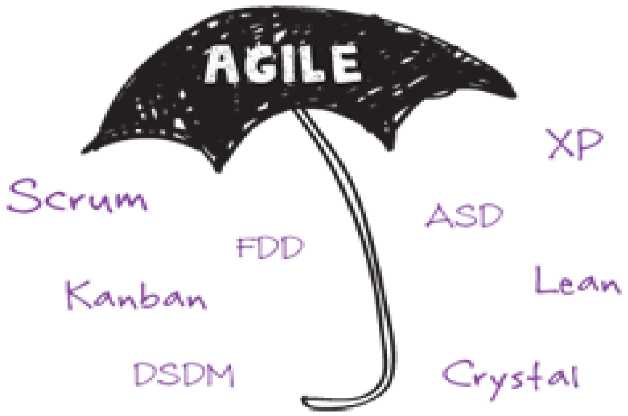


Week-3 : Lesson-1 Agile Model

Abdus Sattar
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University
Email: abdus.cse@diu.edu.bd

Topics Covered

Learning Goals

- ❑ Understand the rationale for agile software development methods, the agile manifesto, and the differences between agile and plan driven development.
- ❑ Know the key practices in extreme programming and how these relate to the general principles of agile methods.
- ❑ Understand the Scrum approach to agile project management.

What is “Agility”?

- Ability to move quickly and easily.
- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Rapid, incremental delivery of software

12 Agility Principles

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 harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to 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.

12 Agility Principles(Cont..)

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.
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.

