CS5002NI

Software Engineering Week 24

Agendas

- Agile Software Development
- Values of Agile
- Principles of Agile
- Stages of Agile
- Different methodologies under Agile

What is Agile?

- Agile is the ability to create and respond to change
- It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment

What is Agile?

- The authors of the Agile Manifesto chose “Agile” as the label for this whole idea because that word represented the adaptiveness and response to change which was so important to their approach

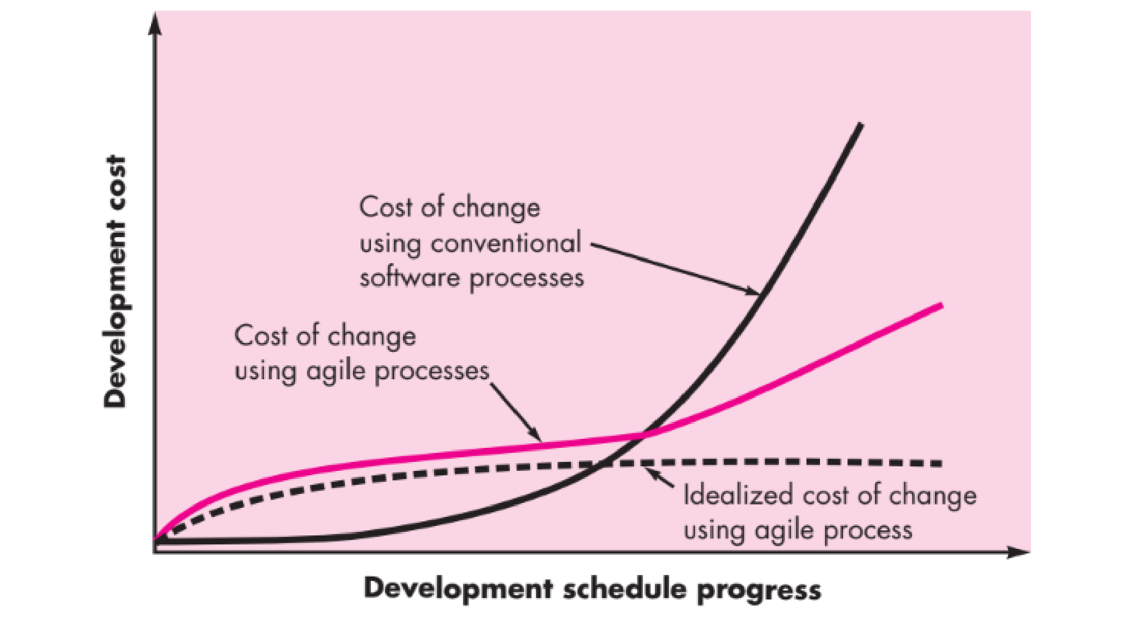
- It’s really about thinking through how you can understand what’s going on in the environment that you’re in today, identify what uncertainty you’re facing, and figure out how you can adapt to that as you go along.

What is Agile?

- The authors of the Agile Manifesto chose “Agile” as the label for this whole idea because that word represented the adaptiveness and response to change which was so important to their approach

- It’s really about thinking through how you can understand what’s going on in the environment that you’re in today, identify what uncertainty you’re facing, and figure out how you can adapt to that as you go along.

Agility and the Cost of Change



Agile Software Development

- Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches.

- Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.

- Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Agile Software Development

- The Agile methodology was created by a group of software developers who wanted a better approach to the traditional development process, which they found to be too complicated and weighed down by documentation requirements.

- In a founding document called the Agile Manifesto, the group outlined 4 values and 12 principles that guide the Agile philosophy

4 Values of Agile

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

12 Principles of Agile

1. Satisfy the customer through early and continuous delivery of valuable software.
2. Welcome and harness changes for the customer's competitive advantage, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for shorter timescales.
4. Have daily collaboration between business people and developers throughout the project.

12 Principles of Agile

5. Build projects around motivated individuals. Create the environment and support developers need, and trust them to get the job done.
6. Prioritize face-to-face conversation as the most efficient and effective method of conveying information to and within a development team.
7. Measure progress by the amount of working software completed.
8. Maintain a constant and sustainable pace of development indefinitely.

