AGILE MODEL OF SOFTWARE DEVELOPMENT

ECE 811 – SOFTWARE ENGINEERING

Monday, July 28, 2025

AGILE DEVELOPMENT PROCESS /01

The agile development process is achieved through the following steps:

1. The requirements are decomposed into many small parts that can be incrementally developed.
2. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed over a short period.
3. Each iteration is planned, developed and deployed to the customers.

AGILE DEVELOPMENT PROCESS /02

Agile model is the combination of iterative and incremental process models. Steps involve in agile SDLC models are:

1. Requirement gathering
2. Requirement Analysis
3. Design
4. Coding
5. Unit testing
6. Acceptance testing

KEY FEATURES OF THE AGILE MODEL/01

The features of agile software development are:

1. Maintains a close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.
2. Relies on working software deployment rather than comprehensive documentation.
3. Frequent delivery of incremental versions of the software to the customer representative in short intervals.
4. Requirement change requests from the customer are encouraged and efficiently incorporated.

GOALS OF THE AGILE MODEL /02

5. It emphasizes on having efficient team members and enhancing communications among them is given more importance.
6. Small development teams of to help the team members have collaborative work environment.
7. Use of pair programming where two programmers work together with one does coding while the other reviews the code.

ADVANTAGES OF AGILE SOFTWARE DEVELOPMENT

1. Working through Pair programming produce well-written compact programs which has fewer errors as compared to programmers working alone.

2. Reduced development time of the whole project.

3. Customer representative get the idea of updated software products after each iteration making it easy to change any requirement if needed.

DISADVANTAGES OF AGILE SOFTWARE DEVELOPMENT

1. Due to lack of formal documents, it creates confusion and important decisions taken during different phases can be misinterpreted by different team members.

2. Due to absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

TYPES OF AGILE MODELS

Some Agile SDLC models are given are:

1. Scrum
2. Extreme programming (XP)
3. Crystal
4. Atern
5. Feature-driven development
6. Lean development
7. Unified process

