may not be familiar how to effectively Self-Organize in a highly collaborative Team. We look at techniques and issues to help create a healthy and skillful Self-Managing team.

15. Introducing Scrum into the Organization

We examine examples and techniques that ScrumMasters and Product Owners use to more effectively introduce Scrum and create positive change in the organization.

PLUS Topics Include...

16. Communities of Practice

In a cross-functional team organization, the concern of a single function (such as architecture, user experience, or testing) remains important. How to handle this cross-team concern? With communities of practice. In this section we examine this critical organizational element.

17. Agile Estimation

With relative point estimation, estimation poker, monte carlo simulation, and more.

18. Retrospective Techniques

With system dynamics analysis, root cause analysis, collaboration exercises, and more.

19. Large-Scale Scrum

How do you apply Scrum to a product group of 1000 people, perhaps in multiple sites? We provide a few pointers to this large subject for large, multisite, and offshore development, and introduce Large-Scale Scrum framework-1 and framework-2 for scaling to massive and multisite product development.

20. Visual Management

In Lean Thinking and Agile methods, using big visible charts on walls, paper cards on walls, and other physical information radiators are important tools to foster collaboration and agility. We explore the motivation and modalities of Visual Management.

21. Lean Thinking

Lean Thinking, or the Toyota Way, is not the same as Scrum, but it has been an influence on many Agile approaches, including Scrum. What are the key elements of Lean Thinking? We explore some of these, including value flow, manager-teachers, managers practicing Go See and Help, continuous improvement, respect for people, and more.

22. Splitting large requirements

How do you take a large requirement and split it into much smaller customer-centric requirements? We will take real requirements from student examples, and show how to split them, using the many splitting perspectives, such as splitting by use cases, by scenarios, by specializations, and more.

23. The 3 Cs of User Stories

User stories are a behavior of Agile and Collaborative requirements analysis, based on card, conversation, and confirmation. We look at the details of these, and how to do stories.

24. Agile Requirements Analysis and Acceptance Testing

We explore skillful and simple techniques such as specification by example, and Acceptance Test-Driven Development.

25. Design workshops

How can team members, and cross-team members, collaborate on a shared design and architecture? We examine the practice of facilitated design workshops and Agile Modeling.

26. Agile Architecture

How to grow a system with agility? We look at some of the behaviors and patterns of Agile architecture.

27. Many more exercises

Related to scenarios and case studies of Scrum in action, benefits, problems, estimation, and more.

AUDIENCE

Anyone interested in adopting Scrum should take CSM PLUS as a foundation – including potential Product Owners, and the Leadership Team. And certainly, anyone wishing to serve as a ScrumMaster.

About this seminar

This information-packed and hands-on course shows developers and technical leaders how to apply Test-Driven Development (TDD) and Refactoring, apply the most popular Open-Source frameworks for TDD and use them within a popular IDE. TDD is powerful and practical. It's the practice of always writing test code before the code to be tested. In addition to the obvious benefit that tests actually get written and executed for most code, a more subtle but important effect is that when we start by thinking very concretely—with code—in the role of a calling client to the new application code before it is written, it clarifies our design and becomes a more fun and creative way to combine writing tests with application code. Hence, TDD is more than just testing—it is a kind of creative micro-design step.

In this course you will learn how to think in and apply Test-Driven design and programming, and establish it as a consistent method for your development team. You'll learn and work with the popular TDD framework JUnit (if Java, or one of the other popular xUnit frameworks if working in another language).

TDD quickly leads developers to see the need for and value of reducing coupling in their code, and for techniques to break dependencies so that tests can be run quickly in isolation. Thus, a critical adjunct skill in TDD is learning how to create and inject alternate "test doubles" (fakes, stubs...). In this course you will learn how to create stubs, fakes, mocks, object factories/mothers, how to break dependencies, and how to apply dependency injection methods.

Learning how to break dependencies for testing in isolation is especially important in the context of legacy code; in this course you will to work with legacy code to "bring it under test" and apply TDD.

Refactoring is a disciplined design skill to improve the structure of code without changing its external behavior. And Refactoring is part of the TDD cycle. Thus, in this course you will learn the various "code smells" and the refactoringsto clean them up. Refactoring depends on automated refactoring tools built into popular IDEs or editors, such as Netbeans, Eclipse, SlickEdit, or emacs; thus in this course you will learn to apply an automated refactoring tool. Finally, the course includes a brief introduction to the companion agile methods practice called Acceptance TDD—executable requirements with automated verification. You will learn about the popular FIT framework and the Acceptance TDD methodology.

