# SDLC

- SDLC stands for **Software Development Life Cycle** and is also referred to as the Application Development life-cycle
- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control
- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

## SDLC Phases

Requirement Analysis

Feasibility Study

Design

Coding

Testing

Install Deploy

Maintenance

## Phase 1: Requirement collection and analysis

- ❖ The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry.
- ❖ This stage gives a clear picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

- ❖ Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

### ❖ Phase 2: Feasibility study

- **There are mainly five types of feasibilities checks:**
- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

# Phase 3: Design

- In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.
- This design phase serves as input for the next phase of the model.

**There are two kinds of design documents developed in this phase:**

**High-Level Design (HLD):**

- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technology details

**Low-Level Design(LLD)**

- Functional logic of the modules
- Database tables, which include type and size
- Complete detail of the interface
- Addresses all types of dependency issues
- Listing of error messages
- Complete input and outputs for every module

## Phase 4: Coding

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

## Phase 5: Testing

During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

## Phase 6: Installation/Deployment

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

## Phase 7: Maintenance

- ❖ Once the system is deployed, and customers start using the developed system, following 3 activities occur
- ❖ Bug fixing - bugs are reported because of some scenarios which are not tested at all
- ❖ Upgrade - Upgrading the application to the newer versions of the Software
- ❖ Enhancement - Adding some new features into the existing software

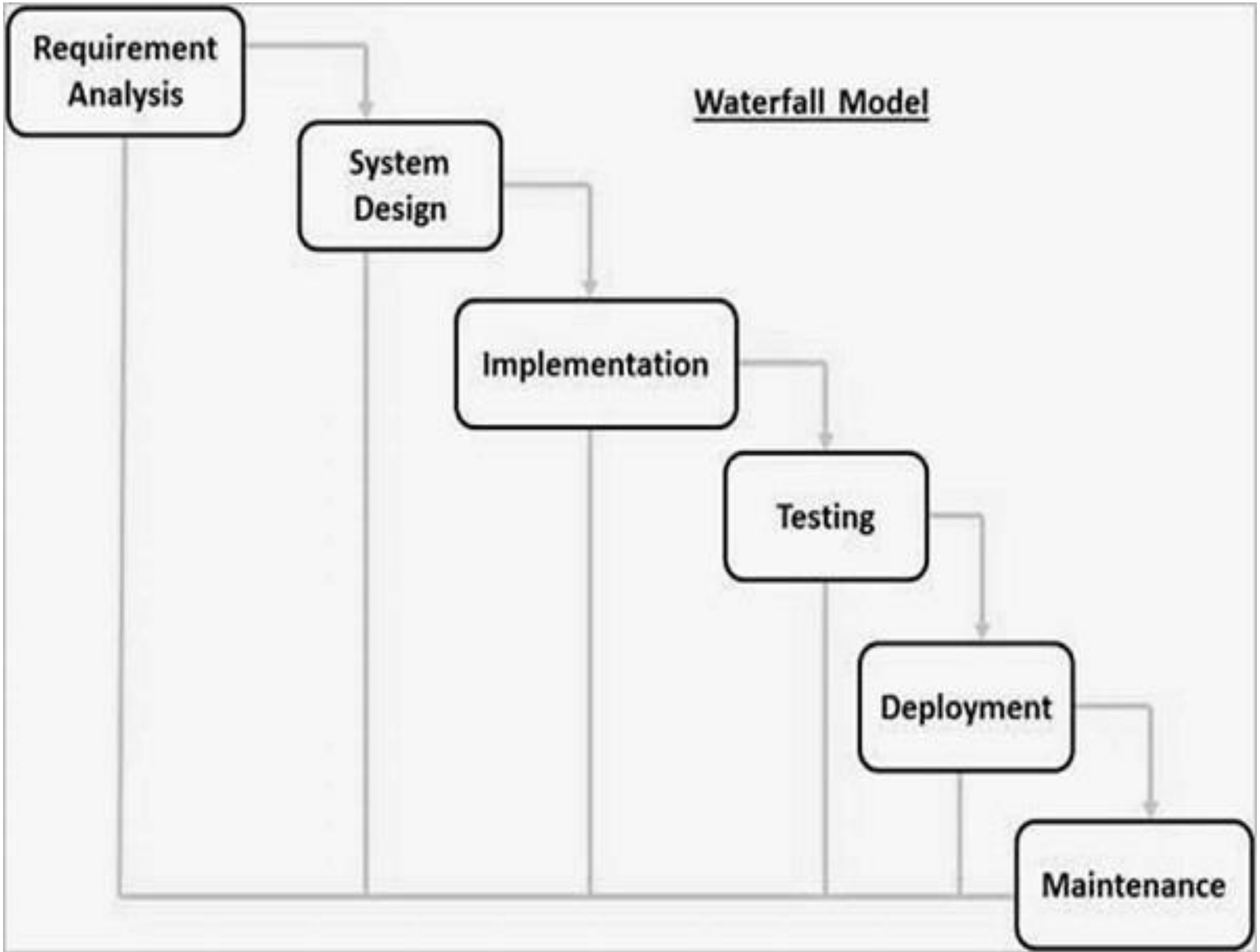
## Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**.

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project.

In "The Waterfall" approach, the whole process of software development is divided into separate phases.

In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.


Requirement Analysis
Waterfall Model
System Design
Implementation
Testing
Deployment
Maintenance

# Applications of Waterfall Model

- ❑Requirements are very well documented, clear and fixed.
- ❑Product definition is stable.
- ❑Technology is understood and is not dynamic.
- ❑There are no ambiguous requirements.
- ❑Resources with required expertise are available to support the product.
- ❑The project is short.