12 Principles of Agile

9. Enhance agility through continuous attention to technical excellence and good design.
10. Keep it simple. Simplicity—the art of maximizing the amount of work not done—is essential.
11. Recognize that the best architectures, requirements, and designs emerge from self-organizing teams.
12. Regularly reflect and adapt behavior for continual improvement.

Benefits of Agile

- Greater stakeholders engagement and collaboration
    - Encourages high degree of input and collaboration between the client and development team
    - Leads to happier clients due to transparency throughout the process and developers are better informed on client needs and wants

Benefits of Agile

- Predictable costs and scheduling
  - By breaking down the development process into iterative sprints, project managers can more accurately estimate costs and set clear, predictable timelines.
  - This makes stakeholders happier because they know what to expect and can plan budgets and marketing strategies more precisely.
  - It also makes the development process easier for teams because they can focus on delivering quickly and reliably and test software regularly for quality and efficacy.

Benefits of Agile

- Flexibility amidst change
  - Agile project management is all about being nimble so teams can adapt to changes quickly while reducing sunk costs.
  - Agile allows teams to pivot due to changing client needs, shifts in market demands, or in response to evolving product requirements.
  - This gives teams the flexibility to refine and prioritize the product backlog so that they are always delivering high-quality, relevant products on time and on budget.

Benefits of Agile

- High quality products
  - Agile product development integrates regular testing into the development process.
  - This makes it easier for the product owner to identify any issues early on and make changes as needed.
  - The result is higher quality products that are relevant and thoroughly vetted.

6 Stages of Agile Lifecycle

- Concept
- Inception
- Iteration
  - Requirements
  - Development
  - Testing
  - Delivery
  - Feedback
- Release
- Production
- Retirement

Agile Methodologies

- Scrum
- Kanban
- Extreme Programming
- DSDM

Scrum

- Scrum is an Agile framework that focuses on cross-functional teamwork, accountability, and iteration in order to develop, deliver, and support complex products.

- It’s primarily used for software development, but its principles can be applied to other project management teams too.

Scrum

- The Scrum framework is organized into key roles, events, and artifacts:
  - Scrum Roles
    - Product Owner
    - Scrum Master
    - Scrum Development Team

Scrum
Sprint 1
Sprint 2
Sprint 3
Spr
Several
incremental
releases
called
Sprints
Potentially Shippable Product

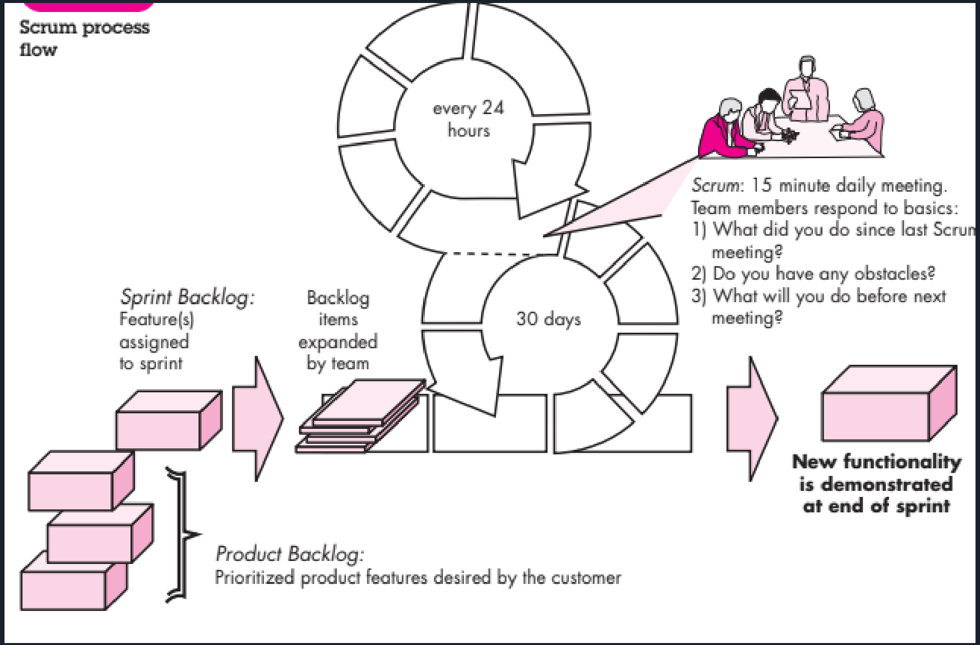
Scrum

- Scrum Events
  - Daily Scrum
  - Sprint Planning Meeting
  - Sprint Review
  - Sprint Retrospective

