

Agile -Developing Agile Requirements with User Stories

COURSE DESCRIPTION:

User Stories are the most common means of representing requirements on agile projects. A user story describes system functionality that is valuable to a user or purchaser of a system. Acceptance tests are written to clarify and verify the behavior of user stories. In this course, you will learn how to write effective user stories and acceptance tests, and how to map your existing requirements processes to an agile approach.

TOPICS COVERED:

- Identify and describe user roles and personas
- Write and evaluate user stories
- Identify and document non-functional requirements and business rules
- Learn how to manage changes to agile requirements
- Learn the characteristics of an effective user representative
- Learn how to use lightweight techniques for iterative requirements gathering
- Learn how to conduct story writing workshops using low-fidelity prototypes
- Write acceptance tests for user stories
- Learn to prioritize and estimate user stories for iterations and releases

AUDIENCE:

This course is intended for business analysts, users, product managers, developers, and testers who are currently responsible, or will be in the near future, for gathering and documenting requirements using agile methods.

DURATION:

2 days

PREREQUISITES:

None

OBJECTIVES:

Learn how to write effective user stories and acceptance tests, and how to map your existing requirements processes to an agile approach.



Agile - Applying Scrum with User Stories

COURSE DESCRIPTION:

Learn the basics of Scrum's project management approach which will give you the ability to quickly develop working software that meets business needs. In Scrum, teams work within time-boxed sprints to produce production-quality software driven by business priorities and using an empirical, adaptive approach. This course explains this approach's values, key techniques, and how each team member contributes to success.

You will learn how to build a product backlog with user stories as input, and how to plan and work in 30 day sprints. You will apply what you learn in a series of exercises that take you through project start-up, sprint planning, and running a sprint.

TOPICS COVERED:

- Understand key concepts and terms
- Be able to explain the methodology's benefits and key techniques
- Become familiar with the progression of key activities and artifacts
- Prepare to participate in an upcoming project

AUDIENCE:

This course is intended for all who will actively participate in a project using the Scrum approach. People on the edges of the project (secondary customers, management staff, and participants in related processes) will also benefit.

DURATION:

2 days

PREREQUISITES:

"Developing Agile Requirements with User Stories" or a basic understanding of user story writing.

Agile - Estimating and Planning

COURSE DESCRIPTION:

A good estimating and planning process is integral to project success. Estimates provide information to make decisions, create performance targets and develop plans. Lack of an estimating process and pressure to over promise can lead to overly optimistic estimates that potentially hurt both your project team and their customers. Adaptive planning reduces budgeting and forecasting complexity. This course provides guidelines for project estimation and planning that are consistent with the core principles of Agile Development (e.g. Scrum, Extreme Programming (XP), Crystal, Feature Driven Development, etc.) and a reference process that satisfies those principles.

You will learn how to construct an estimate using a product backlog as input as well as how to derive and maintain a project schedule and staffing model. You will apply what you learn in a series of exercises that take you through the agile estimating reference process.

TOPICS COVERED:

Upon completion of this course, students will:

- Understand the key concepts and terms of agile estimating and planning
- Be able to explain the benefits and key techniques of agile estimating and planning
- Be familiar with the progression of key activities and artifacts in agile estimation and planning
- Be prepared to adapt the agile estimating reference process to an upcoming project

AUDIENCE:

This course is intended for all project team members, customers and stakeholders who are responsible for deriving, approving or understanding the fundamentals of a project estimate or project plan.

DURATION:

2 days

PREREQUISITES:

- "Developing Agile Requirements with User Stories" or a basic understanding of user stories and product backlogs.
- Applying Scrum with User Stories" or attendance at a Certified ScrumMaster (CSM) course.



Agile - Testing Agile Projects

COURSE DESCRIPTION:

This course teaches students how to implement a high level of software quality within the context of an agile development approach. Students will learn concepts and approaches useful for agile projects in their development of automated tests, code refactoring, and continuous integration. The course will prepare team members to understand the unique challenges and to employ best practices when addressing the testing aspect of agile development.

You will learn how to create and conduct functional testing within an iterative software development lifecycle.

TOPICS COVERED:

Upon completion of this course, the students will have a comprehensive understanding of:

- Test-driven development
- Incorporation of user stories' acceptance criteria into automated tests
- Utilization of a testing framework
- Software development using continuous integration
- Code refactoring
- How test-driven development impacts design

AUDIENCE:

Testers, test leads, developers, project managers, architects, and systems analysts.

DURATION:

1 day

PREREQUISITES:

"Developing Requirements with User Stories" or "Defining & Managing Requirements with Use Cases." Knowledge of object-oriented concepts and programming languages is helpful.

