the andrea group

“Mastering Agile” Course Series

This course series is based on a clear and practical definition of Agile: regular delivery of valuable, running, and tested features to the user community on a regular basis to maximize business return on investment and to respond to timely feedback with strategic product adaptation (see “The Agile Cheat Sheet” on page 2).

On the surface, Agile methods look familiar because they contain the standard set of practices. Teams frequently run into problems when they attempt to adopt an Agile method, and continue to do things the same old way.

In order to truly deliver running, tested features on a regular basis, no practice has been left unchanged. Everything - requirements, design, development, integration, testing, and review - must be condensed in order for an entire cycle to fit into a short, fixed sized iteration (typically two weeks). Physical distance between people and conceptual distance between roles must disappear so that interactions are timely, continuous, collaborative, and concrete. Development practices must be highly focused and disciplined so that the software is simultaneously stable and malleable.

The Mastering Agile course series is designed to introduce Agile concepts to teams, and further help teams master requirement specification and testing practices. All courses are highly interactive and collaborative, interleaving presentations with facilitated group reflections and discussions, and hands-on exercises to provide a deep and enduring learning experience.

When a course is delivered in-house for a single project team, we endeavour to tailor the standard course offerings to best meet the specific needs of the team.

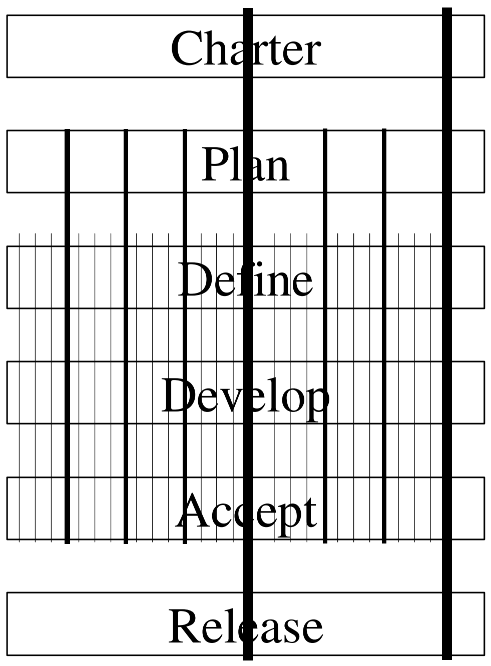
the andrea group

The "Agile Cheat Sheet"

Agile =

regular	• iterations, releases • time-boxed • incremental • regular heartbeat • streamlined collaboration • co-located team • on-site customer • face-face communication
delivery	• disciplined build process • frequent feedback • retrospective
of valuable	•prioritizing •business value •kano analysis
running,	• test driven development • testable design • continuous integration • refactoring
and tested	**Feature level (incremental)** • ATDD • TDD • Exploratory testing • Story acceptance testing **System level:** • End-end business acceptance • Integration testing • Non-functional testing
features.	•span plan •story-o-types •story •examples • incremental requirements

Agile Process Lifecycle



ATDD Workflow

	Author	**Developer**
Red	1. Create Example	
	2. Execute Example	
		3. Execute Example
		4. Create Unit Tests
		5. Execute Unit Tests
Green		6. Write System Code
		7. Execute Unit Tests
		8. Execute Example
	9. Execute Example	
Re-factor		10. Refactor System Code
		11. Execute Examples and unit tests

Foundations of Agile Software Development

Wondering what agile is all about? Skeptical that agile is the same old thing wrapped in new buzz words? Tried agile on a project with disappointing results? You need a solid foundation of what it really means to develop software using an agile approach. Regardless of the brand name, a successful agile project is easily recognized by it's outcomes: delivering valuable, running, tested features on a regular basis. During this full day course you will explore each component of this definition in detail, looking at how specific practices and role collaboration work synergistically to achieve the goal. This foundation enables you to create a personal roadmap of agile topics to pursue for mastering agile software development and for introducing agile to your organization. More importantly, your foundation is rock-solid because you will understand how project context influences decisions about selecting and adapting an agile process.