Scrum

- Scrum Artifacts
    - Product Backlog
    - Sprint Backlog
    - Increment (or Sprint Goal)

Scrum Process Flow



Scrum process flow
every 24 hours
Scrum: 15 minute daily meeting.
Team members respond to basics:
1) What did you do since last Scrum meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog: Feature(s) assigned to sprint
Backlog items expanded by team
30 days
New functionality is demonstrated at end of sprint
Product Backlog: Prioritized product features desired by the customer

Kanban

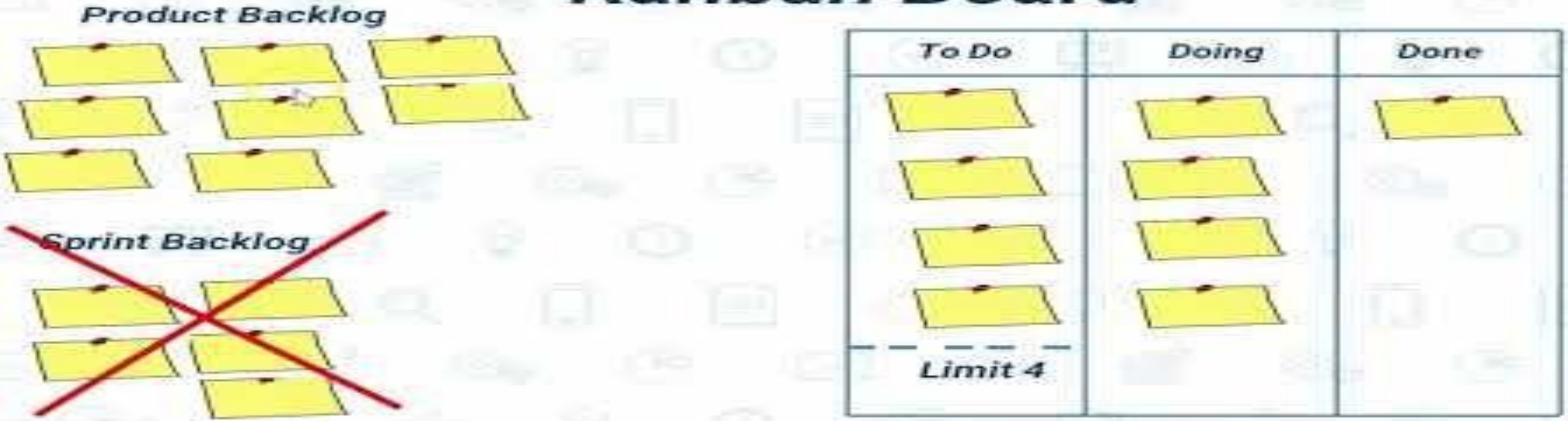
- Kanban is an agile model designed to help teams work together more effectively.

- It follows three guiding principles:
  - Visualize your workflow.
  - Limit the amount of work in progress.
  - Organize the workflow based on priority.