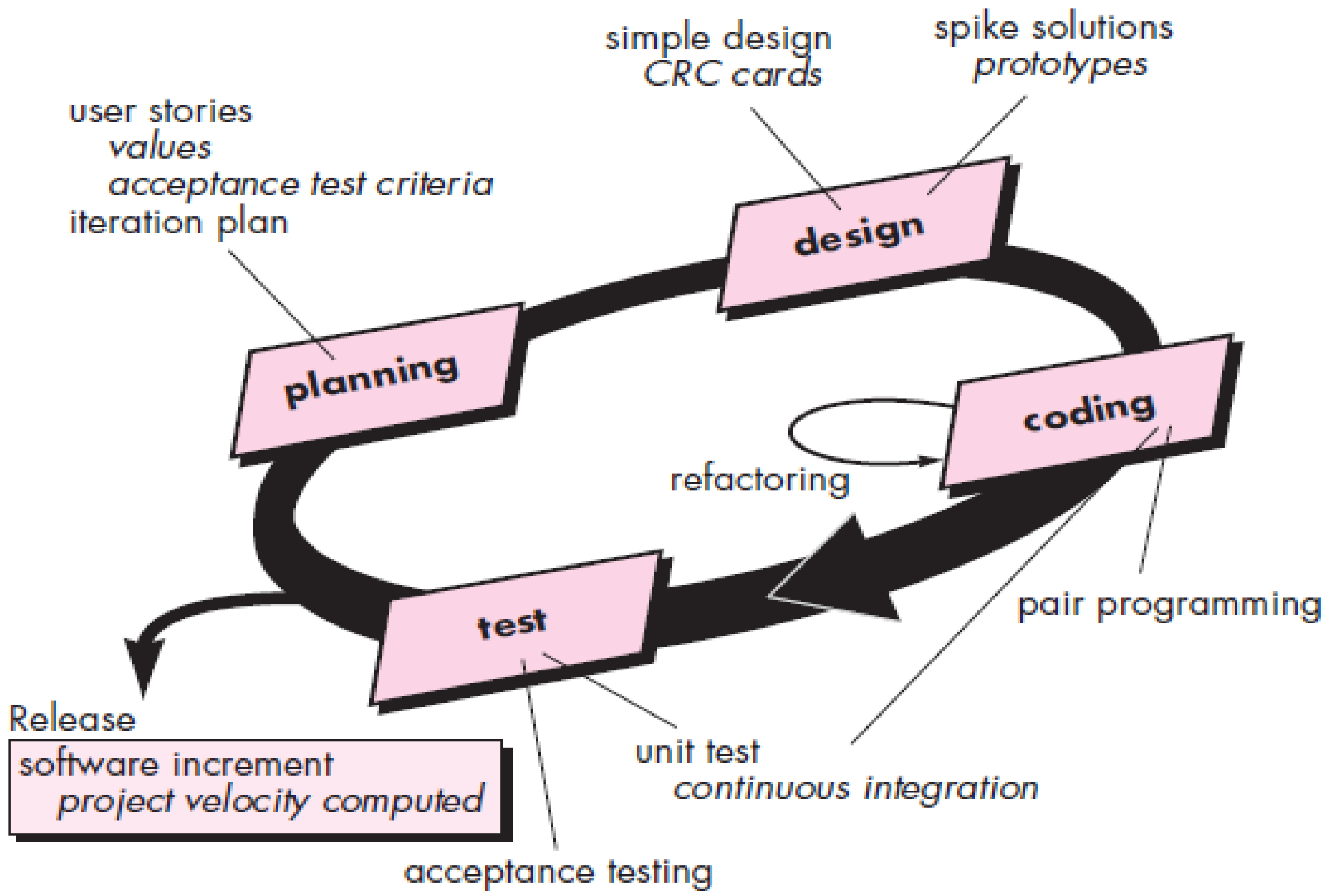
Agility Methodology

- Scrum
- Crystal Methodologies
- DSDM (Dynamic Software Development Method)
- Feature driven development (FDD)
- Lean software development
- Extreme Programming (XP)

Extreme Programming(XP)

- XP uses an object oriented approach.
- XP encompasses a set of rules and practice that occur within the context of four framework activities.
    - XP Planning
    - XP Design
    - XP Coding
    - XP Testing

Extreme Programming(XP)



Extreme Programming(XP)

- **There are four basic activities that XP proposes for software development process :**

1. **XP Planning :**
   - The planning activity begins with the creating of a set of stories that describe required features and functionally for software to be built.
   - Each stories is written by the customer and placed on an index card.
   - The customer assign a value to the story on the overall business value of the feature or function.
   - ¨Members of the XP team the access each story and assign a cost measured in development week to it.
2. **XP Design:**
   - XP design follows the KIS-Keep it simple principle.
   - A simple design is always preferred over a more complex representation.
   - The design provides implementation giddiness for a story as it is written nothing less, nothing more.
   - The XP team conducts the design exercise using a process and the CRC cards are the only design work product produced as the part of XP process.
   - XP recommends the immediate creating of an operational prototype of that portion of the design called spike solution.

Extreme Programming(XP)

3. XP Coding :

- XP recommends that after stories are developed and preliminary design is done, the team should not move to code, but rather develop a series of unit test.
- Once the code is complete, it can be unit tested immediately, thereby providing instaneous feedback to the developers.
- During the coding activity is pair programming.
- XP recommends that two people work together at one computer work station to create code for a story. This provides a mechanism for real time problem solving and real time quality assurance.

4. XP Testing:

- The creation of unit test before coding commence is a key element of the XP approach.
- The unit test that are created should be implemented using a framework the enable them to be automated.
- Integration and validation testing of the system can occur a daily basis.
- XP acceptance test, also called customer test are specified by the customer and focus on overall system.