Foundations of Agile Software Development	
Objective	Introduce beginners to the fundamental concepts about agile software development. This is a foundation for the other courses.
Outline	• **Simulation**: Experience the difference between a Traditional and an Agile approach. • **Overview**: Agile means delivering valuable, running tested features on a regular basis. • **Regular**: Release planning; ripple effect of time-boxing. • **Valuable**: Prioritization and iteration planning; ripple effect of adapting to change. • **Features**: Agile requirements; ripple effect of incremental delivery. • **Tested**: Test-driven development; incremental exploratory and non-functional testing. • **Delivery**: Disciplined development practices. • **Project Context**: Context elements mapped to process decisions (Examples: Cinderella project, distributed team, complex domain, legacy system).
Level	Beginner level.
Audience	All project roles.

Mastering Agile Requirements

An agile software development approach is designed to accommodate changing project priorities and unstable or emerging requirements with minimal impact to budget or schedule. An agile requirements process further aims to minimize the effort specifying requirements without introducing risk or waste.

This delicate balancing act is all the more challenging when the process does not fit the team perfectly. A critical success factor in adopting an agile process is recognizing that every project has different needs, goals, and constraints; one size of requirements process does not fit all agile projects. You will master agile requirements by learning how to tune any requirements process to fit your unique set of project characteristics.

Rather than documenting the details of all of the functional requirements up front, agile requirements are developed incrementally: just enough, just in time, story by story. Through on-going collaboration with subject matter experts, the development team specifies functional requirements as a suite of functional tests. You will master agile requirements by learning how to develop effective user stories and examples that are readable, unambiguous, sufficient, and locatable.

Whether you are new to agile or have worked on several agile projects already, the Mastering Agile Requirements curriculum will help you dramatically improve the efficiency of your requirements process and the effectiveness of your automated examples.

Agile Requirements: Principles, Process, and Practices (1 day, or 2 days)	
Objective	Provide a solid foundation of agile requirements values and principles and build practical experience with the core practices.
Outline	• Foundations: values and principles that govern an agile requirements process • Iterative and incremental requirements specification: use workflow models for generating user stories, and developing a well-organized 'big picture' of the target system. Techniques to guide release and iteration planning. • Advanced topics for improving the effectiveness of agile requirements specifications through domain specific languages, and automated examples. • Optional (day 2): Additional exercises and full simulations provide accelerated learning through direct experience.
Level	Intermediate level; a basic understanding of what agile is about is required. Beginners should consider taking "Introduction to Agile Software Development" first.
Audience	Team members responsible for gathering, documenting, analyzing, or managing functional requirements (SMEs, testers, business/system analysts, and developers).

the andrea group

Requirements Process Adaptation (1/2 day, or 1 day)	
Objective	This course is designed specifically for teams that want to improve the effectiveness and efficiency of their requirements process.
Outline	• Foundations: deep understanding of the components of a requirements process; recognizing 'process smells' as opportunities for improvement. • Strategies: context driven approach to recognizing and eliminating waste, and improving the agility of any requirements process. • Examples: case studies that show how to adapt the agile requirements process for: complex domain, distributed team, legacy systems, framework development, and request for proposal process. • Optional (1/2 day): Review of the team's current context and requirements process and suggest areas for improvement.
Level	Intermediate - advanced level; prior hands-on experience on an agile project is recommended. Consider taking "Mastering Agile Requirements: Principles, Processes, and Practices" if there is no prior formal training in agile requirements.
Audience	Team members responsible for gathering, documenting, analyzing, or managing functional requirements (SMEs, testers, business/system analysts, and developers). It is also recommended for agile coaches and retrospective facilitators, to enable them to guide process adaptation.

the andrea group

Mastering Agile Testing

In an Agile approach, quality is intensely proactive and grounded in business value, involving the entire team from story inception to production release. The Mastering Testing courses are designed to help the analyst and tester understand how their roles have changed, and develop the skills needed to collaborate in the continuous quality assurance of an evolving system.

