Software Engineering

Code-2050302105

Unit-1

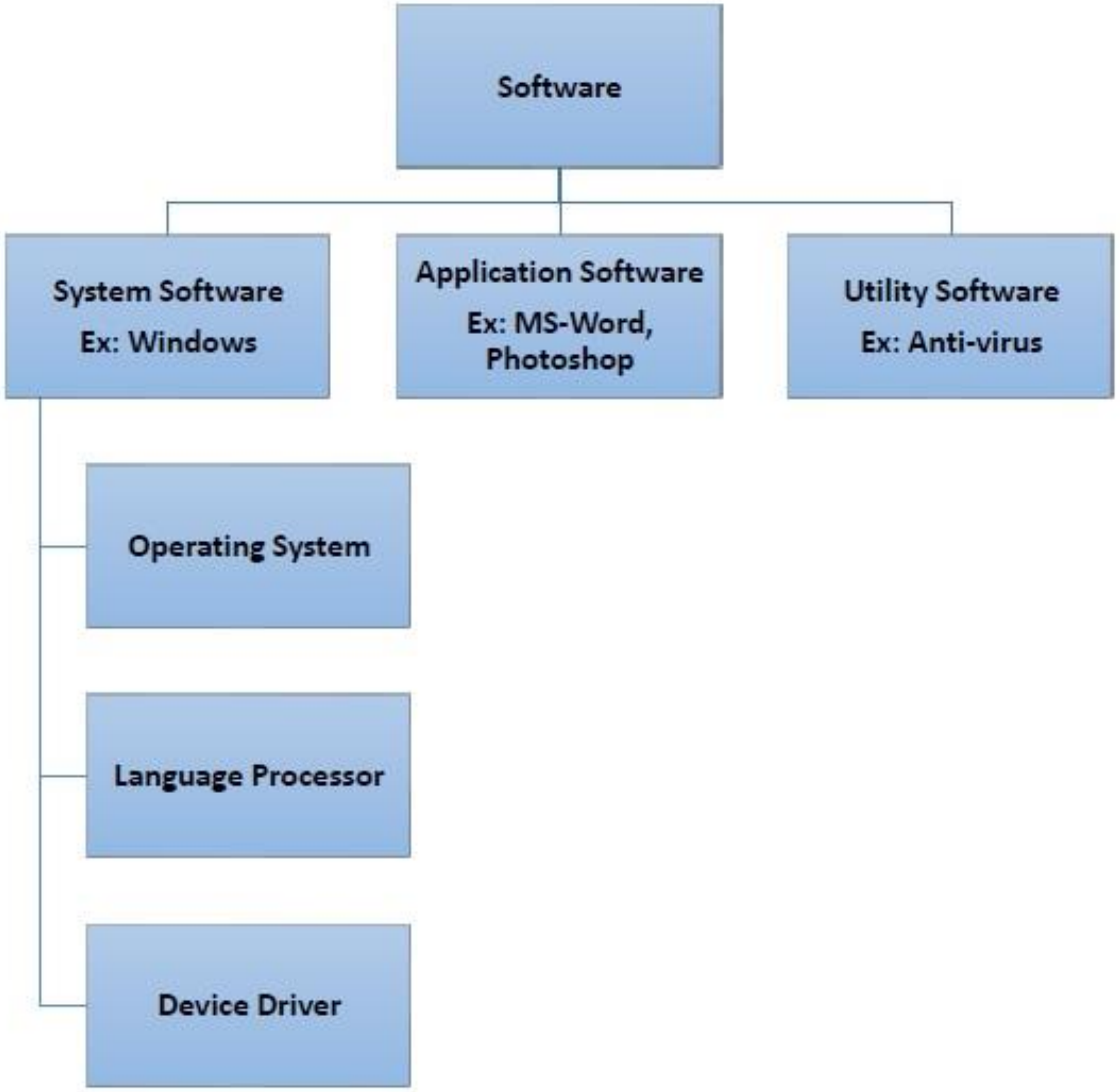
Introduction to SE

BY: Sunil Panchal

Concepts of Software

- Software is a set of instructions, data or programs used to operate computers and execute specific tasks.
- Software is a generic term used to refer to applications, scripts and programs that run on a device.

Conti..


Software
System Software
Ex: Windows
Application Software
Ex: MS-Word, Photoshop
Utility Software
Ex: Anti-virus
Operating System
Language Processor
Device Driver

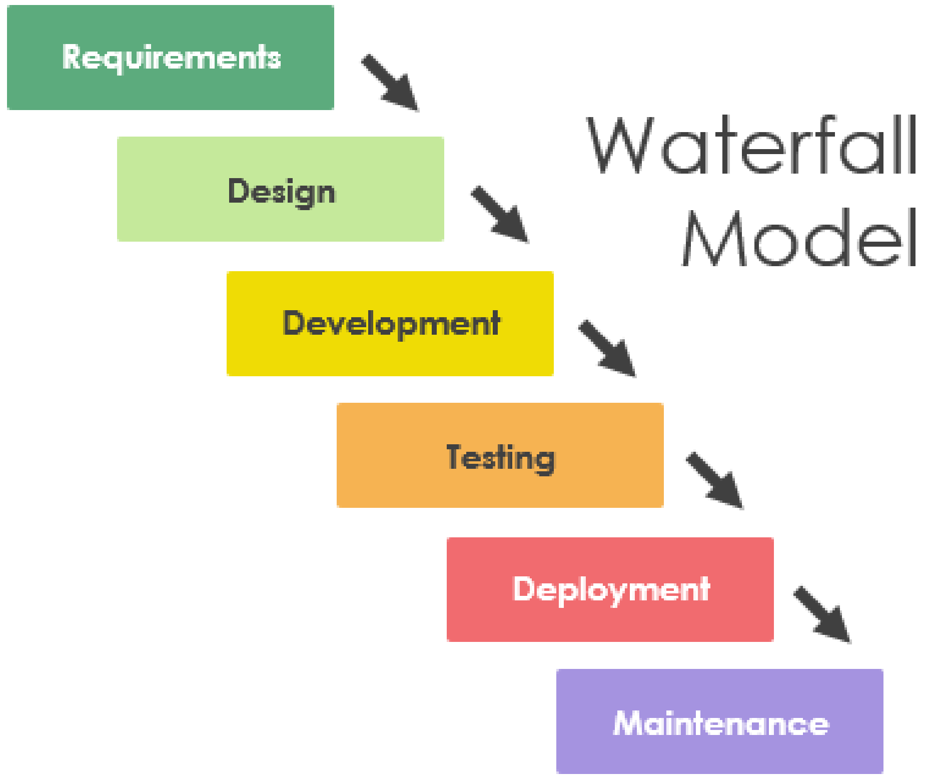
Software Process

- Software Processes is a coherent set of activities for specifying, designing, implementing and testing software systems. A software process model is an abstract representation of a process that presents a description of a process from some particular perspective. There are many different software processes but all involve:
- Specification – defining what the system should do;
- Design and implementation – defining the organization of the system and implementing the system;
- Validation – checking that it does what the customer wants;
- Evolution – changing the system in response to changing customer needs.

