Collaborating and Communicating Agile Requirements

Code:	CC-AGILE-REQ
Length:	2 days
URL:	View Online

Gathering Requirements for an Agile Project

Project failures are often due to poor requirements gathering, analysis and planning. Traditional requirements documents may not contain complete and accurate requirements due to rapidly changing business environments. Agile requirements gathering, by moving detailed requirements closer to implementation, allows for rapid response to change. "Collaborating and Communicating Agile Requirements" will show you how to gather and manage these requirements.

Organizing and Managing Requirements

Traditional requirements are documented in a requirements specification. Changes to the requirements are managed through a change process. This course will demonstrate alternative ways of documenting requirements and managing changes. These alternatives can allow for a less "heavy" process in projects that can benefit from quick changes in direction.

Discover Real-World Techniques

This two-day Agile requirements training course will give you hands-on experience with techniques for gathering Agile requirements. Explanatory lectures with demonstrations, combined with practice exercises will provide you with the experience needed to create requirements that meet business needs.

Skills Gained

- Master writing user stories
- Appreciate how best known methods in traditional requirements processes can apply to Agile methods
- Communicate requirements using agile techniques to bridge the customers' and developers' needs.
- Identify stakeholders and user roles to ensure that all requirement viewpoints are elicited
- Create and maintain a product backlog
- Prioritize requirements so that the most important customer needs are delivered first
- Formulate an iterative project plan with feedback cycles that keep the project on track
- Estimate business value for requirements to track how a project contributes to the enterprise
- Develop requirements in an iterative approach to capture the details at the appropriate time

Who Can Benefit

- Business Customer or Partner
- Product Manager or Customer Representative
- Business or Systems Analyst

- Architect or Developer
- QA Tester or QA Engineer
- Project Manager or Team Leader
- IT Manager/Director

Course Details

Section I: Agile Overview

More than simply a methodology or approach to software development, Agile embraces a set of principles that drive more effective software development. Agile focuses on the customer, embraces the ever changing nature of business environments and encourages human interaction in delivering outstanding software. In this introduction, we'll discuss the following:

- Agile Manifesto
- Agile Principles
- Agile Methodologies
- Agile Benefits

Section II. Project Initiation

Among the key contributing factors leading to project failure is poor communication between the customer and developer groups. It is critical, therefore, that each successful project start out right. In this section we'll cover the following topics:

- Project Charter
- Project Roles
- Project Planning
- Communication
- Class Exercise: Working in small teams, you will establish a project charter including goals and objectives for a sample project. You will participate in defining key roles for project team members and set clear expectations for project communication.

Section III: Focus on the Customer

It is critical that the customer be the focus of a product throughout the development lifecycle. Every requirement should bring some value to the customer. Therefore, prior to defining requirements, it is important to define the customer. This will include the following topics:

- Customer Involvement
- Customer Roles
- Creating and Using Personas
- Constraints
- Class Exercise: Within your teams you will brainstorm some customer roles for your example project. From the brainstorming, you will consolidate the larger list of roles into key roles that will be the focus of your sample project For each of the key roles, each team will create personas and share them with the class.

Section IV: User Stories

User stories are a way to capture requirements from a customer point of view. Stories do not capture all of the detailed requirements, but require enough information to estimate and plan. A proven tool used in Agile teams to capture initial

requirements, in this section we will explore the following topics:

- User Stories
- Goals and Objectives
- Acceptance Criteria and Acceptance Tests
- Non-user Stories
- Class Exercise: Led by the instructor, the class will come up with some user stories for a sample project. We will discuss how to determine as a team what is appropriate for your user stories to be effective.

Section V: Product Backlog

The Product Backlog is the complete list of desired elements, or requirements, for the product. It is NOT a Requirements Specification, but a high level identification of what the software may satisfy. In this section we will discuss effective means of creating, prioritizing and maintaining the Product Backlog. We will peruse the following topics:

- Who owns the Product Backlog?
- Functional and Non-functional Requirements
- Story-Writing Workshop
- Prioritizing the Product Backlog
- Maintaining the Product Backlog
- Techniques for further elaboration
- Class Exercise: In small teams identified previously, you will engage in a story-writing workshop as a means of building a product backlog for your sample project. Subsequently, you will participate in prioritizing your product backlog and present the highest priority stories to the class.

Section VI: Estimating and Planning

Among the greatest challenges in developing software and delivering against stakeholder expectations is estimating accurately and subsequently planning how those expectations can be met. Agile cannot make that challenge disappear but offers some very helpful tools that enable teams to set and meet the appropriate expectations.

- Relative vs. Actual Estimating
- Using Story Points
- Planning Poker (Grenning 2002)
- Five Levels of Planning in Agile
- Estimating Team Velocity
- Class Exercise: Using the estimating techniques taught using story points, you'll enjoy a few rounds of Planning Poker with your team to establish estimates for your highest priority stories. This fun and highly effective method of relative estimating is certain to be a valuable tool for you to incorporate into your own estimating process.

Section VII: Release Plan

The release plan identifies a goal for the stories that will be included for a release of the software. Through the prior processes, the team will have prioritized the stories and estimated the team velocity. These key elements will come together to give the team a level of confidence that they can deliver the necessary requirements for a product release in what is normally a fixed timeframe. We'll examine the following topics:

- Iteration Estimates
- Prioritization Revisited

- Ownership and Participation
- Communication
- Class Exercise: Each team will establish a release plan for their sample project incorporating priority, estimates and velocity as appropriate. We'll discuss how real experiences of fixed time and requirement projects that work well with an Agile release plan.

Section VIII: Use Cases

At the appropriate time, prior to entering into the development of a story, requirements will need to be discussed in more detail. Use cases are a proven method for documenting the appropriate detail from a user interaction point of view. In this section, the instructor will introduce use cases and discuss some of the foundational elements that support the development process.

- Use Case Advantages
- Use Case elements
- Success Path
- Alternate Paths
- Exceptions
- Class Exercise: Teams will discuss and document use cases, including alternate paths and exceptions, for some of their high priority stories. As a class we'll discuss the relationship between use cases and stories, and how they complement each other.

Section IX: Iteration Plan and Execution

An iteration is a fixed amount of time in which stories/requirements will be developed, tested and ready for release. Because the requirements communication process takes you into each iteration throughout the product release, we'll explore the iteration planning and execution process. During this section we will discuss the following topics:

- Iteration Planning
- Defining "Done"
- Test-Driven, Test Often
- Demonstrate Working Software (Delivered Requirements)
- Inspect and Adapt applied to Requirements
- Finding your rhythm

Section X: Retrospective on Communicating Requirements

Using Agile Methods – Retrospectives are a key practice in Agile. We will take an opportunity to review our learning collectively and how we can improve. Each participant will identify one or two things that they will adapt in their working environment based on their learning. The instructor will also identify any elements of the course that should be adapted for a better learning experience, thus benefiting future course participants.