Agile Testing for Analysts (2 days)	
Objective	Designed for analysts that do not have much prior experience in testing.
Outline	• Trace the lifecycle of an agile project from story to iteration to release, and identify the type of testing that is performed at each stage and where their role fits in. • Think like a tester – hands on exercises to become a more creative and effective tester. • DONE: introduce concept of Test Driven Development; specifying acceptance criteria up front. • DONE-DONE: incremental functional story testing • DONE-DONE-DONE: workflow and integrated system testing
Level	Beginner (may want to include 'Introduction to Agile' module first)
Audience	Team members responsible for specifying and accepting the functionality of a system developed in an Agile fashion (business/system analysts, SME, etc).

Agile Testing for Testers (2 days)	
Objective	Designed for testers that would like a clear understanding of how to manage and perform incremental testing on an agile project.
Outline	• Trace the lifecycle of an agile project from story to iteration to release, and identify the type of testing that is performed at each stage, and where their role fits in. • DONE: introduce concept of Test Driven Development, and how it affects all roles, and in particular the traditional testing role. • DONE-DONE: incremental exploratory and non-functional story testing; test environment strategies; test data strategies; defect management • DONE-DONE-DONE: milestone and release testing; incremental non-functional testing • Continual process improvement: retrospectives; mining defects for process smells; process tuning; resource planning to keep the pipeline running smoothly
Level	Beginner (may want to include 'Introduction to Agile' module first)
Audience	Team members responsible for quality management (test leads and testers).

Foundations of Test Driven Development (1 day, or 2 days)	
Objective	Provide a solid foundation of the principles and practices of automated test driven development. The approach and strategies presented are tool-independent, enabling the practitioner to better evaluate and master tools.
Outline	• Beyond red-green-refactor: provide the BIG picture of TDD as it affects requirements, design for testability, iteration/release testing and the handoffs throughout the life of a system. • Design for testability: key patterns and strategies to ensure the system is designed for automated and manual testability • Case studies: review difficult-to-test situations (legacy port, legacy upgrade, thick client, etc). • Tools: how to choose a tool for automating examples; survey of current tool options. • Optional (1 day): A full day lab provides hands-on experience with the full FTDD lifecycle (iterative development, team etiquette, DSL, functional tests, unit tests)
Level	Intermediate – advanced.
Audience	Team members responsible for specifying and automating examples (testers, business/system analysts, and developers). Programming experience is not required.

Instructor

Since graduating from the University of Calgary in 1988 with a B.Sc. (distinction) in Computer Science, Jennitta Andrea's professional career has consistently focused on software processes, specification languages, and automated testing.

In 1988, Jennitta joined the process research group at MPR Teltech, where she developed domain specific languages and automated tools to support both structured and object-oriented software processes.

In 1994, Jennitta joined Object Systems Group (later renamed ClearStream), as an object-oriented practitioner and mentor. She quickly mastered the subject matter, and began designing and teaching advanced courses in business modeling and requirement specificaiton using UML.

Jennitta worked on her first agile project in 2000, and has continued to be a multi-faceted, hands-on practitioner (analyst, tester, developer, manager) and coach on a wide variety of different types of agile projects ever since.

Jennitta has always been a keen observer of teams and processes, and has published many experience-based papers for conferences and software journals. Jennitta delivers practical, simulation-based tutorials and in-house training covering: agile requirements, process adaptation, automated examples, and project retrospectives.

Today, Jennitta's work has culminated in international recognition as a thought leader in the area of agile requirements and automated examples. She is very active in the agile community: serving a third term on the Agile Alliance Board of Directors, director of the Agile Alliance Functional Test Tool Program to advance the state of the art of automated functional test tools, member of the Advisory Board of IEEE Software, and member on many conference committees.

Jennitta founded The Andrea Group in 2007. She remains actively engaged on agile projects as a hands-on practitioner and coach, and continues to bridge theory and practice in her writing and teaching.

Materials Provided

Student manual containing the course slides and published articles for each module. Student handouts with class exercises.

Price

This course is offered periodically as a public course, and is also available on-site. Please contact The Andrea Group for availability and pricing.

Contact

For more information on this or other courses offered by The Andrea Group, please contact us at:

Phone: 4 0 3 . 8 6 1 . 4 7 9 8
Email: info <at> theandreagroup <dot> ca
Web: www <dot> theandreagroup <dot> ca