Process Model

- **Waterfall Model:** The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design.

Waterfall Model

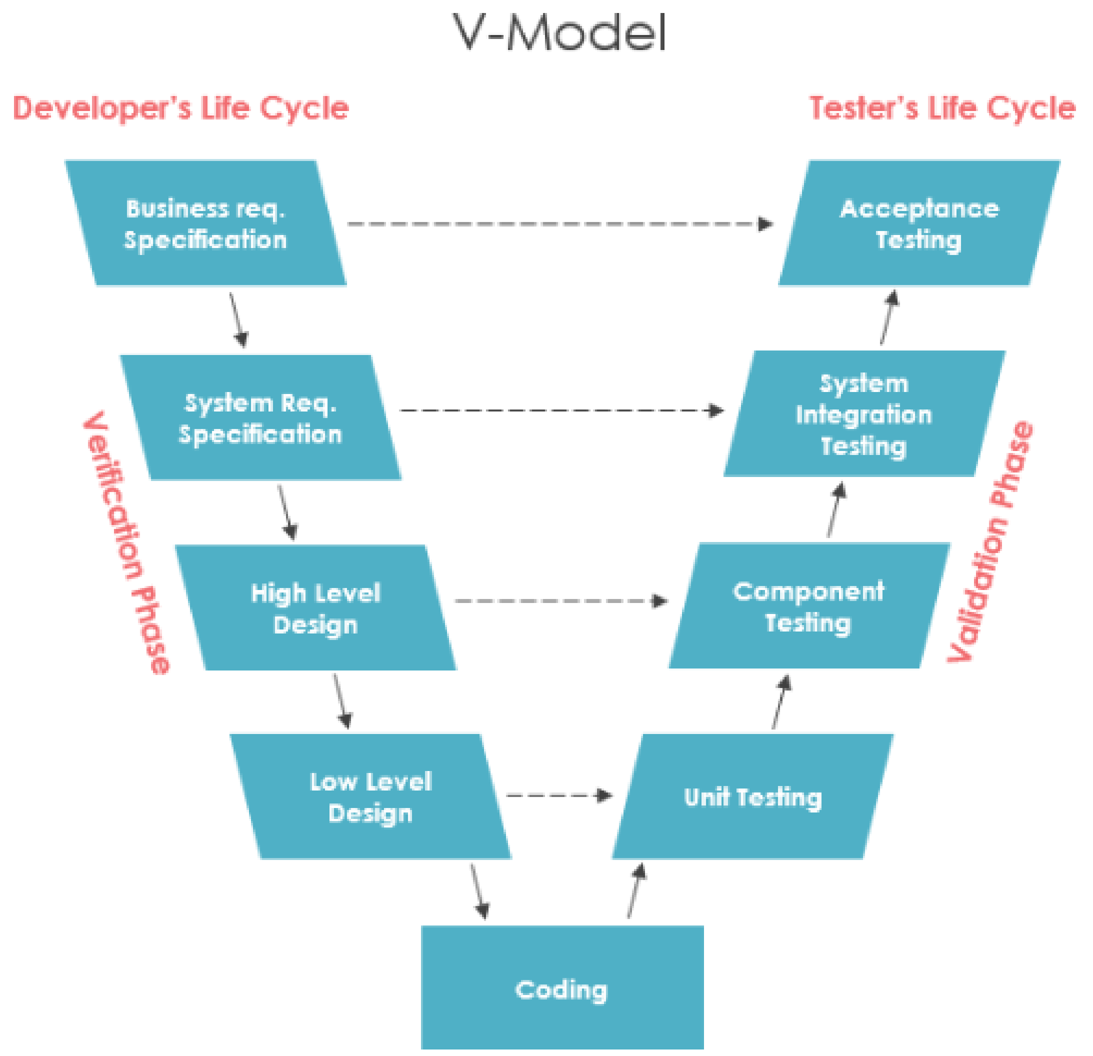


BY: Sunil Panchal

Process Model

- **V Model:** The V-model represents a development process that may be considered an extension of the waterfall model and is an example of the more general V-model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represent time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

V Model

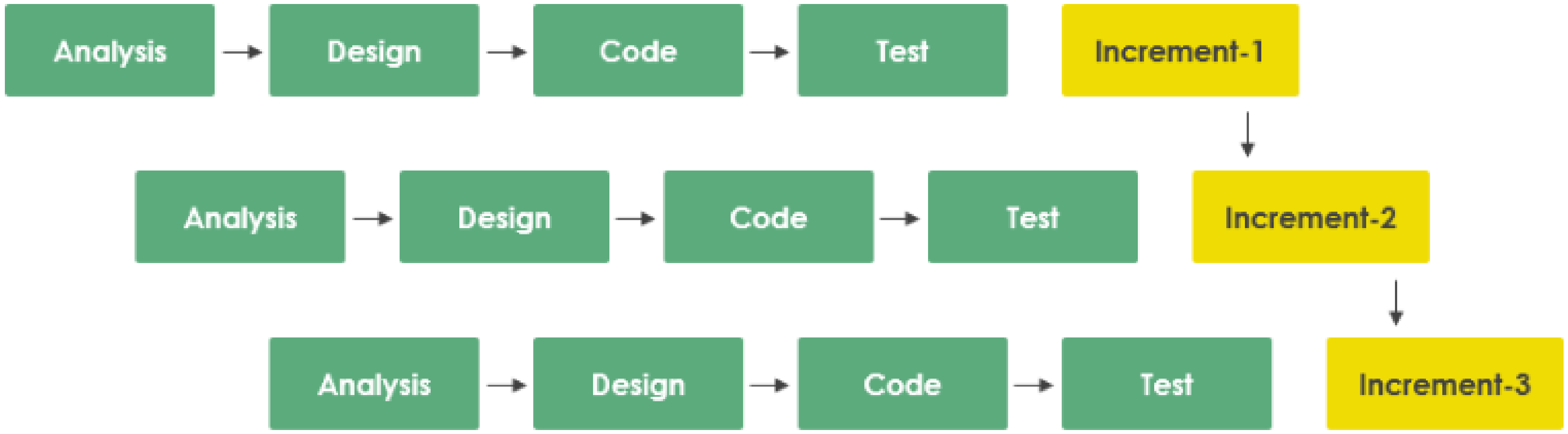


BY: Sunil Panchal

Process Model

- **Incremental Model:** The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionally has been implemented. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