Portfolio Management - Iterative Project Management with the Unified Process

COURSE DESCRIPTION:

Iterative development using the Unified Process provides many competitive benefits: major risks are resolved early in development; requirement changes and new stakeholder requests (inevitable in any project) are accommodated smoothly; skill and resource utilization is maximized; progress and quality are continuously assessed. To obtain these advantages, the Project Manager must overcome many challenges: projects are started with incomplete and changeable requirements, documentation and artifacts evolve throughout the lifecycle; finalized sign-off of one stage does not occur before the next stage begins.

This course presents proven techniques and concepts, balanced with real-world labs and exercises, for planning and managing an iterative project. Project Managers will learn the key skills required to successfully plan and manage iterative projects. Techniques discussed have been developed and successfully applied on actual software projects.

TOPICS COVERED:

- Understand the concepts of the Unified Process and Iterative Development that impact software project management
- Gain practical experience in planning and tracking, resource management, risk management, and scope management for iterative development using the Unified Process
- Learn iterative project planning elements: phase planning, iteration planning, and resource management
- Communicate progress in an iterative project: set expectations of customers and stakeholders and demonstrate objectives are being met
- Estimate iterative projects

AUDIENCE:

Project Managers, Technical Leads and all others currently involved in, or soon to be responsible for, managing or coordinating iterative development projects.

DURATION:

2 - 3 days (*optional 3rd day includes additional exercises*)

PREREQUISITES:

"Introduction to the Unified Process" or comparable SDLC experience.

OBJECTIVES:

- Understand concepts of the Unified Process and Iterative Development that impact software project management
- Learn the effective implementation of process, tools, architecture, and reuse centers
- Gain practical experience in planning and tracking, project staffing, risk management, and scope management using the Unified Process.

Portfolio Management - Advanced Iterative Project Management

COURSE DESCRIPTION:

Iterative development using the Unified Process (UP) provides solutions to many problems and obstacles facing project managers. A PM must manage risk, incorporate change, manage resources, and ensure quality while meeting strict schedule and budget constraints. The UP engages a project manager, requiring intense involvement to plan, control and assess a project objectively. Project managers must adapt their management skills and techniques to succeed with the UP and avoid being overwhelmed with iterative development.

"Advanced Iterative Project Management" builds on topics covered in "Iterative Project Management with the Unified Process" providing the next steps for the PM. This course addresses the issues that a PM must face after the initial process is put in place. The course is a seminar providing in-depth discussion of advanced concepts, with real-world examples to illustrate the concepts, exercises to further understand them, and examination of students' projects to reinforce the application of these concepts.

TOPICS COVERED:

- Applying best practices to iterative project management
- Configuring iterative development to your needs and culture
- Managing the process for different types of projects
- Project manager's role in requirements, change and configuration management
- Metrics for iterative processes
- Enterprise view of the unified process

AUDIENCE:

This course is intended for project managers, managers of project management offices and others responsible for success of the SDLC in the organization.

DURATION:

2 days

PREREQUISITES:

- Project management experience
- "Iterative Project Management with the Unified Process"

OBJECTIVES:

- Provide the project managers with advanced tools to manage iterative projects
- Improve iterative project management skills
- Apply iterative management concepts and principles to current project



Portfolio Management - Service Oriented Architecture for Managers

COURSE DESCRIPTION:

Service Oriented Architecture (SOA) is a leading technology direction for more and more IT groups who are trying to create integrated and consistent solutions throughout their organization. Although many implementation details of SOA, including the supporting Web Services technologies, are well understood, there has been very little analysis of the impact of SOA on the management decision making process. Organizations most successful implementing SOA and Web Services will be those who understand the technology fundamentally, but more importantly, understand the new decision matrix that it creates and how these decisions ultimately impact long term planning. This course covers a range of topics from an understanding of the management issues presented by SOA, to a practical look at the details of SOA legacy enablement.

TOPICS COVERED:

- Understand the basic concepts of SOA and Web Services
- Understand the costs, risks and benefits of the SOA approach
- Understand how to model a Service Oriented Architecture for an organization integrating existing software solutions with new ones
- Understand the skills, practices, and processes related to creating service oriented architectures
- Understand why to use Web Services with SOA
- Understand issues related to SOA operations
- Understand the practical details of legacy enablement

AUDIENCE:

This course is designed for business and technical managers and architects who are interested in what SOA is all about and the practical implications of its implementation.

DURATION:

2 days

PREREQUISITES:

None



Process - Introduction to the Unified Process

COURSE DESCRIPTION:

The Unified Process is globally recognized as the standard development process for mission-critical software systems. Combining the ideas of the foremost thinkers in software process development, the Unified Process is designed to address the six best practices in modern software development. Based upon the notions of Use Case driven requirements, architecturally-centric development, and controlled iterative management, the Unified Process provides a framework for project success. This 2-day introduction explains the key Unified Process concepts including recommended lifecycle phases, core workflows, dependencies between activities, deliverables, and lessons learned from applying the Unified Process. This course ensures your team is able to succeed with formal software processes in the real world.

