

Course Syllabus Agile for Practitioners

I. DESCRIPTION

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 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

After this course, participants will be able to:

1. Explain various Agile frameworks and why they are used—including Scrum, XP, Lean, and Kanban
2. Apply the Scrum framework to develop a new product
3. Create a product backlog and user stories
4. Create Agile communications tools that create transparency and stakeholder trust
5. Demonstrate how teams drive business value using product backlogs
6. Contrast Agile Planning with traditional ways of planning projects
7. Understand how to establish an Agile team and help encourage high-performance
8. Apply agile approaches to manage project scope, schedule, budget and quality
9. Understand the knowledge and experience requirements necessary to apply for and pass the PMI-ACP® certification exam

III. COURSE OUTLINE

1. Course Introduction, Introduction to Agile and a Comparison of Traditional and Agile Ways of Working

- Compare and contrast Agile with Traditional/Waterfall ways of working
- Describe the differences between Iterative and Incremental delivery
- Summarize the 4 Agile Values and 12 Agile Principles
- Describe the changes that would be needed by both team members and leaders to support agile ways of working

2. Benefits of Agile and Popular Agile Frameworks and Methods

- Explain the benefits of using agile approaches
- Explain various Agile frameworks and methods and benefits provided by them

- Demonstrate how teams drive business value using Agile approaches
- Support an argument for delivering a minimum viable product

3. Planning Agile Projects

- Develop charter artifacts for a new projects using agile techniques
- Create an initial product backlog for a new project
- Contrast Agile Planning with traditional ways of planning projects
- Use release planning techniques to plan an agile project

4. Exploring Scrum and High-Performing Teams

- Understand how to establish a Scrum team and help encourage high-performance.
- Apply techniques to continuously improve team performance and productivity
- Apply agile approaches to manage project scope, schedule, budget and quality
- Analyze team performance reports to determine corrective actions

5. Working Together as an Agile Team

- Apply the Scrum framework to develop a new product
- Evaluate whether an organization has implemented the Scrum roles per the Scrum Guide
- Create Agile communications tools that create transparency and stakeholder trust
- Evaluate the level of conflict and recommend strategies to resolve the conflict

6. Miscellaneous Agile Topics and the PMI-ACP Exam

- Compare various communications methods and identify those that will be most effective for an agile team.
- Recognize challenges and approaches to use with distributed teams.
- Engage team members in resolving problems
- Apply value stream mapping to analyze a process
- Revise a project forecast based on actual team progress
- Understand the knowledge and experience requirements necessary to apply for and pass the PMI-ACP® certification exam

IV. 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 the 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. To pass the course, you need to come to every day of class and participate. If you need a letter grade for reimbursement, speak to the instructor at the start of the first class.

V. Course Map – Agile Certificate Program

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

COURSE TOPICS	Agile for Practitioners (PROJ_PMI 403-0)	Leading and Coaching Agile Teams (PROJ_PMI 350-0)	Agile Estimating and Planning (PROJ_PMI 360-0)	Supporting an Agile Transformation (PROJ_PMI 365-0)
Creating your Agile Transformation Roadmap				✓

VI. BIBLIOGRAPHY

12 Source Books for the PMI-ACP:

- Adkins, Lyssa. Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition. Addison-Wesley. 2010.
- Cockburn, Alistair. Agile Software Development: The Cooperative Game, Second Edition
- Cohn, Mike. Agile Estimating and Planning. Pearson Education. 2006
- Cohn, Mike. User Stories Applied: For Agile Software Development.
- Derby, Esther. Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf. 2006.
- Highsmith, Jim. Agile Project Management: Creating Innovative Products. Addison-Wesley. 2010.
- Sliger, Michele and Broderick, Stacia. The Software Project Managers Bridge to Agility. Addison-Wesley. 2008.
- Shalloway, Alan, et al. Lean-Agile Software Development: Achieving Enterprise Agility. Addison-Wesley Professional. 2009
- Wysocki, Robert K. Effective Project Management: Traditional, Agile, Extreme. Wiley. 2013
- Rawsthorne Dan with Doug Shimp Exploring Scrum: The Fundamentals. CreateSpace Independent Publishing Platform. 2013.
- Hammarberg, Marcus and Joakim Sunden. Kanban In Action. Manning Publications. 2014
- Anderson, David J. Kanban: Successful Evolutionary Change for your Technology Business. Blue Hole Press. 2014

Other Helpful Books and Articles:

- Beck, Kent. Extreme Programming Explained. Addison-Wesley. 2004.
- Mersino, Anthony. Emotional Intelligence for Project Managers: The People Skills You Need to Succeed. AMACOM. 2013.
- Mersino, Anthony. Agile Project Management: A Nuts and Bolts Guide to Success. Vitality Chicago. 2015.
- Poppendieck, Mary and Poppendieck, Tom. Lean Software Development: An Agile Toolkit. Addison-Wesley. 2003.
- Project Management Institute. The PMI-ACP Examination Content Outline. December 2014. Retrieve from: http://www.pmi.org/en/Certification/~media/Files/PDF/Agile/PMI_Agile_Certification_Content_Outline.ashx
- Project Management Institute. The PMI-ACP Handbook (<http://www.pmi.org/en/Certification/New-PMI-Agile-Certification.aspx>)
- Sutherland, Jeff and Schwaber, Ken. The Scrum Guide. 2017. (<http://www.scrum.org/Scrum-Guides>)

