Chapter 3

- **Agile Development**

The Manifesto for Agile Software Development

"We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

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

Kent Beck et al

What is "Agility"?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding …

- Rapid, incremental delivery of software

An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adjusts as changes occur

Agility Principles - I

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes binds change for the customer's reasonable advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face–to–face conversation.

Agility Principles - II

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. (work together)
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self–organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Human Factors

- *the process molds to the needs of the people and team,* not the other way around
- key qualities must exist among the people on an agile team and the team itself:
  - **Competence. (capability)**
  - **Common focus.**
  - **Collaboration.**
  - **Decision-making ability.**
  - **Fuzzy (uncertain) problem-solving ability.**
  - **Mutual trust and respect.**
  - **Self-organization.**

Extreme Programming (XP)

- The most widely used agile process, originally proposed by Kent Beck
- Value
  - **Communication** between s/w engineers and stakeholder
  - Need to design only immediate need with intent to create **simple** design that can be easily implemented in code
  - **Feedback** from implemented s/w, customer and other s/w team member
  - **Courage** to design for today with consideration that it may change in future
  - **Respect** among its team member

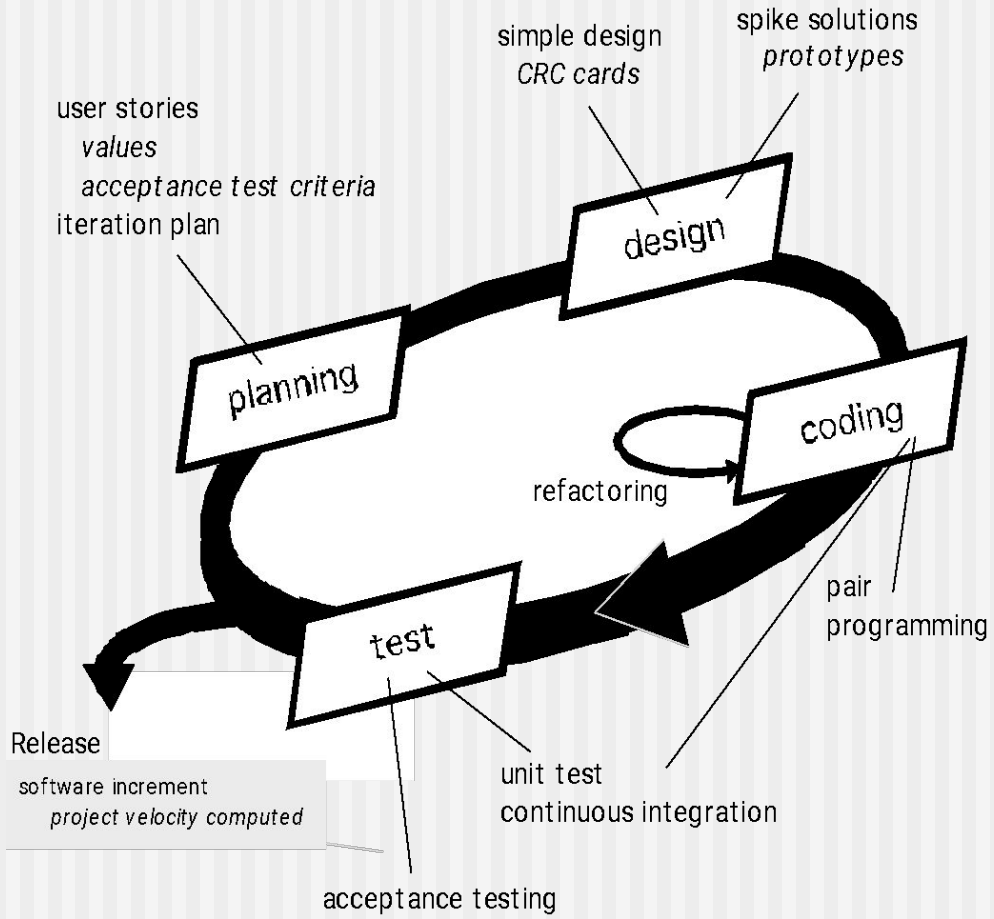
Extreme Programming (XP)

- XP Planning
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for 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 Keep It Simple principle
  - Encourage the use of CRC cards (class responsibility collabrator)
  - 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 excuted to assess customer visible functionality