Kanban Model

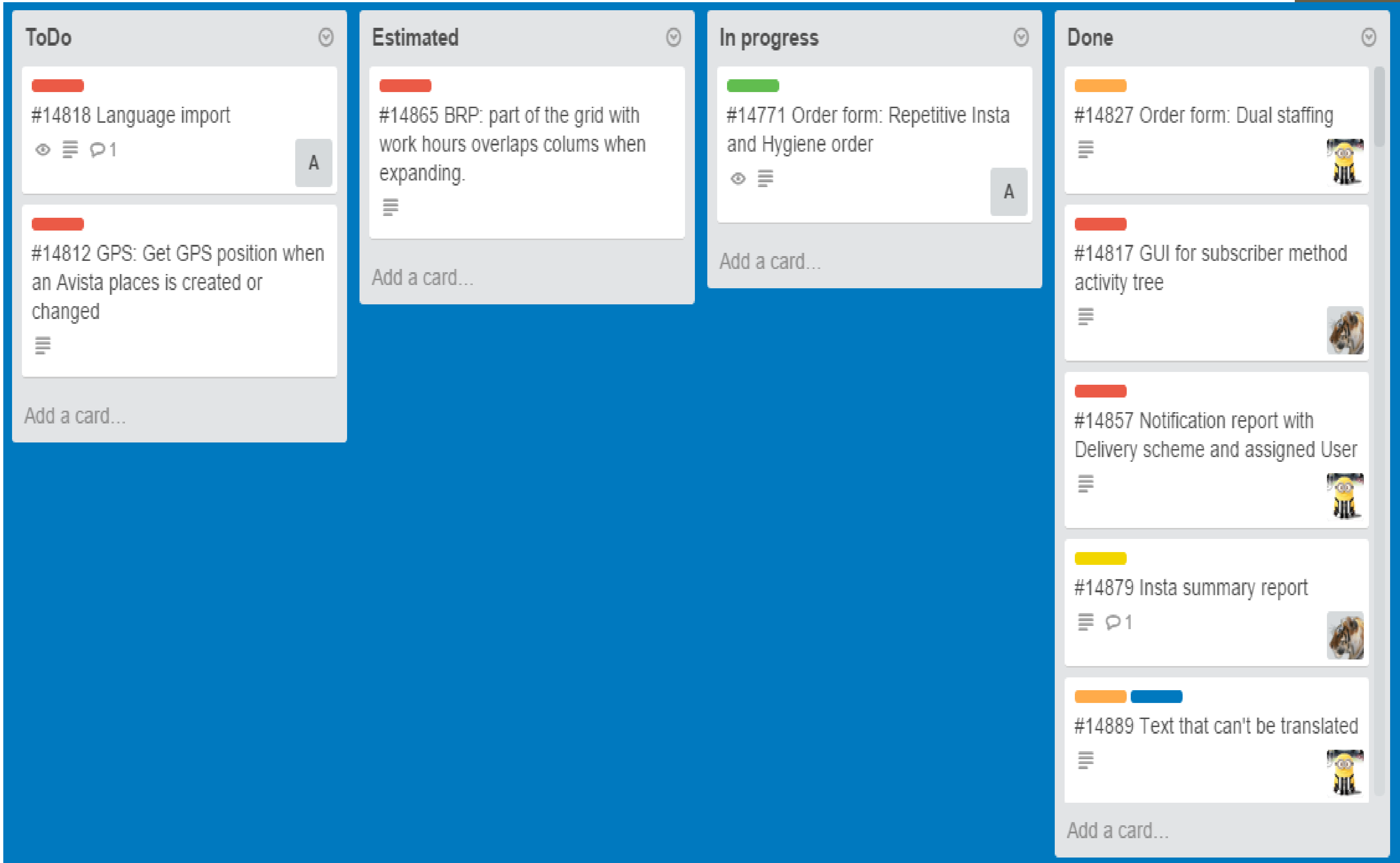
- Kanban is a visual system for managing work as it moves through a process.
- Kanban visualizes both the process (the workflow) and the actual work passing through that process.
- The goal of Kanban is to identify potential bottlenecks in your process and fix them so work can flow through it cost-effectively at an optimal speed or throughput.

The Three Principles of Kanban Development

Three core principles allow you to use Kanban in your project:

- **Visualize what you do today (workflow):** seeing all the items in context of each other can be very informative
- **Limit the amount of work in progress (WIP):** this helps balance the flow-based approach so teams don€™ start and commit to too much work at once
- **Enhance flow:** when something is finished, the next highest thing from the backlog is pulled into play

Kanban Board Example



ToDo

#14818 Language import

1

A

#14812 GPS: Get GPS position when an Avista places is created or changed

Add a card...