- VersionOne. The 10th Annual Survey of Agile Development. May 7, 2013. Retrieve from: <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>
- Royce, Winston. Managing the Development of Large Software Systems. IEEE WESCON Proceedings. 1970
- Womack, James and Jones, Daniel. The Machine That Changed the World. Free Press. 2007.
- Salovey, Peter and Mayer, John. Emotional Intelligence. Psychology Today. 1990.
- Pink, Daniel. Drive: The Surprising Truth About What Motivates Us. Riverhead Books. 2009
- Tabaka, Jean. Collaboration Explained. Addison-Wesley. 2006.
- Herzberg, Frederick. One More Time: How Do You Motivate Employees. Harvard Business School Press. 2008.
- Schein, Edgar. Humble Inquiry: The Gentle Art of Asking Instead of Telling. Berrett-Koehler Publishers. 2013.
- 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:

- Santeo: **A History of Agile**, http://www.santeo.com/download/Santeo_HistoryofAgileSoftwareDevelopment.pdf
- PMI: **7 Domains of the PMI-ACP**, <https://www.pmi.org/-/media/pmi/documents/public/pdf/certifications/agile-certified-exam-outline.pdf>
- VersionOne: **11th Annual State of Agile Report, 2017**, <http://stateofagile.versionone.com>
- Gene Kim: **The Top 10 Things You Need to Know about DevOps**, <https://www.thinkhdi.com/~media/HDI/Corp/Files/White-Papers/whtppr-1112-DevOps-kim.pdf>
- Agile42: **6 Core Kanban Practices**, <https://www.agile42.com/en/training/kanban-pizza-game/>
- Digite: **What is Kanban**, <https://www.digite.com/kanban/what-is-kanban/>
- The Agile Executive: **Standish Group Chaos Reports Revisited**, <http://theagileexecutive.com/2010/01/11/standish-group-chaos-reports-revisited/>
- VersionOne: **Agile Value Proposition**, <https://www.versionone.com/agile-101/agile-software-development-benefits/>
- Science Daily: **Working Together In "War Rooms" Doubles Teams' Productivity, University Of Michigan Researchers Find** <https://www.sciencedaily.com/releases/2000/12/001206144705.htm>
- PMI: **Conflict Resolution**, <https://www.pmi.org/learning/library/coaching-agile-project-teams-navigate-conflict-6760>
- Slideshare: **Qs Approach To Project Cost**, <https://www.slideshare.net/jmerkler/qs-approach-to-project-cost>
- Scrum Primer: **The Scrum Primer**, <http://www.goodagile.com/scrumprimer/scrumprimer20.pdf>
- Scrum Inc: **The Scrum Papers**, <https://www.scruminc.com/scrumpapers.pdf>
- Scrum Inc: **Agile Contracts: Money for Nothing and Your Change for Free**, <https://www.scruminc.com/agile-contracts-money-for-nothing-and/>

Videos Used in Class:

- VersionOne: **The Agile Manifesto - 4 Agile Values Explained**, <https://www.youtube.com/watch?v=rf8Gi2RLKWQ>
- Axosoft: **Intro to Kanban in Under 3 minutes**, <https://youtu.be/R8dYLBjITUE>
- Henrik Kniberg: **Kanban and Scrum – Making the most of Both**, <https://vimeo.com/16918747>

- McDermott Chris: **The Other Side of Kanban**, <https://youtu.be/klpmzhBiMok>
- Construx: **How to Build a Kanban Board**, <https://youtu.be/N3BoLRVXoIQ>
- HatsiekiedeeDotTV: **A Funny Scrum Master Movie with Jeff Sutherland**, <https://youtu.be/oheekef7oJk>
- Adam Weisbart: **Stuff Bad Scrum Masters Say**, <https://youtu.be/GGbsgs611MM>

VII. GLOSSARY OF AGILE TERMS

Term	Definition
Burndown Chart	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.
Product Owner	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.
Daily Scrum / Daily Meeting / Daily Standup	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 all 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?
Definition of Done (DoD)	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.
Definition of Ready (DoR)	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.
Empirical Process Control	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.
Epic	A term for a very large user story that is eventually broken down into smaller stories.
MVP (Minimum Viable Product)	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.
Pair Programming	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.
Pairing	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.
Planning Poker	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.
Potentially Shippable Increment	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.

Term	Definition
Product Backlog	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.
Product Backlog Item	A product backlog item is an individual feature or function for the solution. User stories are one format for a product backlog item.
Product Owner	See Capability Owner.
Refactoring	Refactoring is an XP technical practice. It is the process of improving software design, without changing the functionality.
Scrum	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.
Scrum Master	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.
Specification by Example (SBE)	Specification by example is a method of producing living requirements which can be translated into automated acceptance tests (ATDD).
Spike	A spike is a specific type of user story that represents a short, time-boxed piece of research.
Sprint	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.
Sprint Backlog	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.
Sprint Planning	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.
Sprint Retrospective	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.
Sprint Review	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.
Story Points	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.

Term	Definition
Task	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.
Taskboard	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.
Team	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.).
Technical Debt	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.
User Story	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.
Velocity	Velocity is a relative number which describes how much work the team can get done in story points over a period of time.
WIP	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.