Course Syllabus
Agile for Practitioners

I. DESCRIPTION

Agile skills are in high demand in all industries, and those project professionals who pick up agile skills will stay relevant and increase their market value. This course is an in-depth, workshop-style course that will teach participants the principles and values of Agile, the most popular agile frameworks like Scrum, Kanban, Lean Software Development and Extreme Programming (XP), and help them apply Agile practices and techniques to their current projects. This is a highly interactive course that leverages real world examples from Scrum, Lean, Kanban and other Agile methods. Participants will work in teams to experience and apply Agile to the course exercises.

This course qualifies as education contact hours for the Project Management Institute’s Agile Certified Practitioner (PMI-ACP®) exam. Note that in order to sit for the PMI-ACP® exam, students must meet additional requirements set forth by PMI that can be found on their website.

II. COURSE OBJECTIVES

The purpose of this course is to prepare participants to be able to:

- Explain various Agile frameworks and why they are used—including Scrum, XP, Lean, and Kanban
- Apply Scrum to your projects
- Create a product backlog and user stories
- Explain what leadership looks like in an Agile environment
- Gain insight on how to create transparency using Agile communications tools
- Comprehend various Agile concepts including Value-Driven Delivery, Stakeholder Engagement, Team Performance, Adaptive Planning and Continuous Improvement
- Become a better Agile project practitioner
- Understand how to establish an Agile team and help encourage high-performance
- Learn to effectively manage project scope, schedule, budget and quality using Agile controls
- Learn how to apply servant leadership
- Gain insights, techniques and skills to successfully coach and mentor agile teams
- Understand the knowledge and experience requirements necessary to apply for and pass the PMI-ACP® certification exam

III. COURSE OUTLINE

Unit 0: Introduction
- Agile definitions and historical context
• The seven Knowledge Domains of the PMI-ACP
• Agile Values and Principles found in the Agile Manifesto
• Misconceptions about Agile

Unit 1: Agile Principles and Mindset
• The Agile Manifesto
• Common Agile frameworks including Scrum, Lean, XP, Crystal and Kanban
• Specific Agile techniques
• Cumulative Flow Diagrams
• Variations in Agile Methods

Unit 2A: Value Driven Development
• How Agile focuses on business value
• Backlog Prioritization
• XP Technical Practices
• Tasks Boards
• Minimum Viable Product
• The Agile Value Proposition
• Workshop – Airplane Production

Unit 2B: Value Driven Development
• Release Planning
• User Stories
• Scrum Teams
• Prioritizing the Product Backlog
• Estimating the Product Backlog
• Relative Estimating and Story Points
• Creating the Release Plan

Unit 3: Stakeholder Engagement
• Typical Agile Project Lifecycle
• Stakeholder Engagement
• Typical Chartering Activities & Artifacts
• Personas
• Innovation Games
• Product Roadmaps and MVP
• Wireframes
• Story Mapping
• Team Formation Activities
• Sample Team Agreements
• Develop Initial Time and Cost Estimates
• Product Quality Practices

Unit 4: Team Performance
• Project Performance Factors
• Team Roles
• Vision for a High Performing Team
• Osmotic Communication
• Co-Located Teams vs. Distributed Teams
• Team Maturity Models
• Skills needed by Agile Leaders
• Emotional Intelligence
• Listening Skills
• Negotiation Skills
• Command and Control vs. Servant Leadership
• Adaptive Leadership
• Collaboration, Facilitation, Problem Solving and Participatory Decision-Making Skills
• Motivating Teams
• Building Empowered Teams
• Conflict Resolution

Unit 5: Adaptive Planning
• Timeboxes
• Rolling Wave Planning
• Creating the Initial Release Plan
• Update Plans Frequently
• Learning and Re-planning
• Information Radiators – Task Boards, Burndown Charts
• Re-planning Example

Unit 6: Problem Detection and Resolution
• Team Performance
• Variance and Trend Analysis
• Tracking via Team Task Boards
• Leveraging Team Velocity Charts
• Cycle Time vs. Lead Time
• Change Cost Curve
• Tracking Escaped Defects
• Technical Debt
• Risk Burndown Graph
• Engaging Teams in Problem Solving
• Protecting Teams from Failure
Unit 7: Continuous Improvement
- Adopting an Continuous Improvement Mindset
- Great Retrospectives
- Process Tailoring & Hybrid Agile Approaches
- Agile Modelling
- Systems Thinking
- Value Stream Mapping
- Customer Satisfaction Surveys

