Moving Requirements Forward

SPRING 2015

SOFTWARE REQUIREMENTS TRAINING
The more training you take the greater the savings!

Maximize the impact of your training by combining courses in the same location or attending training in conjunction with a top industry conference on software development.

<table>
<thead>
<tr>
<th>SUNDAY</th>
<th>MONDAY</th>
<th>TUESDAY</th>
<th>WEDNESDAY</th>
<th>THURSDAY</th>
<th>FRIDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements-Based Testing Workshop—$1,995</td>
<td></td>
<td></td>
<td>STAREAST • May 3–8, 2015 • Orlando, FL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Combine a training class with your conference registration and save up to $300!

LEARNING OPTIONS

- **Public P**
  - Instructor-led training in a city near you
  - Network with colleagues

- **ON-SITE ADVANTAGE**
  - Instructor-led training at your location
  - Train your entire team

- **eLearning**
  - Self-paced learning, online
  - Train anytime, anywhere

- **LIVE VIRTUAL TRAINING**
  - Instructor-led training from your desktop
  - Save Money and Travel Time

EASY TO REGISTER

- **Online:**
  - www.sqetraining.com/requirements

- **Email:**
  - sqeinfo@sqe.com

- **Phone:**
  - 888.268.8770/904.278.0524
Requirements-Based Testing Workshop

A DISCIPLINED APPROACH FOR DESIGNING, MAINTAINING, AND EXECUTING TESTS

- Develop and maintain efficient tests that cover all functional requirements
- Design test cases that force defects to appear early in testing
- Learn and practice cause-effect graphing to design more robust tests
- Learn and practice alternative test design approaches—pairwise, equivalence class
- Optimize and reduce the size of your test suite
- Integrate testing in the software development lifecycle

If your testing efforts are not achieving the payback you and your organization expect, this course is for you. Requirements-Based Testing (RBT) delivers a proven, rigorous approach for designing a consistent and repeatable set of highly optimized test cases. Companies employing RBT practices have achieved twice the requirements coverage with only half the tests they previously maintained.

**Design the Test Library**

The RBT process helps you validate that the requirements are clear and complete. Then, it guides you to define a set of tests verifying that the design and code fully meet those requirements. You’ll learn and practice cause-effect graphing, a test design technique that ensures that defects will be fully observable. If there are any defects in the software—even ones that could be hidden from tests by other errors—cause-effect graphing will find them. With this technique, you’ll be able to reduce the number of tests you need and make sure that every test is valuable.

Explore alternative test design techniques and the advantages and disadvantages of each. Learn how to complement functional, black-box testing with code-based, white-box testing to further ensure complete coverage and higher quality. Classroom exercises are employed throughout the course to reinforce your learning.

**Leave With a Testing Process That Integrates With the Development Lifecycle**

Take back a lifecycle testing process that incorporates testing as an integrated—and integral—part of the software development project. With the RBT process, your next project will experience significant time and cost savings while helping the test team develop better estimates and dynamically track test and project progress.

*Bring samples from your own projects to work on and evaluate during class.*

Who Should Attend

Requirements-Based Testing is for test managers, test engineers, QA specialists, software managers, and anyone responsible for developing tests and test suites.

Although the focus of this course is on process and techniques, there will be a brief introduction to the BenderRBT™ software tool, which automates much of the requirements-based testing process.

**Course Link:** www.sqetraining.com/RBT3

**Instructor Spotlight**

**Richard Bender** has over 45 years experience in software with a primary focus on quality assurance and testing. He has consulted internationally to large and small corporations, government agencies, and the military. Richard’s work has included a wide variety of application classes and technology bases from embedded systems to super computer-based systems—and everything in between—consulting to both vendors and IT departments alike. He has been active in establishing industry standards for software quality and is a frequent speaker at conferences, universities, and corporate events. He was one of the first programmers ever awarded IBM’s Outstanding Invention Award for his breakthroughs on code-based testing.

Gary Magyeroodi is an additional instructor for this course.