Incremental Model

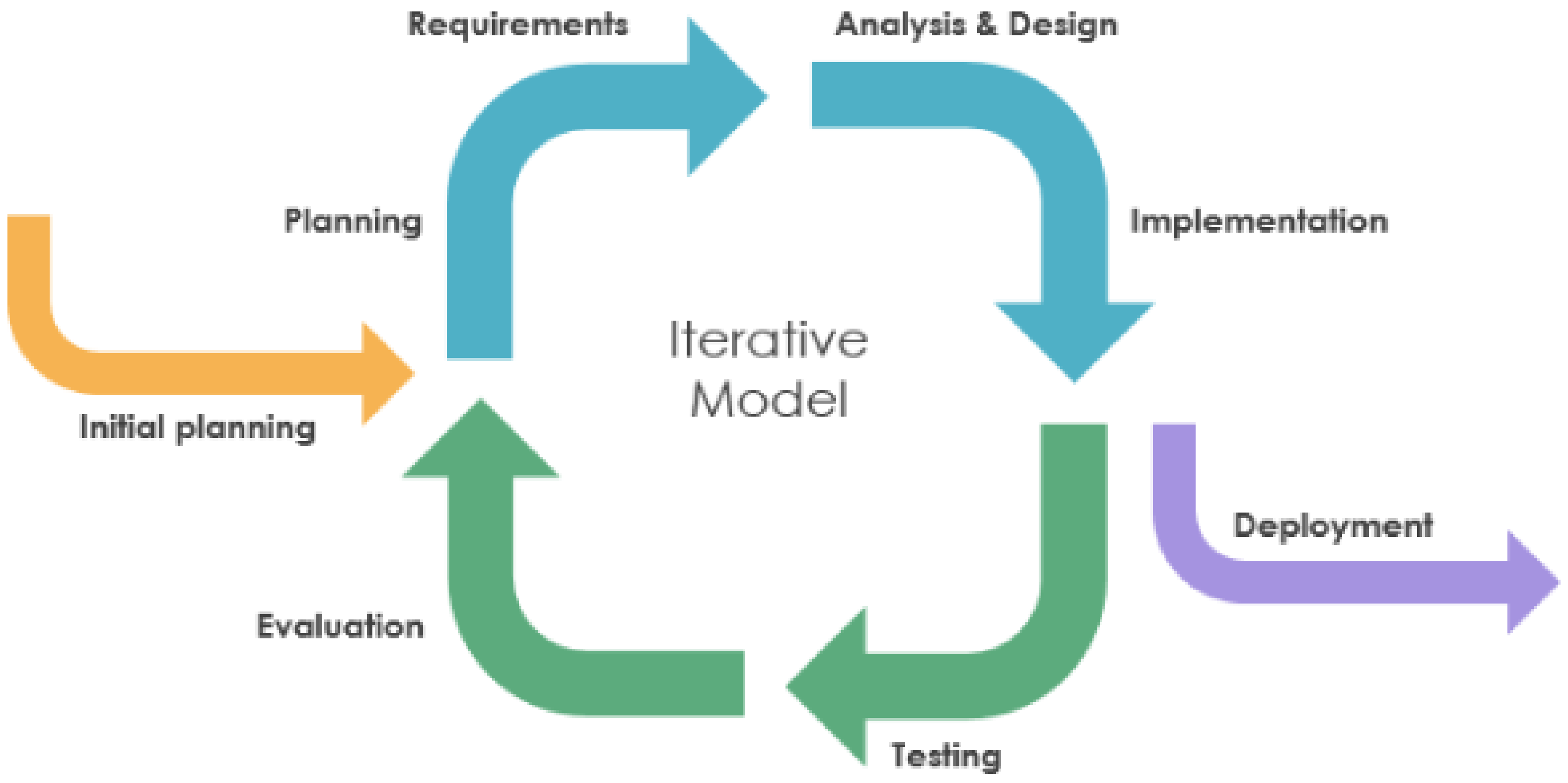


Incremental Model

Process Model

- **Iterative Model:** An iterative life cycle model does not attempt to start with a full specification of requirements by first focusing on an initial, simplified set user features, which then progressively gains more complexity and a broader set of features until the targeted system is complete. When adopting the iterative approach, the philosophy of incremental development will also often be used liberally and interchangeably.
- In other words, the iterative approach begins by specifying and implementing just part of the software, which can then be reviewed and prioritized in order to identify further requirements. This iterative process is then repeated by delivering a new version of the software for each iteration. In a light-weight iterative project the code may represent the major source of documentation of the system; however, in a critical iterative project a formal software specification may also be required.