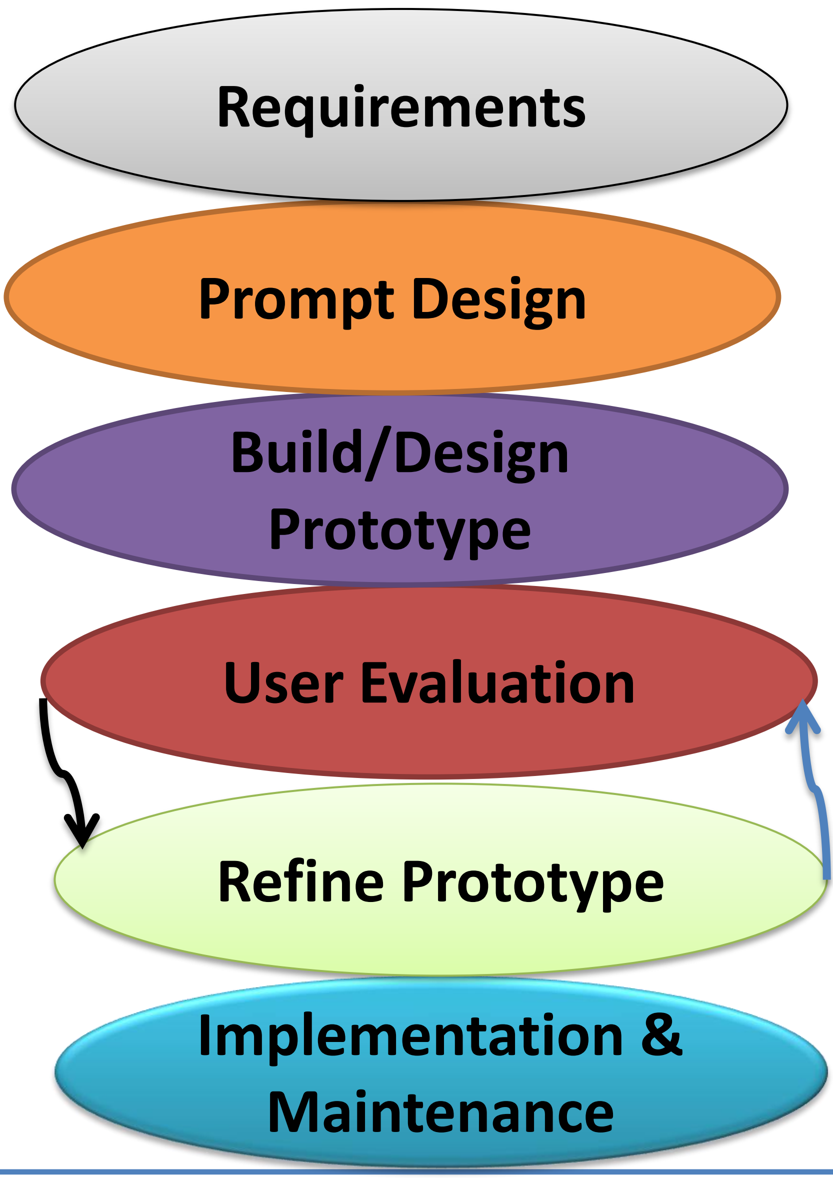
## Waterfall Model - Advantages

- ❑Simple and easy to understand and use
- ❑Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- ❑Phases are processed and completed one at a time.
- ❑Works well for smaller projects where requirements are very well understood.
- ❑Clearly defined stages.
- ❑Well understood milestones.
- ❑Easy to arrange tasks.
- ❑Process and results are well documented.

## Waterfall Model - Disadvantages

- ❑ No working software is produced until late during the life cycle.
- ❑ High amounts of risk and uncertainty.
- ❑ Not a good model for complex and object-oriented projects.
- ❑ Poor model for long and ongoing projects.
- ❑ Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- ❑ It is difficult to measure progress within stages.
- ❑ Cannot accommodate changing requirements.

# Phases in Prototype Model



## Phases of Prototyping Model

- **Requirement**: In this phase requirement is gathered. In this phase the interaction with user is done to know his/her expectation from system.

- **Prompt Design:** This is very second phase of model. In this phase a simple quick design is formed. This is similar to actual design so that user would be able to get the brief idea about system.

- **Design/Build of Prototype:** This is very important phase because in this phase an actual prototype is build that is totally based on the information collected from prompt design.

## Phases of Prototyping Model(Cont..)

- **Initial user evaluation**

In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

- **Refining prototype**

If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions.

- **Implement Product and Maintain**

Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

# Build & Fix Model

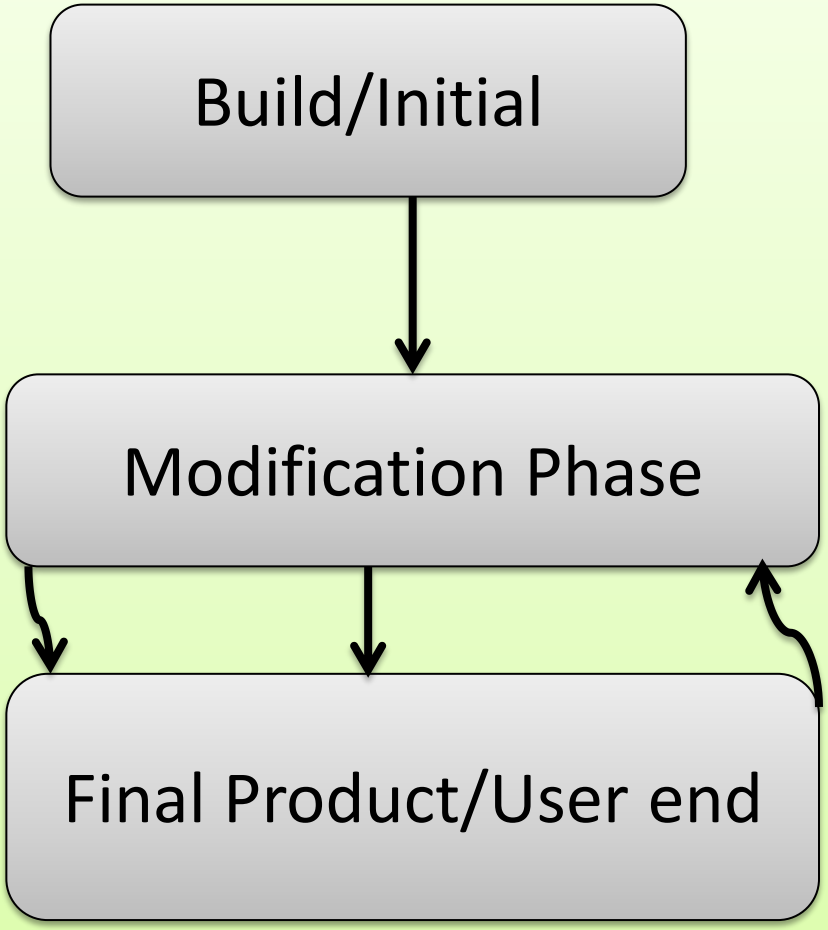
This model is also known as  Ad-Hoc Model

In it software is developed without any specified design.

