Agile Methodology

What is Agile Methodology?
AGILE methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall mode

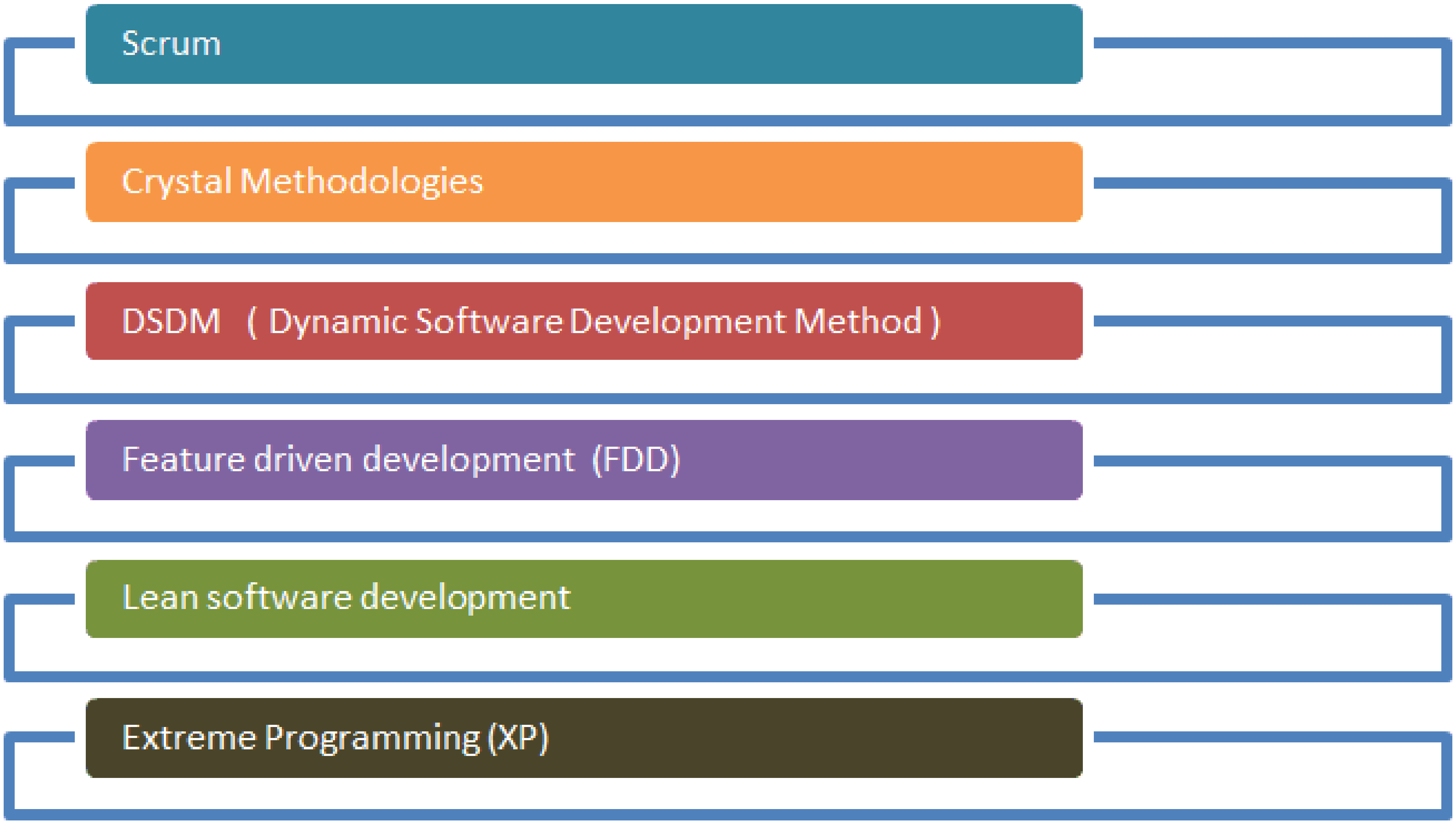
The agile software development emphasizes on four core values.

