Agile model

Mrs. Veena Kulkarni

Agile Process

- Agility
  - The ability to both create and respond to change in order to profit in a turbulent business environment
    - Companies need to determine the amount of agility so they need to be competitive
- Why do we go for Agile process

1) the customer did not give all of the expected requirements and

2) engineers did not always understand the requirements.

Some Agile Methodologies

- Scrum
- Extreme Programming (XP)-With XP

programming the steps are only planning, design, coding and testing.

e.g. pair programming

Principles of Agility

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

3. Deliver working software frequently, from a couple of weeks to a couple of months.

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.

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. *Working oftware is important, but don't forget that it must also exhibit a variety of quality attributes including reliability, usability, and maintainability.*

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity is essential.

11. The best architectures, requirements, and designs emerge from self– organizing teams.

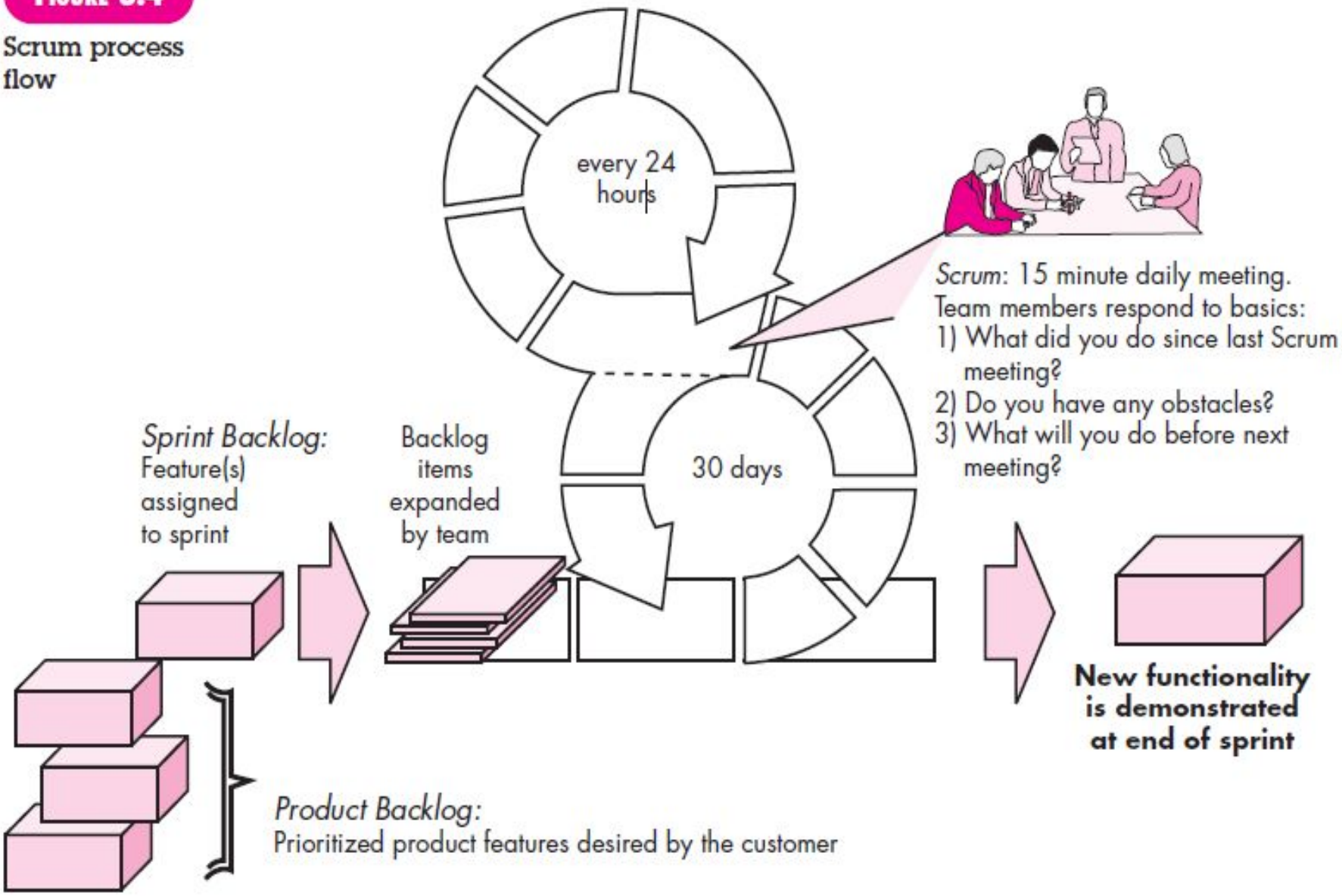
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

other agile process models used in industry

- Adaptive Software Development (ASD)
- Scrum
- Dynamic Systems Development Method (DSDM)
- Crystal
- Feature Drive Development (FDD)
- Lean Software Development (LSD)
- Agile Modeling (AM)
- Agile Unified Process (AUP)

FIGURE 3.4

Scrum process flow



Agile Management components

- 1. Visual control
- This is a “cards-on-the-wall” method of planning to assist a team in organizing the work of the project. For example, one successful Agile project team placed different color groups of cards that represented the features of the solution on the wall. The features that were designed, developed, tested and in production were one color, the features that were designed, built, tested but not yet put in production (but ready to go) were another color. The team was able to see at a glance where they were with each feature set. Visual control is a valuable technique for all projects, since it ensures that every member of the team views the project the same way.