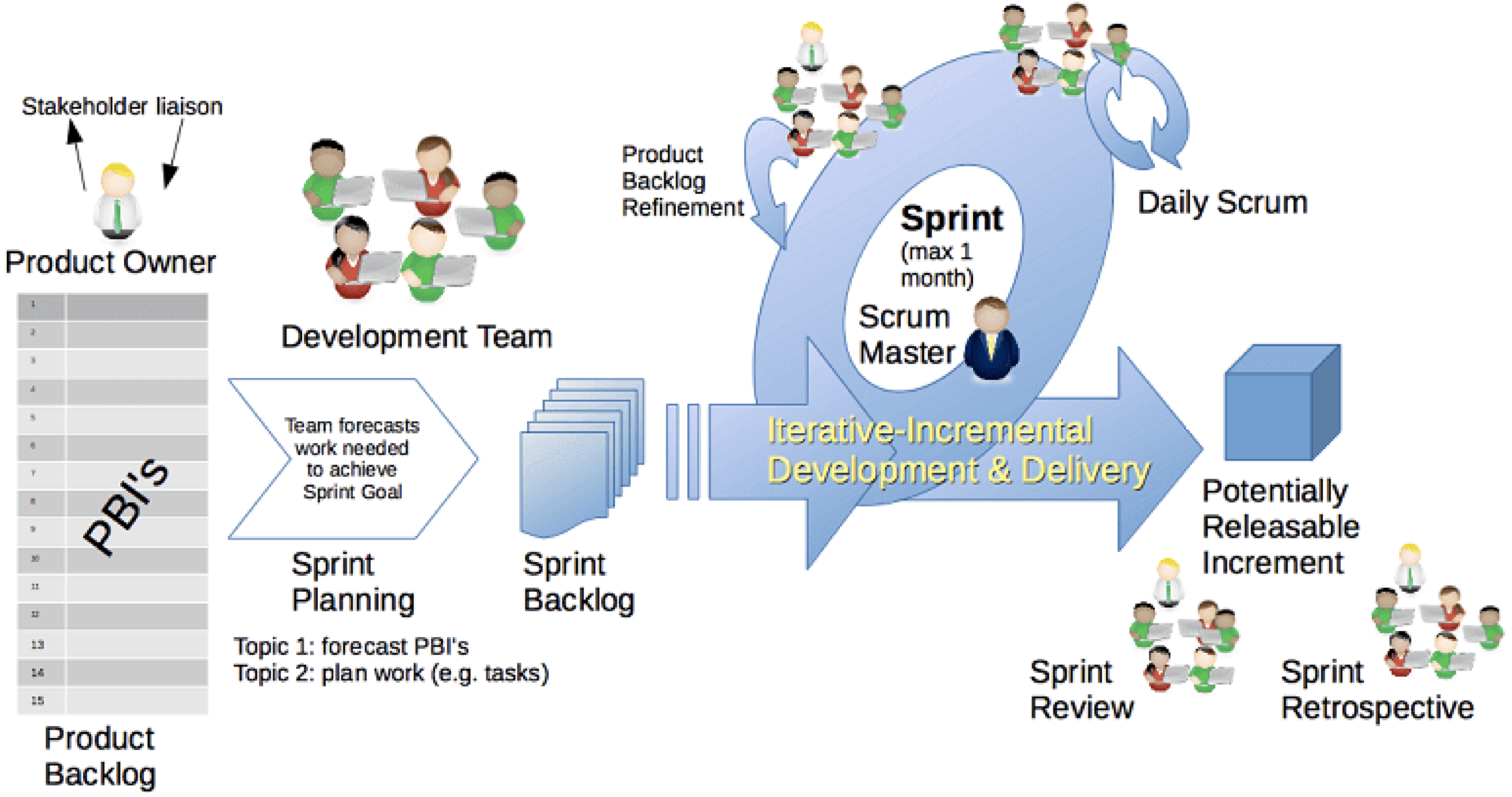
SCRUM SOFTWARE DEVELOPMENT MODEL

1. Scrum is an agile framework for developing, delivering, and sustaining complex products, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies.
2. It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called sprints, no longer than one month and most commonly two weeks. The Scrum Team track progress in 15-minute time-boxed daily meetings, called daily scrums.
3. At the end of the sprint, the team holds sprint review, to demonstrate the work done, and sprint retrospective to improve continuously.
4. The software development term scrum was first used in a 1986 paper titled "The New Product Development Game". The term is borrowed from rugby, where a scrum is a formation of players

ROLES IN THE SCRUM FRAMEWORK

In software development, three roles are defined in the scrum framework:

1. The scrum team does the work.
2. Individuals work together in the sprints to produce the products.
3. The scrum master is part of the scrum team makes sure the team works in compliance with the scrum rules. This is not a manager.
4. The product owner represents the customer. He prioritizes the backlog and coordinates the scrum teamwork. His role is similar to project manager in more traditional project management frameworks.


Stakeholder liaison
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
PBI's
Product Backlog
Development Team
Team forecasts work needed to achieve Sprint Goal
Sprint Planning
Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)
Sprint Backlog
Product Backlog Refinement
Sprint
(max 1 month)
Scrum Master
Daily Scrum
Iterative-Incremental Development & Delivery
Potentially Releasable Increment
Sprint Review
Sprint Retrospective

ADVANTAGES OF SCRUM MODEL /01

The benefits of the scrum model are:

1. Developers who want the freedom to make decisions thrive in scrum teams. Team morale tends to be high.
2. Each sprint produces a product that is ready for market even though the project is ongoing.
3. The highest priority requirements are addressed first so a high-quality, low-risk product can be on the market.
4. The incremental process shortens the time to market by about 30 percent to 40 percent.
5. Because the product owner is part of the scrum team, requirements can be delivered as they are needed.

