CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# Object Oriented Software Engineering

## LECTURE 5

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## Today's Outline:

**An Agile View of Process:**

- What is agility
- What is an agile process
- Agile Process Models:
  - extreme programming (XP)
  - ASD, Scrum

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## Introduction

- The meaning of Agile is

**swift or versatile.**

**What is Agile?**

- Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## What is Agile Methodology?

- Agile Methodology meaning a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project.

- Agile is a time-bound, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver all at once.

# Why Agile?

- Technology in this current era is progressing faster than ever, enforcing the global software companies to work in a fast-paced changing environment. Because these businesses are operating in an ever-changing environment, it is impossible to gather a complete and exhaustive set of software requirements. Without these requirements, it becomes practically hard for any conventional software model to work.
- The conventional software models such as Waterfall Model that depends on completely specifying the requirements, designing, and testing the system are not geared towards rapid software development. As a consequence, a conventional software development model fails to deliver the required product.

# Why Agile?

- This is where the agile software development comes to the rescue. It was specially designed to curate the needs of the rapidly changing environment by embracing the idea of incremental development and develop the actual final product.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

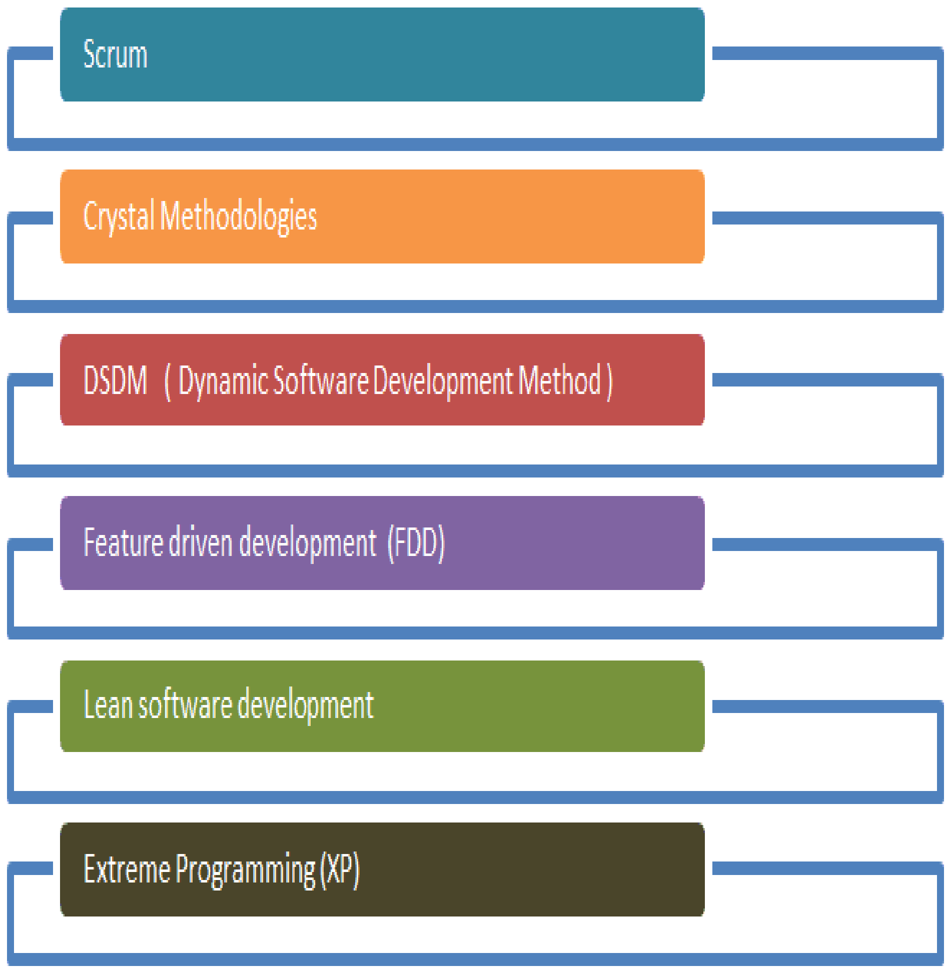
## History

- In 2001, this new management paradigm began to pick up momentum, agile was formalized when 17 pioneers of the agile methodology met at the Snowbird Ski Resort in Utah and issued the Agile Manifesto.
- 1.Kent Beck 2.Mike Beedle
- 3. Arie van Bennekum 4. Alistair Cockburn
- 5.Ward Cunningham 6.Martin Fowler
- 7.James Grenning 8.Jim Highsmith
- 9.Andrew Hunt 10.Ron Jeffries
- 11.Jon Kern 12.Brian Marick
- 13.Robert C. Martin 14.Steve Mellor
- 15.Ken Schwaber 16.Jeff Sutherland
- 17.Dave Thomas

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# Popular Agile Methods

- The most popular Agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as **Agile Methodologies**, after the Agile Manifesto was published in 2001.


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

# Agile Model

- "**Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.
- Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# Agile Model