- 2. Co-located high-performing teams
- In Agile development, all the key team members are co-located, including the customer/end-user, preferably in a work room. This approach greatly increases the quality of coordination and communication. However, this may represent a significant cultural change for IT developers. In traditional development methods, the developers typically work independently, and rarely interact with the customer until the solution is fully developed. Since project managers are responsible for building a high performing team, they need to ensure everyone is working well together and that they have been assigned developers who truly can work in this collaborative manner.

- team performance.

- 3. Test-driven development

- In cases when the customer is having a difficult time articulating requirements, Agile teams often use test-driven development. Using the same successful Agile project team mentioned above as an example, the test cases were often developed first, and then the team backed into the requirement. This obviously requires more iteration between requirements, design, development and test. The entire four-stage cycle is collapsed. In any case, Agile teams almost always develop test plans at the same time they define requirements; if a requirement isn't testable, then the requirement is not yet fully developed. This is a best practice that can be used in traditional development to ensure requirements are complete, accurate and testable.
-
- 4. Adaptive control
- Everyone on the team is constantly adapting, which may make some team members nervous, especially those that crave structure. Because of this dynamic environment, the project manager needs to be seen as a leader, not a taskmaster. Instead of setting rigid instructions for the entire team to follow, the project manager facilitates the team in establishing working relationships, setting ground rules and fostering collaboration. Agile team members continuously adapt to improve their methods as they incorporate lessons learned from the previous cycle into the next iteration, also a best practice for any project.

- 5. Collaborative development

- APM relies on collaboration among all team members to deliver the results, capture candid feedback and implement learnings on the next iteration of the solution. This is one of the strengths of APM - constant feedback and improvement. The project manager completes the initial planning and the business analyst defines and prioritizes the solution features in collaboration with the customer and technology representatives. Then the Agile project teams collaborate on the design, development, testing and reworking of each incremental build. It is this constant collaboration with the customer that promotes project success.

- 6. Feature-driven development

- This practice greatly reduces complexity, because it allows the team to focus on one feature and only one feature at a time. For example, one team is working on Feature #4 and that's the team's only focus. They don't concern themselves about Features #1-3. It is the business analyst and project manager who ensure the next feature in the backlog is truly the next priority, based on business value and risk. Typically, high-risk components or core infrastructure pieces are built first, and then everything else is prioritized based on business value. The goal is to build the featuredriven components with only a one-way dependency to the core system; therefore, specialized components are independent of each other and can be created in any order or even in parallel.

- 7. Leadership and collaboration rather than command and control
- "The principles of APM are timeless. If you look at APM, it links much more closely to leadership. It addresses a lot of the steps that facilitate leadership much more than traditional management,"

- The project manager works with the client management ,the IT management and the key stakeholders to ensure they know the project's status. Additionally, the project manager removes any barriers hindering the core Agile teams. The business analyst manages the business benefits of the project and continually focuses the Agile team on the business need.

- 8. Move from C (cost) to R (revenue) focus
- Features are prioritized based on value, such as increased revenue or market share. It's the business analyst's role to ensure the Agile project team is not investing too much into the development of the new solution. If so they will have eroded the business case and the project will cost more than the organization will gain. While the project manager focuses on project costs, the business analyst focuses on the total cost of ownership that includes not only the development or acquisition costs of the new solution, but also the cost of operating the system after it is deployed.

- 9. Lessons learned
- After each cycle, the team holds a lessons learned session to determine what they can do better on the next iteration. As the team learns, it adapts how the members are working together to continuously improve

Scrum

- requirements, analysis, design, evolution, and delivery are the activities.
- Scrum-it is incremental and iterative in nature
- Sprints- basic unit of development in scrum. It produces a working and tested software within given time limits.
- Sprints tend to last between on week to month
- *Backlog*—a prioritized list of project requirements or features

- *Scrum meetings*—are short (typically 15 minutes) meetings held daily by
- A team leader, called a *Scrum master,* leads the meeting and assesses the responses from each person. the Scrum team.
- Team size according to 2 pizza rule.