**STAR EAST • May 3-5, 2015 • Orlando, FL**

**Introduction**

Making the business case for quality
Definitions of testing
The 12-step RBT test approach

**Initial RBT Steps**

Validating requirements against objectives
Validating the scope of requirements using scenarios and tours
Exercise: Identifying scenarios

**Finding Ambiguities in Requirements**

The ambiguity review checklist
Managing the ambiguity process

**Cause-effect graphing**

Basic logical operators
Exercise: Identifying variables, states, and relations
Five graphing constructs of functional requirements
Exercise: Create cause-effect graphs for numerous requirements

**Data Constraints**

Boundary condition data constraints
Processing sequence imposed constraints
Exercise: Determine what constraints apply
Inconsistencies in processing rules
Exercise: Determine why the requirements are logically inconsistent

**Test Case Design from Graphs**

Strategies for test case design
Concept of fault detection
Exercise: Identifying functional variations
Exercise: Determine the required functional variations to test
Packaging functional variations into test cases
Exercise: Complete the test designs from the variations

**Alternative Test Designs**

Equivalence class testing
Exercise: Determine the states to tests
Review of other model-based testing techniques
Optimized pairs and orthogonal pairs
Exercise: Design tests using optimized pairs
Comparing the various test design approaches

**Points of Integration**

Integrating testing throughout development
Developing user acceptance tests before coding starts

**Code-based Testing**

White-box test completion criteria
Data flow-based testing
Integrating black-box and white-box testing

**Management Considerations**

Planning and estimating guidelines
Change control
Test team organization
Tracking the testing effort
Contract management

**Test Automation**

Test automation issues
How the RBT process integrates with the rest of test automation
Studies show that there is no way to significantly improve software quality and productivity without improving the quality of the requirements. Testing, by definition, is comparing the expected results to the observed results. The majority of software testing does not meet this criteria since the requirements are neither detailed enough nor clear enough to pre-determine the test results.

This course addresses how to ensure that your requirements are correct, complete, unambiguous, and logically consistent. The approaches defined in this course work with all methodologies and automation suites.

**Focus Up-Front on Problem Avoidance**

Most software projects are over budget, over schedule, and result in zero to negative return on investment. Studies have shown that poor requirements are the single biggest factor in this. The cost of system development and the time required to deliver an application is driven up by scrap and rework as requirements deficiencies are discovered late in the project. Writing Testable Requirements focuses on problem avoidance—ensuring that the project scope is focused on meeting the organization’s needs and writing requirements accurately the first time before coding starts. This training course offers guidelines for describing software specifications of processes and data and ensuring that requirements have the clarity and detail needed as the basis of test cases.

This information is critical to designers, coders, testers, and technical writers. The techniques can be applied to requirements written to various company or industry standards. The course also addresses compliance with common industry guidelines and the effects of automated repositories on requirements writing styles. You are encouraged to bring samples from your own projects to work on and evaluate during class.

Who Should Attend

This course is intended to help those who write and review detailed functional specifications and those who must develop and test systems based on those requirements. The intended audience includes testing staff, requirements analysts, developers, and project managers. No specific prerequisites are assumed, but you are expected to be knowledgeable with at least basic test or software development experience.

Instructor Spotlight

Richard Bender has over 45 years experience in software with a primary focus on quality assurance and testing. He has consulted internationally to large and small corporations, government agencies, and the military. Richard’s work has included a wide variety of application classes and technology bases from embedded systems to super computer-based systems—and everything in between—consulting to both vendors and IT departments alike. He has been active in establishing industry standards for software quality and is a frequent speaker at conferences, universities, and corporate events. He was one of the first programmers ever awarded IBM’s Outstanding Invention Award for his breakthroughs on code-based testing.

Gary Magyarik is an additional instructor for this course.

Course Link: [www.sqetraining.com/wtr](http://www.sqetraining.com/wtr)
Requirements-Based Testing

A DISCIPLINED APPROACH FOR DESIGNING, MAINTAINING, AND EXECUTING TESTS

- Develop and maintain efficient tests that cover all functional requirements
- Design test cases that force defects to appear early in testing
- Learn and practice cause-effect graphing to design more robust tests
- Learn and practice alternative test design approaches—pairwise, equivalence class
- Optimize and reduce the size of your test suite
- Integrate testing in the software development lifecycle

If your testing efforts are not achieving the payback you and your organization expect, this course is for you. Requirements-Based Testing (RBT) delivers a proven, rigorous approach for designing a consistent and repeatable set of highly optimized test cases. Companies employing RBT practices have achieved twice the requirements coverage with only half the tests they previously maintained.

**Design the Test Library**

The RBT process helps you validate that the requirements are clear and complete. Then, it guides you to define a set of tests verifying that the design and code meet fully those requirements. You’ll learn and practice cause-effect graphing, a test design technique that ensures that defects will be fully observable. If there are any defects in the software—even ones that could be hidden from tests by other errors—cause-effect graphing will find them. With this technique, you’ll be able to reduce the number of tests you need and make sure that every test is valuable.