Estimated

#14865 BRP: part of the grid with work hours overlaps colums when expanding.

Add a card...

In progress

#14771 Order form: Repetitive Insta and Hygiene order

A

Add a card...

Done

#14827 Order form: Dual staffing

#14817 GUI for subscriber method activity tree

#14857 Notification report with Delivery scheme and assigned User

#14879 Insta summary report

1

#14889 Text that can't be translated

Add a card...

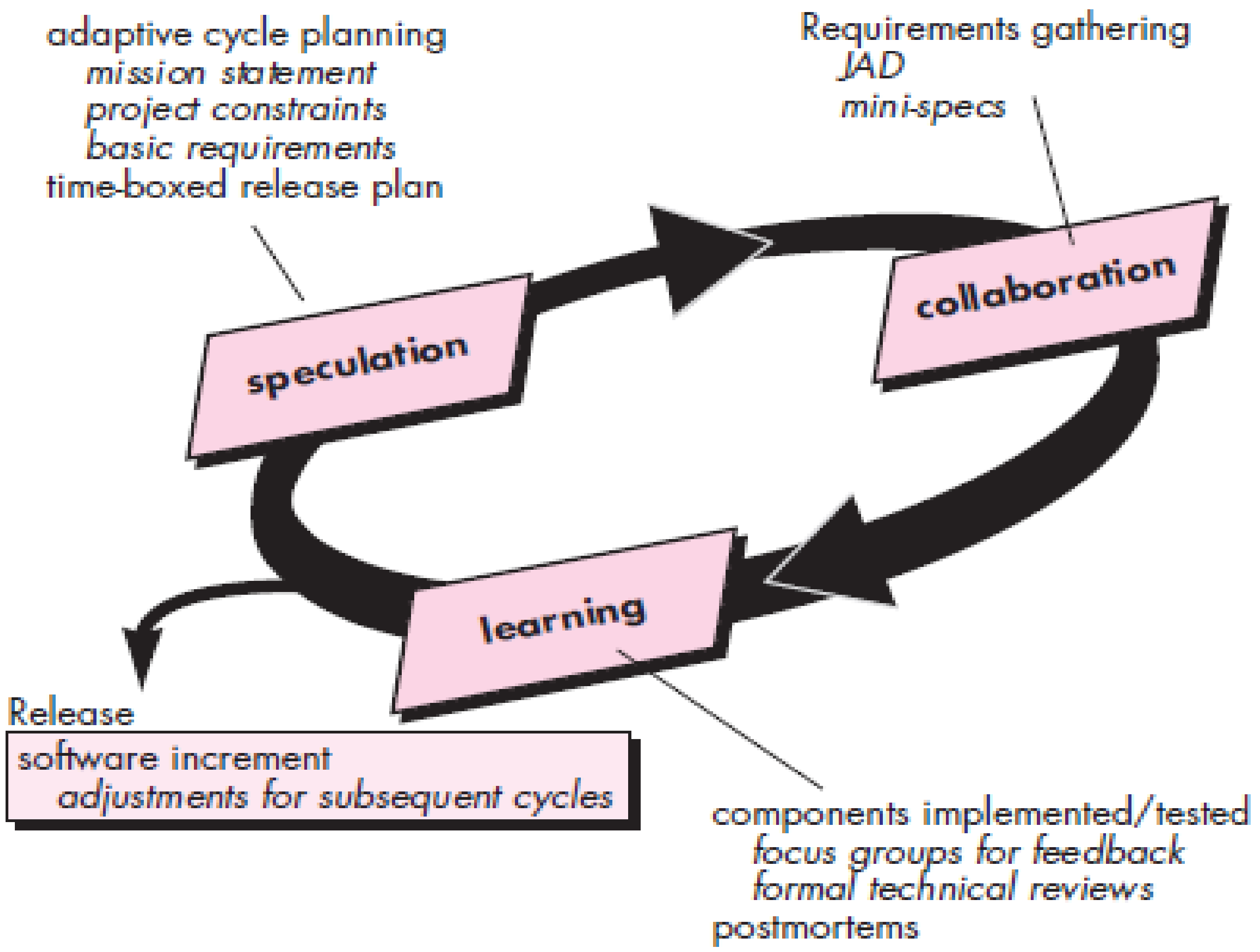
Kanban Board Example(Cont…)