AUDIENCE:

Technical managers, project leads, analysts, designers, software engineers, configuration management personnel, testing and quality assurance personnel, and anyone needing an overview of the Unified Process will benefit from this course.

DURATION:

2 days

PREREQUISITES:

There are no prerequisites. No prior knowledge of these technologies is assumed.

OBJECTIVES:

- Understand the best practices for modern software engineering and the benefits they provide
- Understand the key concepts and terminology used within the Unified Process
- Describe the full software development lifecycle in terms of Phases and Workflows
- Understand the benefits of a Use Case driven, architecture-centric approach
- Describe an iterative, risk-driven approach to planning a software development project
- Understand the importance of supporting the process with automated tools



Process - Unified Process Overview

COURSE DESCRIPTION:

This course provides an overview of the Unified Process. The process uses an iterative and incremental approach to development. The Unified Process is use case-based and architecture centric. It leverages object-oriented and component-based development methods. The process promotes risk management and risk-based project planning, as well as formal requirements management practices. Many, if not all, of these concepts, methods and techniques are new to project staffs. This course provides a brief, but informative introduction to the Unified Process and how it ties all these elements together to create a modern software development process that enables project objectives.

TOPICS COVERED:

- The key structural elements of the Unified Process
- The goals and impacts of iterative, incremental development
- What use cases are and their benefits
- How object-oriented development differs from traditional methods
- How the process encourages a component-based architectural approach
- The project management implications of risk management, use cases, and iterative development

AUDIENCE:

Entire project staff

DURATION:

4 hours

PREREQUISITES:

None

Process - Extending the Unified Process for the Web

COURSE DESCRIPTION:

To meet business and user expectations of the Web, you need more than the Unified Process. You must integrate strategy, branding, marketing, usability, information architecture, creative design, and content with software engineering activities. Learning how these varied roles can productively work together is a significant challenge for many teams.

This course reveals our best practices for Web development and how they integrate with the Unified Process framework. You will learn about new roles, activities, and artifacts typically produced in Web projects and how they relate to use cases and other UML artifacts. The result is a process that smoothly integrates business strategy, user-centered design, and software engineering principles. You leave the course full of practical advice for enhancing the Unified Process and streamlining it for different types of Web development projects.

TOPICS COVERED:

- Modified objectives for the Inception, Elaboration, Construction, and Transition phases
- New roles such as Business Strategist, Usability Evaluator, Information Architect, and Content Manager
- Vital business strategy, user experience, and deployment activities
- Adapting the process to your project and organization

AUDIENCE:

Project managers, project leaders, systems analysts, information architects, business strategists, business analysts, architects, HCI specialists, content developers, software developers, architects, testers, and others who know the Unified Process and want to learn how to make it more useful for Web development projects.

DURATION:

2 days

PREREQUISITES:

Knowledge of, or experience applying, the Unified Process.

OBJECTIVES:

- Identify best practices for Web development
- Define new roles, activities, and artifacts used in Web projects
- Understand where the new roles, activities, and artifacts fit into the Unified Process
- Learn how to tailor the process for different types of projects

Requirements - Business Modeling Using UML

COURSE DESCRIPTION:

A business model is an abstraction of how a business functions. It models the goals, organization, processes, objects, and events of a business. A business model can tell us how a business functions today or how we plan for it to function in the future. We do business modeling to analyze a business and mold it to meet our business goals. Alternatively, we model pertinent areas of the business to inform a systems development project about the business processes, business objects, and business rules to be automated. This course teaches how to create business models using the Unified Modeling Language (UML).

TOPICS COVERED:

Upon completion of this course, students will be able to:

- Write a Business Vision document defining the business strategy and goals of the business.
- Model business processes and business rules using UML business use case diagrams and business use cases.
- Model workflow in business processes using UML activity diagrams.
- Model business objects using UML class diagrams.
- Model the states of business objects using UML state machine diagrams.

AUDIENCE:

This course is designed for the business process analysts and business designers who perform business modeling. This course is also of benefit to project managers, systems analysts, and other project roles that are recipients of the artifacts produced.

DURATION:

3 days

PREREQUISITES:

None

Students who take this course will also benefit from the following courses:

- **Requirements Elicitation and Facilitation:** teaches information elicitation and meeting facilitation techniques that are invaluable throughout business modeling activities.
- **Advanced Use Case Lab:** teaches advanced concepts and guidelines for writing high quality use cases. While this course is targeted at system use cases, most of the information applies equally well to business use cases.