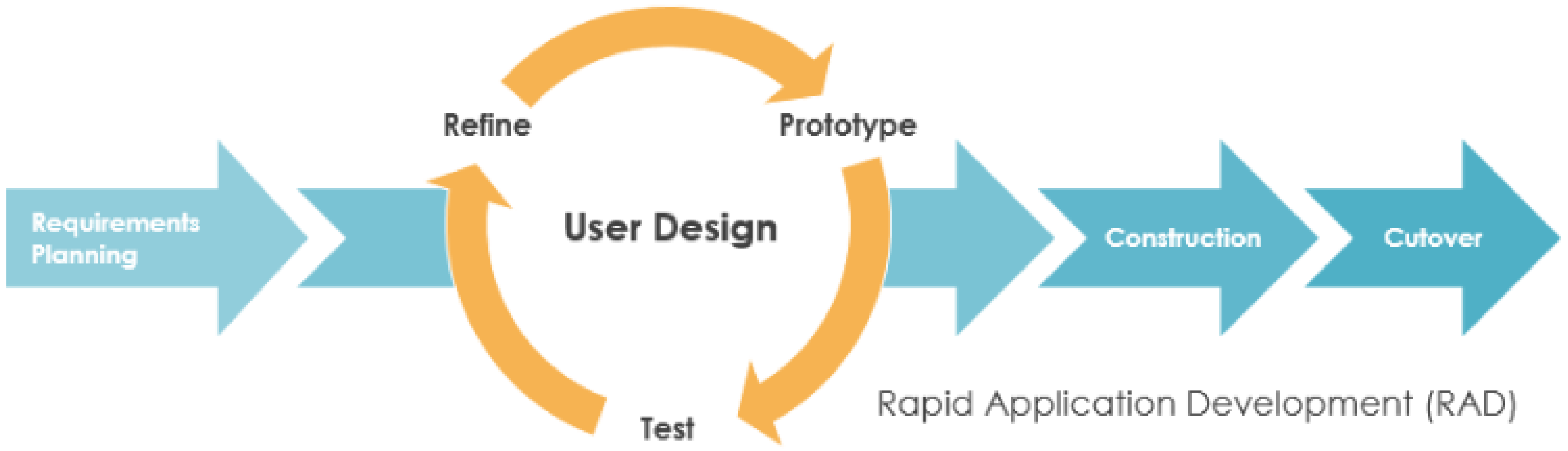
Iterative Model



Process Model

- **RAD Model:** Rapid application development was a response to plan-driven waterfall processes, developed in the 1970s and 1980s, such as the Structured Systems Analysis and Design Method (SSADM). Rapid application development (RAD) is often referred as the adaptive software development. RAD is an incremental prototyping approach to software development that end users can produce better feedback when examining a live system, as opposed to working strictly with documentation. It puts less emphasis on planning and more emphasis on an adaptive process.
- RAD may resulted in a lower level of rejection when the application is placed into production, but this success most often comes at the expense of a dramatic overruns in project costs and schedule. RAD approach is especially well suited for developing software that is driven by user interface requirements. Thus, some GUI builders are often called rapid application development tools.

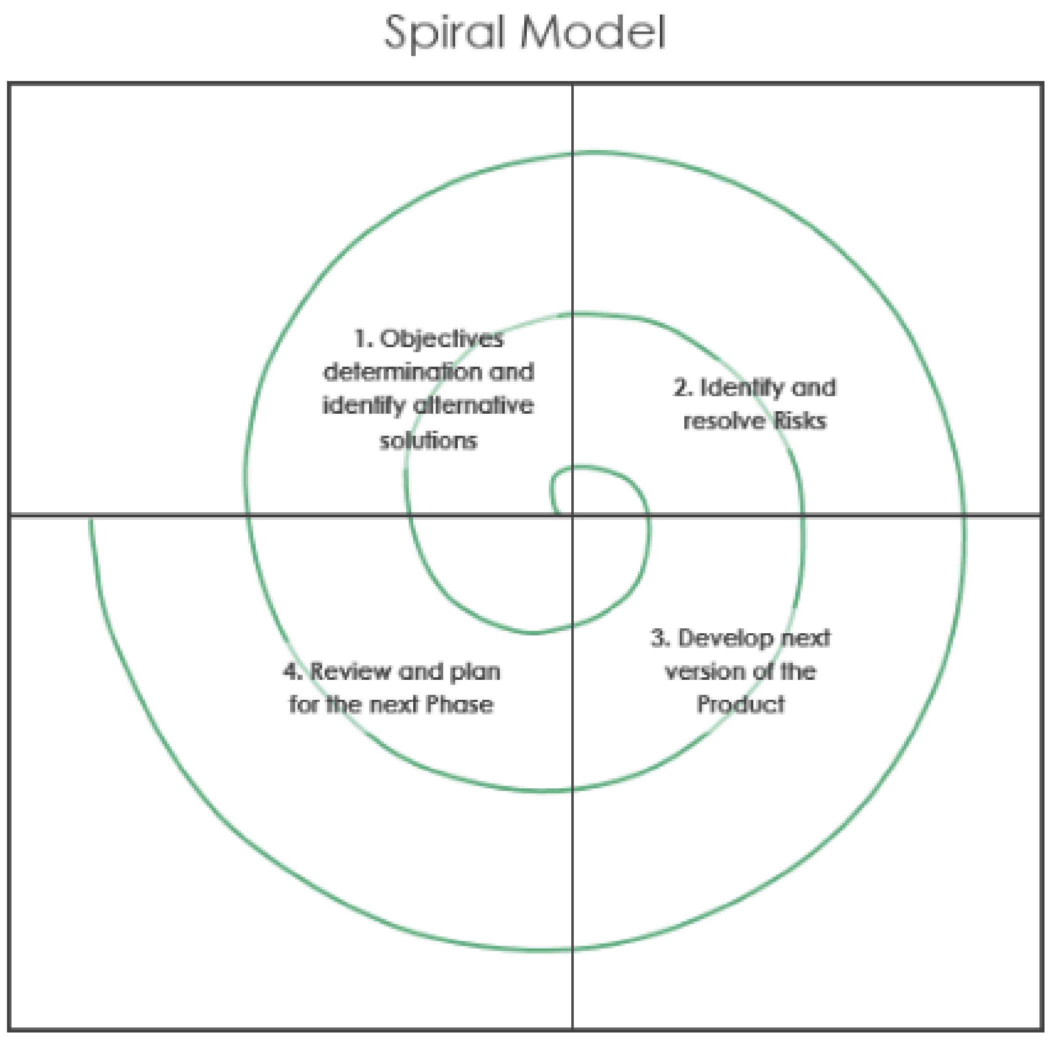
RAD Model


Requirements Planning
Refine
Prototype
User Design
Test
Construction
Cutover
Rapid Application Development (RAD)

Process Model

- **Spiral Model:** The spiral model, first described by Barry Boehm in 1986, is a risk-driven software development process model which was introduced for dealing with the shortcomings in the traditional waterfall model. A spiral model looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. This model supports risk handling, and the project is delivered in loops. Each loop of the spiral is called a Phase of the software development process.
- The initial phase of the spiral model in the early stages of Waterfall Life Cycle that is needed to develop a software product. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using a spiral model.