Explore alternative test design techniques and the advantages and disadvantages of each. Learn how to complement functional, black-box testing with code-based, white-box testing to further ensure complete coverage and higher quality. Classroom exercises are employed throughout the course to reinforce your learning.

**Leave With a Testing Process That Integrates With the Development Lifecycle**

Take back a lifecycle testing process that incorporates testing as an integrated—and integral—part of the software development project. With the RBT process, your next project will experience significant time and cost savings while helping the test team develop better estimates and dynamically track test and project progress.

Bring samples from your own projects to work on and evaluate during class.

Who Should Attend

Requirements-Based Testing is for test managers, test engineers, QA specialists, software managers, and anyone responsible for developing tests and test suites. Finding Ambiguities in Requirements is a prerequisite for this class.

Although the focus of this course is on process and techniques, there will be a brief introduction to the BenderRBT™ software tool, which automates much of the requirements-based testing process.

**Instructor Spotlight**

Richard Bender has over 45 years experience in software with a primary focus on quality assurance and testing. He has consulted internationally to large and small corporations, government agencies, and the military. Richard’s work has included a wide variety of application classes and technology bases from embedded systems to super computer-based systems—and everything in between—consulting to both vendors and IT departments alike. He has been active in establishing industry standards for software quality and is a frequent speaker at conferences, universities, and corporate events. He was one of the first programmers ever awarded IBM’s Outstanding Invention Award for his breakthroughs on code-based testing.

Gary Mogiroti is an additional instructor for this course.

**Course Links:**

Requirements-Based Testing — www.sqetraining.com/rbt

---

---

2-Day Course Outline:

- **Introduction**
  - Making the business case for quality
  - Definitions of testing
  - The 12-step RBT test approach

- **Initial RBT Steps**
  - Validating requirements against objectives
  - Validating the scope of requirements using scenarios and tours
  - Exercise: Identifying scenarios

- **Cause-effect graphing**
  - Basic logical operators
  - Exercise: Identifying variables, states, and relations
  - Five graphing constructs of functional requirements
  - Exercise: Create cause-effect graphs for numerous requirements

- **Data Constraints**
  - Boundary condition data constraints
  - Processing sequence imposed constraints
  - Exercise: Determine what constraints apply
  - Inconsistencies in processing rules
  - Exercise: Determine why the requirements are logically inconsistent

- **Test Case Design from Graphs**
  - Strategies for test case design
  - Concept of fault detection
  - Identifying functional variations
  - Exercise: Determine the required functional variations to test
  - Packaging functional variations into test cases
  - Exercise: Complete the test designs from the variations

- **Alternative Test Designs**
  - Equivalence class testing
  - Exercise: Determine the states to test
  - Review of other model-based testing techniques
  - Optimized pairs and orthogonal pairs
  - Exercise: Design tests using optimized pairs
  - Comparing the various test design approaches

- **Points of Integration**
  - Integrating testing throughout development
  - Developing user acceptance tests before coding starts

- **Code-based Testing**
  - White-box test completion criteria
  - Data flow-based testing
  - Integrating black-box and white-box testing

- **Management Considerations**
  - Planning and estimating guidelines
  - Change control
  - Test team organization
  - Tracking the testing effort
  - Contract management

- **Test Automation**
  - Test automation issues
  - How the RBT process integrates with the rest of test automation

---

---

Easy to Register

Online: www.sqetraining.com/requirements

Email: sqeinfo@sqe.com

Phone: 888.268.8770
904.278.0524
Finding Ambiguities in Requirements

TECHNIQUES FOR IMPROVING REQUIREMENTS AND SOFTWARE

- Practice performing ambiguity reviews on requirements documents
- Improve requirements and reduce errors in software
- Design the right tests with clear and unambiguous requirements
- Take back a practical ambiguity review checklist

Studies have shown that poor requirements are one of the most significant contributors to project failure—and half of all defects have their origin in bad requirements. If specifications are ambiguous, there is nearly a 100% chance that there will be one or more defects in the corresponding code.

Techniques for Quickly Reviewing Requirements for Ambiguities

Finding Ambiguities in Requirements explores ways to review specifications quickly and quantitatively to identify what is unclear about them. This powerful, yet practical, method helps you ensure that requirements documentation is clear, concise, and unambiguous.

