PROCESS IMPACT

11491 SE 119TH DRIVE
HAPPY VALLEY, OR 97086-8278
PHONE: 503-698-7879
WWW.PROCESSIMPACT.COM

Learning Objectives for “In Search of Excellent Requirements” 1-Day Version

The student will learn how to:

- Recognize different types of requirements information, and classify and document them appropriately
- Document requirements at all three levels: business, user, and functional
- Recognize and select appropriate good practices for effective requirements elicitation, analysis, specification, validation, and management
- Understand why customer involvement is so critical to software development and see how to work with customer representatives through a “product champion” model
- Understand the 10 characteristics of high-quality requirements
- Apply the use case approach for defining user needs and system functions
- Use prototypes to clarify and refine user needs
- Elicit and define quality attribute requirements
- Model a user interface architecture using a dialog map
- Apply the key practices of requirements management
- Create a requirements traceability matrix to connect requirements to design elements, code, and tests

Additional Learning Objectives for 2-Day Version:

- Create an effective customer-developer partnership
- Understand the role and responsibilities of the business analyst
- Identify and document use cases for a sample project (in a practice workshop)
- Classify and record the business rules that affect a software system
- Identify and correct poor requirement statements
- Prioritize requirements using a simple tool
- Use peer reviews and other techniques to find requirements errors
- Implement improved requirements change practices
- Adapt requirements practices to projects following an agile development approach
- Define a path forward for improving your current requirements processes.

Additional Learning Objectives for 3-Day Version:

- Write and review different types of requirements statements

Course Description: “In Search of Excellent Requirements” (1 day)

Objectives:

- Be able to recognize and classify different types of requirements information
- Learn about many “good practices” for requirements elicitation, analysis, specification, validation, and management
- Learn how to apply the use case technique for eliciting user requirements
- Select appropriate techniques for representing requirements on your projects
- Be able to critically evaluate requirements statements for ambiguity and other problems

Abstract: Requirements form the foundation for all the software work that follows. Arriving at a shared vision of the product to be developed is one of the greatest challenges facing the software project team, and customer involvement is among the most critical factors in software quality. The objective of this course is to give attendees a tool kit of practices, reinforced with practice sessions and group discussions, that they can begin applying to improve the quality of the requirements development and requirements management processes in their organization.

This 1-day course describes tested methods that can help any organization improve the way it elicits, documents, and analyzes software requirements. Characteristics of excellent requirements statements and requirements specifications are presented and used to evaluate some sample functional requirements. The course emphasizes several practical techniques, including:

- Customer involvement through a “product champion” model
- The application of use cases for defining user needs and system functions
- Writing software requirements specifications using a standard template
- Construction of dialog maps to model user interfaces
- Using prototypes to clarify and refine user needs
- Using peer reviews to find errors in requirements
- Using a requirements traceability matrix to connect requirements to design elements, code, and tests

The basic concepts of requirements management are described, as are practical methods for managing changes to requirements. These techniques can reduce project risk by improving the quality and control of the software requirements, thereby increasing the likelihood of a successfully completed project.

Audience: This course will be useful to requirements and business analysts, project and product managers, user representatives, software engineers, marketers, and anyone else engaged in eliciting, documenting, analyzing, or managing customer requirements for software applications.

Format: Blend of lecture, class discussion, group discussions on requirements problems and solutions, and practice sessions. Practice sessions give attendees some experience in working with use cases, drawing a dialog map, and reviewing requirements.

Outline for “In Search of Excellent Requirements” Course (1 day)

(Practice activities are shown in *italics*)

I. Introduction to Requirements Engineering
   A. Introduction to course, objectives, participant expectations
   B. Define three levels of software requirements: business, user, and functional
   C. Describe characteristics of high-quality requirements
   D. Requirements development vs. requirements management

II. Practice session: Small group discussions on requirements problems in their projects

III. Software Requirements Development
   A. A requirements development process
   B. The role of the business analyst
   C. The vision and scope document
   D. Sources of requirements
   E. Classifying requirements into categories
   F. Customer involvement in the requirements process: the product champion model
   G. Eliciting user requirements through use cases
      Practice session: Describing a use case for an airline reservation kiosk
   H. Documenting requirements: the software requirements specification
   I. Software quality attributes
   J. Practice session and discussion: reviewing a portion of an SRS
   K. Modeling user interfaces with dialog maps
      Practice session: Drawing a dialog map from use cases
   L. Reducing the expectation gap through prototyping
   M. Other methods of analyzing and validating requirements

IV. Software Requirements Management
   A. Basic requirements management practices
   B. Version and change management
   C. Requirements change impact analysis
   D. Requirements tracing