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

Five reasons

- **Early Time to Market need**
- **Ever changing requirement - difficult to adopt.**
- Basic Systems in place (thru waterfall). Most of the projects were on incremeantal modifications. Need of quick implementation.
- Need of continous interaction with Customer.
- Waterfall was heavy on documentation.

Agile Testing Methodology



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

SCRUM

Scrum

The Scrum framework consists of Scrum Teams and their associated roles, events (time boxed), artifacts, and rules.

Sprint

- The heart of Scrum is a Sprint, a time-box of two weeks or one month during
- A potentially releasable product increment is created.

Roles in Scrum

Scrum Master
Responsible for setting up the team, sprint meeting.

Product owner
Creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration.

Scrum Team
Team organizes and manages its own work to complete the sprint.

Activities in Sprint

- Sprint planning
- daily scrums meeting
- the development work
- the Sprint review
- the Sprint retrospective

Scrum Methodology

Sprint

- A new Sprint starts immediately after the conclusion of the previous Sprint. Sprints consist of the
  - **Sprint planning** - the work to be performed in the **Sprint**
daily scrums meeting - 15-minute time-boxed meet daily
  - the **development work** - develop and test the change
  - the **Sprint review** - Review the increment
  - the **Sprint retrospective** - inspect itself and create a plan for improvements