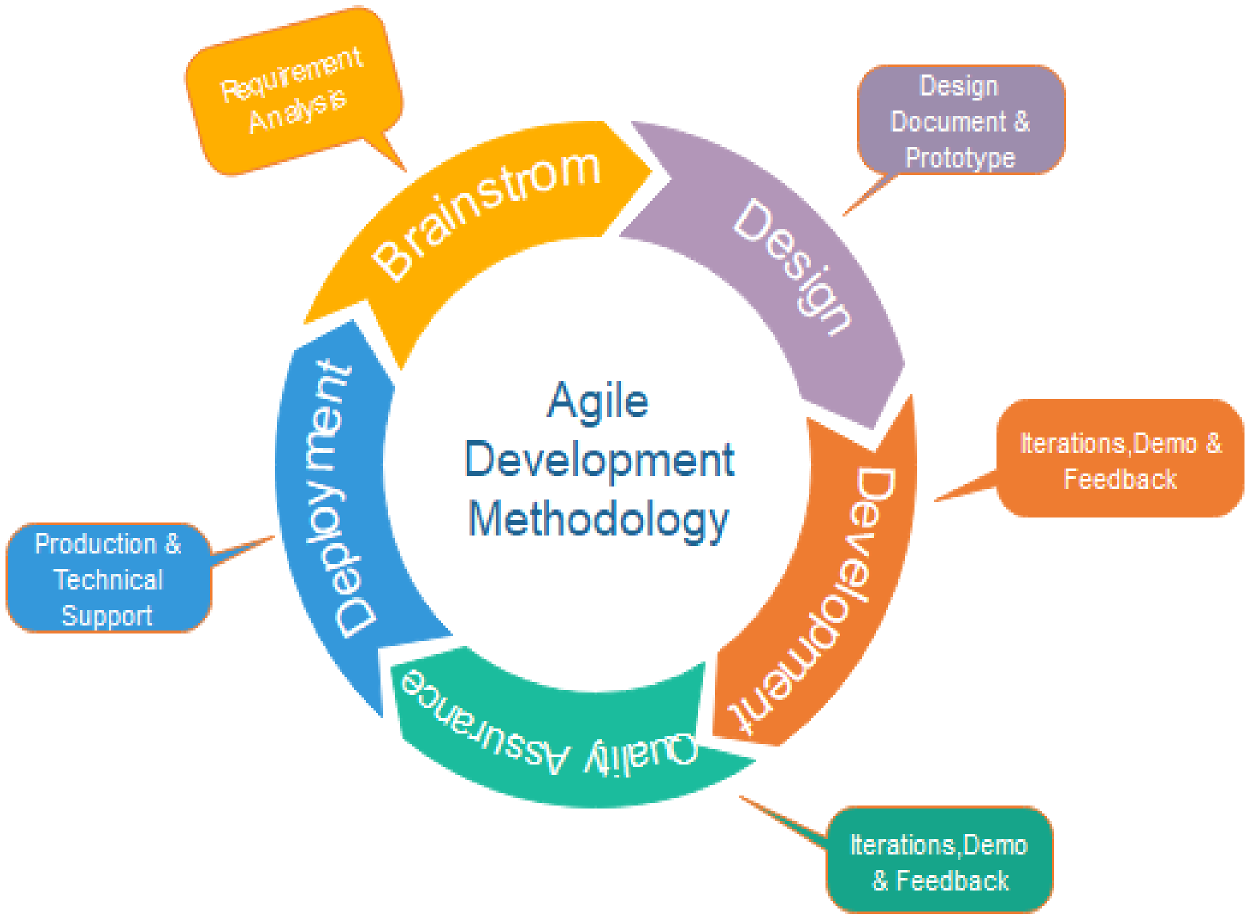

Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Agile Development Methodology
Development
Iterations,Demo & Feedback
Quality Assurance
Iterations,Demo & Feedback
Deployment
Production & Technical Support

***Fig. Agile Model***

## Phases of Agile Model:

Following are the phases in the Agile model are as follows:

- Requirements gathering
- Design the requirements
- Construction/ iteration
- Testing/ Quality assurance
- Deployment
- Feedback

## Phases of Agile Model:

1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
2. **Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## Phases of Agile Model:

**3. Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

**4. Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

**5. Deployment:** In this phase, the team issues a product for the user's work environment.

**6. Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

# Agile Values

- **Individuals and interactions** – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration** – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** – Agile Development is focused on quick responses to change and continuous development.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# Agile Principles

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

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## Agile Principles

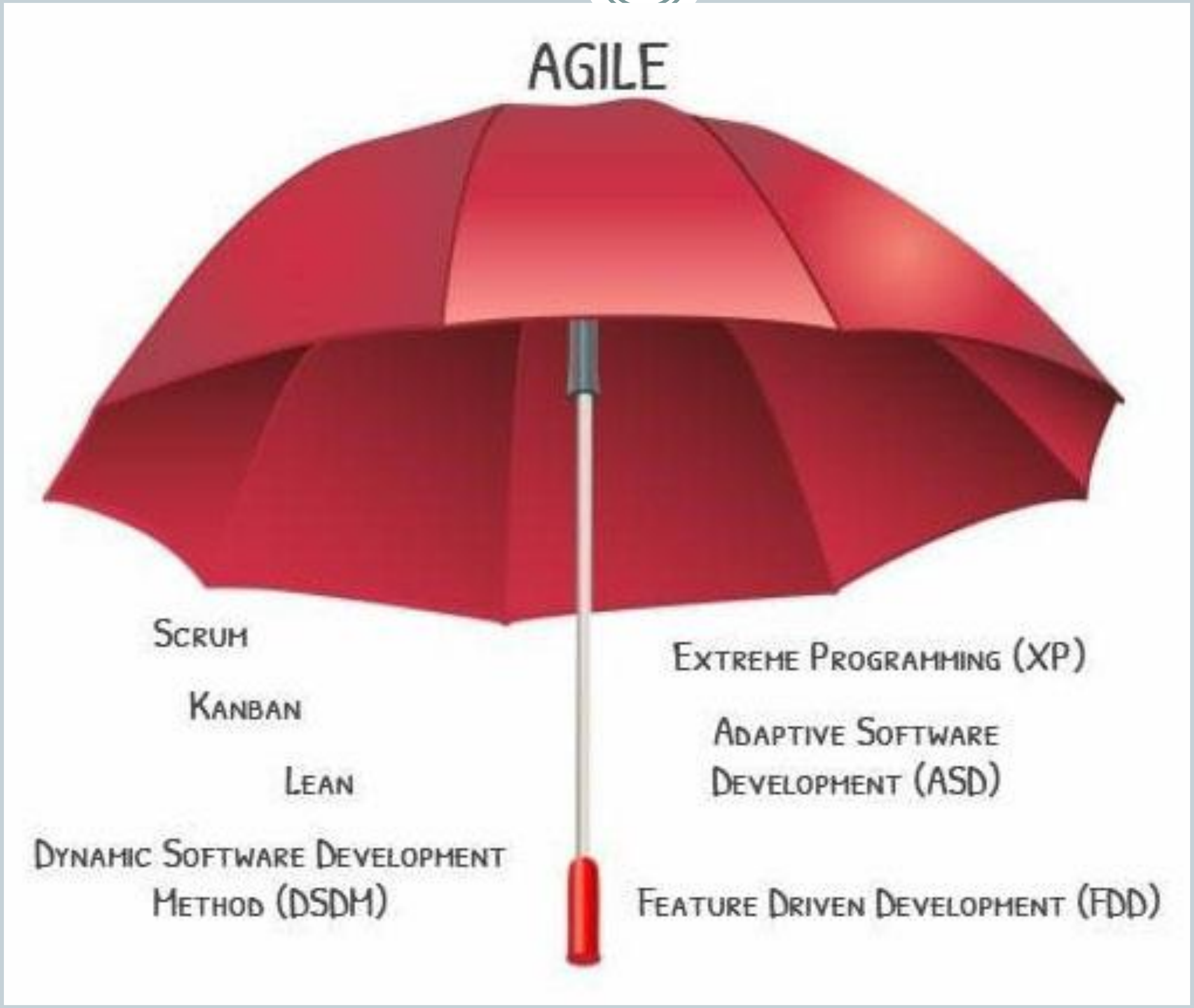
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art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

## Agile Testing Methods:

- Scrum
- eXtreme Programming(XP)
- Adaptive Software Development (ASD)
- Crystal
- Dynamic Software Development Method(DSDM)
- Feature Driven Development(FDD)
- Lean Software Development

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# Agile Methodology


AGILE
Scrum
Kanban
Lean
Dynamic Software Development Method (DSDM)
Extreme Programming (XP)
Adaptive Software Development (ASD)
Feature Driven Development (FDD)

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## SCRUM

- What is Scrum?

- Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.
- ***Scrum fall under the word "Team Centric".***

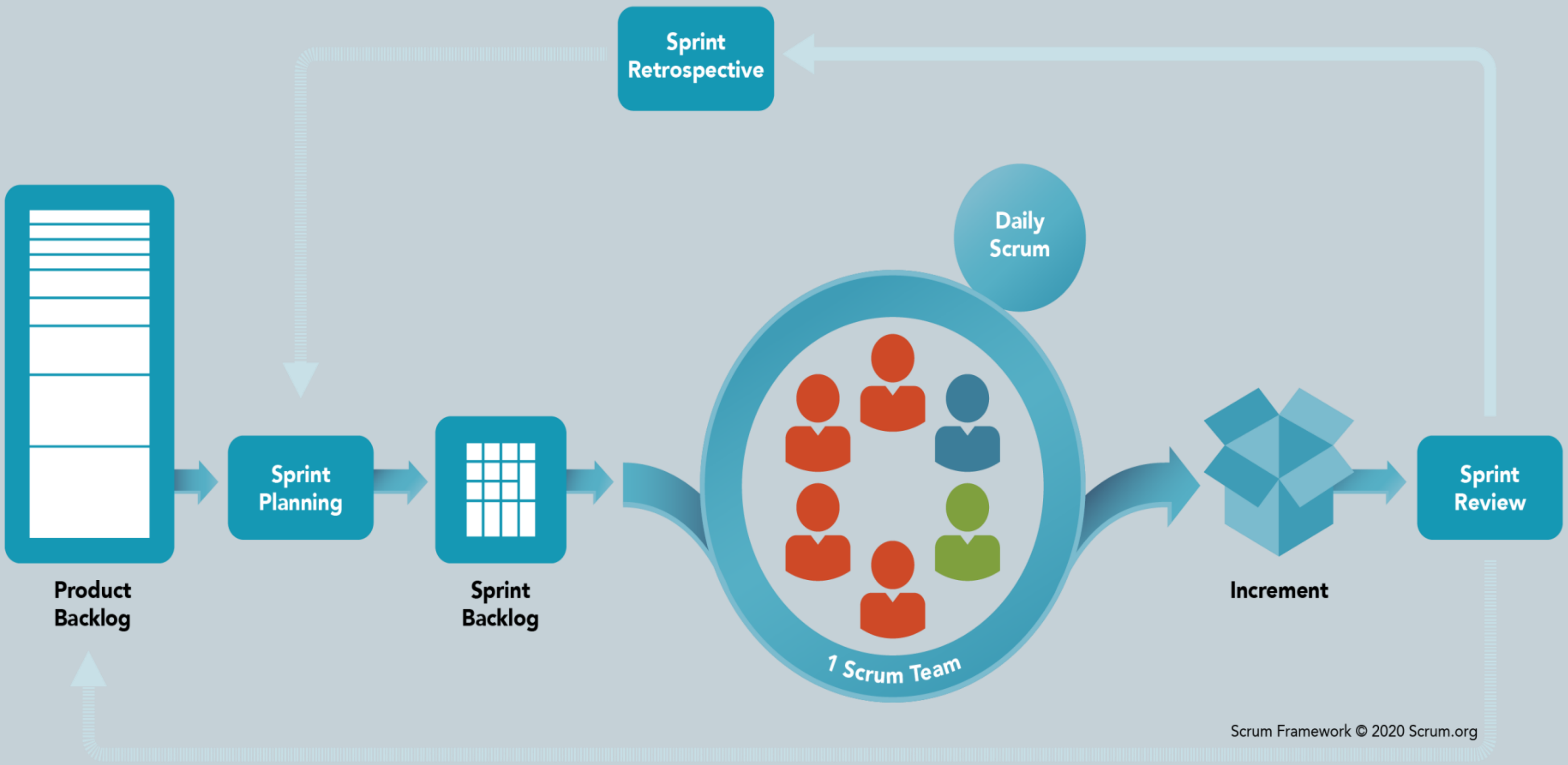
## Scrum

- SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, Scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members). Agile and Scrum consist of three roles, and their responsibilities are explained as follows:

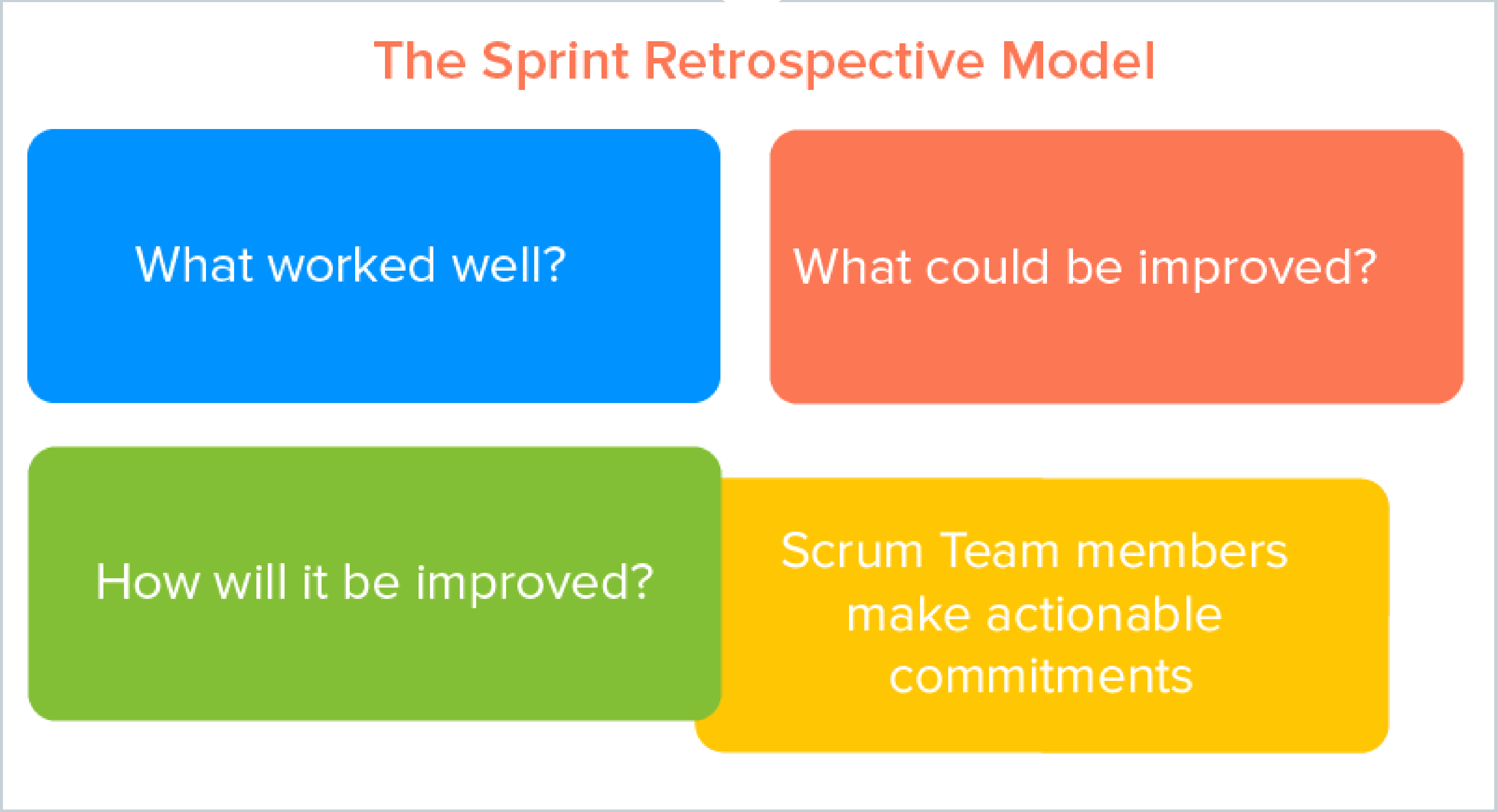
## Scrum

- Scrum Master
  - Master is responsible for setting up the team, sprint meeting and removes obstacles to progress
- Product owner
  - The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration
- Scrum Team
  - Team manages its own work and organizes the work to complete the sprint or cycle

## Sprint Framework



# The Retrospective scrum model


The Sprint Retrospective Model

What worked well?

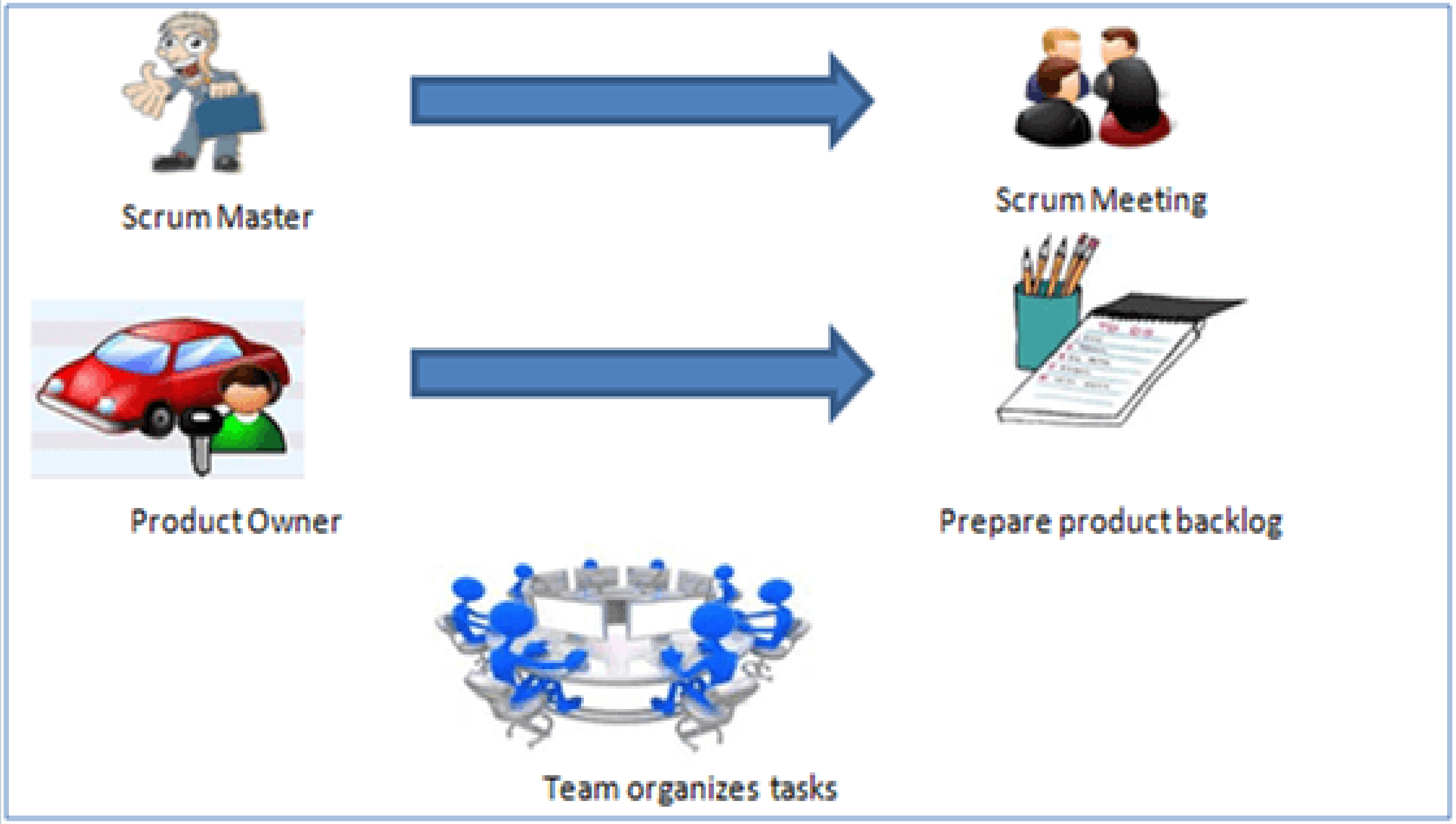
What could be improved?

How will it be improved?