Requirements - Defining & Managing Requirements with Use Cases

COURSE DESCRIPTION:

Requirements are the foundation for any software development project. How well you define requirements is the key to building software that users want to use because it meets their needs. Use Cases are an industry proven approach to capture end user requirements. This course explores the process of eliciting, analyzing, documenting, communicating, and managing business requirements and the requirements of a software system. Students will learn how to gather requirements using several techniques, create the Use Case Model, write Use Case Specifications, and manage changing requirements.

TOPICS COVERED:

- Putting requirements into perspective
- Defining the right problem
- Eliciting techniques
- Actors
- Use Case Model
- Writing a Use Case
- Non-Use Case requirements
- Scope and Changing Requirements

AUDIENCE:

This course is designed for business analysts, systems analysts, requirements analysts, managers, subject matter experts, users, team leads, architects and software developers who require knowledge of requirements management techniques and use cases.

DURATION:

2-3 days

PREREQUISITES:

None

OBJECTIVES:

- Define requirements management within the context of the Unified Process
- Understand different types of requirements and where Use Cases fit in
- Learn techniques for eliciting, capturing, and communicating requirements
- Understand and practice the process to develop use case models and use case specifications
- Understand how to manage changing requirements

Requirements - Elicitation & Facilitation

COURSE DESCRIPTION:

This course teaches students how to elicit, analyze, and communicate requirements for software development projects. Many elicitation techniques rely upon the collaborative work of individuals with diverse backgrounds. Students learn and apply facilitation skills and strategies to get the most out of collaborative requirements definition sessions. Labs allow students to practice and refine elicitation techniques and facilitation skills in groups of various sizes. All labs simulate typical and difficult software requirements situations and issues. Students will apply their acquired skills by facilitating sessions for developing a vision, use case modeling, and drafting and reviewing use case specifications. Students receive instructor and peer feedback throughout the course. This enhances students' listening skills and provides constructive feedback.

TOPICS COVERED:

- Understanding facilitation
- How to make a meeting work
- JAD sessions
- Team and group dynamics
- Effective communication skills
- Problem solving and decision making
- Facilitator techniques
- Role playing

AUDIENCE:

This course is designed for students who want to improve their requirements elicitation and facilitation skills, especially for conducting use case sessions. These roles include: Business analyst, systems analyst, project manager, team leader, and software architect

DURATION:

3 days

PREREQUISITES:

"Defining & Managing Requirements with Use Cases," "Introduction to Use Cases," or other project-related use case exposure (writing or reviewing use case artifacts).

OBJECTIVES:

- Provide students with the essential skills to effectively elicit and refine requirements
- Learn & apply facilitation skills for use case and other requirements sessions
- Assess and improve listening skills
- Increase awareness of what assists and inhibits clear communication
- Understand the essentials of effective meetings and group dynamics
- Select and plan different types of meetings used with a use case driven requirements process
- Design meeting agendas and select effective techniques to accomplish goals

- Select and apply elicitation techniques appropriate to the project's circumstances
- Practice facilitation techniques in mock use case sessions
- Accept constructive feedback
- Improve skills based upon feedback



Requirements - Advanced Use Case Lab

COURSE DESCRIPTION:

This course teaches techniques for writing high quality Use Cases. Student retention of these techniques is optimized as the course spends the majority of time in hands-on labs applying the techniques discussed.

TOPICS COVERED:

This in-depth, focused training class equips students with skills needed to be highly productive use case analysts. The course will increase team members' understanding of what a good use case is and how high quality use cases are developed quickly and effectively.

AUDIENCE:

Business Analysts, Requirements Analysts, System Analysts, Software Designers, Information Architects, Project Managers, Testers, and Quality Assurance team members will increase their understanding of what a good use case is and how high quality use cases are developed.

DURATION:

2-3 days

PREREQUISITES:

"Defining & Managing Requirements with Use Cases," "Introduction to Use Cases," or at least 2 months of hands-on, project experience writing use cases.

OBJECTIVES:

- Improve use case writing skills
- Learn how to handle common use case issues
- Learn how to identify and correct common mistakes in use cases
- Learn useful patterns for handling common types of requirements
- Apply these techniques by revising and writing use cases

Requirements - Effective Use Case Development

COURSE DESCRIPTION:

Without proper capture of requirements software projects have a high probability of failure. This 2-day course presents the use case technique that has become a norm for capturing functional requirements in object-oriented projects, yet can be used just as successfully with procedural development projects. Structured around a problem-driven approach, students will learn use case techniques and concepts in a "just-in-time" manner, emphasizing the semantics of each technique and not just its notation. Numerous examples, and two parallel case studies let students see how a concept is described in a use case model, and then immediately apply their understanding.

