

# Software Engineering Approaches

**J. Kamiri**

**Department of Information Technology**

**Murang'a University of Technology**

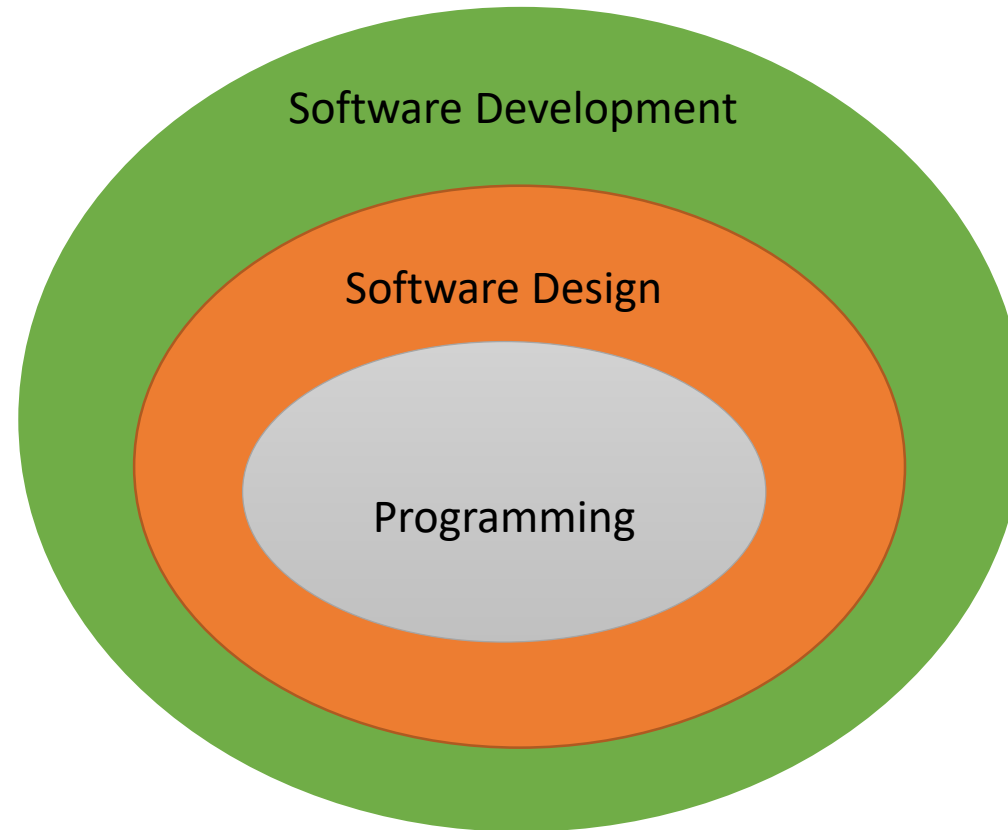
# What is Software Engineering?

- The term is made of two words, software and engineering.
- **Software** is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product**.
- **Engineering** on the other hand, is all about developing products, using well-defined, scientific principles and methods.
- According to IEEE Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software

# Software Paradigms

- Software paradigms refer to the methods and steps, which are taken while designing the software.
- There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand.
- These can be combined into various categories, though each of them is contained in one another:

# Visual Representation of Software Paradigms



# Cont.

- **Software Development Paradigm**

- It includes various researches and requirement gathering which helps the software product to build. It consists of –
  - Requirement gathering
  - Requirement analysis

- **Software Design Paradigm**

- This paradigm is a part of Software Development and includes –
  - Design
  - Documenting designs

- **Programming Paradigm**

- This paradigm is related closely to programming aspect of software development. This includes –
  - Coding
  - Testing
  - Integration