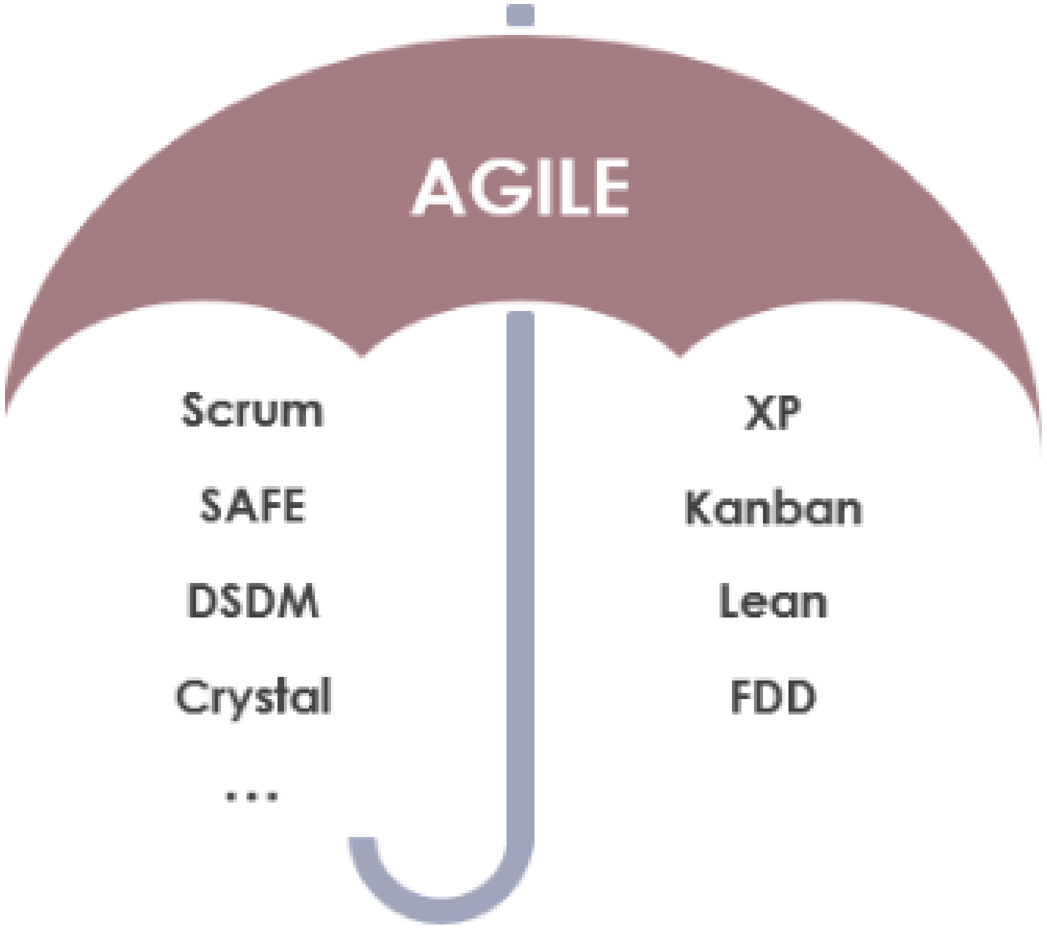
Spiral Model


Spiral Model
1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

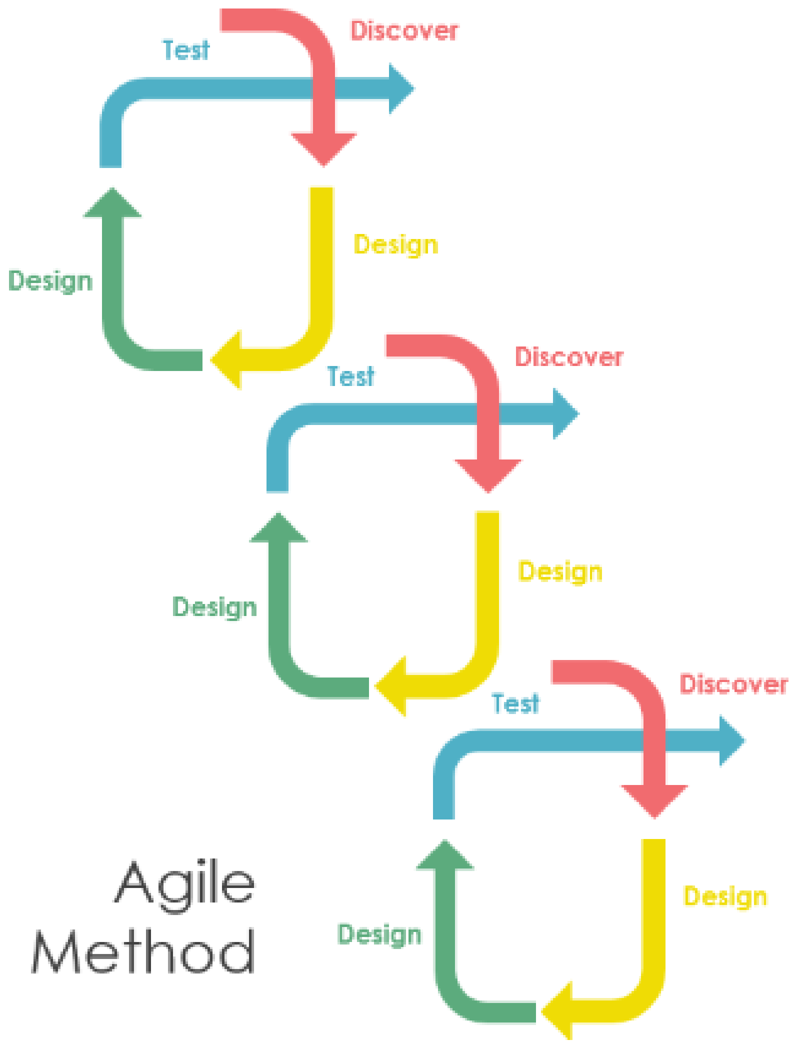
Process Model

- **Agile Model:**Agile is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto that is a way of thinking that enables teams and businesses to innovate, quickly respond to changing demand, while mitigating risk. Organizations can be agile using many of the available frameworks available such as Scrum, Kanban, Lean, Extreme Programming (XP) and etc.
- The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability which refer to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
- The primary goal of being Agile is empowered the development team the ability to create and respond to change in order to succeed in an uncertain and turbulent environment. Agile software development approach is typically operated in rapid and small cycles. This results in more frequent incremental releases with each release building on previous functionality. Thorough testing is done to ensure that software quality is maintained.