This course is designed to provide students with intensive, practical training in the concepts of requirements specification through use cases. During the course the students write at least three non-trivial use cases. At course completion, students will have a comprehensive understanding of use cases, how to write useful and effective use cases, and both their value and limitations in either procedural or object-oriented development life cycles.

TOPICS COVERED:

At the end of this course, students will be able to:

- Write effective use cases for gaining consensus on the desired behavior of a software system
- Recognize and understand the UML use case notation and semantics
- Describe the iterative process for developing use cases
- Describe the degree of detail needed to write successful use cases
- Recognize improper use case descriptions
- Understand how to apply use cases directly to software testing

AUDIENCE:

This course is designed for business analysts, systems analysts, requirements analysts, technical managers, and software developers who wish to learn techniques for capturing requirements for software system development.

DURATION:

2 days

PREREQUISITES:

Experience in requirements gathering or systems analysis is desirable, but not mandatory.

OBJECTIVES:

Course enables students in critical skills emphasized in the Business Analysis Body of Knowledge (BABOK) expressed as necessary for Business Analysts to sit for the International Institute of Business Analysis (IIBA) certification exam in order to be a Certified Business Analysis Professional or CBAP (TM).

Requirements - Business Analysis: Requirements Discovery and Facilitation

COURSE DESCRIPTION:

Acquiring skill for the role of business analyst (BA) in a modern software development organization is often left to “on-the-job” experience, or to chance. The BA is often a “middleman” between the customer and the IT organization who must produce solutions meeting the customer’s expectations. The BA role is complex and requires skill in multiple dimensions, including: harvesting of requirements, serving as a Subject Matter Expert, satisfying business strategy defined in multiple constituencies, and communicating business process goals and process details to technical groups.

This course is a highly interactive curriculum that focuses on the role of the BA within the technical process of software development. Through numerous examples, it enables BAs to work more effectively with the IT team by teaching how to express business requirements in forms that IT can directly use. In the course exercises, the BAs learn to identify the business entities in the business domain, how to express these concepts in both visual and textual means, and how to specify the semantic relationships among those entities.

TOPICS COVERED:

At the end of this course, students will be able to:

- Identify and organize various types of business requirements
- Explain the significance and purpose of core artifacts useful to the BA
- Express business requirements using diverse presentations including Unified Modeling Language artifacts that capture both static and dynamic views of those requirements
- Fulfill the role of mediator between the business and IT constituencies
- Ask the right questions to discover the project’s requirements

AUDIENCE:

This course is designed for business analysts, systems analysts, requirements analysts, technical managers, and software developers who wish to learn techniques for successful business analysis in software development.

DURATION:

2 days

PREREQUISITES:

Experience in requirements gathering or systems analysis is desirable, but not mandatory.

OBJECTIVES:

Course enables students in critical skills emphasized in the Business Analysis Body of Knowledge (BABOK) expressed as necessary for Business Analysts to sit for the International Institute of Business Analysis (IIBA) certification exam in order to be a Certified Business Analysis Professional or CBAP™.

Requirements - CBAP Certification Preparation Boot Camp



COURSE DESCRIPTION:

The CBAP certification only debuted in late 2006, but already there's tremendous interest in it. We'll start the CBAP Prep Boot Camp with an overview of what "Certified Business Analyst Professional" means—what it is and what it isn't. Under your instructor's guidance, you'll have a chance to discover some of the most important philosophies and aspects of the CBAP examination, setting the tone for this fast-paced and interactive learning experience. And to get you ready for the workshop to come, we'll give you a preview of the workshop content and our process for delivering a valuable learning experience.

TOPICS COVERED:

- Components of the CBAP certification
- Introduction to the CBAP examination
- Philosophy of the CBAP exam
- Overview of CBAP exam content
- Overview of the CBAP Boot Camp

AUDIENCE:

Business Analysts and all others interested in taking the IIBA(c) Certified Business Analysis Professional (CBAP) examination.

DURATION:

4 days.

PREREQUISITES:

Background in Business Analysis. IIBA endorsed training courses, "Effective Use Case Development" and "Business Analysis - Requirements Discovery and Facilitation".

Requirements - Testing Use Case-Driven Projects

COURSE DESCRIPTION:

This course teaches students how to leverage their projects' use cases, requirements and requirements artifacts to develop test plans, test cases, and test procedures to support iterative development. Students will learn the concepts and a practical step-by-step approach for developing test plans, test cases and test procedures. How to apply these concepts and techniques in an agile methodology approach is also discussed. Lab exercises, in which students practice the techniques, are a central part of the course.