Or

Initial build up is made and necessary corrections are made until desire or expected output does not meet.

# Build & Fix Model(cont…)



# Phases in Build & Fix Model

## Build Phase

This is very initial phase, in it software code is create/developed and moved to another phase for verification.

## Fix Phase

This is very important phase. All errors are fixed in this phase to meet expected output/result.

## Advantages of Phases in Build & Fix Model

There is no need of high project planning.

Suitable model for small project.

No large management activities are required.

Less management experience is required

# Disadvantages of Phases in Build & Fix Model

Required more time.

Required high cost.

Unplanned model.

No documentation is required.

Maintenance is also tough.

# Iterative Enhancement Model

1:It is also known as increment model.

2:It comprises the features of waterfall model in iterative manner.

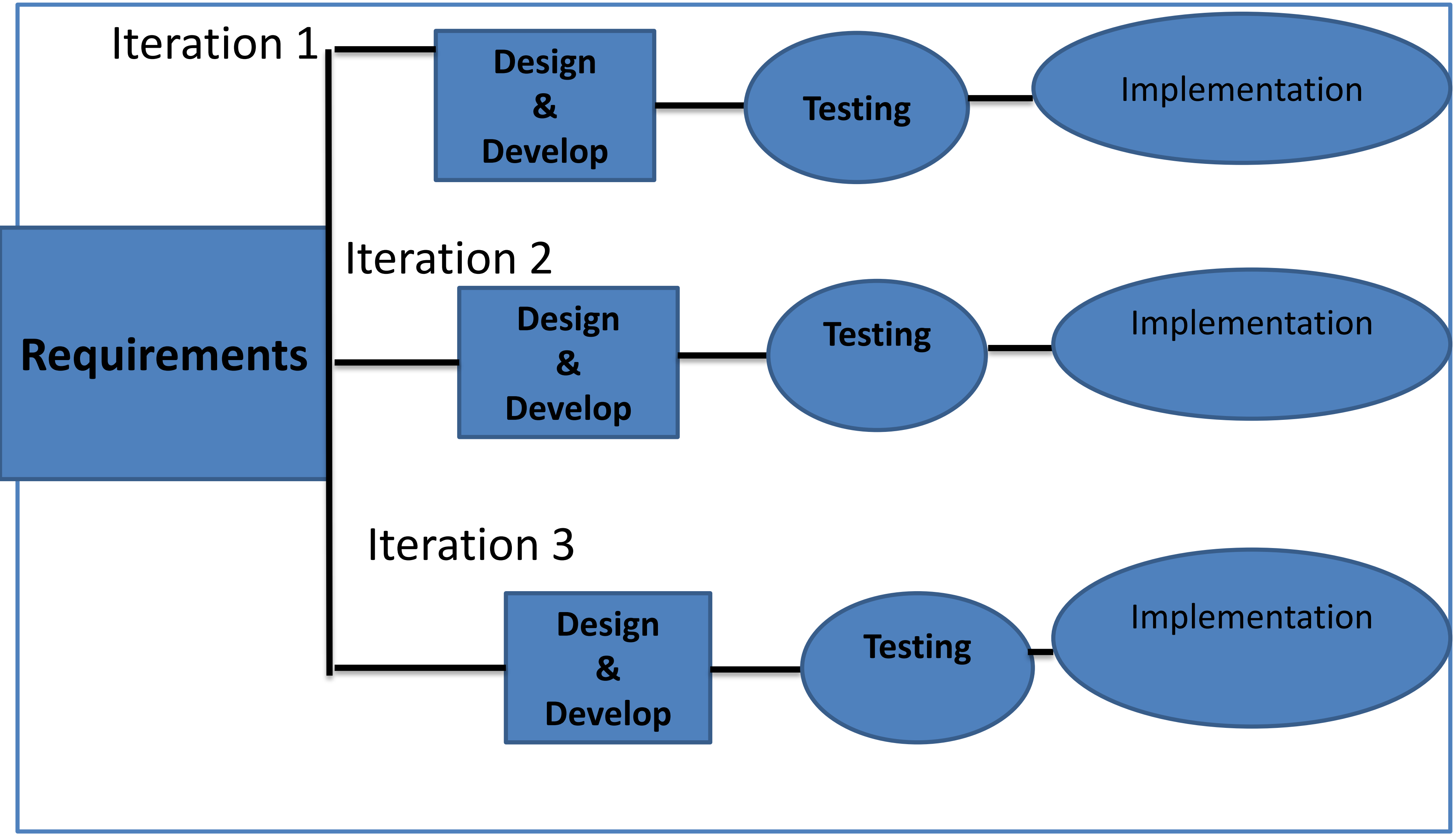
3:At each iteration, modifications in design are made and new functionalities are added.

4:The basic idea is to develop a product in iterative and incremental manner.