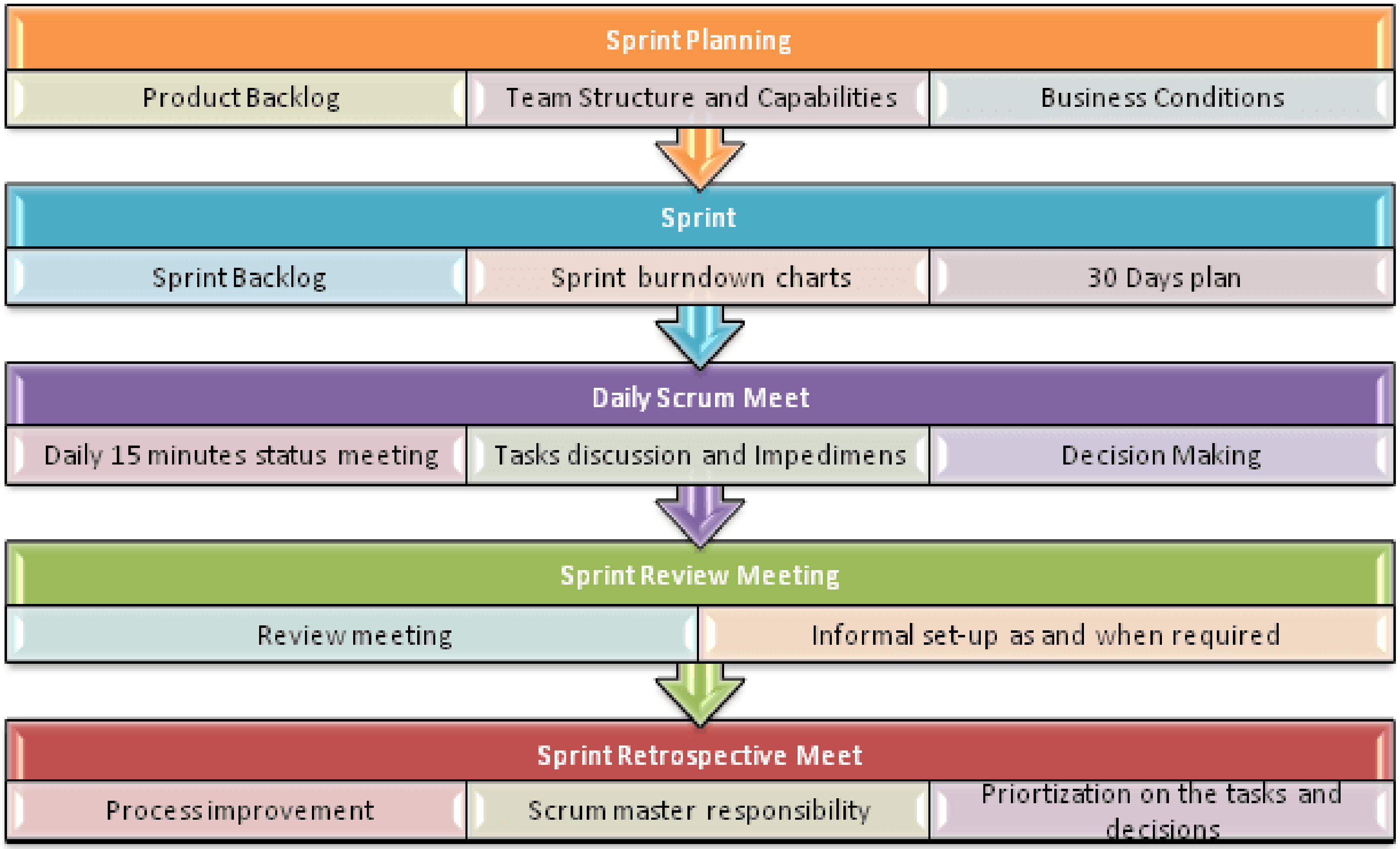
Scrum Team members make actionable commitments

## Team Meetings

# Scrum


Scrum Master
Scrum Meeting
Product Owner
Prepare product backlog
Team organizes tasks

# Scrum Practices


Sprint Planning
Product Backlog
Team Structure and Capabilities
Business Conditions
Sprint
Sprint Backlog
Sprint burndown charts
30 Days plan
Daily Scrum Meet
Daily 15 minutes status meeting
Tasks discussion and Impedimens
Decision Making
Sprint Review Meeting
Review meeting
Informal set-up as and when required
Sprint Retrospective Meet
Process improvement
Scrum master responsibility
Priortization on the tasks and decisions

## Process flow of Scrum Methodologies:

Process flow of scrum testing is as follows:

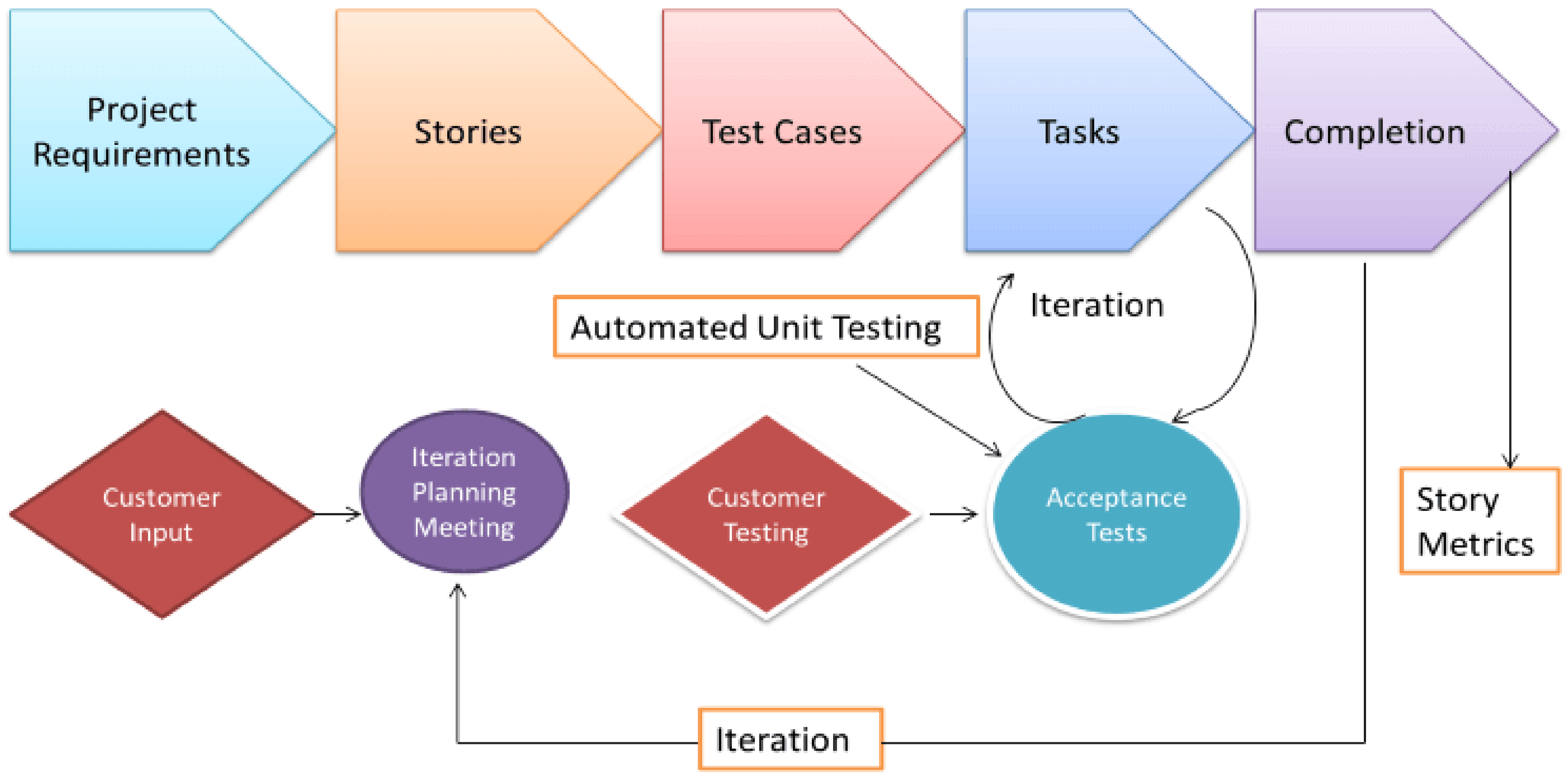
- Each iteration of a scrum is known as Sprint
- Product backlog is a list where all details are entered to get the end-product
- During each Sprint, top user stories of Product backlog are selected and turned into Sprint backlog
- Team works on the defined sprint backlog
- Team checks for the daily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