Learn about and practice simple, effective review techniques that can reduce the ambiguity rate by 95% on subsequent specifications. In addition you’ll learn to determine if the requirements are detailed enough to produce a sufficient set of test cases to validate the system’s functionality. To reinforce lectures and discussions, you’ll practice your newly acquired knowledge and skills in classroom exercises.

You can apply these same techniques to design specifications, user manuals, training materials, and online help, as well as agreements and contracts for software development projects.

Who Should Attend

This course is intended to help those who write and review functional requirements and those who develop and test systems based on those requirements. The audience includes business analysts, test analysts, requirements engineers, developers, technical writers, and project managers. No specific prerequisites are assumed.

Course Link: www.sqetraining.com/far

Instructor Spotlight

Richard Bender has over 45 years experience in software with a primary focus on quality assurance and testing. He has consulted internationally to large and small corporations, government agencies, and the military. Richard’s work has included a wide variety of application classes and technology bases from embedded systems to super computer-based systems—and everything in between—consulting to both vendors and IT departments alike. He has been active in establishing industry standards for software quality and is a frequent speaker at conferences, universities, and corporate events. He was one of the first programmers ever awarded IBM’s Outstanding Invention Award for his breakthroughs on code-based testing.

Gary Mogiorodi is an additional instructor for this course.

Instructor Spotlight

Learning Options

1-Day Course Outline:

Introduction
- Definition of good requirements
- Testable requirements
- Deterministic results and requirements

Confusing Constructs
- Limitations of the English language
- Examples of ambiguity
- Ambiguity review checklist
- Performing an ambiguity review
  - Exercise: Identify ambiguities in various mini-specs

Jargon and Complexity
- The language barrier
- Carelessness
- Assumed functional knowledge
  - Exercise: Translate jargon into plain English
  - Exercise: Simplify overly complex writing

Defining Clear Objectives
- Objectives vs. requirements
- Quantitative vs. qualitative objectives
- Identifying the objectives of all stakeholders
  - Product vs. project objectives
  - Templates for the objectives specification
  - Exercise: Identify the true objectives

Introduction to Writing Testable Requirements
- Alternative styles
- Guidelines for writing clear specifications
- Summary of how ambiguities are addressed

Introduction to Requirements-Based Testing
- Quality filters
- Advantages of rigorous testing
- Using test cases to validate requirements
Agile Requirements Workshop

- Use a lightweight but disciplined requirements approach to speed time to market while increasing quality
- Progressively elaborate from a high-level vision to a user story supported with lightweight specifications
- Commit to a feature breakdown structure instead of a work breakdown structure
- Use a readiness approach to continually mature requirements and spec for release and sprint planning
- Create testable business goals and use them as the true measure of success
- Write user stories, and split or combine to represent a full slice of customer value
- Use collaborative story writing, story mapping, personas, and scenarios to drive the requirements: elicitation, elaboration, and prioritization
- Create acceptance criteria and testable examples instead of lengthy text descriptions to create clarity
- Use Acceptance Test-Driven Development (ATDD) to get better requirements
- Prioritize work using lightweight techniques like selection matrices, MoSCoW, and voting models
- Apply the basics of automated testing

Agile development methods remove barriers between customers and the development team. Using agile approaches, your organization will more easily meet market and customer needs while attaining its ROI objectives. Agile relies on lightweight but disciplined approaches to requirements, and by attending this training, your team will increase alignment with business objectives, get to market faster, and unleash creativity, achieving both “quick wins” as well as long term sustainable success, all while having more fun.

Learn the Essential Concepts and Tools of Effective Agile Requirements

This workshop provides the participants experience to quickly deliver value in a dynamically changing environment. After this course, attendees will be able to prioritize, define, and refine requests into user stories, requirements, and lightweight specifications to simplify delivery and maximize project value. The workshop presents an easy to understand, lightweight model for agile requirements, specification by example, and functional specifications. The workshop also provides a maturation approach so that teams can balance sprint readiness with just-in-time specifications.

This course is taught by leading agile practitioners with decades of real-world industry experience at companies ranging from small businesses to the Fortune 100. Exercises, demonstrations, facilitated discussions, case studies, tool and template examples, and more are interwoven throughout the course to illustrate the principles being taught in a comprehensive fashion and are interactively tailored to each class’s particular needs.

Who Should Attend

B2-100 business customers, product managers, business analysts, quality analysts, and others aiming to maximize the benefit that they receive from their agile projects by learning how to better prioritize and define requirements.

Bonus

Attendees are eligible for 14 PDU credits with the Project Management Institute (PMI).