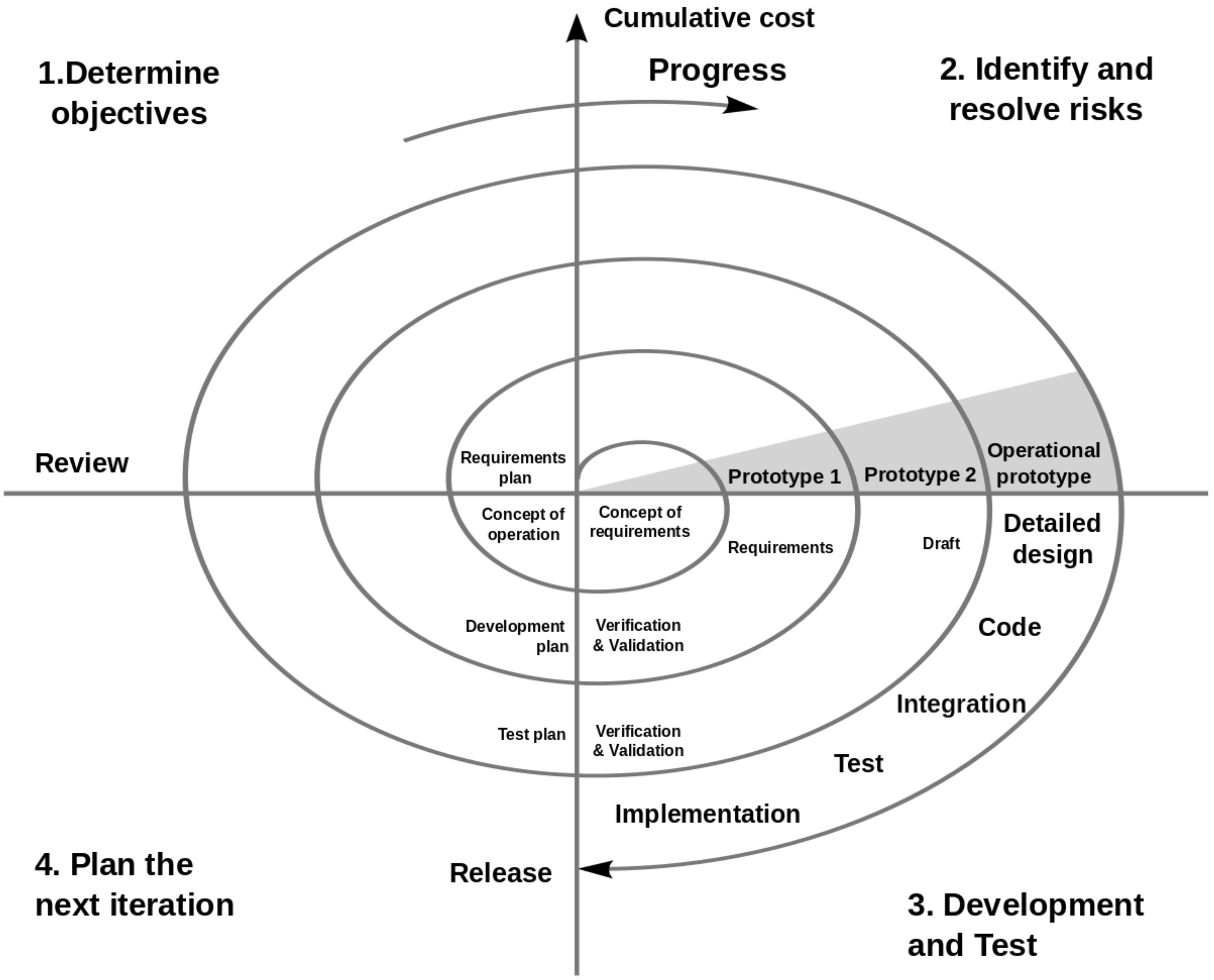
Iterative: Repeated Cycles

Incremental: Smaller portion at a time.

# Iterative Enhancement Model



# Spiral Model



## Spiral Model

**1:Determination of Objectives and identification of solution:** All the requirements are collected from users and the objectives are identified. Analysis of objective is done at each phase.

**2:Identification the solution of risk:** In this quadrant the main task is to collect all possible solutions and select optimal solution among those. Now next task is to evaluate the risk associated with that solution. If risk persist then it is solved with the best method.

**3:Development and Test:** In this quadrant, all the features which have been identified are now tested for verification purpose.

**4:Planning of next phase:** In this phase product is reviewed and planning for next phase is started.

## Spiral model is also called Meta Model

- It subsumes all other SDLC.
- It incorporates the features of waterfall model.
- It incorporates the features of prototype model.
- It incorporates the features of iterative model.

## Spiral Model-Advantages

1:In spiral model risk handling and risk analysis is possible at every phase, because there is continuous and repeated development process.

2:If additional functionalities are required then these can be added or done at later stage.

3:Development process is fast.

4:Customer can observe the development of product at early stage.

5:It is suitable for large product.

## Spiral Model-Disadvantages

1:If we compare this model with other SDLC models than it is complex.

2:Due to its complex nature expert resources are required which lead to high cost. So we can say that it is not suitable for small product.

It has so many intermediate phases so there are large documents in it.

Estimation of time is not so easy at the starting of this model due to unknown number of phases.

## Agile Software Development Model

It is combination of iterative and incremental process

Basically, the main motive of agile is that every project must be dealt with different approaches/methods.

In agile task is broken into small iterations.

Plan for each iteration is clearly defined in advance.

The motive to divide project in small part is to minimize risk and to deliver the project on time.