This course focuses on functional testing; performance, load, stress, and other types of tests are also discussed. The class will not cover basic testing overview concepts, except as needed. Familiarity with the testing role and test activities is assumed.

TOPICS COVERED:

Upon completion of this course, the students will be able to:

- Develop an overall Test Plan for the project
- Develop an Iteration Test Plan
- Group and trace requirements by Use Cases
- Identify potential tests for a Use Case
- Define the Test Cases for a Use Case
- Develop a Test Procedure for a Test Case
- Organize the testing artifacts
- Track defects found during test execution
- Manage change control items, both defects and change requests, across iterations of the project.

AUDIENCE:

Test managers, testers, quality assurance, business analysts, project managers, developers, programmers, and/or subject matter experts responsible for testing or overseeing testing activities.

DURATION:

2 days

PREREQUISITES:

- Knowledge of, or experience with, use cases is highly recommended
- Familiarity with the testing role and test activities is highly recommended.

Requirements - Introduction to Use Cases

COURSE DESCRIPTION:

This fast-paced, introductory course explores the basic concepts needed to develop use case models and use case specifications for software requirements. You learn how use cases fit into the software development process and how they become the basis for subsequent work such as software design and test case development. Lectures and labs focus on developing, reading, and reviewing use case models and specifications. The need for additional artifacts (such as a business strategy, vision, business process models, supplementary specification or architectural requirements, user interface specification or wireframes, and prototypes) is briefly discussed, however details on the artifacts and related process are not provided in this introductory course.

TOPICS COVERED:

- Identifying and analyzing actors
- Fundamentals of developing a use case model
- Fundamentals of writing use case specifications
- Why teams choose use cases instead of traditional functional requirement specifications
- Relationship between use cases and user interfaces, test cases, and other Unified Modeling Language (UML) diagrams

AUDIENCE:

This course is intended for project managers, subject matter experts, executives, testers, business analysts, requirements analysts, systems analysts, software designers, architects, information architects, user interface designers, quality assurance engineers, and others who want to understand use cases.

DURATION:

1 day

PREREQUISITES:

None

OBJECTIVES:

- Learn the essential elements of use case models
- Describe the process for developing use case models and specifications
- Apply the techniques during labs
- Describe and practice the process of reviewing use cases
- Explain where use cases fit into the software development process

Analysis & Design - Introduction to Modeling with UML

COURSE DESCRIPTION:

This program offers a highly concentrated look at the major forces, technologies, processes, tools, risks, and benefits associated with adopting object and component technologies. An expert practitioner facilitates the program, bringing real-world experience in strategizing, architecting, designing, and delivering actual object-oriented systems. The curriculum is intentionally broad enough to introduce the major issues, while offering sufficient detail to be directly usable by emerging teams.

TOPICS COVERED:

- Understand how to identify and classify the objects in business problems
- Understand how to model business data, behavior, rules and constraints using object-oriented constructs
- Be introduced to the notation and semantics of class modeling and sequence diagramming
- Understand the basic concepts of object-oriented software architecture
- Define the processes used in analysis and design
- Be introduced to design patterns

AUDIENCE:

This course is designed for systems analysts, architects, designers, developers, and testers who are directly responsible for developing object-oriented systems. This course also benefits managers, technical leads and software quality assurance personnel who oversee development of UML-driven systems and require an understanding of the process and associated artifacts.

DURATION:

1-3 days

PREREQUISITES:

None

OBJECTIVES:

- Leveraging model-based development.
- Fundamentals of objects and components.
- Key techniques such as use case modeling and software analysis & design.
- Importance of methodology and process.
- Introduction to distributed architectures.
- Integrating legacy systems.
- Realistic benefits and common pitfalls.

Analysis & Design - Analysis & Design Using UML 2.0

COURSE DESCRIPTION:

The features, principles and techniques of object-oriented technology mitigate the complexities of modern software systems. Successful projects have learned that object-oriented programming is insufficient; that object-oriented analysis, architecture, and design are required for robust, scalable, maintainable web-based and conventional business systems, as well as embedded systems.

This course teaches the processes, techniques, and artifacts necessary for modern object-oriented analysis and design. Students will learn the chief diagrams, symbols, and concepts of the Unified Modeling Language (UML) v2.0, the de facto international standard for modeling and specifying software systems. UML 2.0 added composite structure, timing and interaction overview diagrams, and significantly enhanced class, component, sequence, and state machine diagrams. Students will learn through detailed lecture and hands-on labs the core competencies in object-oriented analysis and design through the use of UML 2.0 diagrams and semantics.