Agile Model



Agile Model



BY: Sunil Panchal

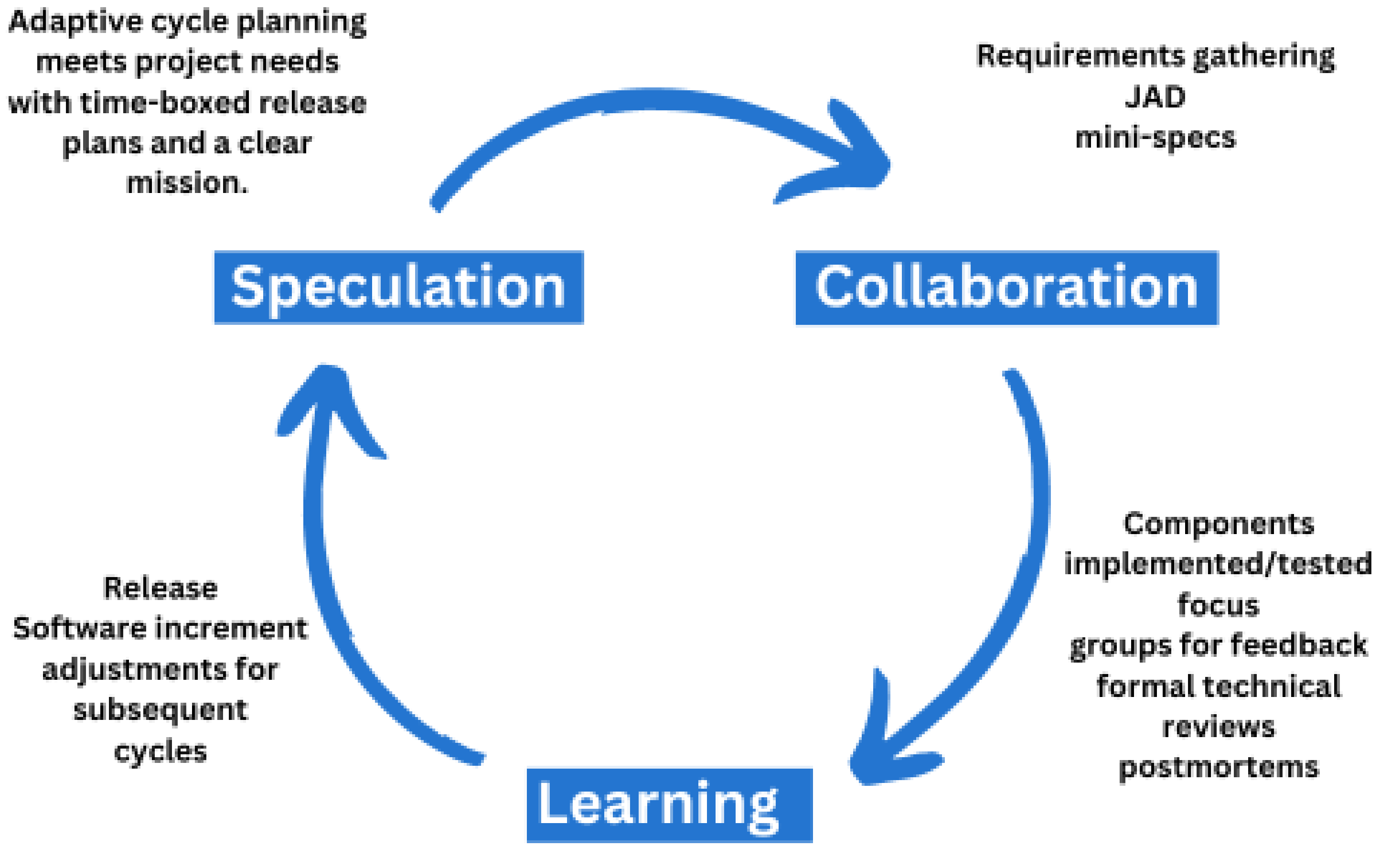
Steps of Software Development

1. **Planning:**
2. **Analysis:**
3. **Design:**
4. **Implementation:**
5. **Testing:**
6. **Deployment:**
7. **Maintenance**

Adaptive Software Development

- Adaptive Software Development (ASD) is a method for building complex software systems that emphasizes teamwork and the team’s ability to organize itself.
- So, primarily, the ASD process covers three phases:

Conti..


Adaptive cycle planning meets project needs with time-boxed release plans and a clear mission.
Requirements gathering
JAD
mini-specs
Speculation
Collaboration
Components implemented/tested
focus groups for feedback
formal technical reviews
postmortems
Learning
Release
Software increment
adjustments for subsequent cycles

Conti..

- **Speculation:**

This phase kicks off the project and involves planning based on project requirements and user needs.

Further, it sets up a series of stages for releasing parts of the project.

- **Collaboration:**

This phase is challenging because it requires motivated workers.

In addition, it encourages communication and teamwork while also valuing individual creativity.

- **Trust among team members is essential for:**
  - Giving helpful feedback
  - Supporting each other
  - Working hard
  - Having the right skills
  - Solving problems together effectively

Conti..

- **Learning:**

Workers might think they know more about technology than they do, which can cause problems.

Also, learning helps improve their understanding of the project.

- **Learning happens through:**
  - Group discussions
  - Technical reviews
  - Looking back on the project afterward

Conti..

- **Mission-driven:** Focuses on achieving specific goals.
- **Feature-based:** Builds software by adding features step by step.
- **Iterative:** Develops in repeated cycles of planning, building, and reviewing.
- **Time-boxed:** Sets fixed timeframes for each phase of development.
- **Risk-driven:** Manages risks actively throughout the process.
- **Change tolerant:** Adapts well to changes during development.

Conti..

- **Advantages of adaptive software development:**
- **User-focused:** ASD prioritizes end users, resulting in more intuitive products.
- **Timely delivery:** It often ensures projects are completed on time, sometimes even earlier.
- **Transparency:** ASD also promotes clear communication between developers and clients, fostering trust.