V. Practice session: Small group discussions on solving requirements problem

VI. Wrap-up

Course Description: “In Search of Excellent Requirements” (2 days)

Objectives:

- Be able to recognize and classify different types of requirements information
- Learn about many “good practices” for requirements elicitation, analysis, specification, validation, and management
- Learn how to apply the use case technique for eliciting user requirements
- Select appropriate techniques for representing requirements on your projects
- Be able to critically evaluate requirements statements for ambiguity and other problems

Abstract:

Requirements form the foundation for all the software work that follows. Arriving at a shared vision of the product to be developed is one of the greatest challenges facing the software project team, and customer involvement is among the most critical factors in software quality. The objective of this course is to give attendees a tool kit of practices, reinforced with practice sessions and group discussions, that they can begin applying to improve the quality of the requirements development and requirements management processes in their organization.

This two-day course describes dozens of tested methods that can help any organization improve the way it elicits, analyzes, documents, validates, and manages software requirements. Characteristics of excellent requirements statements and requirements specifications are presented and used to evaluate some sample functional requirements. The course emphasizes many practical techniques, including:

- Creating an effective customer-developer partnership
- Customer involvement through a “product champion” model
- The application of use cases for defining user needs and system functions
- Specifying requirements using a standard template
- Classifying and recording business rules that affect a software system
- A simple model for prioritizing requirements
- Constructing dialog maps and other analysis models to provide alternative views of the requirements
- Using prototypes to clarify and refine user needs
- Using peer reviews to find requirements errors
- Using a requirements traceability matrix to connect requirements to design elements, code, and tests
- Adapting requirements practices to agile projects

The basic concepts of requirements management are described, as are practical methods for managing changes to requirements. These techniques can reduce project risk by improving the quality and control of the software requirements, thereby increasing the likelihood of a successfully completed project.

Audience:

This course will be useful to business analysts, project and product managers, user representatives, software engineers, marketers, and anyone else engaged in eliciting, documenting, analyzing, or managing customer requirements for software applications.

Format:

Blend of lecture, class discussion, group discussions on requirements problems and solutions, and practice sessions. Practice sessions give attendees some experience in working with use cases, drawing a dialog map, reviewing requirements, and writing an action plan to improve their group’s requirements practices.

Outline for “In Search of Excellent Requirements” Course (2 days)

(Practice activities are shown in *italics*)

I. Introduction to Requirements Engineering
   A. Introduction to course, objectives, participant expectations
   B. Define three levels of software requirements: business, user, and functional
   C. Describe characteristics of high-quality requirements
   D. Requirements development vs. requirements management

II. Practice session: Small group discussions on requirements problems in their projects

III. Software Requirements Development
   A. A requirements development process
   B. The role of the business analyst
   C. The customer-development partnership
   D. The vision and scope document
      Practice session: Drawing a context diagram

IV. Requirements Elicitation
   A. Sources of requirements
   B. Classifying requirements into categories
   C. User classes
   D. Customer involvement in the requirements process: the product champion model
   E. Eliciting user requirements through use cases and user stories
      Practice session: Describing a use case for an airline reservation kiosk
   F. Business rules
      Practice session: Writing business rules

V. Requirements Specification
   A. Documenting requirements: the software requirements specification
   B. Writing clear requirements
   C. Requirements management tools
   D. Practice session and discussion: Reviewing a short set of requirements
   E. Practice session: Examining requirements for problems and rewriting them

VI. Requirements Analysis
   A. Prioritizing requirements
   B. Software quality attributes
      Practice session: Writing quality attributes
   C. Using analysis models to represent requirements visually
      Practice session: Drawing a dialog map
   D. Reducing the expectation gap through prototyping

VII. Requirements Validation
   A. Requirements validation techniques
   B. Reviewing requirements

VIII. Software Requirements Management
   A. Requirements management goals and practices
   B. Version and change management
   C. Requirements change impact analysis
   D. Requirements tracing
   E. Requirements and software risk management
   F. Practice session: Designing a requirements change control process

IX. Practice session: Small group discussions on solving requirements problems

X. Improving Your Requirements Processes
   A. The process improvement change cycle
   B. The learning curve
   C. Practice session: Writing a requirements process improvement action plan

Course Description: “In Search of Excellent Requirements” (3 days)

Objectives:

- Be able to recognize and classify different types of requirements information
- Learn about many “good practices” for requirements elicitation, analysis, specification, validation, and management
- Learn how to apply the use case technique for eliciting user requirements
- Select appropriate techniques for representing requirements on your projects
- Be able to critically evaluate requirements statements for ambiguity and other problems
- Be able to write clear, unambiguous, and actionable requirements.