The shift : TSD to ASD

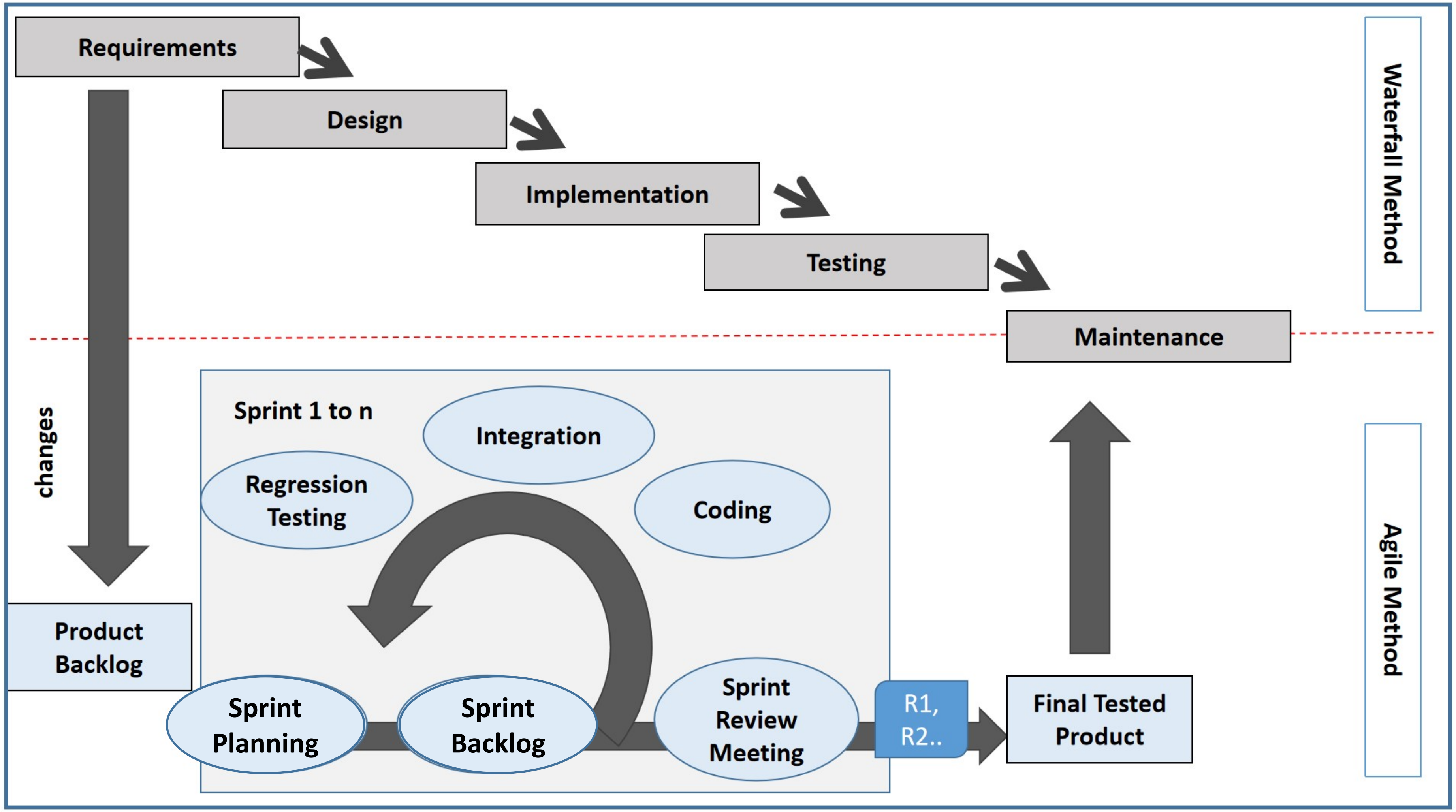


Fig 3: Comparison of Traditional Software Development (TSD) and Agile Software Development (ASD) process

- In Agile method, **software is developed in short iterations and the team incorporated all new knowledge gained through feedback** from preceding iterations.

***Testing that is done to verify that a code change in the software does not impact the existing functionality of the product.**