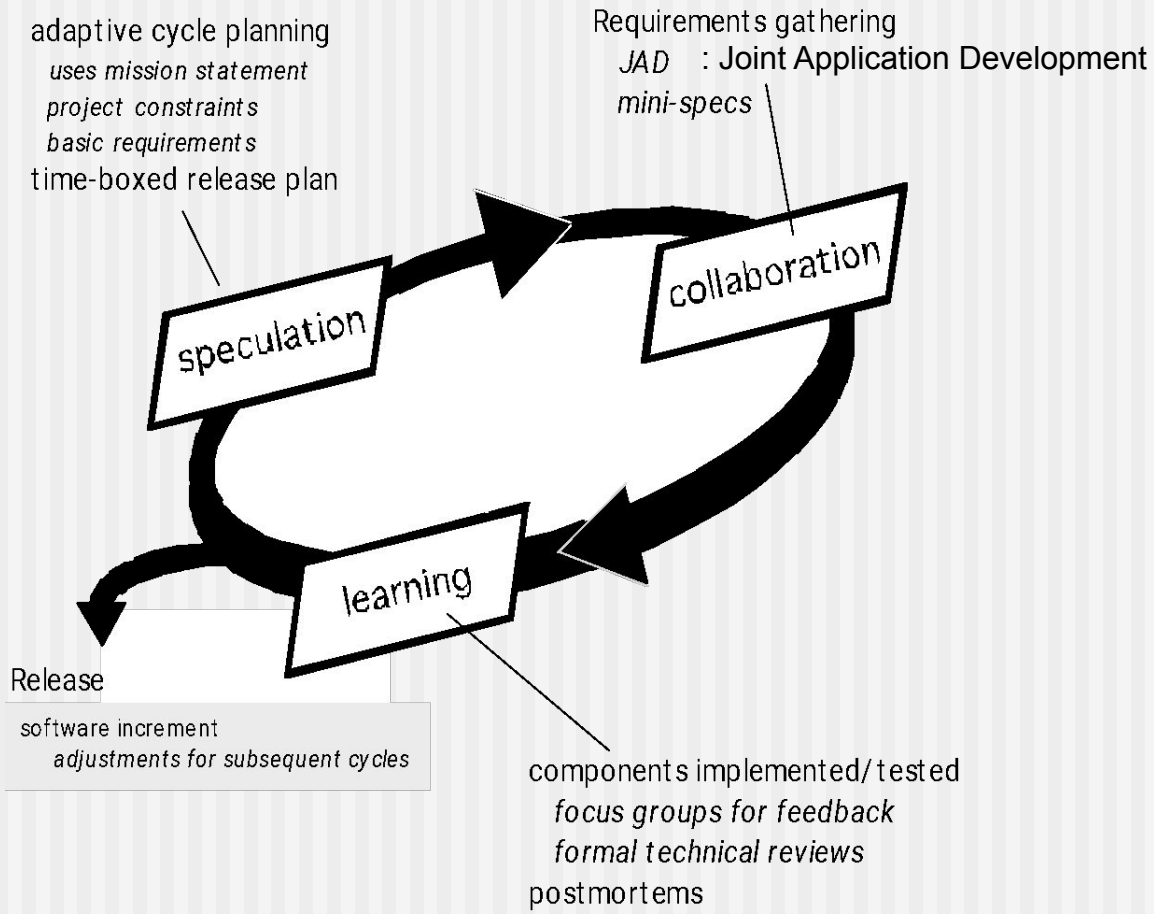
Extreme Programming (XP)



Adaptive Software Development

- Originally proposed by Jim Highsmith
- ASD — distinguishing features
  - Mission-driven planning
  - Component-based focus
  - Uses "time-boxing" (See Chapter 24)
  - Explicit consideration of risks
  - Emphasizes collaboration for requirements gathering
  - Emphasizes "learning" throughout the process