Instructor Spotlight

David Bulkin is a strong leader, technologist, and process engineer with over 20 years experience in applying lean processes to manage portfolios, projects, people, process, and technology for competitive advantage. He has helped numerous teams apply agile project management and engineering methods as a practitioner and coach. He is a frequent speaker at user groups and blogs on the subject. David’s career has spanned both the public and private sector. He built an electronic commerce startup from the ground up and also managed large scale, mission critical technology projects as a VP at JP Morgan Chase. He is frequently engaged in both strategic board-level and hands-on implementation (analysis, design, coding) activities, keeping his agile coaching and training skills sharp and relevant. He is on the Board of Directors at the Center for Program Transformation (CPT) and Ocean 20 Technologies Group. With the CPT, David prepared Senate testimony on IT Oversight and contributed to federal legislation. In the 1990s David was a key member of the Software Program Managers Network (SPMN) where he identified proven software best practices, conveyed them to managers of large-scale DoD system acquisition programs, and consulted on numerous projects.

Arlen Bankston is an additional instructor for this course.

Introduction to Agile

A case for change

Scrum process overview

Prioritization

- From vision to high concept
- Creating tangible goals
- Scenarios
- Requirements brainstorming
- Product box
- Projects on a page
- Release planning and product backlog
- Sprints
- Planning and sprint backlog
- Sprint
- Review and retrospective

Grooming, Prioritization, and Readiness

Story splitting

- Story mapping
- Prioritization
- MoScow
- Impact matrices
- Economic/voting models
- Sustainable discovery
- Problems with JIt elaboration
- Sprint look-a-head
- Continuous discovery

Course Link:

www.sqetraining.com/arw
Mastering the Requirements Process
BUILD THE RIGHT SOFTWARE THE FIRST TIME

- Learn the complete process of eliciting, writing, and testing requirements
- Write universally understandable requirements
- Understand exactly what your customer wants—and needs

Why Requirements—What’s in It for You?
The problem is that people rely on software to help them do their work, but other people build it. Solving the problem means understanding the actual work of the business users—and what they need to do it. Requirements is about deducing the product that will add long-term value to the organization—and then writing requirements that lock the developers into the exact product.

Getting It Right the First Time
Building software today means that you are in it for the long haul. And you know that there are more demands—and fewer resources—to meet those demands. Getting the software right—the first time—is the only way to succeed under these circumstances. Today’s requirements process is incremental with quick cycle times. It uses prototypes and scenarios, and the requirements process ensures that you get the right result by writing a fit criterion to ensure that the requirement is testable.

Your Requirements
Requirements are the most misunderstood—but the most crucial—part of systems development. If the requirements are wrong, you end up with loads of late re-work, or even worse, with the wrong system. Your requirements process must be your own, but it should be based on field-proven techniques and templates. This course presents the Volere process—used and improved by thousands of organizations around the world—and then shows you how to make it your own process. As a participant, you receive the Volere Requirements Specification Template—downloaded by more than 13,000 users—to take home with you. Your instructor has written requirements for dozens of projects and brings you insights that only come from real-world experience. You will learn insights and techniques that you can put to work right away.

Instructor Spotlight

James Robertson and Suzanne Robertson are additional instructors for this course.

Take Home Bonus:

Course Link: www.sqetraining.com/mrp

Costs of Development
- The requirements process defined
- Overview of the requirements process
- Cyclical requirements

Project Blast-Off
- Scope of the business area
- Identifying and using stakeholder maps
- Testable project goals
- Ensuring a viable requirements project

Trawling for Requirements
- Finding the real requirements
- Requirements for agile projects
- Techniques for eliciting requirements
- Using business events and use cases

Functional Requirements
- Identifying what the product must do
- Establishing the product use case
- Writing use case scenarios
- Requirements, not solutions

Non-functional Requirements
- Qualities of the product
- Usability, look and feel, security, etc.
- How to find non-functional requirements

Managing Your Requirements
- Specification templates
- Tracing requirements
- Prioritizing requirements
- Automated requirements tools

The Quality Gateway
- Stopping requirements creep
- Defining fit criteria
- Testing requirements

Prototyping and Scenarios
- Using prototypes to drive out requirements
- Low- and high-fidelity prototypes

Your Requirements Process
- Deciding your own process
- How to use a fast-track approach
- Planning iterative development
- Knowing when you have all the requirements