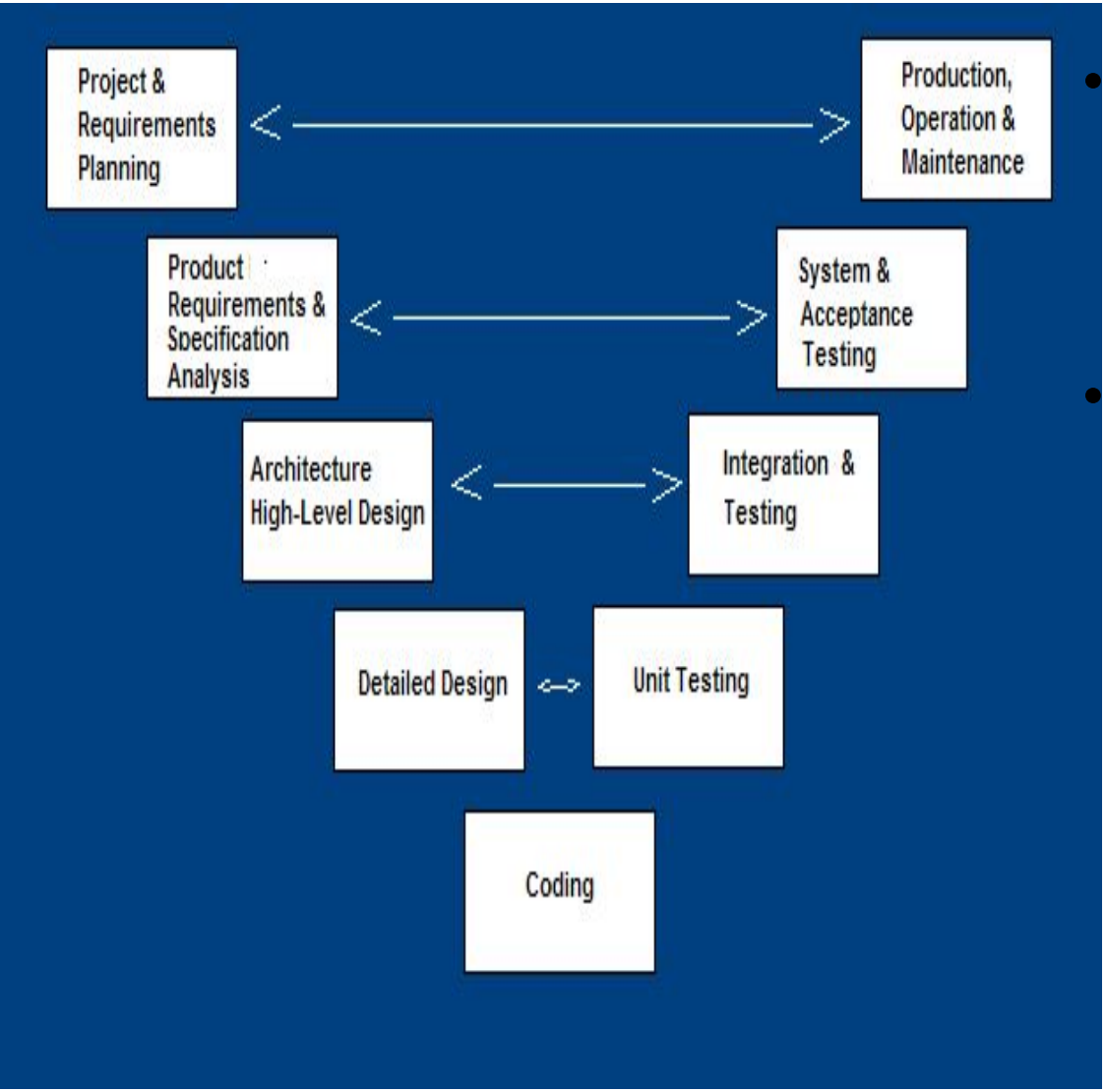
Advantages of Agile model

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Disadvantages of agile model

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

V-Shaped SDLC Model



- A variant of the Waterfall that emphasizes the verification and validation of the product.
- Testing of the product is planned in parallel with a corresponding phase of development

V-Shaped Steps

- Project and Requirements Planning – allocate resources
- Product Requirements and Specification Analysis – complete specification of the software system
- Architecture or High-Level Design – defines how software functions fulfill the design
- Detailed Design – develop algorithms for each architectural component
- Production, operation and maintenance – provide for enhancement and corrections
- System and acceptance testing – check the entire software system in its environment
- Integration and Testing – check that modules interconnect correctly
- Unit testing – check that each module acts as expected
- Coding – transform algorithms into software

V-Shaped Strengths

- Emphasize planning for verification and validation of the product in early stages of product development
- Each deliverable must be testable
- Project management can track progress by milestones
- Easy to use

V-Shaped Weaknesses

- Does not easily handle concurrent events
- Does not handle iterations or phases
- Does not easily handle dynamic changes in requirements
- Does not contain risk analysis activities

When to use the V-Shaped Model

- Excellent choice for systems requiring high reliability – hospital patient control applications
- All requirements are known up-front
- Solution and technology are known

- Courtesy
- Softaware Engineering Pressman