Essential Software Requirements

- Take away powerful techniques for identifying, documenting, and verifying requirements
- Understand the best of both the formal Plan-Driven and Agile requirements approaches
- Use the Product Vision as a roadmap to success
- Discover how to elicit and document system requirements
- Learn new skills with practical, interactive exercises

Clear, concise, and accurate requirements will help avoid late, over budget, or cancelled projects. Too much documentation or inflexible requirements can bog down a project. Find the right mix of formal written requirements and agile documentation—user stories, use cases, prototypes, and visual models—that works best for you.

This practical, hands-on course will provide a flexible requirements development approach customized to your environment and the skills needed to successfully discover, analyze, communicate, and evaluate requirements.

Powerful Techniques for Identifying, Documenting, and Verifying Requirements

Many acknowledge that their processes need some improvement but feel helpless to do much about the problem. In this course you will learn how to fill the critical information gaps and freshen up those stale requirements processes in a highly practical way. Take away a new awareness of what "good" requirements are really about and the skills to help you complete your project on time and on budget.

The Best of Formal Plan-Driven and Agile Requirements Development Approaches

Learn how the plan-driven and agile development approaches differ in terms of timing, depth, and documentation of these valuable references. The plan-driven approach values product and process documentation. The agile approach values individuals and collaboration, working software, and the ability to swiftly accommodate change.

Practice New Skills with Interactive Exercises

This course offers interactive exercises to provide practical experience and improve your requirements development skills. Use a real-world case study to identify stakeholders, develop a vision statement, and produce concise, accurate, and usable requirements documentation. Find ways to transfer the newly learned techniques back to your organization’s requirements process and take away a framework for understanding business and user needs to develop a suitable software solution.

Who Should Attend

Whether you are a requirements or business analyst, software engineer, developer, test engineer, user, stakeholder, or a member of the QA staff responsible for gathering, analyzing, documenting, confirming, and maintaining requirements, this course is for you.

Course Outline

Overview of Essential Software Requirements

Types of requirements
The benefits of "good" requirements
When and how much to document requirements
The WebPhlyx Case Study
Exercise – Create requirements for the case study

Development Approaches and Requirements

Plan-Driven—values, core practices, and documentation
Agile—values, core practices, and documentation

The System Requirements

User, functional, and non-functional requirements and business rules
Mandatory vs. preferred requirements
Exercise – Specifying non-functional requirements
Business rules—facts, constraints, action enablers, computations, and terms
Information sources and the discovery process
Formal documentation and tools
Plan-driven documentation

The Product Vision

Product vision—the foundation of the project effort
Understanding business requirements
The role of the product champion
Identifying and involving stakeholders
User classes and user representatives
Exercise – Identifying project stakeholders
Developing the product vision document
Exercise – Create a Product Vision statement

Exercise – Create part of a System Requirements Specification
Agile documentation
Exercise – Create user stories and a use case
Visual models
Exercise – Create a decision table
Exercise – Create a state-transition diagram

Organizational Processes

Working together
Joint Application Development (JAD)
Reviews
Exercise – Creating and revising ambiguous requirements

Course Summary