ADVANTAGES OF SCRUM MODEL /02

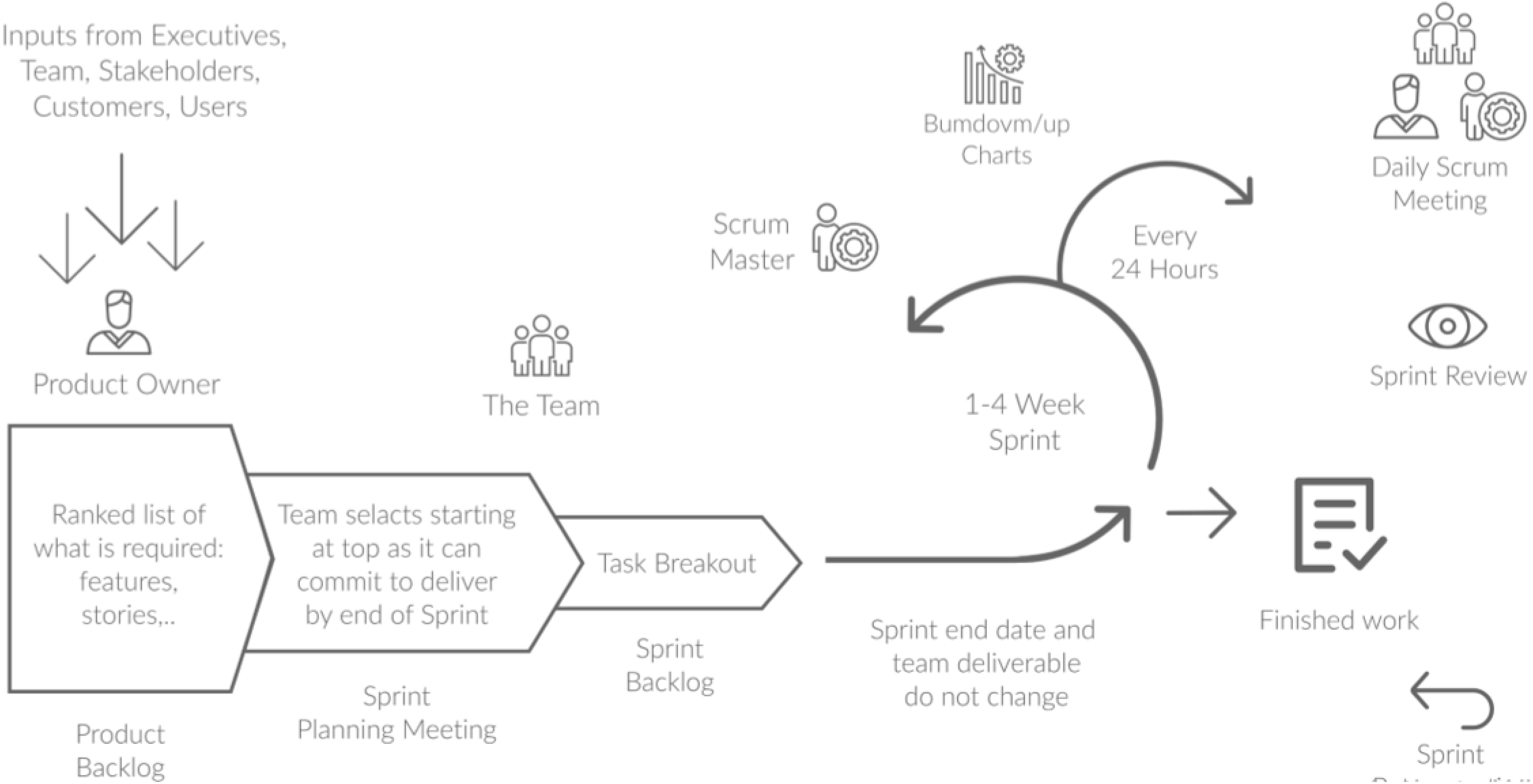
6. Scrum projects often realize a higher return on investment (ROI). This is attributed to:
    a) Decreased time to market.
    b) Early and regular feedback that prompts course corrections early when they are less costly.
    c) Defects that are fewer and less costly.
    d) Projects failing early and quickly when it’s less costly.
7. Reviewing each sprint before the team moves on to the next sprint spreads testing throughout development.
8. Project focus and goals can change with evolving business goals

DISADVANTAGES OF SCRUM MODEL /02

1. There is a danger of scope creep if stakeholders keep adding functionality to the backlog.
2. Scrum works best with small teams of experienced software developers. They need to be able to work quickly.
3. Scrum teams do not work well when the scrum master micromanages their work.
4. Losing any team members can hurt the progress of the project.

SCRUM BEST PRACTICES

1. Define requirements just in time to keep product features as relevant as possible.
2. Test and incorporate product owner feedback daily.
3. Sprint reviews with stakeholders need to be regular.
4. The scrum team needs to use the sprint retrospectives to improve how they work.
5. Conduct face-to-face conversations to reduce miscommunications.
6. Trust the teams to do the best job possible.
7. Allow the teams to self-organize around people’s skills, work styles and personalities.
8. Don’t burn out the team members. Respect the balance between their personal and professional lives to ease stress.



Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Scrum Master

Bumdovm/up Charts

Every 24 Hours

Daily Scrum Meeting

Sprint Review

Ranked list of what is required: features, stories,..

Product Backlog

Team selacts starting at top as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog

1-4 Week Sprint

Sprint end date and team deliverable do not change

Finished work

Sprint

EXTREME PROGRAMMING (XP)

- Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.
- XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

FEATURES OF XP

Key features of XP model includes:

1. The Planning Game
2. Small Releases
3. Metaphor
4. Simple Design
5. Testing
6. Refactoring
7. Pair Programming
8. Collective Ownership
9. Continuous Integration
10. 40-hour week
11. On-site Customer
12. Coding Standard

APPLICATIONS OF EXTREME PROGRAMMING (XP)

XP should be used in the following situations.

1. Dynamically changing software requirements
2. Risks caused by fixed time projects using new technology
3. Small, co-located extended development team
4. The technology you are using allows for automated unit and functional tests

DIFFERENCES BETWEEN XP & SCRUM

1. Scrum is more high level, focusing on the management of the project (e.g. the requirements or features are managed) rather than specifying or defining engineering practice such as pair programming or test driven development
2. The length of an iteration in XP is usually 1-3 weeks whereas, in Scrum sprints are 1-4 weeks
3. Once sprint (or iteration in XP) starts, customer cannot change the requirements, in other words the customer will have to wait until the sprint (or iteration in XP) finishes. In XP however, requirements can change anytime
4. In XP features are developed in a strict order whereas in Scrum, the team is free to choose the features to be developed. Sequence does not matter
5. Both XP and Scrum define the role of a coach, In Scrum it is called Scrum Master and requires (or strongly recommended) certification, whereas, XP defines the role of coach quite informally and the role may float between members of the team.