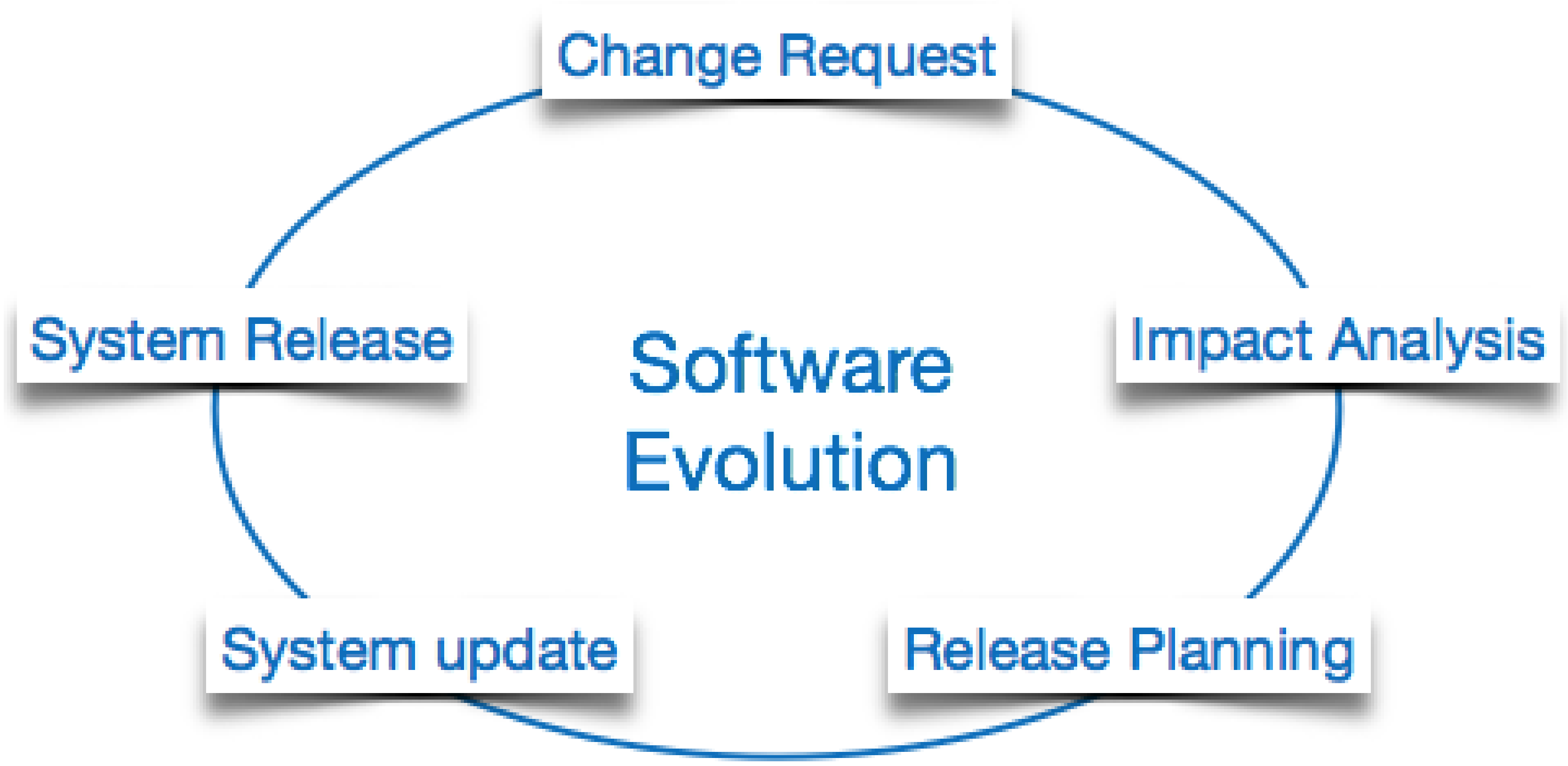
Conti..

- **Disadvantages of adaptive software development:**
- **High user involvement:** ASD requires extensive user participation, which can be challenging to manage.
- **Increased costs:** Integrating testing throughout the process can raise project expenses.
- **Scope creep:** Rapid iterations and constant feedback may lead to the project expanding beyond its initial scope.

Software Evolutionary

- **Software Evolution** refers to the process of software development over time, encompassing changes, updates, and enhancements made to software systems after their initial release. It's a natural and continuous process driven by various factors such as changing user requirements, technological advancements, bug fixes, and improvements in functionality and performance.

Software Evolutionary



Scrum

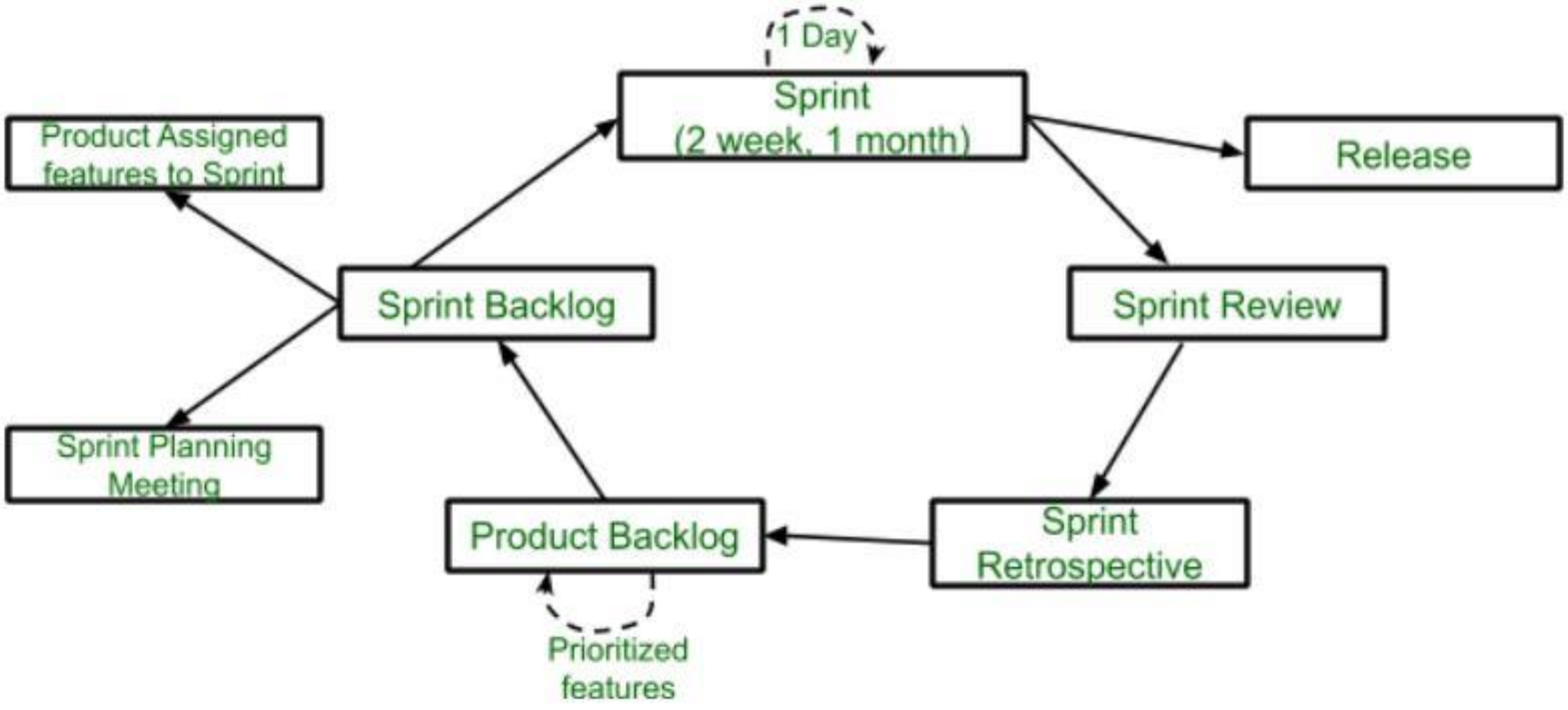
- Scrum is a management framework that teams use to self-organize tasks and work towards a common goal. It is a framework within which people can address complex adaptive problems while the productivity and creativity of delivering products are at the highest possible value. Scrum is a management framework that teams use to self-organize and work towards a common goal.
- Scrum allows us to develop products of the highest value while making sure that we maintain creativity and productivity.

