

# LECTURE 14

## Software requirements Engineering

Requirements engineering for agile methods

**Instructor:**


Dr. Natalia Chaudhry,

Assistant Professor, PUCIT, University of the Punjab, Lahore.

## **Reading:**

Chapter 20 Agile projects

Software Requirements, Wiegers K. & Beatty J., 3rd Ed. Microsoft Press, 2013

- 
- Agile development refers to a set of software development methods that encourage continuous collaboration among stakeholders and rapid and frequent delivery of small increments of useful functionality.
  - There are many different types of agile methods; some of the most popular are Scrum, Extreme Programming, Lean Software Development, Feature-Driven Development, and Kanban.

- Organizations often think of a waterfall development process as involving a linear sequence of activities, where project teams fully specify (and sometimes overspecify) the requirements, then create designs, then write code, and finally test the solution.
- Large projects that use a **waterfall approach** are often *delivered late, lack necessary features, and fail to meet users' expectations*. Waterfall projects are susceptible to this kind of failure because of the layers of dependency built upon the requirements.

- Agile development methods attempt to address some limitations of the waterfall model.
- Agile methods focus on iterative and incremental development, breaking the development of software into short cycles called iterations (or, in the agile method known as Scrum, “sprints”).
- Iterations can be as short as one week or as long as a month.
- During each iteration, the development team adds a small set of functionality based on priorities established by the customer, tests it to make sure it works properly, and validates it with acceptance criteria established by the customer.



# **Essential aspects of an agile approach to requirements**

## Customer involvement

- The main difference between the two approaches (waterfall vs agile) is in the timing of the customer involvement.
- On waterfall projects, customers typically dedicate considerable time up front, helping the BA understand, document, and validate requirements.
- However, during the construction phase, there is generally little customer involvement, which makes it difficult for a project to adapt to changing customer needs.
- On agile projects, customers (or a product owner who represents them) are engaged continuously throughout the project.
- customers work with the project team to identify and prioritize user stories that will serve as the preliminary road map for the development of the product.
- customers must be available during iterations to provide input and clarification during the design and construction activities.
- They should also test and provide feedback on the newly developed features when the construction phase of the iteration is complete.

## Documentation detail

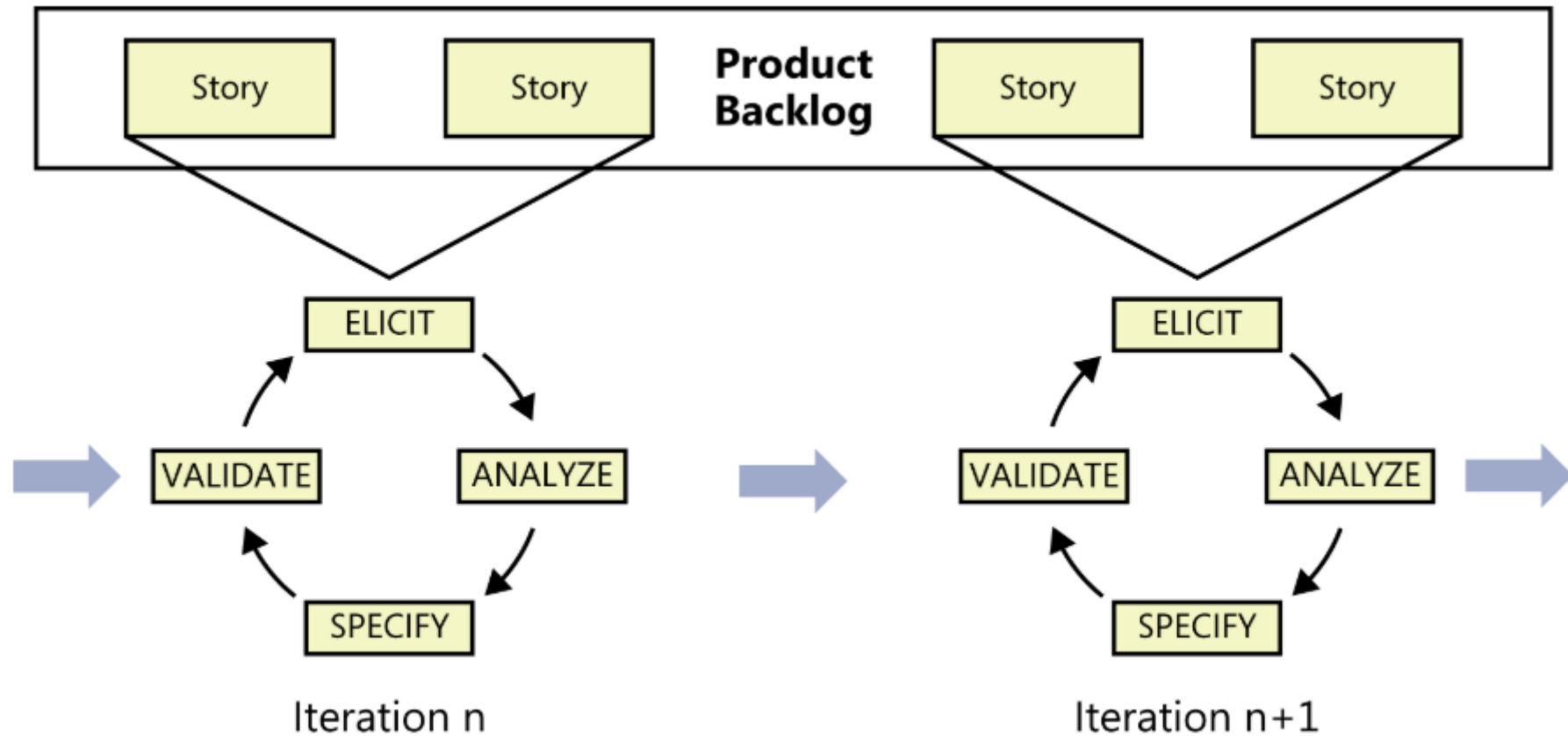
- Because developers have little interaction with customers after construction begins on waterfall projects, the requirements must specify system behavior, data relationships, and user experience expectations in considerable detail.
- The close collaboration of customers with developers on agile projects generally means that requirements can be documented in less detail than on traditional projects.
- People sometimes think that agile project teams are not supposed to write requirements. That is not accurate. Instead, agile methods encourage creating the minimum amount of documentation needed to accurately guide the developers and testers.