TOPICS COVERED:

- Understand how to identify and classify the objects in business problems and model their business data, behavior, rules and constraints
- Master UML 2.0 diagram using proven analysis and design methods
- Learn and apply key principles that facilitate repeatable, quality designs such as the Liskov Substitution Principle, the Law of Demeter, the Information Expert Principle, High Cohesion, Loose Coupling, and many others
- Understand the basic concepts of object-oriented software architecture including logical partitioning of systems into layers and subsystems, process and thread architecture, and hardware architecture modeling
- Learn what analysis, architecture, and design patterns are and apply them to improve designs
- Acquire hands-on experience in these methods and diagrams through case study exercises

AUDIENCE:

This course is designed for systems analysts, architects, designers, developers, and testers who develop object-oriented systems. Technical leads and software quality assurance personnel who oversee development of object-oriented systems will also find this course vital.

DURATION:

5 days (Analysis and Design can be taught separately, each as a 3 day class)

PREREQUISITES:

Prior software development experience.

Analysis & Design - Analysis Modeling with UML Workshop

COURSE DESCRIPTION:

Up to a 1 week tailored session with facilitator/mentor to guide participants through activities using the project's assets: requirements (use cases), decisions, models, and CASE tools. The workshop is customized to focus on a single project's needs. The workshop is structured to provide brief mentor presentations of concepts and techniques to refresh students' memories of prior training followed by hands-on modeling using client project assets.

Brief workshops build upon classroom training, turning learned concepts into actual skills. Longer workshops will produce project modeling assets that can serve as the project's initial development iteration analysis model.

TOPICS COVERED:

WORKSHOP INPUTS: The following artifacts from the project under development are preferred inputs:

- Use cases (preferred)
- Other requirements documentation
- Existing analysis and design models
- User interface designs

The following artifacts from legacy or related projects are also helpful:

- System requirements documentation
- Analysis and design models
- Data models
- User interface designs

WORKSHOP OUTPUTS: The scope of outputs depend on length of workshop and activities chosen by the client.

- Domain class model detailed with business attributes and relationships
- Class, attribute, and relationship specifications
- System sequence diagrams revealing system operations
- System operation contracts (specifications)
- State models for key domain classes

AUDIENCE:

System Analysts, Designers, Architects, Subject Matter Experts

DURATION:

2-5 days

PREREQUISITES:

"Analysis & Design with UML" or "Analysis with UML"

Analysis & Design - Design Modeling with UML Workshop

COURSE DESCRIPTION:

Up to a 1 week tailored session with facilitator/mentor to guide participants through activities using the project's assets: requirements (use cases), decisions, models, and CASE tools. The workshop is customized to focus on a single project's needs. The workshop is structured to provide brief mentor presentations of concepts and techniques to refresh student's memory of prior training followed by hands-on modeling using client project assets. Brief workshops build upon classroom training, turning learned concepts into actual skills. Longer workshops will produce project modeling assets that can serve as the project's initial development iteration design model.

TOPICS COVERED:

WORKSHOP INPUTS: The following artifacts from the project under development are preferred inputs. Similar artifacts from legacy or related projects are also helpful.

- Use cases (preferred)
- Other requirements documentation
- Business or analysis domain class model, system sequence diagrams, and system operation contracts
- Any existing other analysis, architecture, and design models
- Data models
- User interface designs

WORKSHOP OUTPUTS: The scope of outputs depends on length of workshop and activities chosen by the client.

- Component models at the enterprise or project-level with service-oriented interfaces
- Class models at the enterprise, project domain, and subsystem levels
- Component, class, service, and operation specifications
- Sequence diagrams demonstrating inter-class/component dynamics
- State models for key classes

AUDIENCE:

Designers, Architects, System Analysts

DURATION:

3-5 days (private, on-site workshop)

PREREQUISITES:

"Analysis & Design with UML" or "Design with UML" and "Service-Oriented Architecture & Web Services"



Analysis & Design -Agile Methodology Workshop

COURSE DESCRIPTION:

One to 4 week workshop tailored to guide an organization through activities exploring the various agile methodologies, experimenting with techniques, and building organizational support to ensure the new methodology is embraced. Completing all segments of the workshop will produce a clear plan for piloting and using an agile methodology in your organization.

TOPICS COVERED:

Typical workshop activities include:

- Assess the organizational context
- Develop methodology recommendations
- Define pilot project or implementation approach

AUDIENCE:

Entire IT organization.

DURATION:

1 to 4 weeks.

PREREQUISITES:

None

OBJECTIVES:

By attending this workshop your team will gain a solid understanding of agile modeling techniques, how they fit together, and how they may be applied simply and effectively by your teams.



Tools - Introduction to ClearCase

COURSE DESCRIPTION:

This class teaches the fundamental skills that streamline and simplify the software development process through the use of the ClearCase tool. Students learn concepts and specific usage of ClearCase that positively affect development teams and accelerate development cycles. This covers both Base ClearCase and Unified Change Management (UCM).