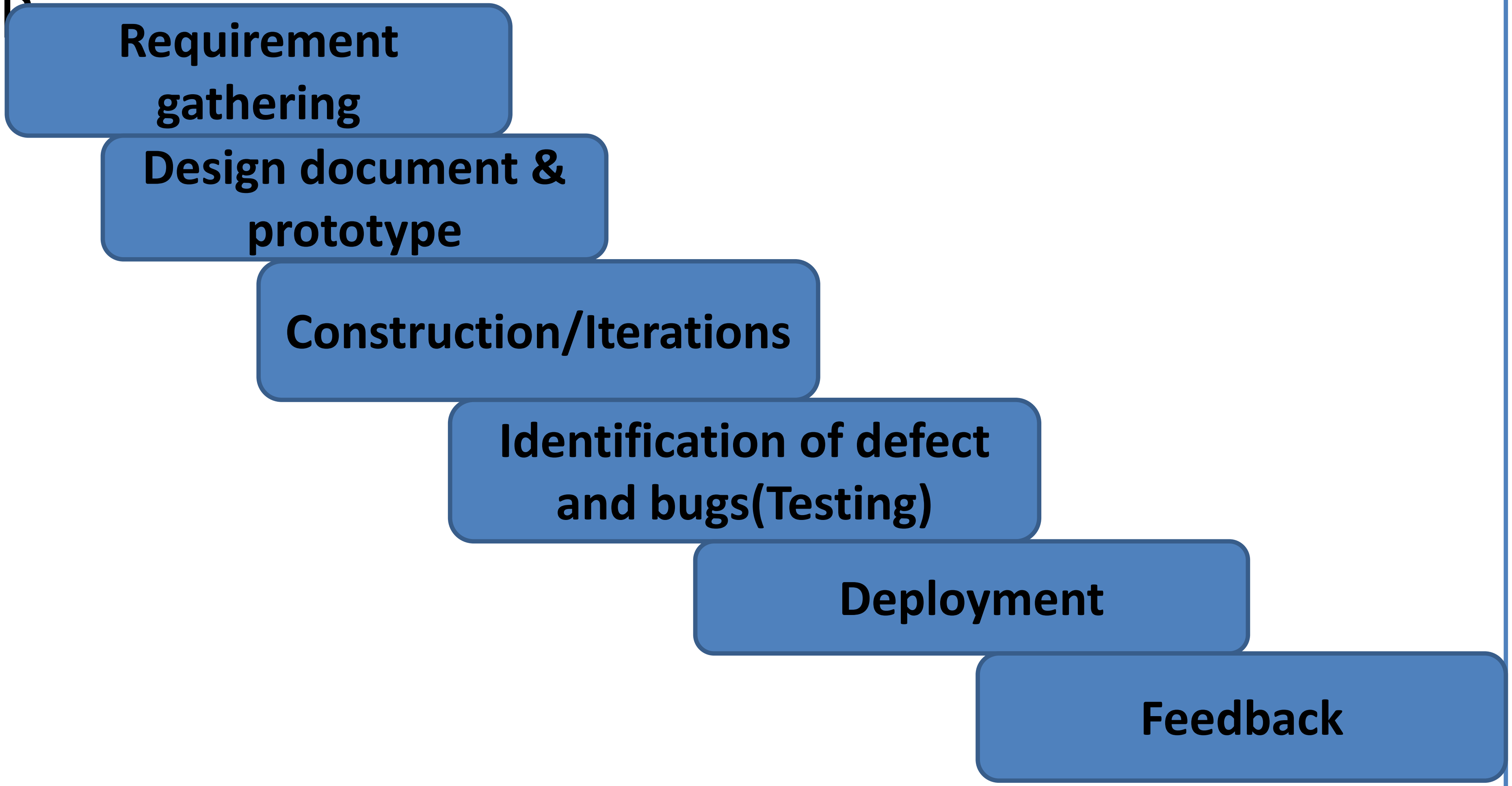
## Agile Software Development Model

As development process is iterative, so the project may be executive in very short time period and no long planning is required.

There is a testing phase with each iteration.

Both testers and developers work together.

# Phases in agile Model


Requirement gathering
Design document & prototype
Construction/Iterations
Identification of defect and bugs(Testing)
Deployment
Feedback

# Agile Software Development Model

**1: Requirements gathering:** This is most important phase of any model. All the future requirements of product is collected ,evaluated .Based on this information the feasibility of project is evaluated.

**2:Design the requirements:** After requirement gathering design of requirement is started. This design can be achieved using various data flow diagram.

**3:Construction/ iteration:** In this phase team members start working on assigned module or part.

**4:Testing:** This is the phase related to quality assurance. This is done to check the working capability of the product and to make a comparison between expected and actual output.

## Agile Software Development Model

**5: Deployment:** In this phase product is issued to work in user's environment.

**6:Feedback:** This is last step in which team receive feedback from user and work on that.

## Scrum

This is a agile methodology or we can say agile framework.

It is widely used by software development team.

Its main focus is on time business delivery(two to three week time).

Basically scrum is used where requirement is rapidly change, because due to its flexibility its quickly adopt changes.

## Features of Scrum

- Light Weighted
- Self-organized
- Easy to understand

## Scrum(cont…)

The meaning of light weighted is that the overhead of the process is kept as small as possible.

Self organization principles to complete work and to coordinate with other teams.

# Scrum(cont…)

Scrum is a framework used to manage product development.

Scrum is run on time boxes of one month or less.

Scrum team consist of **product owner**, **development team** and **scrum master.**

**Product owner**: They are responsible for maintaining the product backlog and ensuring the entire scrum team understands the overall vision for the product

**Development team:** It is a team who has every thing to deliver the product on time without a dependency on outsider team. It is a self-organizing team.

## Scrum(cont…)

**Scrum master:**

Responsible to ensure scrum process.

Responsible to train the team as and when required.

Responsible to remove all barriers.

Responsible for the practice of defined rules.

## Scrum Events and Artifacts

**Scrum Event:** In scrum all events are time box events. Every event has maximum duration.

**Scrum Artifact:** It may also said as “work of art”.It provides information to be aware about the product development.

***What activities have  done?***

***What activities planned?***

***What activities to be?***

# List of Scrum Events and Artifacts

| Scrum Event | Scrum Artifacts |
|---|---|
| Sprint | Product backlog |
| Sprint planning | Sprint backlog |
| Daily Scrum | Increments |
| Sprint review | |
| Sprint retrospective | |

**Sprint:** Scrum project is divided into small iterations. Normally the duration of these iterations is one to three week. This is called Sprint.

**Sprint Planning:** Meaning of Sprint planning is to give answer.

What,we are doing?

How, we are doing?

**Daily Scrum:** Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.

**Sprint Review: The sprint review** is an informal meeting with the development team, the scrum master, the product owner and the stakeholders will attend.

**Sprint Retrospective: Sprint Retrospective** is an opportunity for the **Scrum** Team to inspect itself and create a plan for improvements to be enacted during the next **Sprint**. The **Sprint Retrospective** occurs after the **Sprint** Review and prior to the next **Sprint** Planning.