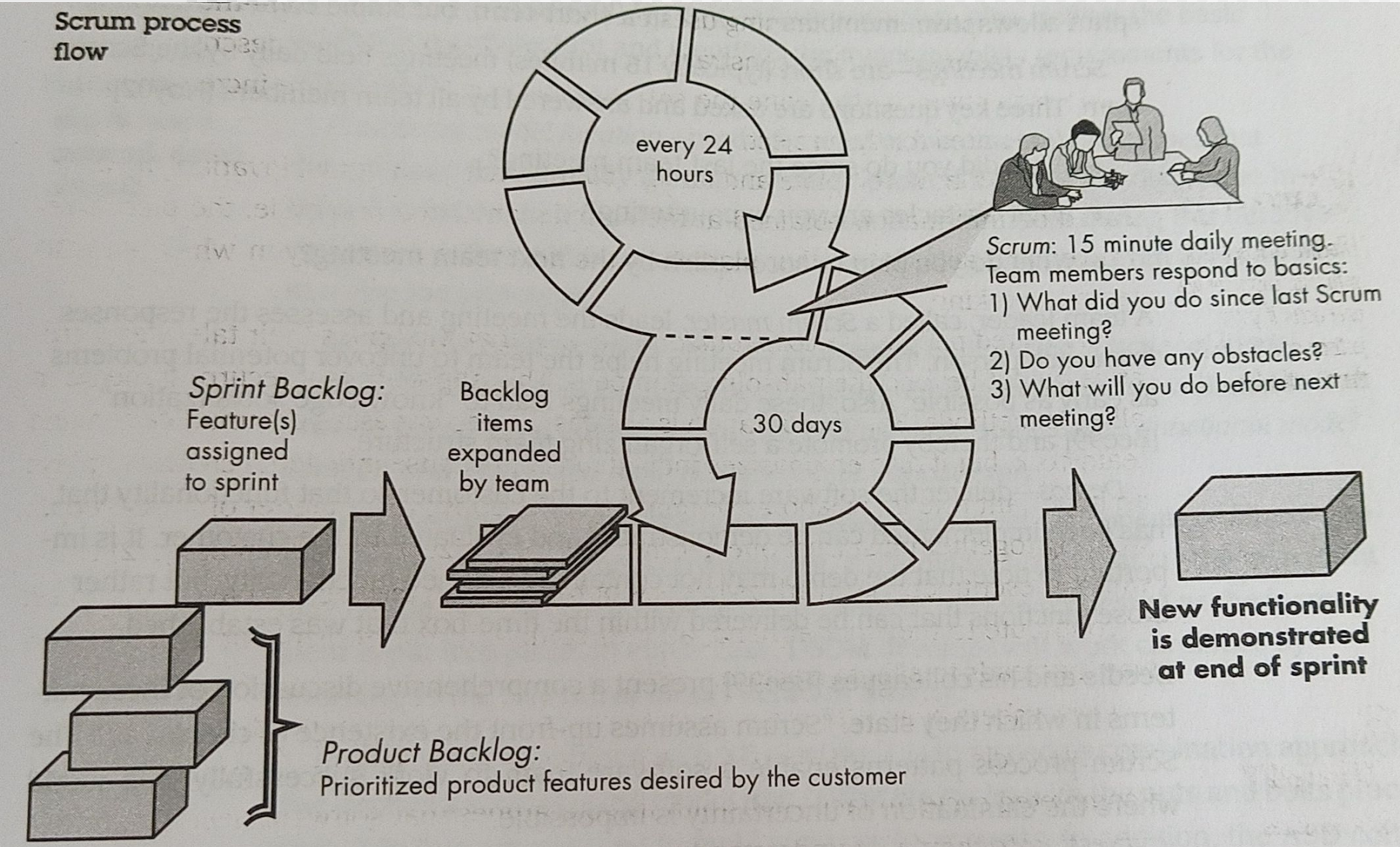
Adaptive Software Development



Scrum

- Originally proposed by Schwaber and Beedle
- Scrum—distinguishing features
  - Development work is partitioned into "packets"
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in "sprints" (consist of work units that are required to achieve in predefine time) and is derived from a "backlog"(a prioritized list) of existing requirements , changes are not allow during sprint
  - Meetings are very short (15 min. daily) leads by scrum master, and sometimes conducted without chairs,
  - "demos" are delivered to the customer with the time-box allocated

Scrum process flow



- Scrum delivers the software after each sprint

In short, Scrum is an Agile process aimed at delivering business value in the shortest possible time, whereas Agile is a continuous iteration of development and testing.

- Scrum is a more rigid method with less flexibility for change, and it’s ideal for those who need to produce results as quickly as possible. Agile is more suited for smaller teams and for those who prefer a more straightforward design and execution, while Scrum is used more for creative and experimental approaches.
- It's best to look at it this way: Scrum is always Agile, but Agile is not always Scrum. This means Scrum will encompass the same methodologies of Agile, but Agile may not share some of the same qualities as Scrum.

Excursive :

1. What is Agility ? Explain in brief the manifesto for Agile software development.
2. Explain the Agile process and key quality of the people in an agile team in brief.
3. Explain Extreme Programming with figure
4. Explain Adaptive Software Development.
5. Explain Scrum process flow with figure.