IIBA®, the IIBA® logo, BABOK® and Business Analysis Body of Knowledge® are registered trademarks owned by International Institute of Business Analysis. These trademarks are used with the express permission of International Institute of Business Analysis.
Requirements Modeling

**USE MODELS TO IMPROVE YOUR REQUIREMENTS GATHERING AND SYSTEMS ANALYSIS**

- Find and verify requirements with models
- Model as-is and will-be processes
- Model data and states to better understand requirements

All engineering disciplines use models to develop the products they intend to build. The reason is simple: Models help to uncover—and then clarify—the functional and control requirements for any product, including software systems. A poor set of requirements will cripple any project. Modeling the business and its requirements is a proven way of finding all the requirements and guaranteeing their accuracy. Once you know they are correct, you can use the requirements models as specifications for the designers and builders of the system.

**Learn How to Show What a System Is—Not Just What It Does**

You can describe a system by what it is and by what it does. For example, consider this typical statement from a requirements specification: “The product must calculate the cheapest fare.” Beyond this innocent description of what the system must do lies a complex set of rules, procedures, data, and functions. Requirements modeling discovers the rules for calculating the cheapest fare, the algorithms needed, and the data necessary to complete and support those calculations. In other words, you build a complete model of the system.

You also can use models when eliciting requirements. A quickly sketched process model can be indispensable for displaying your understanding of the system during interviews. A data or class model reveals the policy of the system and any gaps are quickly discovered by a model constructed with your customer. A state model explains how a system behaves and thus clarifies the consequences of requirements.

**Put Techniques into Practice**

This course includes frequent opportunities to apply the illustrated techniques. You work with your instructor to build models and prove or disprove example requirements. You construct models to elicit requirements and then feed them back to see if your understanding matches that of your customer. Learn to evaluate when each of the models is useful and determine what degree of detail is necessary.

**Instructor Spotlight**


James Robertson and Suzanne Robertson are additional instructors for this course.

**Course Link:** [www.sqetraining.com/rm](http://www.sqetraining.com/rm)

---

**Modeling Objectives**

- Requirements and systems analysis
- How modeling is used in product development

**The Context Model**

- Describing the scope of the problem
- How the work connects to the outside world
- Ensuring that the scope is accurate

**Event Partitioning**

- Need to partition large systems into smaller parts
- Business events
- Work’s response to the business event
- How to find the business events

**Process Modeling**

- Event response process models
- Process specifications and data definitions
- Modeling viewpoints and how to use them

**Data Modeling**

- Class or entity model
- Classes, attributes, and associations
- Heuristics for finding classes and associations
- Relationship between process and data
- Using attribution to construct first-cut models

**State Modeling**

- State models and why we want them
- Link between state transitions and business events
- Modeling the states of classes

**Modeling the Product**

- Building models of the software product
- UML models
- Role of business events and adjacent systems
- Determining the product scope
- Business events and use cases

**TAKE HOME BONUS:**

Each course participant receives a copy of James and Suzanne Robertson’s book Complete Systems Analysis for class work and later review of course material.
Essential Software Requirements
TECHNIQUES AND PRACTICES FOR SUCCESSFUL PROJECTS

- 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.

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 Link: www.sqetraining.com/esr

Instructor Spotlight
Robert Sabourin has more than 30 years of management experience leading teams of software development professionals. A well-respected member of the software engineering community, Robert has managed, trained, mentored, and coached thousands of top professionals in the field. He frequently speaks at conferences and writes on software engineering, SQA, testing, and documentation. The author of I am a Bug!, the popular software testing children’s book, Robert is an adjunct professor of Software Engineering at McGill University.

Lee Copeland is an additional instructor for this course.

Instructor Spotlight

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 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

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
- 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

3-Day Course Outline:
Extending Requirements  
TAKE YOUR REQUIREMENTS TO THE NEXT LEVEL

This workshop extends the foundations laid in the Mastering the Requirements Process course by showing you how to choose the best set of requirements to give you a competitive edge—and still get your product to market on time. The instructor covers techniques for quantifying the business value of investing in requirements as well as the need to anticipate market opportunities by creating and inventing visionary requirements. You’ll also learn to deal with requirements for existing systems along with techniques for managing meta projects (large projects made up of a number of smaller ones).

A good requirements engineer has the skill of helping a wide variety of people communicate with each other. The project sociology structure in this seminar helps you to discover the correct stakeholders for your project. You’ll also walk away with proven techniques to facilitate their involvement in the appropriate parts of the project and, even more importantly, how to help them stay involved throughout.