Unit 8: Scrum Simulation
- Scrum Simulation Exercise
- Scrum Overview
- The Scrum Framework
- The Scrum Roles – Product Owner, Team and Scrum Master
- Creating Your Backlog
- The Scrum Meetings
- Complete Sprint Planning
- Develop Your Definition of Done
- Sprinting
- Sprint Review
- Sprint Retrospective
- Simple but difficult
- Common misunderstandings

Unit 9: Other PMI-ACP Topics
- PMI Code of Ethics
- Agile project accounting
- Earned Value Management
- Agile contracting

Unit 10: PMI-ACP Certification
- Scrum Alliance Certification
- Scrum.org
- ICAgile Certifications
- The PMI-ACP Exam
- PMI-ACP – Detailed Requirements
- Reference books for the PMI-ACP
- Practice PMI-ACP Exam
IV. EXERCISES

- Exercise 1: Course Goals
- Exercise 2: Agile Values
- Exercise 3: Agile Principles
- Exercise 4: XP Practices
- Exercise 5: Lean Wastes
- Exercise 6: Kanban Exercise
- Exercise 7: Agile Workshop – Airplane Production
- Exercise 8: Estimating with Planning Poker
- Exercise 9: Release Planning – Plan a trip
- Exercise 10: eCommerce Website – Part 1
- Exercise 11: eCommerce Website – Part 2
- Exercise 12: eCommerce Website – Part 3
- Exercise 13: Multi-tasking
- Exercise 14: Team Communications
- Exercise 15: Command and Control Litmus Test
- Exercise 16: Agile Team Leadership Scenarios Exercise
- Exercise 17: eCommerce Website – Part 4
- Exercise 18: eCommerce Website – Part 5
- Exercise 19: Scrum Simulation Exercise
- Exercise 20: Who Does the PM Role?
- Exercise 21: Value Stream Mapping
- Exercise 22: Sample PMI-ACP Exam

V. GRADING POLICY

Participation is the key to learning in this class. To facilitate your learning, there will be numerous team and class discussion topics, and 21 exercises. You will be graded on your participation in the team and class discussions, Q&A throughout the day, and exercises. It is important that you show up to every class in order to get the most out of your learning experience.

Grading for this class is based on in-class participation and engagement with your group. You will receive 1 point for every day that you come to class (Up to 4 total). You will receive an additional point for showing leadership in terms of participating in activities, discussion and Q&A. To pass the course, you need to come to every class and earn 1 leadership point (5 or more). If you need a letter grade for reimbursement, speak to the instructor at the start of the first class. To earn an “A” you will need 8 points. To earn a “B” you will need 6 points; a C will be 5 points and anything less is an F.
### VI. Course Map – Agile Certificate Program

<table>
<thead>
<tr>
<th>COURSE TOPICS</th>
<th>Agile for Practitioners (PROJ_PMI 403-0)</th>
<th>Leading and Coaching Agile Teams (PROJ_PMI 350-0)</th>
<th>Agile Estimating and Planning (PROJ_PMI 360-0)</th>
<th>Supporting an Agile Transformation (PROJ_PMI 365-0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Introduction, Values &amp; Principles</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Lean Principles</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Overview of Scrum, Lean, Kanban, &amp; XP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Benefits of Agile</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kanban</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value Driven Development</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning Agile Projects</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum Framework &amp; Simulation</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMI-ACP® Certification</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leading Beyond the Agile Team</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leading Agile Teams</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile vs Traditional Approaches</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misconceptions about Agile</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling Agile</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile Estimating Techniques</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building Your Agile Schedule</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coaching Agile Teams</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Agile Approaches</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile Metrics and Reporting</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Organizational Impediments</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>An Agile Leaders Toolkit</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile Adoption Patterns</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration (Sprint) Planning</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A Change Agent’s Toolkit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assessing Your Organizational Culture for Agile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating your Agile Transformation Roadmap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VII. BIBLIOGRAPHY

12 Source Books for the PMI-ACP:

- Cohn, Mike. Agile Estimating and Planning. Pearson Education. 2006
- Cohn, Mike. User Stories Applied: For Agile Software Development.
- Sliger, Michele and Broderick, Stacia. The Software Project Managers Bridge to Agility. Addison-Wesley. 2008.
- Wysocki, Robert K. Effective Project Management: Traditional, Agile, Extreme. Wiley. 2013

Other Helpful Books and Articles:

- Sutherland, Jeff and Schwaber, Ken. The Scrum Guide. 2017. (http://www.scrum.org/Scrum-Guides)
• Larman, Craig & Vodde, Bas. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley Professional. 2008

Web Posts Used in Class:

- Digithe: *What is Kanban*, [https://www.digite.com/kanban/what-is-kanban/](https://www.digite.com/kanban/what-is-kanban/)
- Slideshare: *Qs Approach To Project Cost*, [https://www.slideshare.net/jmerkler/qs-approach-to-project-cost](https://www.slideshare.net/jmerkler/qs-approach-to-project-cost)