## eXtreme Programming (XP)

- Extreme Programming technique is very helpful when there is constantly changing demands or requirements from the customers or when they are not sure about the functionality of the system. It advocates frequent "releases" of the product in short development cycles, which inherently improves the productivity of the system and also introduces a checkpoint where any customer requirements can be easily implemented. The XP develops software keeping customer in the target.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

# eXtreme Programming (XP)



Extreme Programming (XP)

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## eXtreme Programming (XP)

- 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.
- The outcome of the XP is always a quality product meeting the exact customer requirements.
- ***XP is Customer Centric***

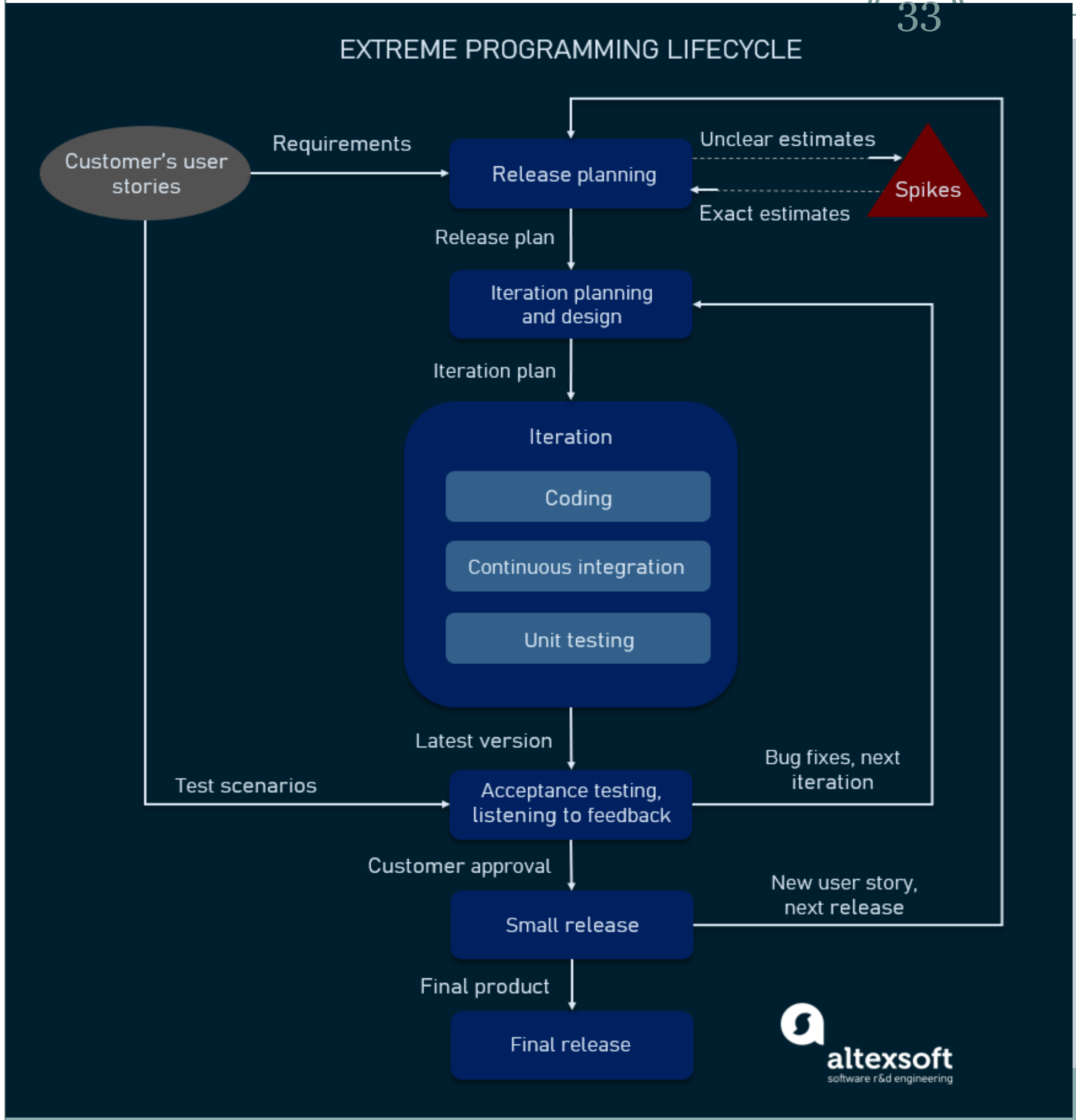
# The process and roles of extreme programming

- he XP framework normally involves 5 phases or stages of the development process that iterate continuously:
- **Planning,** the first stage, is when the customer meets the development team and presents the requirements in the form of user stories to describe the desired result. The team then estimates the stories and creates a release plan broken down into iterations needed to cover the required functionality part after part. If one or more of the stories can't be estimated, so-called *spikes* can be introduced which means that further research is needed.
- **Designing** is actually a part of the planning process, but can be set apart to emphasize its importance. It's related to one of the main XP values that we'll discuss below — simplicity. A good design brings logic and structure to the system and allows to avoid unnecessary complexities and redundancies.