## Timing

- Agile projects require fundamentally the same types of requirements activities as traditional development projects
- detailed requirements are not documented all at once at the beginning of an agile project.
- Instead, high-level requirements, typically in the form of user stories, are elicited to populate a product backlog early in a project for planning and prioritization.

## Standard requirements activities occur within each agile iteration



## User stories

- User stories are sized so as to be fully implementable in a single iteration.
- An epic as being a user story that is too large to fully implement in a single iteration.
- Because epics span iterations, they must be split into sets of smaller stories.
- Sometimes epics are large enough that they must be subdivided into multiple epics, each of which is then split into multiple stories until each resulting story can be reliably estimated and then implemented and tested within a single iteration

## Expect change

- The biggest adaptation that BAs need to make when a requirement change arises on an agile project is to say not, “Wait, that’s out of scope” or “We need to go through a formal process to incorporate that change,” but rather, “Okay, let’s talk about the change.”
- This encourages customer collaboration to create or change user stories and prioritize each change request against everything else that’s already in the backlog.
- As with all projects, agile project teams need to manage changes thoughtfully to reduce their negative impact, but they anticipate and even embrace the reality of change.



# **Agile usage and tools**

## Jira:

- **Best for:** Software development teams.
- **Features:** Customizable Kanban and Scrum boards, sprint planning, burndown charts, advanced reporting, and extensive integrations with other tools like Bitbucket and GitHub.
- **Pros:** Strong community support, robust features tailored for Agile practices.
- **Cons:** Steep learning curve, especially for new users.

**Jama Connect:** This tool is excellent for complex and large-scale projects, offering features like real-time collaboration, traceability, and risk management. It supports agile methodologies by allowing teams to manage requirements iteratively and incrementally

**IBM Engineering Requirements Management DOORS Next:** Ideal for large enterprises, DOORS Next excels in scalability and comprehensive traceability. It integrates well with other IBM tools and supports agile practices by providing real-time collaboration and advanced configuration management.

**Modern Requirements4DevOps:** Integrated with Azure DevOps, this tool provides a seamless experience for managing requirements within a DevOps pipeline. It supports agile methodologies by offering features such as version control, traceability, and collaboration tools that are essential for continuous development and integration