KANBAN IN 2 MINUTES

DEVELOPMENT THAT PAYS

Kanban Board



Product Backlog

~~Sprint Backlog~~

To Do

Doing

Done

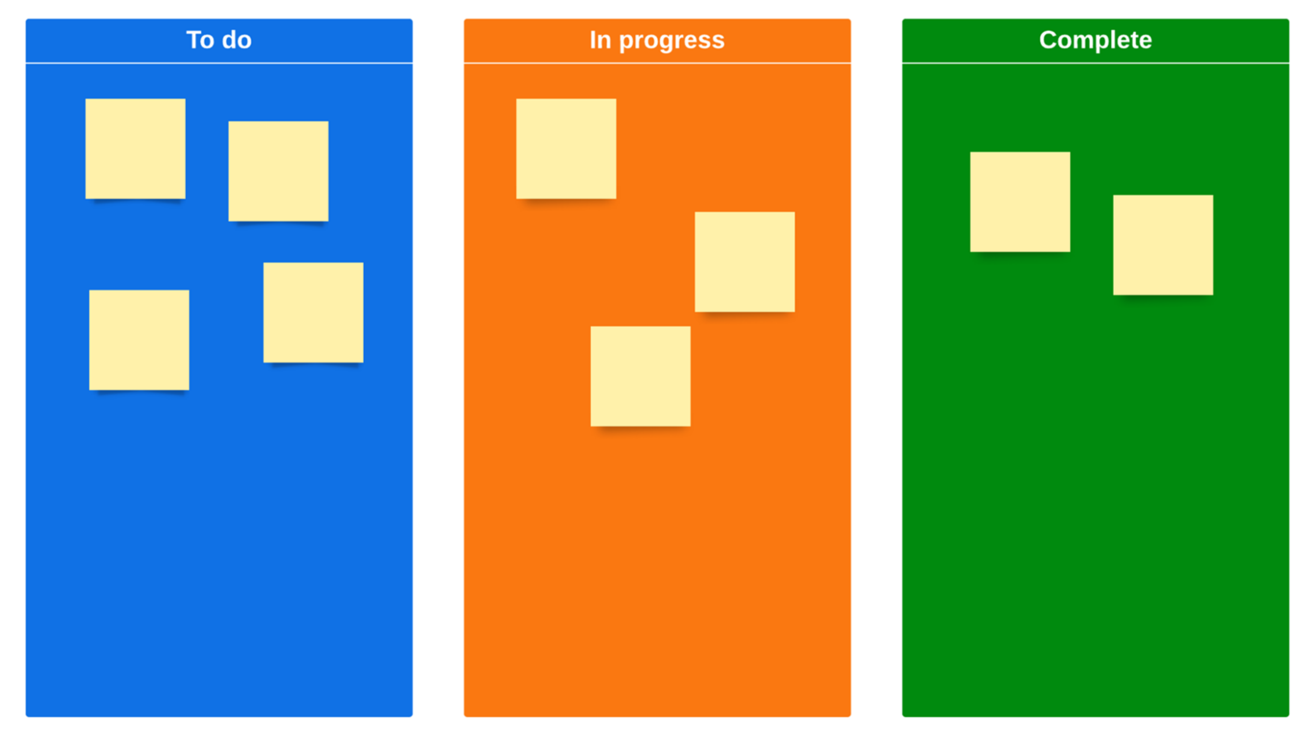
Limit 4

Kanban

- Unlike Scrum, Kanban doesn’t have prescribed roles or timeboxed sprints.

- Instead, Kanban focuses on shorter cycles for faster delivery and transparency throughout development so everyone understands who is responsible for what and when.

- Tools like an online Kanban board give team members the chance to contribute ideas, change the status of tasks, and track their progress so everyone works more efficiently and effectively together.



To do

In progress

Complete

Extreme Programming (XP)

- XP is the most specific Agile framework for software development practices.

- It aims to not only produce high-quality software but to make the entire process easier on the development team itself.

- XP values communication, feedback, simplicity, courage, and respect.

Extreme Programming (XP)

- It’s best applied when:
  - There are constantly changing requirements
  - Teams have tight deadlines
  - Stakeholders want to reduce risk under deadlines
  - Teams can automate unit and functional tests

Extreme Programming (XP)