OBJECTIVES

Upon completion of this course, students should be able to:

- Apply TDD
- Break dependencies and create "test doubles" (fakes, mocks, stubs, ...)
- Inject dependencies with flexible techniques
- Separate test set up code into object factories or "object mothers"
- Identify code smells
- Apply Refactorings
- Use an xUnit framework such as Junit
- Define acceptance TDD and know tools to apply it

Outline

- Test-Driven Development
- Method and motivation
- Writing tests first
- The TDD lifecycle
- Testing in an iterative and agile method
- Categories of TDD: unit, acceptance
- TDD tools and frameworks: xUnit, ...
- Testing in different architectural layers
- Unit TDD
- Practice with XUnit
- Code smells
- Refactorings, including Extract Method, Introduce Explaining Variable, and more
- Test Doubles: Fakes, Stub and Mock Objects
- Integration vs. unit testing
- Setting up the test environment
- Mock generation tools
- Object factories
- Object Mother pattern
- Dependency injection
- Continuous Integration and TDD
- Acceptance TDD
- TDD and Legacy Code
- Characterization tests
- The Legacy TDD life cycle

AUDIENCE

- Developers
- Architects
- Test engineers
- Technical leaders

Please, bring your laptop to the seminar.

INFORMATION

PARTICIPATION FEE

Certified ScrumMaster Course PLUS
€ 1700

Agile TDD and Refactoring
€ 1300

The fee includes all seminar documentation, luncheon and coffee breaks.

VENUE

Roma, Residenza di Ripetta
Via di Ripetta, 231
Rome (Italy)

SEMINAR TIMETABLE

9.30 am - 1.00 pm
2.00 pm - 5.00 pm

HOW TO REGISTER

You must send the registration form with the receipt of the payment to:
TECHNOLOGY TRANSFER S.r.l.
Piazza Cavour, 3 - 00193 Rome (Italy)
Fax +39-06-6871102

within
October 10, 2015

PAYMENT

Wire transfer to:
Technology Transfer S.r.l.
Banca: Cariparma
Agenzia 1 di Roma
IBAN Code:
IT 03 W 06230 03202 000057031348
BIC/SWIFT: CRPPIT2P546

GENERAL CONDITIONS

DISCOUNT

The participants who will register 30 days before the seminar are entitled to a 5% discount.

If a company registers 5 participants to the same seminar, it will pay only for 4.

Those who benefit of this discount are not entitled to other discounts for the same seminar.

CANCELLATION POLICY

A full refund is given for any cancellation received more than 15 days before the seminar starts. Cancellations less than 15 days prior the event are liable for 50% of the fee. Cancellations less than one week prior to the event date will be liable for the full fee.

CANCELLATION LIABILITY

In the case of cancellation of an event for any reason, Technology Transfer's liability is limited to the return of the registration fee only.

CRAIG LARMAN

☐ **CERTIFIED SCRUMMASTER COURSE PLUS**

Rome October 24-26, 2015
Residenza di Ripetta - Via di Ripetta, 231
Registration fee: € 1700

☐ **AGILE TDD AND REFACTORING**

Rome October 27-28, 2015
Residenza di Ripetta - Via di Ripetta, 231
Registration fee: € 1300

If anyone registered is unable to attend, or in case of cancellation of the seminar, the general conditions mentioned before are applicable.

first name .....

surname .....

job title .....

organisation .....

address .....

postcode .....

city .....

country .....

telephone .....

fax .....

e-mail .....

Stamp and signature

Send your registration form
with the receipt of the payment to:
Technology Transfer S.r.l.
Piazza Cavour, 3 - 00193 Rome (Italy)
Tel. +39-06-6832227 - Fax +39-06-6871102
info@technologytransfer.it
www.technologytransfer.it

SPEAKER

Craig Larman, he serves as a management consultant, with a focus on organizational redesign and systems thinking, for flexible, high-value-throughput product organizations. His emphasis is scaling Agile principles and practices and Lean thinking to very large, multisite, and Agile offshore development (often, embedded systems, telecommunications, or investment banking), and coaching executive teams to succeed with larger Enterprise-level Agile and Lean methods adoption; these topics are the subject of his latest two books: "**Scaling Lean & Agile Development: Thinking & Organizational Tools**" and "**Practices for Scaling Lean & Agile Development: Successful Large, Multisite & Offshore Product Development with Large-Scale Scrum**". Mr. Larman has served as the Lead Coach of Lean Software development adoption at Xerox, and serves or has served as a consultant for large-scale Scrum and Enterprise Agile adoption at Ericsson, Bank of America Merrill Lynch, Alcatel-Lucent, UBS, Nokia Networks and Siemens Networks (now, NSN), Thomson Reuters, Statoil, Cisco-Tandberg, and at Schlumberger, among many other clients. His work includes the best-seller globally on Agile methods*:* "**Agile and Iterative Development: A Manager's Guide**"**.** Mr. Larman is the author of "**Applying UML and Patterns–An Introduction to OOA/D and Iterative Development**", the world's best-selling text on OOA/D, Iterative Development, Modeling and the UML, translated to many languages and used worldwide in industry and colleges as the standard for introducing software design, modeling, design patterns, architecture, and OOA/D.

He is one of the earliest Practicing ScrumMastersand one of the first worldwide authorized to coach and certify new ScrumMasters and Product Owners, as a Certified Scrum Trainer. He has helped lead Scrum adoption for organizations of over 25,000 developers.