**Product Backlog:** **A product backlog** is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

**Sprint Backlog:** The **sprint backlog** is a list of tasks identified by the **Scrum** team to be completed during the **Scrum sprint.**

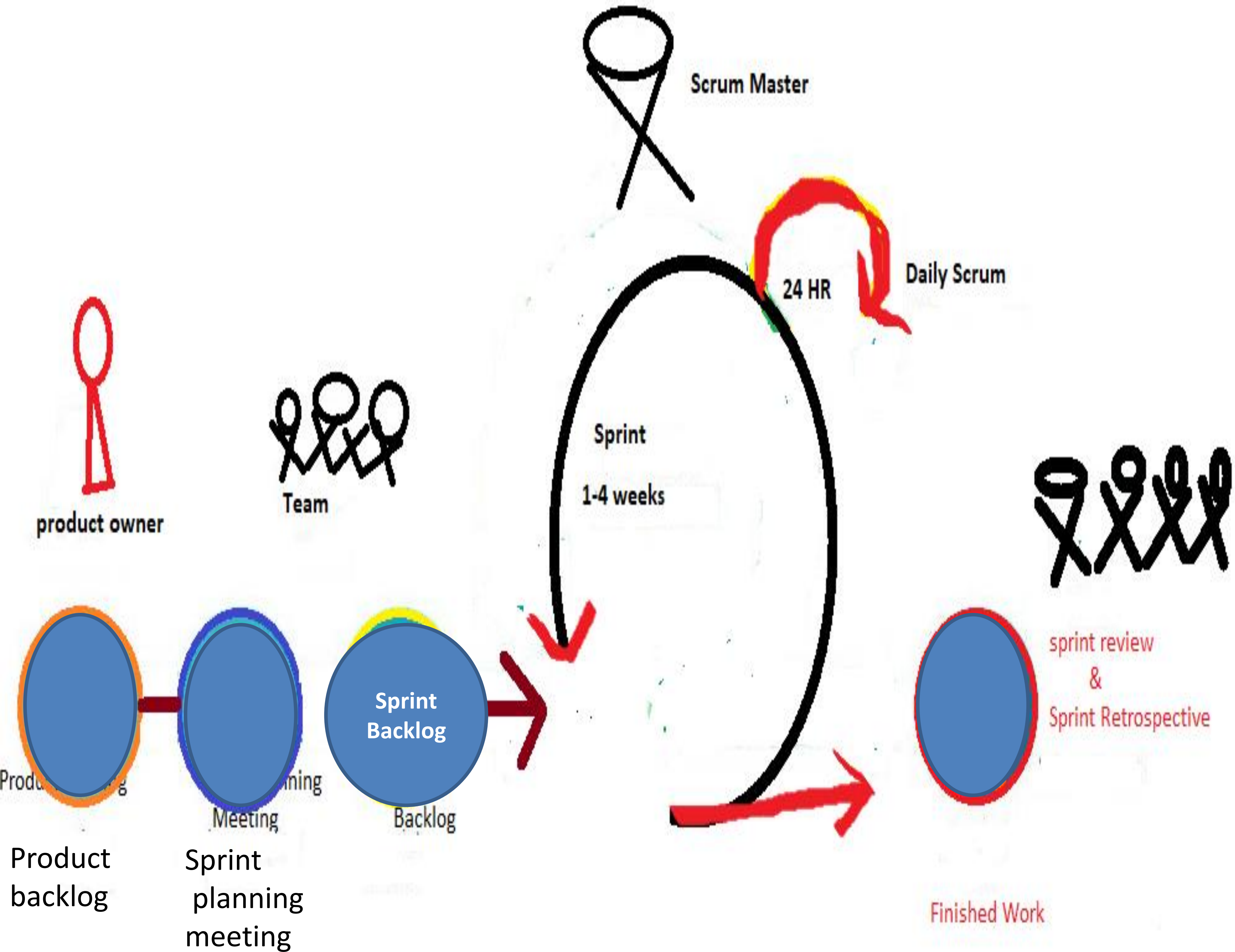
**Increment:** The **Increment** is the sum of all the Product Backlog items completed during a Sprint and the value of the **increments** of all previous Sprints

## Sprint

Scrum project is divided into small iterations. Normally the duration of these iterations is one to three week. This is called Sprint.

It makes project more manageable.

Due to its flexibility it adopt changes easily.


Scrum Master
24 HR
Daily Scrum
product owner
Team
Sprint
1-4 weeks
Sprint Backlog
Meeting
Backlog
sprint review
&
Sprint Retrospective
Finished Work
Product backlog
Sprint planning meeting

# Sprint Planning

Sprint planning is a collaborative event. Which can be done with the collaborative effort of whole Scrum team.

It is a time boxed to a maximum eight hours for a one month Sprint or Sprint planning should be constrained no more than two hours for each week of the sprint.

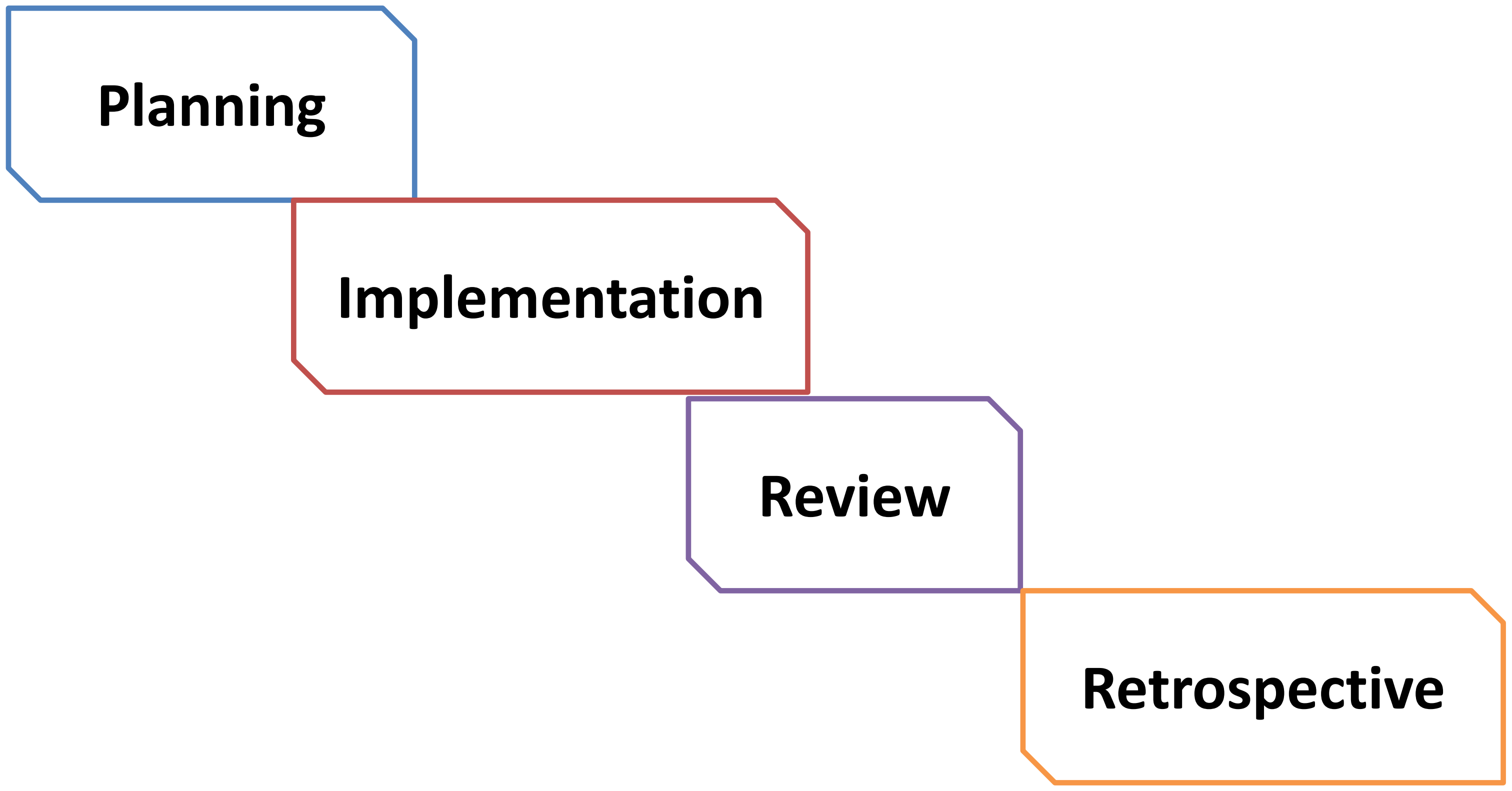
It provides some flexibility to developer regarding the functionality implemented within Sprint.