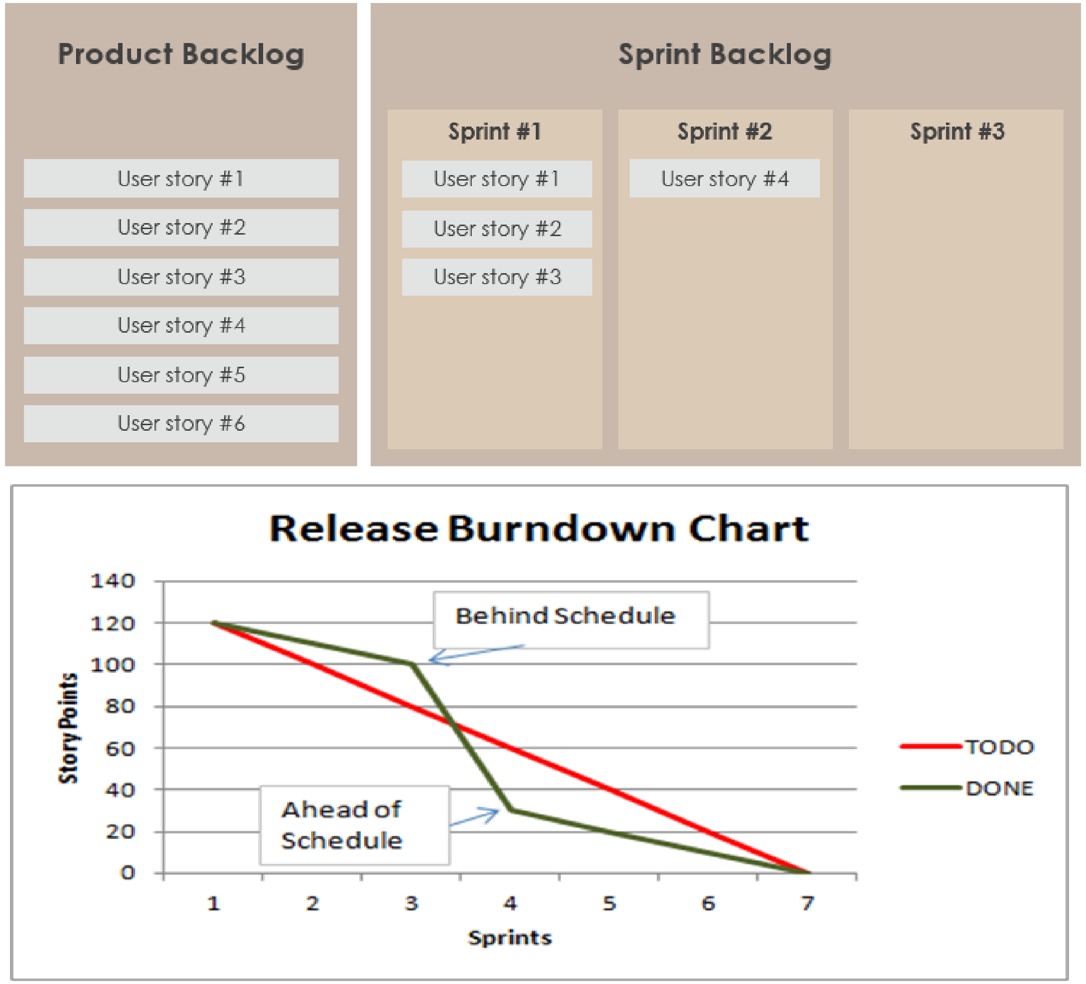
Artifacts in Scrum

- **Product Backlog** : A product backlog is a list of items(features/user stories) to be done.

- **Sprint Backlog:** Subset of Product Backlog items selected for the Sprint.

- **Sprint/Release Burn-Down Chart :** shows the rate at which work is completed and how much work remains to be done.

- **Increment** :The Increment is the sum of all the Product Backlog items completed during a Sprint combined with the increments of all previous Sprints.



Product Backlog	Sprint Backlog		
	Sprint #1	Sprint #2	Sprint #3
User story #1	User story #1	User story #4	
User story #2	User story #2		
User story #3	User story #3		
User story #4			
User story #5			
User story #6			