Extending Requirements expands what you know about requirements and shows you how to take your projects to a higher level of requirements excellence.

Recommended Prerequisite: Mastering the Requirements Process

Benefits
Discover ways to select the lightest requirements process that will work for your project. Effectively determine the cost and value of a requirement. Gain techniques for building and maintaining good stakeholder relationships.

Who Should Attend
Business analysts, systems managers, project leaders, consultants, systems analysts, and planners. The material is designed for experienced requirements engineers who are already familiar with business events, product use cases, context models, functional requirements, non-functional requirements, constraints, and the Volere requirements template.

Instructor Spotlight

Suzanne Robertson is an additional instructor for this course.

Course Link: www.sqetraining.com/er

2-Day Course Outline:

- **eXtreme Requirements**
  - Agile processes
  - Critical success factors
  - Needed requirements attributes

- **Project Sociology**
  - Who are the stakeholders?
  - Keeping them interested
  - Collaborative projects

- **Inventing Better Products**
  - People may not know what they want or what is possible
  - Inventing something better
  - How to inspire new ideas

- **Using Stories in Requirements**
  - Telling stories to communicate
  - Using stories to discover requirements
  - Scenarios and other stories

- **Costs and Benefits**
  - Quantifying requirements
  - Prioritizing and negotiating requirements
  - How to take advantage of requirements reuse

- **Meta Management and Multi-Technology**
  - Large projects
  - Connections between interdependent projects
  - Multiple technologies in the same project

- **Requirements for Existing Systems**
  - Changes to existing systems
  - Analyzing new requirements
  - Assessing the impact (from the point of view of cost, benefit, effort, delay)
  - People affected by changes

- **Skinny Framework**
  - A minimal framework of deliverables and checkpoints
  - Identifying potential requirements black holes
  - Feedback loops
eFoundation for Requirements Development and Management
A ROADMAP TO SUCCESS

- Learn key requirements development and management skills
- Discover the ways to elicit and document requirements
- Understand the fundamentals for using the Requirements Roadmap to analyze requirements
- Realize steps to adapt and improve requirements practices on your project

If you currently develop and manage requirements, manage people who do, or plan to do either in the future, this course is for you. This course teaches essential requirements development and management skills in a flexible eLearning format. The curriculum is a series of eight self-paced courses that build the foundation you need to successfully develop and manage requirements for business projects and software products. You’ll learn how to develop and manage requirements, gain tips for eliciting and documenting requirements, discover how to use the EBG Requirements Roadmap to analyze requirements, and study the steps for adapting your requirements practices to your project.

Who Should Attend
This course is appropriate for anyone who needs a solid foundation in the basics of requirements to drive success, including business analysts, developers, project leads/managers, product owners, subject matter experts, and testing/QA staff.

Curriculum Length
8–12 hours (depending on learner experience and familiarity)

IIBA® Certified Course
This curriculum is endorsed by the International Institute of Business Analysis (IIBA®), and it aligns with the knowledge areas within the IIBA’s Business Analysis Body of Knowledge (BABOK®). A detailed cross-reference guide allows reference to the IIBA BABOK® Knowledge Areas Techniques and Tasks. You’ll earn 24 CDUs (Continuing Development Units) for completing this curriculum.

Course Outline:

- Course 1: Introduction to Requirements
- Course 2: Setting the Stage for Developing Requirements
- Course 3: Requirements Development: Elicitation
- Course 4: Requirements Development: Analysis
- Course 5: Requirements Development: Specification
- Course 6: Requirements Development: Validation
- Course 7: Requirements Management
- Course 8: Adapting Requirements Practices

Course Link: www.sqetraining.com/erts

IIBA®, the IIBA® logo, BABOK® and Business Analysis Body of Knowledge® are registered trademarks owned by International Institute of Business Analysis. These trademarks are used with the express permission of International Institute of Business Analysis.

Learn Anywhere!
Get Requirements Right the First Time
AN EFFECTIVE APPROACH TO REQUIREMENTS DISCOVERY

- Characterize the scope of the requirements work in a graphic form
- Define goals, constraints, facts, and assumptions
- Partition the requirements work
- Get down to the detailed requirements

An Efficient and Effective Approach
Whether you are just getting the requirements for a new feature or you are on a large team trying to define the requirements for a brand new system, you are always under time pressure. Getting the requirements work off on the right foot is critical. The Volere (meaning “to wish” or “to want” in Italian) process for requirements discovery has a front-end defined with efficiency in mind and can be tailored to best fit your specific situation. It is a systematic way to get to the point where there are no unpleasant surprises when you go into the study of detailed requirements.