- **To Do section:** contains tasks that were received from the customer and required to be analyzed. Each task is marked with color according to its priority.
- **Estimated Section:** When tasks from the first section have been analyzed and estimated by the Team, they are moved to the Estimated section.
- **In Progress:** When the developer takes a task to be developed, he moves it from Estimated to In Progress section and marks it with his tag to show who handles each task.
- **Done:** When a task is done, it’s moved to the Done section.

Adaptive Software Development(ASD)

- Originally proposed by Jim Highsmith
- ASD technique proposed for building complex software and system.
- ASD focus on human collaboration and team-self-organization.
- ASD incorporates three phases:-
    - Speculation
    - Collaboration
    - Learning

Adaptive Software Development



ASD Three Phases

1. **Speculation:**
   - During speculation , the project is initiated and adapted cycle planning is conducted.
   - Adapting cycle planning uses project initiation – information the customers mission statement, project constraints and basic requirements to define the set of release cycle.
2. **Collaboration**
   - The collaboration approach is requiring theme in all agile methods, but collaboration is not easy.
   - It is not simply communicate, although communicate is a part of it.
   - It is nota rejection individualism, because individual creativity plays on important role in collaboration thinking.
   - People working together must trust one another to:-
     - Criticize without animosity
     - Assist without resentment.
     - Work as hard of harder as they do.
     - Have the skill set to contribute to the work at hard
     - Communicate problem