■The most widely used agile process, originally proposed by Kent Beck

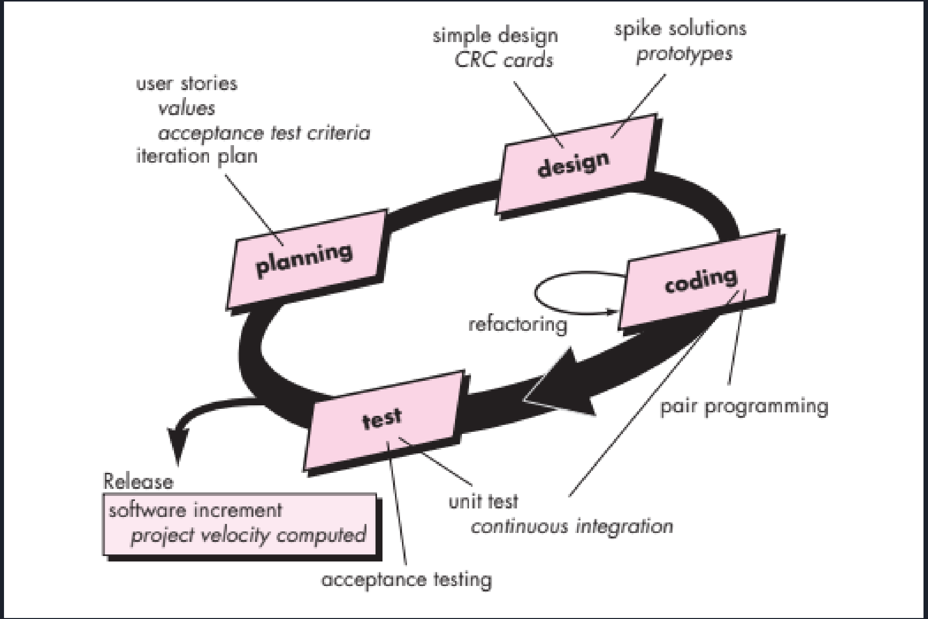
■XP Planning

- Begins with the creation of “user stories”
- Agile team assesses each story and assigns a cost
- Stories are grouped to form a deliverable increment
- A commitment is made on delivery date
- After the first increment “project velocity” is used to help define subsequent delivery dates for other increments

Extreme Programming (XP)

- XP Design
  - Follows the KIS principle (Keep It Simple)
  - Encourage the use of Class Responsibility Cards CRC cards
  - For difficult design problems, suggests the creation of “spike solutions”—a design prototype
  - Encourages “refactoring”—an iterative refinement of the internal program design
- XP Coding
  - Recommends the construction of a unit test for a store *before* coding commences
  - Encourages “pair programming”
- XP Testing
  - All unit tests are executed daily
  - “Acceptance tests” are defined by the customer and executed to assess customer visible functionality

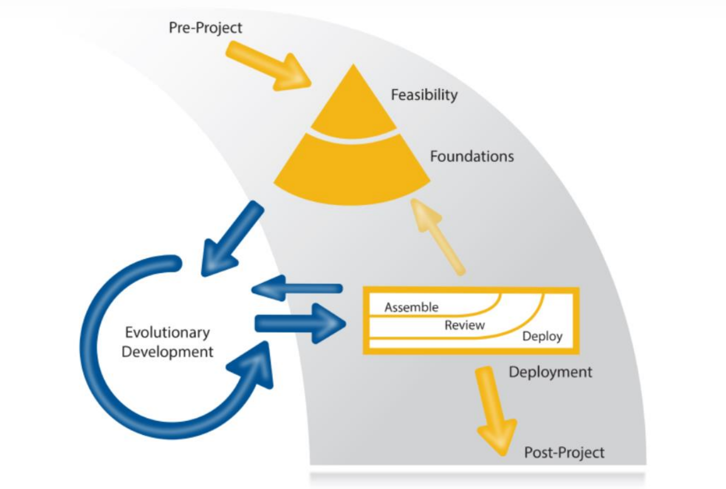
The Extreme Programming Process



Dynamic System Development Method

- (DSDM) is an agile software development approach that “provides a framework for building and maintaining systems which meet tight time constraints through the use of incremental prototyping in a controlled project environment”.

- The DSDM philosophy is borrowed from a modified version of the Pareto principle — 80 percent of an application can be delivered in 20 percent of the time it would take to deliver the complete (100 percent) application.



Dynamic System Development Method

- Promoted by the DSDM Consortium.
- DSDM — distinguishing features
  - Similar in most respects to XP and/or ASD(**Adaptive software development**)
  - Nine guiding principles
    - Active user involvement is imperative.
    - DSDM teams must be empowered to make decisions.
    - The focus is on frequent delivery of products.
    - Fitness for business purpose is the essential criterion for acceptance of deliverables
    - Iterative and incremental development is necessary to converge on an accurate business solutions.
    - All changes during development are reversible.
    - Requirements are baselined at a high level.
    - Testing is integrated throughout the life-cycle.

THANK YOU!