TOPICS COVERED:

- Work comfortably and unassisted with the ClearCase tool
- Understand all major functionality of ClearCase, and how to get help
- Have a basic understanding and an awareness of best practices in collaborative, parallel development environments
- Perform checkin/checkout, merging of files and directories, and navigate version trees
- Use command line interface and GUI tools
- Understand the major integrations with Commercial IDEs such as Visual Studio, VB, C++

AUDIENCE:

This course is intended for first-time ClearCase users requiring a practical introduction to the ClearCase commands and operations used every day.

DURATION:

2 days (private, on-site presentation only)

PREREQUISITES:

Familiarity with either Windows or UNIX operating systems. Basic understanding of software development lifecycle methods. Familiarity with version control concepts is beneficial, but not critical.

OBJECTIVES:

- Use of ClearCase in Projects
- Taking advantage of UCM
- How to create policy and structures for automated software development processes

Tools - Introduction to Rational SoDA

COURSE DESCRIPTION:

SoDA is Rational Software's document generation tool. It provides automated document generation with the ability to integrate Rose model diagrams, requirements, code, and text in the same document.

TOPICS COVERED:

- Documentation processes
- SoDA for Word product overview and concepts
- SoDA for Word commands and terminology
- Fundamentals of generating and regenerating SoDA for Word documents and reports

AUDIENCE:

This course has been designed for software engineers who may be technical leads, object modelers, requirements analysts, configuration managers, quality assurance personnel, technical writers, or developers who require a thorough understanding of SoDA's document generation capability.

DURATION:

1/2 day

PREREQUISITES:

Some knowledge of Rose and UML artifacts is preferred. Knowledge of Microsoft Word is required.

OBJECTIVES:

- Overview of documentation processes
- Define SoDA for Word terminology and functions
- Generate and edit SoDA for Word documents from existing SoDA templates
- Generate web pages
- Build SoDA for Word templates and extract information from information sources such as RequisitePro and Rose
- Create and customize SoDA Word templates
- Modify SoDA for Word commands within a template and from the Template View
- Create SoDA for Word tables



Tools - Introduction to RequisitePro

COURSE DESCRIPTION:

This class teaches how to capture and manage the changing requirements of a system using IBM Rational's RequisitePro tool. Hands-on exercises and activities are used to enhance the concepts presented in the "Defining & Managing Requirements with Use Cases" class. The student will gain a general understanding of how to use the RequisitePro tool to apply software development best practices for using requirement attributes and traceability to help manage project scope and assess the impact of the changing requirements of a project.

TOPICS COVERED:

- Define and create a project within RequisitePro.
- Understand how to capture and document project related functional requirements within RequisitePro.
- Understand how to set up a documentation hierarchy and standards for defining the different levels of requirements for a product using RequisitePro.
- Understand the architecture of a simple requirements management system, linking documents to a central requirement repository, which maintains requirement attributes and traceability as presented in RequisitePro.
- Understand how to use requirement attributes and traceability links in RequisitePro to help manage scope and manage change throughout the life cycle of the product.

AUDIENCE:

Technical, Quality Assurance, and documentation managers, project leads, executives, application experts, business analysts, salespersons and analysts/designers who are involved in requirements capture, specification, use or management.

DURATION:

1 day

PREREQUISITES:

"Defining & Managing Requirements with Use Cases" or hands-on project experience writing use cases.

Tools - Introduction to Rose

COURSE DESCRIPTION:

Rose is a visual modeling tool used to develop object-oriented designs. It supports all Unified Modeling Language (UML) diagram types. This course teaches students to use the tool to develop all supported diagrams types and to specify design details. Students also learn to use the tool's advanced features for forward and reverse engineering, web publishing, and script usage.

TOPICS COVERED:

- Rose Modeling Basics
- Create Use Case Model
- Create Interaction Diagrams
- Add Several Diagrams
- Using Rose in a Team
- Rose Scripts

AUDIENCE:

This course is intended for software engineers who may be technical leads, object modelers, requirements analysts or developers who require a thorough understanding of Rational Rose's visual modeling capability.

DURATION:

1+ Days

PREREQUISITES:

The training class, "Analysis & Design using UML," at least 3 months real-world modeling experience on a software development project, or solid understanding the Unified Modeling Language (UML) is required to successfully complete this course.

OBJECTIVES:

- Use the Rational Rose tool to develop several UML diagram types
- Enter design detail into model element specifications
- Generate code, XML DTDs, and database DDL from models
- Set up and use Rational Rose in a team with concurrent modeling
- Print diagrams and specifications
- Publish a Rose model to HTML suitable for publication on the corporate intranet
- Run Rose scripts that extend the capabilities of the Rose tool