# The process and roles of extreme programming

- **Coding** is the phase during which the actual code is created by implementing specific XP practices such as coding standards, pair programming, continuous integration, and collective code ownership (the entire list is described below).
- **Testing** is the core of extreme programming. It is the regular activity that involves both unit tests (automated testing to determine if the developed feature works properly) and acceptance tests (customer testing to verify that the overall system is created according to the initial requirements).
- **Listening** is all about constant communication and feedback. The customers and project managers are involved to describe the business logic and value that is expected.

# Phases of eXtreme programming:



Such a development process entails the cooperation between several participants, each having his or her own tasks and responsibilities. Extreme programming puts people in the center of the system, emphasizing the value and importance of such social skills as communication, cooperation, responsiveness, and feedback. So, these roles are commonly associated with XP:

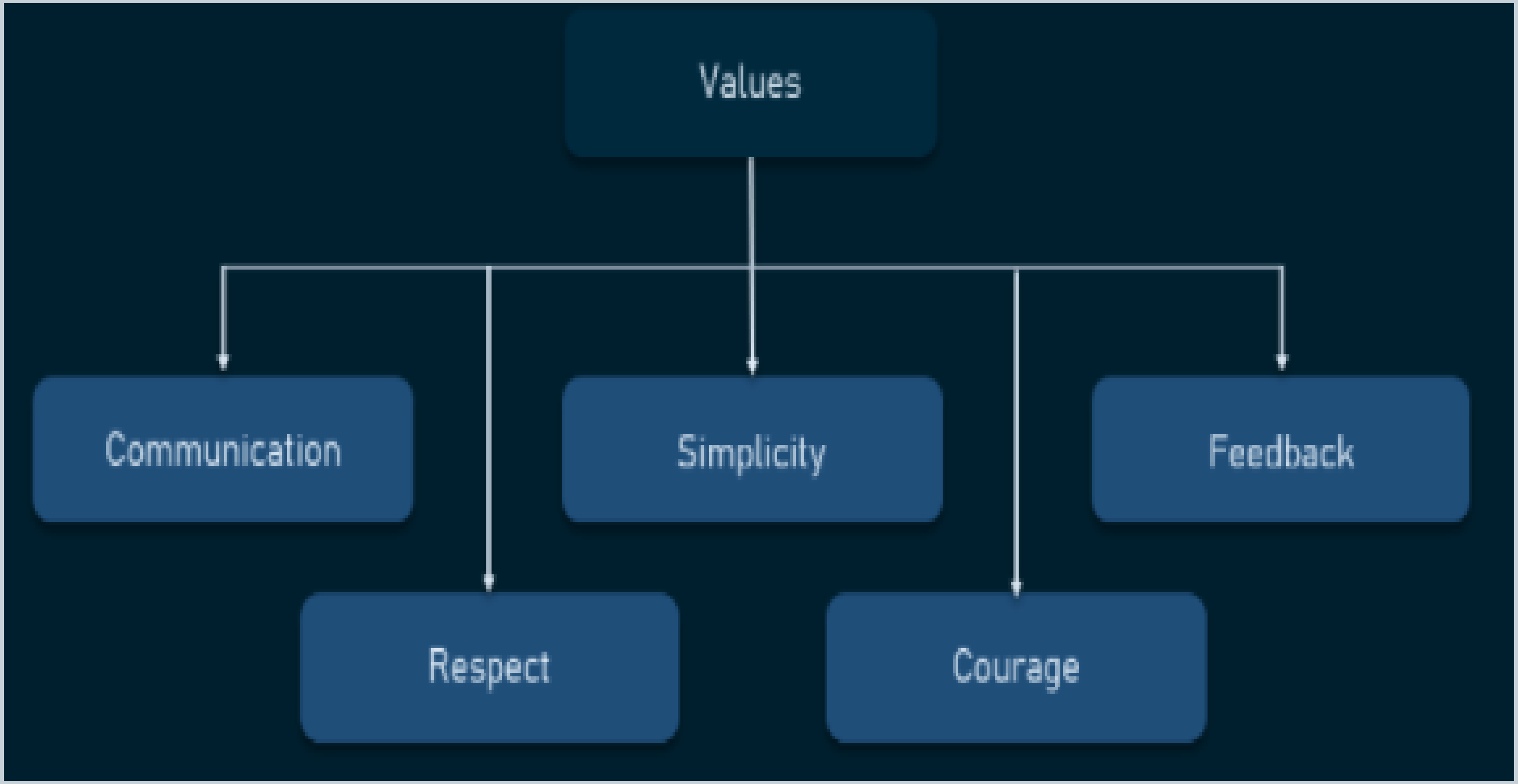
## XP Roles

- **Customers** are expected to be heavily engaged in the development process by creating user stories, providing continuous feedback, and making all the necessary business decisions related to the project.
- **Programmers or developers** are the team members that actually create the product. They are responsible for implementing user stories and conducting user tests (sometimes a separate **Tester** role is set apart). Since XP is usually associated with cross-functional teams, the skill set of such members can be different.

## XP Roles

- **Trackers or managers** link customers and developers. It's not a required role and can be performed by one of the developers. These people organize the meetups, regulate discussions, and keep track of important progress KPIs.
- **Coaches** can be included in the teams as mentors to help with understanding the XP practices. It's usually an outside assistant or external consultant who is not involved in the development process, but has used XP before and so can help avoid mistakes.

# Values of extreme programming



## Values of extreme programming

- XP has simple rules that are based on 5 values to guide the teamwork:
- **Communication.** Everyone on a team works jointly at every stage of the project.
- **Simplicity.** Developers strive to write simple code bringing more value to a product, as it saves time and effort.
- **Feedback.** Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements.
- **Respect.** Every person assigned to a project contributes to a common goal.
- **Courage.** Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

## Principles of extreme programming

- Most researchers denote 5 XP principles as:
- **Rapid feedback.** Team members understand the given feedback and react to it right away.
- **Assumed simplicity.** Developers need to focus on the job that is important at the moment and follow YAGNI (You Ain’t Gonna Need It) and DRY (Don’t Repeat Yourself) principles.
- **Incremental changes.** Small changes made to a product step by step work better than big ones made at once.
- **Embracing change.** If a client thinks a product needs to be changed, programmers should support this decision and plan how to implement new requirements.
- **Quality work.** A team that works well, makes a valuable product and feels proud of it.