A Proven Approach
The Volere process has been defined for almost 20 years and is practiced by organizations around the world. The text, Mastering the Requirements Process, is now in its 2nd edition. The Volere requirements specification template is now in its 15th edition.

Who Should Attend
Anyone participating in defining requirements will find value in this course, whether you are a business analyst, project manager, or subject matter expert.

Instructor Spotlight

Course Outline:
The requirements process inside the development process
Some requirements work is always done before the project starts
Requirements work in an agile environment
A first look at Volere
Getting your ducks all in a row
Getting agreement on scope
Getting agreement on goals and constraints
Getting key stakeholders involved
The Volere process from scope to detailed requirements
How detailed is a detailed requirement?
Dividing up the requirements by event response
Building a business use case
Finding the product use case
Functional and non-functional requirements

Course Link: www.sqetraining.com/vGRR

Easy to Register
Online: www.sqetraining.com/requirements
Email: sqeinfo@sqe.com
Phone: 888.268.8770 904.278.0524
Looking for ways to save training and travel dollars? Consider the on-site advantages:

- Train your team at your location
- Develop the talent already on your team and increase employee productivity
- Schedule training around your current projects
- Focus training on your team’s challenges in a small group environment
- Implement best practices and dramatically improve your business processes
- Customize course content to meet your company’s business needs
- Receive expert instruction from consultants with years of real-world experience

Corporate Volume Pricing
Companies needing to train teams to become proficient in a number of areas can benefit from corporate volume pricing. Let us help you build a training curriculum to suit your company’s needs.

Courses Available for On-site Training:

- Mastering the Requirements Process
- Requirements Modeling
- Writing Testable Requirements
- Finding Ambiguities in Requirements
- Essential Software Requirements

Extending Requirements
- Requirements-Based Testing
- Requirements-Based Testing Workshop
- Agile Requirements Workshop

Get a FREE quote and details about how easy it is to bring on-site training to your location. Call Julie at 888.268.8770 ext. 212 or email onsitetraining@sqe.com.
Easy to Register

Online:
www.sqetraining.com/requirements

Email:
sqinfo@sqe.com

Phone:
888.268.8770
904.278.0524

Public Training Course
3 Days—$1,995

Live Virtual Courses—$599

eLearning Course—$499

The more training you take the greater the savings!
Maximize the impact of your training by combining courses in the same location. Combine a full week of training for the largest discount! For a complete list of courses available, visit www.sqetraining.com or call 888.268.8770 or 904.278.0524 for pairing discount options.

For more information about on-site training courses, contact SQE Training at 888.268.8770 or 904.278.0524 or email onsitetraining@sqe.com.

*3-Day Requirements-Based Testing Workshop course in conjunction with STAREAST

Training Details

Training Course Fee Includes
• Tuition
• Course Notebook
• Continental Breakfasts and Refreshment Breaks
• Lunches
• Certificate of Completion

Training Course Schedule
7:30am–8:30am   Registration (on first day of course) and Continental Breakfast
8:30am–12:00pm  Course
12:00pm–1:00pm  Lunch
1:00pm–5:00pm   Course

Satisfaction Guarantee: SQE Training is proud to offer a 100% satisfaction guarantee. It’s part of our commitment to provide you with the highest quality education and training products. If we are unable to satisfy you, we will gladly refund your registration fee in full.

Public Training Policies: SQE Training reserves the right to make changes in course schedules, dates, locations, and accommodations. We will make every effort to notify students within a reasonable period of time. However, SQE Training is not responsible for personal travel, accommodations, or other incidental expenses in connection with changes to a course.

Cancellation Policy: Attendee substitutions are permitted. Registrants who fail to attend are subject to the full fee if they have not obtained a cancellation code from SQE Training at least 14 business days prior to the event start date. To obtain a cancellation code, call 904.278.0524 or 888.268.8770.

Register Early: The number of students per course is limited, and many courses fill to capacity. Register early to ensure your space in your preferred course.

Payment Policy: Payment is due upon registration. Seats are not confirmed until payment is received.

Forms of Payment Accepted:
• Visa, MasterCard, or American Express
• Check or company purchase order is accepted. However, payment must be received before course registration is confirmed.

Confirmation: After payment, you will receive a confirmation notice containing course details (e.g., hotel, accommodations). Please bring a copy of the letter to the course for admittance.