- The iterative and incremental approach used in scrum allows the teams to adapt to the changing requirements.

Features of Scrum

- Scrum is a light-weighted framework
- Scrum emphasizes self-organization
- Scrum is simple to understand
- Scrum framework helps the team to work together
- Lifecycle of Scrum

Lifecycle of Scrum



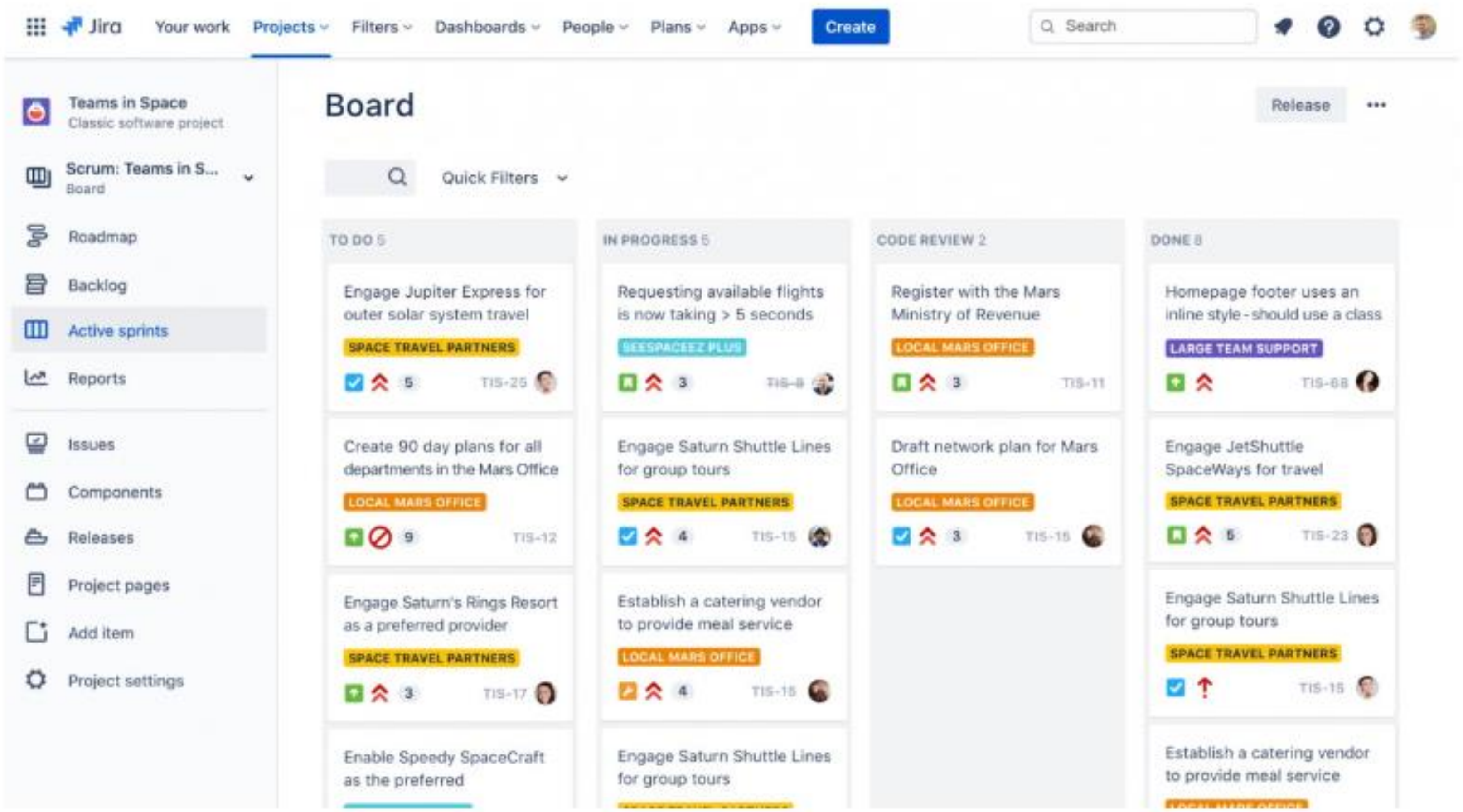
Advantages of Scrum

- Scrum framework is fast moving and money efficient.
- Scrum framework works by dividing the large product into small sub-products. It’s like a divide and conquer strategy
- In Scrum customer satisfaction is very important.
- Scrum is adaptive in nature because it have short sprint.
- As Scrum framework rely on constant feedback therefore the quality of product increases in less amount of time

Dis- Advantages of Scrum

- Scrum framework do not allow changes into their sprint.
- Scrum framework is not fully described model. If you wanna adopt it you need to fill in the framework with your own details like Extreme Programming(XP), Kanban, Dynamic Systems Development Method (DSDM).
- It can be difficult for the Scrum to plan, structure and organize a project that lacks a clear definition.
- The daily Scrum meetings and frequent reviews require substantial resources.

Scrum Framework



Board

TO DO 5

- Engage Jupiter Express for outer solar system travel — SPACE TRAVEL PARTNERS — 5 — TIS-25
- Create 90 day plans for all departments in the Mars Office — LOCAL MARS OFFICE — 9 — TIS-12
- Engage Saturn's Rings Resort as a preferred provider — SPACE TRAVEL PARTNERS — 3 — TIS-17
- Enable Speedy SpaceCraft as the preferred

IN PROGRESS 5

- Requesting available flights is now taking > 5 seconds — SEESPACEEZ PLUS — 3 — TIS-8
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — 4 — TIS-15
- Establish a catering vendor to provide meal service — LOCAL MARS OFFICE — 4 — TIS-15
- Engage Saturn Shuttle Lines for group tours

CODE REVIEW 2

- Register with the Mars Ministry of Revenue — LOCAL MARS OFFICE — 3 — TIS-11
- Draft network plan for Mars Office — LOCAL MARS OFFICE — 3 — TIS-15

DONE 8

- Homepage footer uses an inline style - should use a class — LARGE TEAM SUPPORT — TIS-68
- Engage JetShuttle SpaceWays for travel — SPACE TRAVEL PARTNERS — 5 — TIS-23
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — TIS-15
- Establish a catering vendor to provide meal service

BY: Sunil Panchal