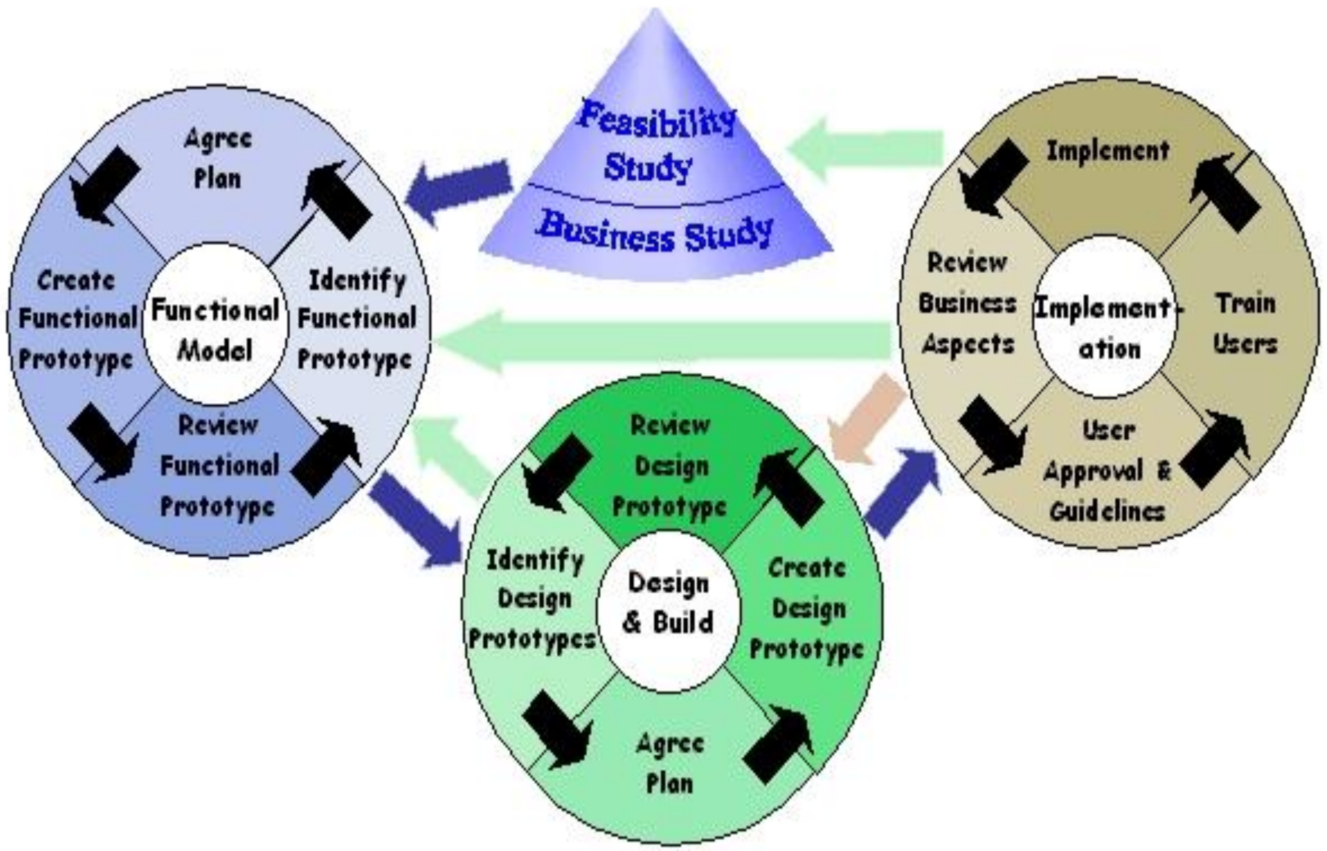
Adaptive Software Development

3. **Learning**
   - ASD teams learns three ways:-
   - Focus Group:
     - The customer or end user provide feedback on software increments that are being delivered.
   - Formal technical review:
     - ASD team members review the software components that are develop, improving quality and learning as they proceed.
   - Postmaster
     - The ASD team becomes introspective , addressing its own performance and process.

Dynamic Systems Development (DSD) Method

- The Dynamic System Development method (DSDM) is an agile software development approach that provides a framework for building and maintain system which meet tight time constraints through the use of incremental prototyping in a controlled project environment.
- The DSDM life cycle defines, three iterative cycle precede by two additional life cycle.
  - Feasibility study
  - Business study
  - Functional model iteration
  - Design and iteration
  - Implementation