Meaning of Sprint planning is to give answer.

What are we doing?

How are we doing?

# Sprint Lifecycle



# Sprint Planning Meeting

**The main motive of Sprint planning meeting is to provide structure, guidelines, criteria, expectations and define backlog for forthcoming backlog.**

**There are following people involved in Sprint planning meeting.**

**Product Owner:** **This is most demanding position. Product owner is responsible for managing the product backlog and product backlog visibility. product owner looks at the project from the customer's perspective. product owners are responsible for managing product backlogs**

**Scrum Master:** **Scrum Master is responsible  to motivate the team in tough time and also responsible for timely completion of project. Scrum Master mainly focus on team's operation.**

**Scrum master improve the quality of product.**

## Product Backlog

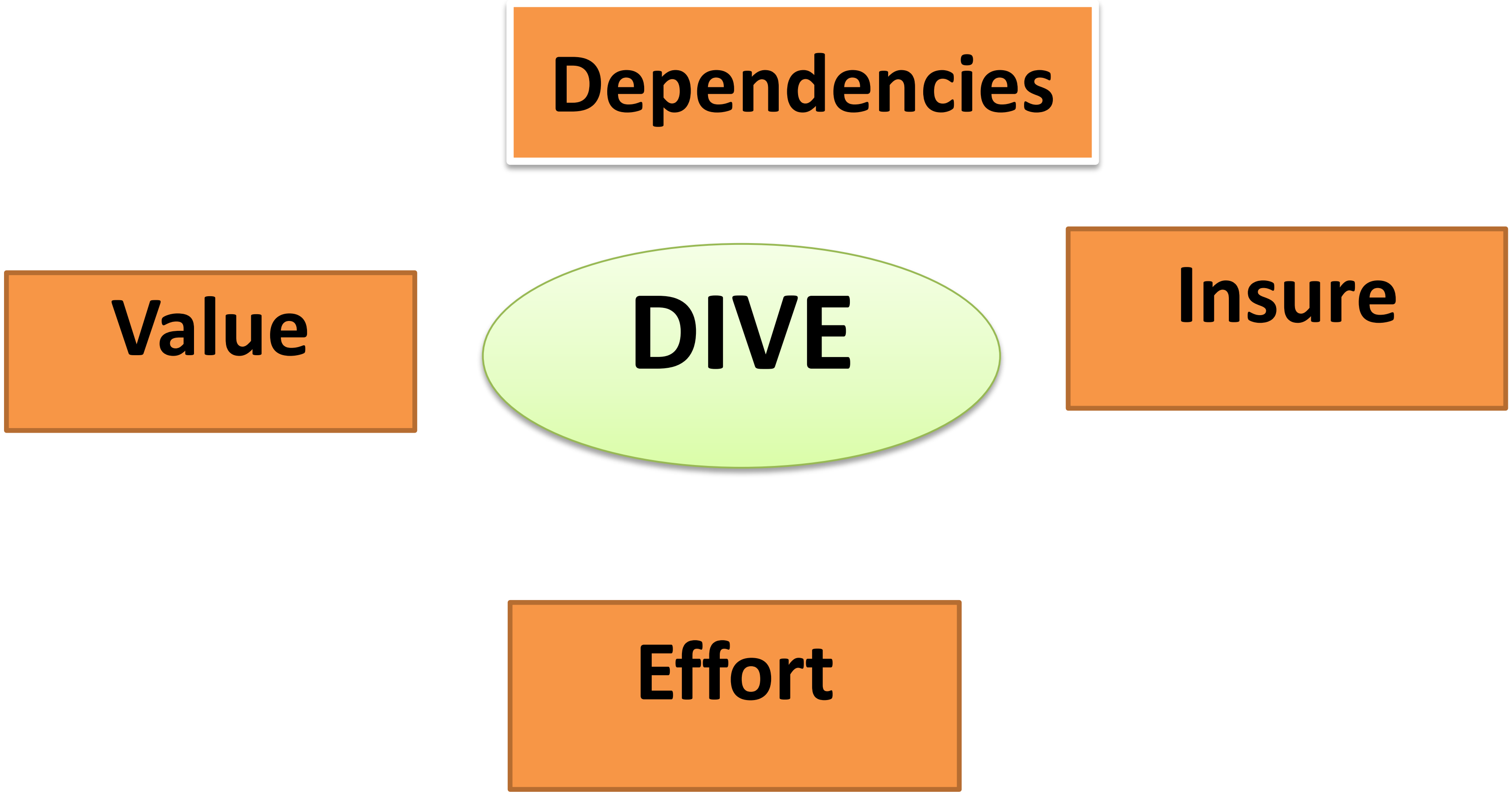
It is compiled of all the things that must be done to complete the whole project.