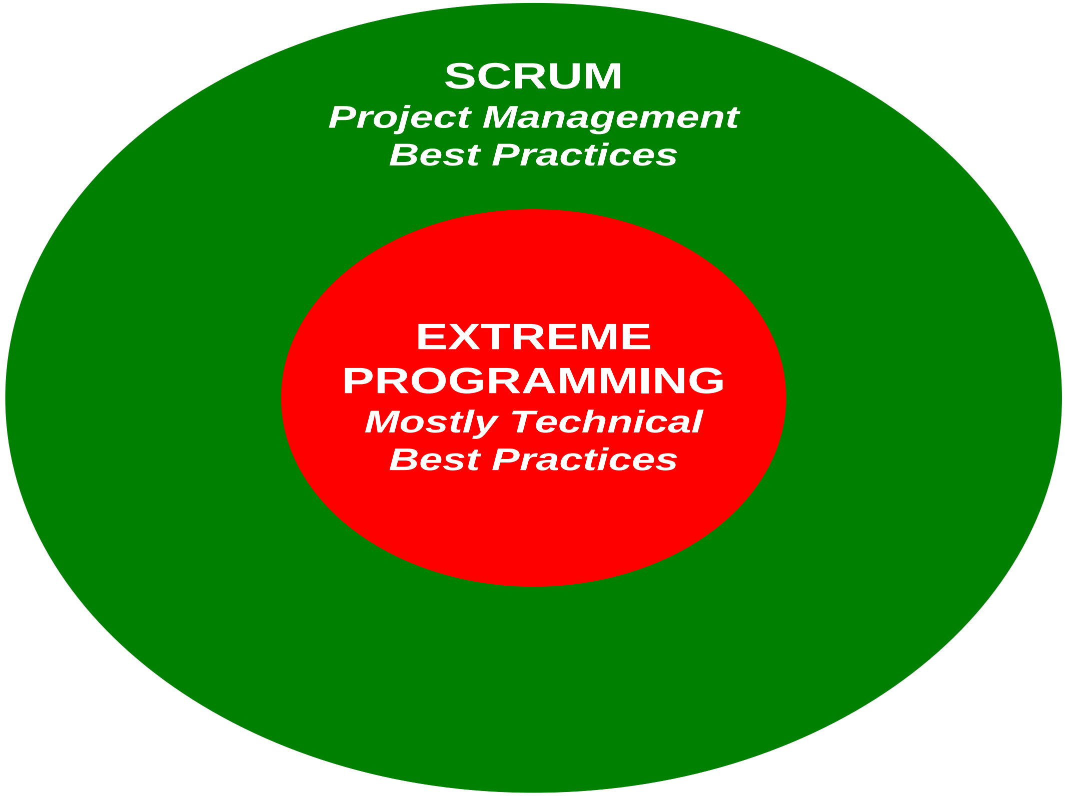
Fig 4 : Important artifacts of Scrum

Process flow of Scrum Methodologies

Process flow of scrum testing is as follows:

- Each **iteration** of a scrum is known as **Sprint**
- **Product backlog** is a **list** where all details are entered to get end product
- During each Sprint, top items of Product backlog are selected and turned into Sprint backlog
- Team works on the defined sprint backlog
- Team checks for the daily work
- At the end of the sprint, team delivers product functionality

Scrum with Extreme Programming



Scrum works well as a *wrapper* around Extreme Programming

eXtreme Programming (XP)

Extreme Programming (XP) is based on the five values –

Communication
Simplicity
Feedback
Courage
Respect
Extreme Programming is a systematic approach with a set of values, rules and practices for rapidly developing high quality software that provides the highest value for customers.

XP Value: Communication

- **Poor communication** in software teams is one of the root causes of **failure of a project**
- Stress on **good communication between all stakeholders-**-customers, team members, project managers
- XP emphasizes value of communication in many of its practices:
  - On-site customer, user stories, **pair programming**, collective ownership (popular with open source developers), daily standup meetings, etc.

XP Value: Simplicity

- ‘Do the Simplest Thing That Could Possibly Work’
  - Implement a new capability in the simplest possible way
  - Refactor the system to be the simplest possible code with the current feature set
- ‘You Aren’t Going to Need It’
  - Never implement a feature you don’t need now

*refactoring is to improve internal code by making many small changes without altering the code's external behavior.

XP Value: Feedback

- Programmers produce new releases every 2-3 weeks for customers to review
- Unit tests tell programmers status of the system
- Small iteration and pair programming help a great deal to give a proper understanding of where they stand
- Hence, Feedback is repetitive and frequent in XP

XP Value: Courage

- The courage to communicate and accept feedback
- The courage to throw code away (prototypes)
- The courage to refactor the architecture of a system

- Business requirements are gathered in terms of stories. All those stories are stored in a place called the parking lot.

- In this type of methodology, releases are based on the shorter cycles called Iterations with span of 14 days time period. Each iteration includes phases like coding, unit testing and system testing where at each phase some minor or major functionality will be built in the application.

Roles

- Developer (also called Programmer by some teams)
- Customer
- Manager (also called tracker)
- Coach

Process of eXtreme programming

- User stories are the heart of planning in Extreme Programming (XP).
- High level designs are created from stories.
- Architectural spikes or prototypes are used to create a simple overall design.
- High code quality is essential on an XP project.
- Developers receive feedback constantly by working in pairs and testing code as it is written.
- Managers get feedback on progress and obstacles at the daily stand up meeting

Scrum vs Extreme Programming

- Scrum teams typically work in **iterations** (called sprints) that are from **two weeks to one month long.** XP teams typically work in iterations that are one or two weeks long.
- Scrum teams **do not allow changes into their sprints**. Once the sprint planning meeting is completed and a **commitment made to delivering a set of product backlog items**, that set of items remains unchanged through the end of the sprint.
- XP teams are much more **amenable** (open and responsive to suggestion) to change within their iterations. As long as the team hasn't started work on a particular feature.
- Extreme Programming teams work in a **strict priority order**. Features to be developed are prioritized by the customer and the team is required to work on them in that order.
- By contrast, the Scrum product owner prioritizes the product backlog but the team determines the sequence in which they will develop the backlog items.