Dynamic Systems Development (DSD) Method



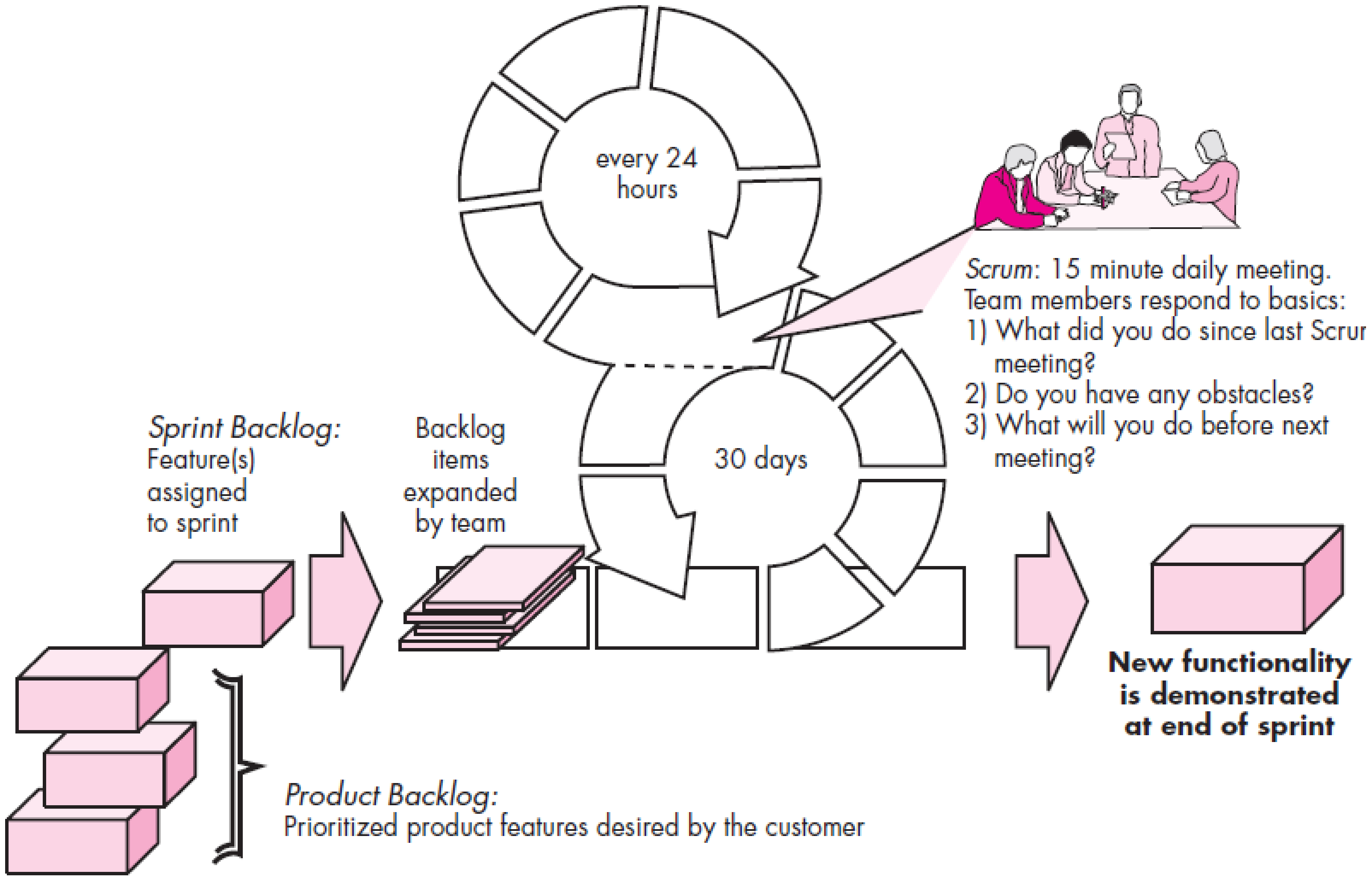
DSDM Iterative life cycle

- **Feasibility study**
  - Established the basic business requirements and constraints associated with the applicants to be built.
- **Business study**
  - Establishes the functional information requirements that will allow the applicants to provide business value.
- **Functional model iteration**
  - Produce a set of incremental prototype that demonstrate functionality for the customer.
- **Design and iteration**
  - Revisits prototype built during the functional model iteration to ensure that each has been engineered in a manner.
- **Implementation**
  - Places the latest software increment into the operational environment.
    - It should be noted that-
      - The increment may not be 100 percent complete
      - Changes may be requested as the increment is put into place.

Scrum Agile Process

- Scrum is an agile software development method that was conceived by Jeff Sutherland and his development team in the early 1990s.
- In recent years, further development on the Scrum methods has been performed by Schwaber and Beedle
- Scrum principles are consistent with the agile manifesto and are used to guide development activities within a process that incorporates the following framework activities: requirements, analysis, design, evolution, and delivery.

Scrum Agile Process



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

Fig: Scrum Process Flow

Scrum Agile Principles

- Small working teams are organized to maximize communication, minimize overload and maximize sharing tacit, informal knowledge.
- The process must be adaptable to both technical and business changes – to ensure the best possible product is produced.
- The process yields frequent software increment that can be inspected, adjusted, tested , documented and built on.
- Development work and people who perform it are partitioned into clean low coupling partitions or packets.
- Constant testing and documentation is preferred as the product is built.
- The scrum process provides the ability to declare a product done whenever required.

❑References:

1. **Software Engineering A practitioner's Approach**

   by Roger S. Pressman, 7th edition, McGraw Hill, 2010.

2. **Software Engineering by Ian Sommerville,**

   9th edition, Addison-Wesley, 2011