Videos Used in Class:

- VersionOne: *The Agile Manifesto - 4 Agile Values Explained*, [https://www.youtube.com/watch?v=rf8Gj2RLKWQ](https://www.youtube.com/watch?v=rf8Gj2RLKWQ)
- Henrik Kniberg: *Kanban and Scrum – Making the most of Both*, [https://vimeo.com/16918747](https://vimeo.com/16918747)
- McDermott Chris: *The Other Side of Kanban*, [https://youtu.be/klpzmzhBIMk](https://youtu.be/klpzmzhBIMk)
- Construx: *How to Build a Kanban Board*, [https://youtu.be/N3BoLRVXol0](https://youtu.be/N3BoLRVXol0)
- HatsiekedeeDotTV: *A Funny Scrum Master Movie with Jeff Sutherland*, [https://youtu.be/oheekef7oJk](https://youtu.be/oheekef7oJk)
**VIII. GLOSSARY OF AGILE TERMS**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burndown Chart</td>
<td>A burndown chart is a visual tool for measuring and displaying team progress. The most common burndown chart represents remaining work in hours or tasks in a sprint. Burndown or Burnup charts may also be used to measure the progress of completion of user stories at an iteration or release level.</td>
</tr>
<tr>
<td>Product Owner</td>
<td>The Product Owner is a member of the Scrum team that represents the voice of the customer and is accountable for ensuring that the team delivers value to the business. The Capability or Product Owner writes customer-centric items (typically user stories), prioritizes them, and adds them to the Product Backlog.</td>
</tr>
<tr>
<td>Daily Scrum / Daily Meeting / Daily Standup</td>
<td>A Daily Standup or Scrum is a meeting of the scrum team (5-9 members, cross functional) that happens at the same time every day and should last 15 minutes or less. The meeting is designed to allow the team to coordinate their efforts, and plan their days based on the flow and challenges of the development process. Each team member should answer 3 questions: what did I do yesterday, what am I planning to do today, and what impediments do I currently have?</td>
</tr>
<tr>
<td>Definition of Done (DoD)</td>
<td>The Definition of Done is the team’s agreement on what constitutes done for a user story or backlog item. This agreement determines the tasks that the team needs to do to consider something done. This is to encourage being as close to production ready as possible within a sprint.</td>
</tr>
<tr>
<td>Definition of Ready (DoR)</td>
<td>The Definition of Ready is the team’s agreement on the characteristics of a user story that would make it ready to bring into a sprint to be worked on. User stories are made ready during backlog refinement sessions which take place in advance of the sprint the items will be completed in.</td>
</tr>
<tr>
<td>Empirical Process Control</td>
<td>Empirical process control is used with processes that are highly variable and unpredictable. It is based on inspecting the results of the process and making regular adjustments. It is often contrasted with predictive approaches which assume results can be predicted.</td>
</tr>
<tr>
<td>Epic</td>
<td>A term for a very large user story that is eventually broken down into smaller stories.</td>
</tr>
<tr>
<td>MVP (Minimum Viable Product)</td>
<td>A Minimum Viable Product represents a version of the product which has just those features that allow the product to be deployed, and no more.</td>
</tr>
<tr>
<td>Pair Programming</td>
<td>Pair programming is an Agile software development technique in which two programmers work together at one workstation. One types in code while the other reviews each line of code as it is typed in. The person typing is called the driver. The person reviewing the code is called the observer (or navigator). The two programmers switch roles frequently. Benefits include better design, fewer bugs, and fewer key person dependencies.</td>
</tr>
<tr>
<td>Pairing</td>
<td>Pairing is a variation of pair programming where two people work together. This could include cross-functional (tester and developer), or even within the same discipline (analyst + analyst). Benefits include knowledge transfer, defect reduction, and team-building.</td>
</tr>
<tr>
<td>Planning Poker</td>
<td>Planning poker is a consensus-based technique for estimating based on relative size of user stories. It is based on the wideband Delphi technique which uses crowd-sourcing to develop more consistent and accurate estimates of work. This technique is similar to the planning game in Extreme Programming.</td>
</tr>
<tr>
<td>Potentially Shippable Increment</td>
<td>The Potentially Shippable Increment (PSI) or Potentially Shippable Product Increment (PSPI) is a small vertical slice of functionality that results from each sprint or iteration of an Agile project.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>The Product Backlog is a rank ordered list of the user stories that serve as the product or project requirements. The Product Owner is maintained by the Product Owner.</td>
</tr>
<tr>
<td>Product Backlog Item</td>
<td>A product backlog item is an individual feature or function for the solution. User stories are one format for a product backlog item.</td>
</tr>
<tr>
<td>Product Owner</td>
<td>See Capability Owner.</td>
</tr>
<tr>
<td>Refactoring</td>