Abstract:

Requirements form the foundation for all the software work that follows. Arriving at a shared vision of the product to be developed is one of the greatest challenges facing the software project team, and customer involvement is among the most critical factors in software quality. The objective of this course is to give attendees a tool kit of practices, reinforced with practice sessions and group discussions, that they can begin applying to improve the quality of the requirements development and requirements management processes in their organization.

This three-day course describes dozens of tested methods that can help any organization improve the way it elicits, analyzes, documents, validates, and manages software requirements. Characteristics of excellent requirements statements and requirements specifications are presented and used to evaluate some sample functional requirements. The course emphasizes many practical techniques, including:

- Creating an effective customer-developer partnership
- Customer involvement through a “product champion” model
- The application of use cases for defining user needs and system functions
- Writing software requirements specifications using a standard template
- Classifying and recording business rules that affect a software system
- A simple model for prioritizing requirements
- Constructing dialog maps and other analysis models to provide alternative views of the requirements
- Using prototypes to clarify and refine user needs
- Using peer reviews to find requirements errors
- Using a requirements traceability matrix to connect requirements to design elements, code, and tests
- Writing use cases, functional requirements, data dictionary entries, quality attributes, and business rules
- Adapting requirements practices to agile projects

The basic concepts of requirements management are described, as are practical methods for managing changes to requirements. These techniques can reduce project risk by improving the quality and control of the software requirements, thereby increasing the likelihood of a successfully completed project.

Audience:

This course will be useful to requirements and business analysts, project and product managers, user representatives, software engineers, marketers, and anyone else engaged in gathering, documenting, analyzing, or managing customer requirements for software applications.

Format:

Blend of lecture, class discussion, group discussions on requirements problems and solutions, and practice sessions. Practice sessions give attendees some experience in working with use cases, drawing a dialog map, reviewing a requirements specification, writing good requirements, and writing an action plan to improve their group’s requirements practices.

Outline for “In Search of Excellent Requirements” Course (3 days)

(Practice activities are shown in *italics*)

I. Introduction to Requirements Engineering
   A. Introduction to course, objectives, participant expectations
   B. Define three levels of software requirements: business, user, and functional
   C. Describe characteristics of high-quality requirements
   D. Requirements development vs. requirements management

II. Practice session: Small group discussions on requirements problems in their projects

III. Software Requirements Development
   A. A requirements development process framework
   B. The role of the business analyst
   C. The customer-development partnership
   D. The vision and scope document
      Practice session: Writing a vision statement
      Practice session: Drawing a context diagram

IV. Requirements Elicitation
   A. Sources of requirements
   B. Classifying requirements into categories
      Practice session: Classifying requirements
   C. User classes
      Practice session: Identifying your user classes
   D. Customer involvement in the requirements process: the product champion model
      Group discussion: Who are your product champions?
   E. Eliciting user requirements through use cases
      Practice session: Describing a use case for an airline reservation kiosk
   F. Event-response tables
   G. Business rules
      Practice session: Writing business rules

V. Requirements Specification
   A. Documenting requirements: the software requirements specification
   B. Requirements management tools
   C. Practice session and discussion: Reviewing a portion of an SRS
   D. Practice session: Examining requirements for problems and rewriting them

VI. Requirements Analysis
   A. Prioritizing requirements
   B. Software quality attributes
      Practice session: Writing quality attributes
   C. Using analysis models to represent requirements graphically
   D. Modeling user interfaces with dialog maps
      Practice session: Drawing a dialog map from use cases
   E. Reducing the expectation gap through prototyping
   F. Requirements analysis and finding missing requirements

VII. Requirements Validation
   A. Requirements validation techniques
   B. Peer reviews and inspections
   C. Reviewing requirements

VIII. Software Requirements Management
   A. Requirements management goals and practices
   B. Version management
   C. Change management
   D. Requirements change impact analysis
   E. Requirements tracing
   F. Requirements and software risk management

  - *G. Practice session: Designing a requirements change control process*

- *IX. Practice session: Small group discussions on solving requirements problems*

- X. Improving Your Requirements Processes
  - A. The process improvement change cycle
  - B. The learning curve
  - *C. Practice session: Writing a requirements process improvement action plan*

- *XI. Requirements Writing Workshop*
  - *A. Eliciting requirements*
  - *B. Writing use cases*
  - *C. Writing functional requirements*
  - *D. Writing quality attributes*
  - *E. Writing data dictionary entries*
  - *F. Reviewing requirements*

- XII. Wrap-up