Product owner owns product backlog.

The product backlog acts as an input to the sprint backlog when comes to functionality

There are also bugs/issues, epic, user stories and themes are included in the product backlog

# Techniques to maintain Product Backlog



# Techniques to maintain Product Backlog

**DIVE**

**Product backlog items are prioritized in linearly ordered based upon DIVE criteria**

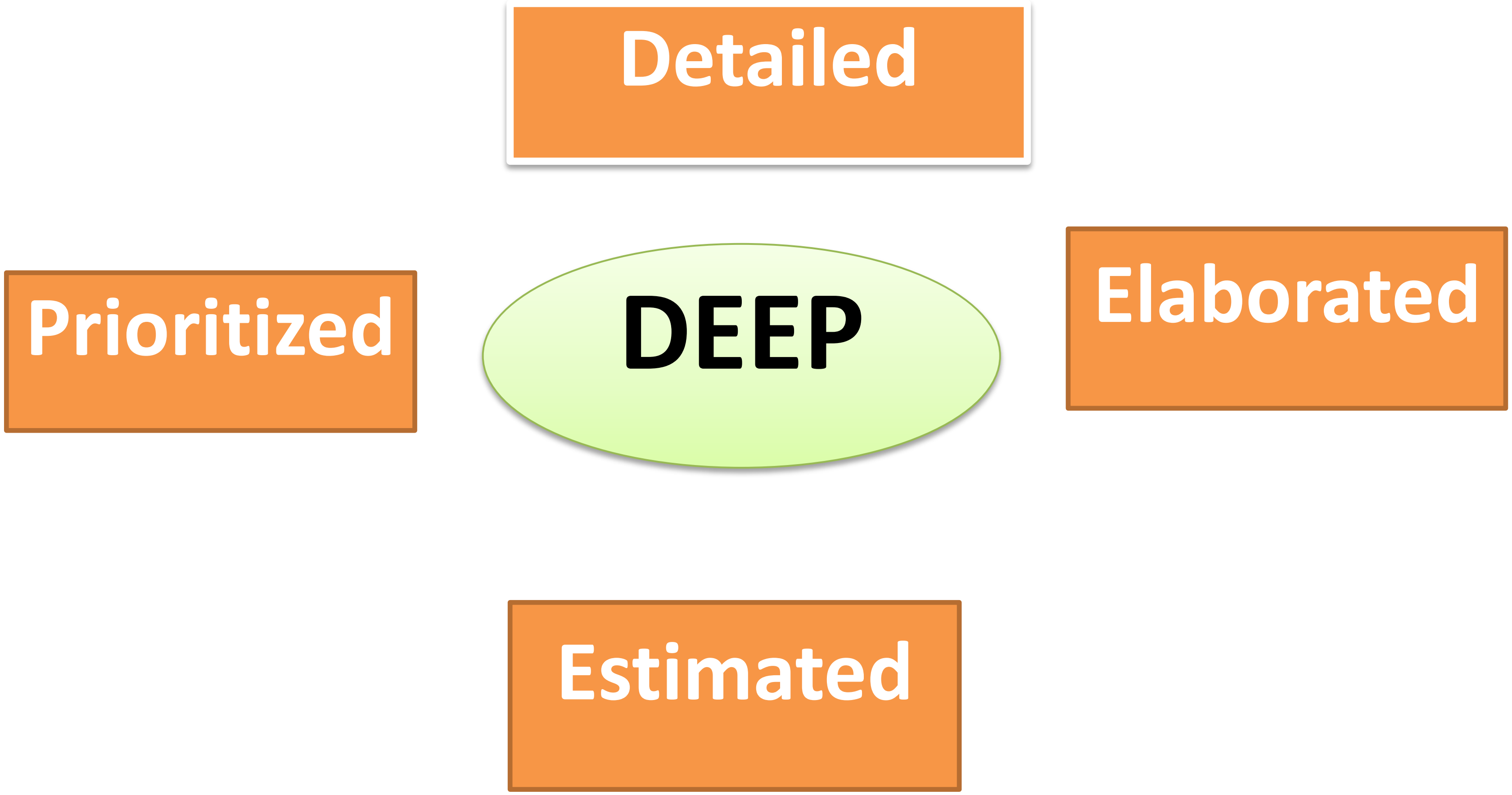
**Dependencies** – keep it linear with fewer dependencies with other user stories, epic or themes. It’s OK to have horizontal dependencies.

**Insure against risk (business and technical)**

**Value(Business)**

**Effort(Estimated)**

# Techniques to maintain Product Backlog



## Techniques to maintain Product Backlog

### INVEST

**Independent :**The user story should be self-contained, in a way that there is no inherent dependency on another user story.

**Negotiable:** User stories, up until they are part of an iteration, can always be changed and rewritten.

**Valuable:** A user story must deliver value to the end user.

**Estimable:** You must always be able to estimate the size of a user story. Small User stories should not be so big as to become impossible to plan/task/prioritize with a certain level of certainty.

**Testable:** The user story or its related description must provide the necessary information to make test development possible.

## Sprint Backlog

Sprint backlog is the subset of the product backlog.

Product Owner set the sprint’s goal for the team, scrum team pick the user stories from product backlog fulfilling those goals.

## Characteristic of a Sprint Backlog

Sprint backlog is dynamic in nature.

Sprint backlog is a subset of product backlog

Sprint backlog is an output of a sprint planning meeting.

In Sprint backlog, scrum team works on how the user stories would be implemented in a sprint by dividing it further into tasks and estimating it.

# Characteristic of a Sprint Backlog

Assuming Product Backlog has stories: 1, 2, 3, 4 , 5 and 6. The team decides to do stories 1,2 and 4.

As during sprint planning team realized that there still some question which is not answered well by the Product Owner, so they decided not to include the user story no: 3 and jump to user story no:4, which was defined well.

Sprint backlog is owned by the Development Team and contains what and how it get's delivered

# Software Requirement Specification

It is  a description of a software system to be developed.

A **software requirements specification** (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements.

The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system.

The software requirement specification document consistent of all necessary requirements required for project development.

To develop the software system we should have clear understanding of Software system.

To achieve this we need to continuous communication with customers to gather all requirements.