<td>Refactoring is an XP technical practice. It is the process of improving software design, without changing the functionality.</td>
</tr>
<tr>
<td>Scrum</td>
<td>Scrum is a development framework developed by Ken Schwaber and Jeff Sutherland which is used to address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It is based on the adaptive and iterative methodology of software development. The name was taken from the game of rugby.</td>
</tr>
<tr>
<td>Scrum Master</td>
<td>The Scrum Master is a role on the scrum team with accountability for removing impediments to the team’s ability to deliver the sprint goal/deliverables. The Scrum Master is a servant leader. The Scrum Master ensures that the Scrum process is followed by the team, and they protect the team and keep them focused on the tasks at hand.</td>
</tr>
<tr>
<td>Specification by Example (SBE)</td>
<td>Specification by example is a method of producing living requirements which can be translated into automated acceptance tests (ATDD).</td>
</tr>
<tr>
<td>Spike</td>
<td>A spike is a specific type of user story that represents a short, time-boxed piece of research.</td>
</tr>
<tr>
<td>Sprint</td>
<td>A Sprint (or iteration) is a fixed timebox that serves as a container for all the Scrum events. During the Sprint, the Scrum team plans their work, produces working software, reviews their output, and then holds a retrospective. The term Sprint comes from the Scrum framework and is analogous to the term Iteration. Most Scrum teams use sprints that are 2-4 weeks long.</td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>The Sprint Backlog is an output of Sprint Planning, where the team forecasts the backlog items and tasks that they will complete during the sprint.</td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>Sprint or iteration planning is a key scrum meeting that occurs at the start of each iteration. The meeting is in two parts; during the first part of this meeting, the Product Owner describes the highest priority features to the team as described on the Product Backlog. In the second part of the meeting, the team then agrees on the number of features they can accomplish in the sprint and plans out the tasks required to achieve delivery of those features.</td>
</tr>
<tr>
<td>Sprint Retrospective</td>
<td>The Retrospective is the Scrum event that happens at the end of every Sprint to review lessons learned and to discuss how the team can be more efficient in the future. It is based on the scrum principles of inspect and adapt.</td>
</tr>
<tr>
<td>Sprint Review</td>
<td>The Sprint Review is a Scrum Meeting that is held at the end of each iteration, this serves as a brief review of the solution developed in the previous Sprint. During the review, the software increment is reviewed and accepted or rejected. Newly identified business needs from the Sprint Review are added to the Product Backlog.</td>
</tr>
<tr>
<td>Story Points</td>
<td>Story points are a relative measure used by Agile teams to represent the complexity and size of a user story. Points are used to provide a relative measure between two user stories, and cannot be compared between two teams.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>A product backlog item can be broken down into one or more tasks. Tasks are estimated in hours during iteration planning, and then re-estimated daily once a team member begins working on them.</td>
</tr>
<tr>
<td>Taskboard</td>
<td>A taskboard is a visual representation of the Sprint Backlog. It is generally a wall chart with cards and/or sticky notes that represents all the work (user stories and tasks) for a given iteration. The notes are moved across the board to show progress.</td>
</tr>
<tr>
<td>Team</td>
<td>The Dev team is a cross-functional and self-organizing group responsible for delivering the product. A team is typically made up of 5–9 people who do all the actual work needed to be done with a particular piece of functionality (analyze, design, develop, test, technical communication, document, etc.).</td>
</tr>
<tr>
<td>Technical Debt</td>
<td>Technical Debt is a consequence of poor or evolving software architecture and software development within a codebase. Technical debt can be thought of as work that needs to be done before a particular job can be considered complete. It is relevant because of the cost and risk of making changes to products with high technical debt.</td>
</tr>
<tr>
<td>User Story</td>
<td>A user story is a very high-level definition of a business need, containing just enough information so that the team can produce a reasonable estimate of the effort to implement it. A user story tells the who, what and why, in business language. A user story is not detailed or a substitute for conversation; in fact, it is a placeholder to have a conversation between the users and the team.</td>
</tr>
<tr>
<td>Velocity</td>
<td>Velocity is a relative number which describes how much work the team can get done in story points over a period of time.</td>
</tr>
<tr>
<td>WIP</td>
<td>Work in Progress, WIP represents any work that has been started but has yet to be completed. Agile teams strive to minimize work in progress so that they maximize throughput.</td>
</tr>
</tbody>
</table>