## Adaptive Software Development

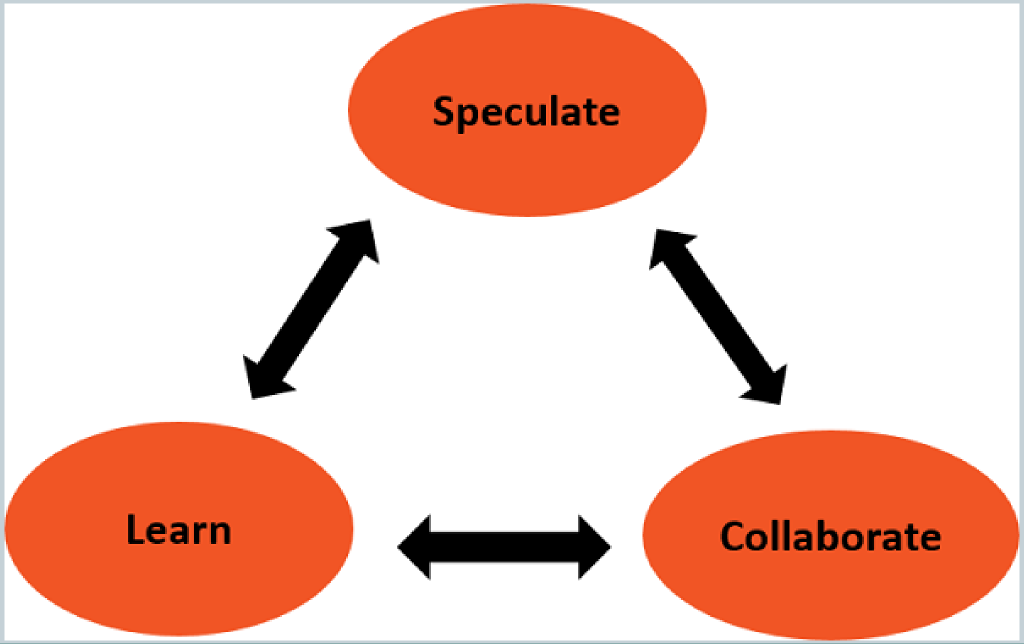
- What is Adaptive Software Development (ASD)?

- Adaptive Software Development (ASD) is a direct outgrowth of an earlier agile framework, Rapid Application Development (RAD). It aims to enable teams to quickly and effectively adapt to changing requirements or market needs by evolving their products with lightweight planning and continuous learning. The ASD approach encourages teams to develop according to a three-phase process: speculate, collaborate, learn.

# Adaptive software development

Adaptive software development grew out of the work by Jim Highsmith and Sam Bayer, two project managers, on RAD in the early 1990s. ASD was designed to replace the traditional waterfall cycle with a three-element repeating series of:

1. Speculation
2. Collaboration
3. Learning



## Adaptive software development

**These are explained as following below.**

**1. Speculation:**

During this phase project is initiated and planning is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement etc, to define set of release cycles that the project wants.

- The term plan is too deterministic and indicates a reasonably high degree of certainty about the desired result. The implicit and explicit goal of conformance to plan, restricts the manager's ability to steer the project in innovative directions.
- In Adaptive Software Development, the term plan is replaced by the term speculate. While speculating, the team does not abandon planning, but it acknowledges the reality of uncertainty in complex problems. Speculate encourages exploration and experimentation. Iterations with short cycles are encouraged.

## Adaptive software development

- **2. Collaboration:**
- Complex applications are not built, they evolve. Complex applications require that a large volume of information be collected, analyzed, and applied to the problem. Turbulent environments have high rates of information flow. Hence, complex applications require that a large volume of information be collected, analyzed, and applied to the problem. This results in diverse Knowledge requirements that can only be handled by team collaboration.

- Collaborate would require the ability to work jointly to produce results, share knowledge or make decisions.
- In the context of project management, Collaboration portrays a balance between managing with traditional management techniques and creating and maintaining the collaborative environment needed for emergence.

# Adaptive software development

- **3. Learning:**
  The workers may have an overestimate of their own understanding of the technology which may not lead to the desired result. Learning helps the workers to increase their level of understanding over the project.
  Learning process is of 3 ways:
- Focus groups
- Technical reviews
- Project postmortem

ASD's overall emphasis on the dynamics of self-organizing teams, interpersonal collaboration, and individual and team learning yield software project teams that have a much higher likelihood of success.

# Advantages of adaptive software development

There are several benefits to the ASD approach. These include:

- A better and stronger overall end product.
- Increased transparency between developers and customers.
- A user-first approach which leads to a more intuitive piece of software.
- Higher likelihood of on-time (or early!) delivery, thanks to the repeating three-step process which allows potential problems to be identified and solved early.

## Disadvantages of adaptive software development drawbacks

No agile framework is perfect! With ASD, there are a few drawbacks that need to be weighed against the benefits:

- Extensive testing can lead to higher project costs.
- The level of user involvement required can be difficult to resource.
- An emphasis on constant product iteration and feedback can lead to burnout.
- Lastly, ASD works better where teams can be dedicated solely to a single project.

## Crystal Methodologies

Crystal Methodology is based on three concepts

- **Chartering:**
- Various activities involved in this phase are creating a development team, performing a preliminary feasibility analysis, developing an initial plan and fine-tuning the development methodology
- **Cyclic delivery:** The main development phase consists of two or more delivery cycles, during which the
  - Team updates and refines the release plan
  - Implements a subset of the requirements through one or more program test integrate iterations
  - Integrated product is delivered to real users
  - Review of the project plan and adopted development methodology
- **Wrap Up:** The activities performed in this phase are deployment into the user environment, post- deployment reviews and reflections are performed

# Dynamic Software Development Method (DSDM)

- DSDM is a rapid application development strategy for software development and gives an agile project distribution structure. The essential features of DSDM are that users must be actively connected, and teams have been given the right to make decisions. The techniques used in DSDM are:
- Time Boxing
- MoSCoW Rules
- Prototyping
- **The DSDM project contains seven stages:**
- Pre-project
- Feasibility Study
- Business Study
- Functional Model Iteration
- Design and build Iteration
- Implementation
- Post-project

## Feature Driven Development (FDD)

- This method is focused around "designing & building" features. Unlike other Agile methods in software engineering, FDD describes very specific and short phases of work that has to be accomplished separately per feature. It includes domain walkthrough, design inspection, promote to build, code inspection and design. FDD develops product keeping following things in the target
- Domain object Modeling
- Development by feature
- Component/ Class Ownership
- Feature Teams
- Inspections
- Configuration Management
- Regular Builds
- Visibility of progress and results

# Lean Software Development

- Lean software development method is based on the principle "Just in time production". It aims at increasing speed of software development and decreasing cost. Lean development can be summarized in seven steps.
- Eliminating Waste
- Amplifying learning
- Defer commitment (deciding as late as possible)
- Early delivery
- Empowering the team
- Building Integrity
- Optimize the whole

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

THANK YOU