

# Transitioning from Waterfall to Agile

---

<b>Code:</b>	AGILE-WF
<b>Length:</b>	2 days
<b>URL:</b>	<a href="#">View Online</a>

---

## Plan Your Agile Transition Strategy

After considering all of the ways in which the Agile methods will affect your organization and considering the benefits and challenges associated with each, you are ready to plan your Agile adoption strategy. Which steps should you take first? Who will need to be involved? How will you capitalize on the benefits you will experience? In this course you will gain all the tools, skills and knowledge to return to your office and successfully implement an Agile transition strategy that best fits your environment.

## Easy to Implement Training

Each concept and technique covered in this course is discussed not only in theory, but will also cover how the information works in practice. Each section will encompass the most important information that you need to know in order to be prepared to put this knowledge to use, and then will be followed with hands-on exercises to demonstrate precisely how to implement it. Each of the sections will also include a discussion around best practices for transitioning and implementation to help participants avoid the pitfalls commonly encountered when making the change to an Agile approach.

## Where Agile Methods Can Help the Organization

There are significant benefits available when utilizing an Agile approach that can address the risks, unknowns, and uncertainties that affect nearly all software development projects. These complexities can best be addressed with a flexible and adaptable model that turns traditional problems into advantages and provides the tools to change the way work is done through addressing organizational issues head on. In this course you will experience, through hands-on exercises, how Agile addresses these traditional project challenges and finally resolve these ever-present constraints. Learn to overcome these hurdles and interweave your traditional practices with Agile practices to develop the best software for your organization and your customer.

## Skills Gained

- Understand the key differences between a waterfall and an Agile approach to software development, then identify the areas you will benefit from most
- Identify and eliminate the traditional practices that undermine your project success
- Learn the 5 measures of Agile enablers then map implementing them to Agile practices to project success
- Uncover the organizational problems that most companies never discover they have, then learn the Agile techniques that address these deficiencies
- Discover how transitioning to Agile provides better techniques to manage the value and quality of your project and product development efforts
- Create a customized, hybrid approach to software development that takes into account your company's unique challenges and constraints
- Uncover the pitfalls that teams will encounter in an Agile transition and understand how to overcome those challenges

- Lay the foundation upon which you can build a learning team and organization

## Who Can Benefit

This course is valuable for anyone who is contemplating making their software projects more agile.

- Software Development Manager
- Software Project Manager
- Software Team Lead
- Quality Assurance Specialist
- Process Engineer
- Software Developer or Tester
- Software Project Customer
- IT Director or Manager

## Course Details

### Introduction to Agility

Agility is comprised of a unique set of principles and values that must be understood and embraced before the organization can employ Agile practices. In this section, we will survey the essence of Agility.

- Exercise: Identify the problems that you are experiencing with software projects in your organization. You will use this list throughout the course to evaluate the benefits of Agility. Compare notes with other students.
- The Essence of Agility
- The Agile Lifecycle
- Learning and Adaptation
- Collaboration
- Customer Focus
- Self-Directed Teams
- Lean Principles
- Progressive Requirements Elaboration
- Incremental Delivery
- Iterative Planning and Adaptation
- Exercise: Identify the Agile concepts that have already gained acceptance in your organization. Compare notes with other students.

### Us vs. Them Teams

The heart of every project is the team. Yet in traditional approaches, the team often operates as a collection of competing individuals, rather than as a unit. Even worse, many organizations have divided responsibilities in ways that pit different groups against each other, when the project desperately needs cooperation. Agility means establishing a single team for each project, and ensuring that it comprises all of the necessary people. Agile practices ensure that all team members are constantly collaborating and driving the project toward success.

- Traditional practices

- The project manager
- Shielding developers & customers from each other
- Building silos of responsibility
- Documents as the primary means of communication
- Lessons Learned at project end
- Contrast w/the Agile approach
- Product Owner
- The Agile coach
- One small co-located team
- Continuous collaboration
- Face-to-face communication
- Regular feedback
- Regular retrospectives

## **My Project Plan, Right or Wrong**

Traditional approaches are often referred to as “plan-based” because of the importance they put on up-front planning and then controlling the project so it conforms to the plan. Of course, some organizations go to the other extreme, paying little attention to their plans after the project starts, or even foregoing planning altogether. Agility means planning just enough, doing that planning when it is needed, and accepting the fact that reality often works out differently from our plans. Agile projects value achieving the project goals; the plan is merely a tool to help them do that.

- Traditional practices
- Predictive planning
- Command and control management
- Corrective action (Conformance to plan)
- Periodic Status Reporting
- Document review/sign-off as milestones
- Contrast w/the Agile approach
- Time-boxed projects & iterations
- Incremental planning & estimation
- Self-directed teams
- Daily status checks & periodic review
- Adaptation to new information
- Information Radiators
- Delivered software as milestones

## **The Insidious Creeping Scope**

Traditional approaches usually begin with an important (and sometimes long and drawn out) Requirements phase during which all of the product requirements are elicited, analyzed and documented. Everyone in the project commits to the resulting specification, and then significant effort is expended in Change Control. Conformance to the Requirements Specification becomes the measure of project success. (Until the customer complains!) Agility means documenting requirements just enough, eliciting more detailed information when it is needed, and accepting the fact that things will change before the project is over. Agile projects value delivering what the customer needs; the requirements are merely a tool to help them do that.

- Traditional practices
- Big-bang, up-front requirements definition
- Requirements sign-off
- Little customer involvement after requirements
- Restrictive change control
- Conformance to specification
- Product delivered/accepted at project end
- Contrast w/the Agile approach
- Progressive requirements elaboration
- Prioritization by the customer
- Incremental product acceptance & feedback
- Welcome changing requirements

## Where's the Quality?

Traditional projects are often quality-challenged. The testing phase at the end of the project seems to be never-ending, and in spite of all that time and effort, a defective product is delivered. This results in high support costs, unhappy customers and out of control costs. Agility means keeping the focus on quality from the very beginning of the project, testing continuously and ensuring that every piece of code is technically excellent. And because testing is not saved for the end, quality surprises are eliminated. Agility means producing production-ready software regularly throughout the project!

- Traditional practices
- Little focus on developer verification
- Separation of responsibilities (testers vs. developers)
- Testers responsible for quality
- Testing postponed to the project end
- Contrast w/the Agile approach
- Developers focused on technical excellence
- Developer/tester/customer collaboration/feedback
- Testers as members of the development team
- Early and regular "System" & "Acceptance" testing
- Each product increment production-quality

## Course Wrap-Up

There is too much to Agility for you to adopt all at once, so you will need an action plan. In this section, you will review what we have covered in the course and prepare a viable action plan for your organization to become more Agile.

- Review the Essence of Agility
- Prioritize the Agile concepts that you could introduce in your organization. For the three highest-priority concepts, create an action plan to make those things a reality on your projects. Compare notes with other students.

---

## Schedule (as of 3 )

Date	Location	
Jun 24, 2024 - Jun 26, 2024	Virtual	<a href="#">Enroll</a>
Sep 4, 2024 - Sep 6, 2024	Virtual	<a href="#">Enroll</a>
Dec 16, 2024 - Dec 18, 2024	Virtual	<a href="#">Enroll</a>

---

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy  
Get Your Free Copy Now