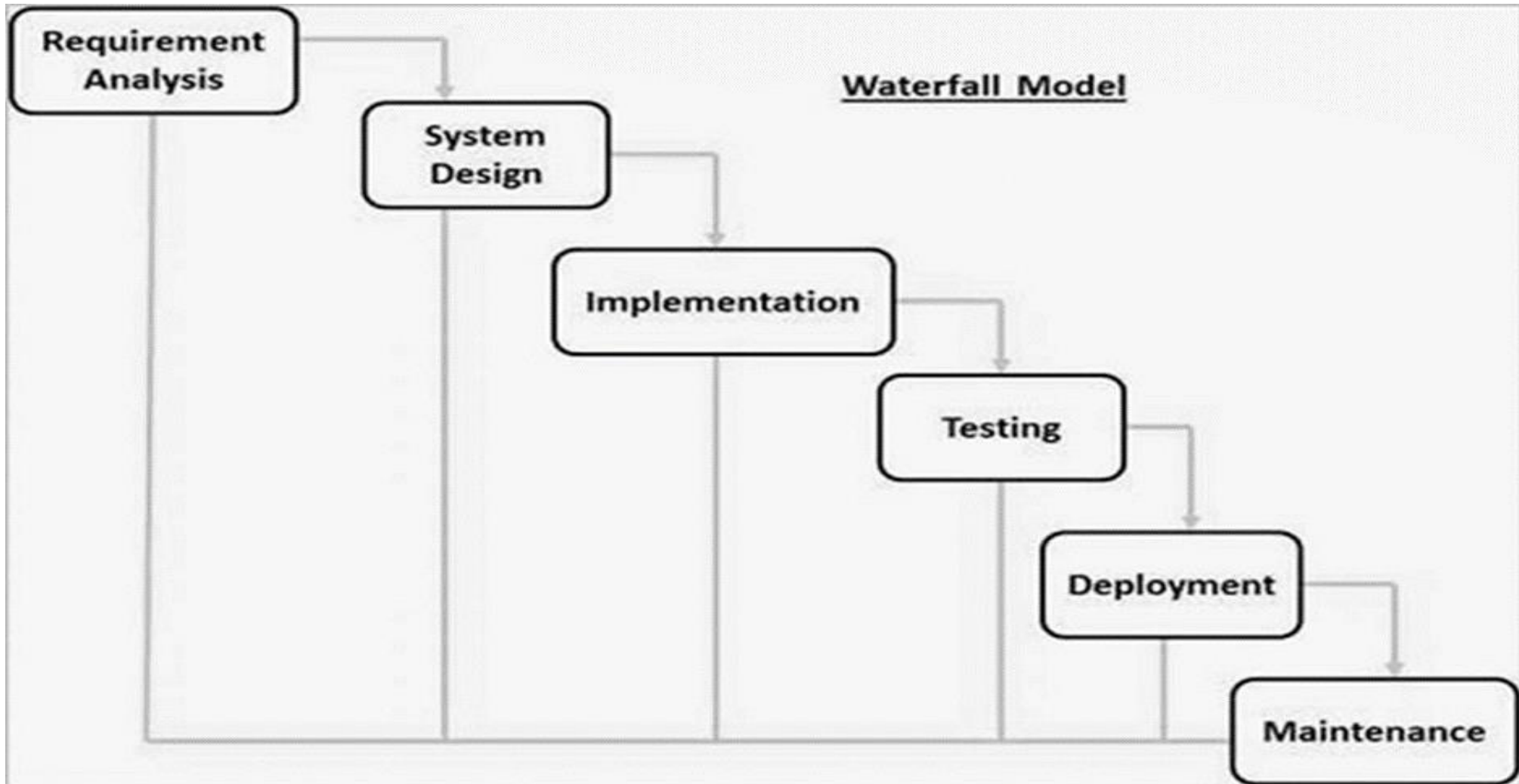
# Software Development Methodologies

- The software development methodology is a framework that is used to structure, plan, and control the process development of an information system.
- Software development methodology can also be referred to as the software development lifecycle (SDLC) since it informs about the phases that a software will go through before the final product is achieved.
- There are various software development methodologies that exist which include:
  - Waterfall
  - Prototyping
  - Agile
  - Spiral

# 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**.
- It is organized into phases and each phase must be completed before the next phase can begin and there is no overlapping in the phases.
- This is so because in Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.
- All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases.

# Stages of the waterfall Model





## Cont. (Phases of Waterfall Model)

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared using UML diagrams. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed (programmed) in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

# Cont.

- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer's environment.

# Advantages of Waterfall Model

- Waterfall model is very simple and easy to understand. That is why it is beneficial for the beginner or novice developer.
- It is easy to manage the projects because of the rigidity of the model. Moreover, each phase has specific deliverables and an individual review process.
- This model saves a significant amount of time at all the phases processed and completed at a given time.
- The requirements are very well understood/defined in this type of development model. Also, it works effectively for smaller projects.
- You can easily do the testing that refers to the defined scenarios in the earlier functional specification.

# Disadvantages of Waterfall Model

- Useful only when the requirements are well known and available
- This model is not applicable to projects that demand continuous maintenance.
- Once an application is in the testing stage, it is not advisable to go back and do any amendments changes for completed software, it may cause a lot of problems.
- There is no possibility that we can to develop any working software until it reaches the last stage of the cycle
- You cannot include the client's valuable feedback within the ongoing development phase.
- In this model, Documentation occupies a lot of time for developers and testers.

# Prototyping Methodology

- It is iterative in nature since it involves developing a prototype (crude version of the software) then giving it to the users for review who after review either accept it and recommend adjustments or reject it.
- Unlike Water fall model and spiral model prototyping makes it possible to develop software that are acceptable by customers.

# Cont.

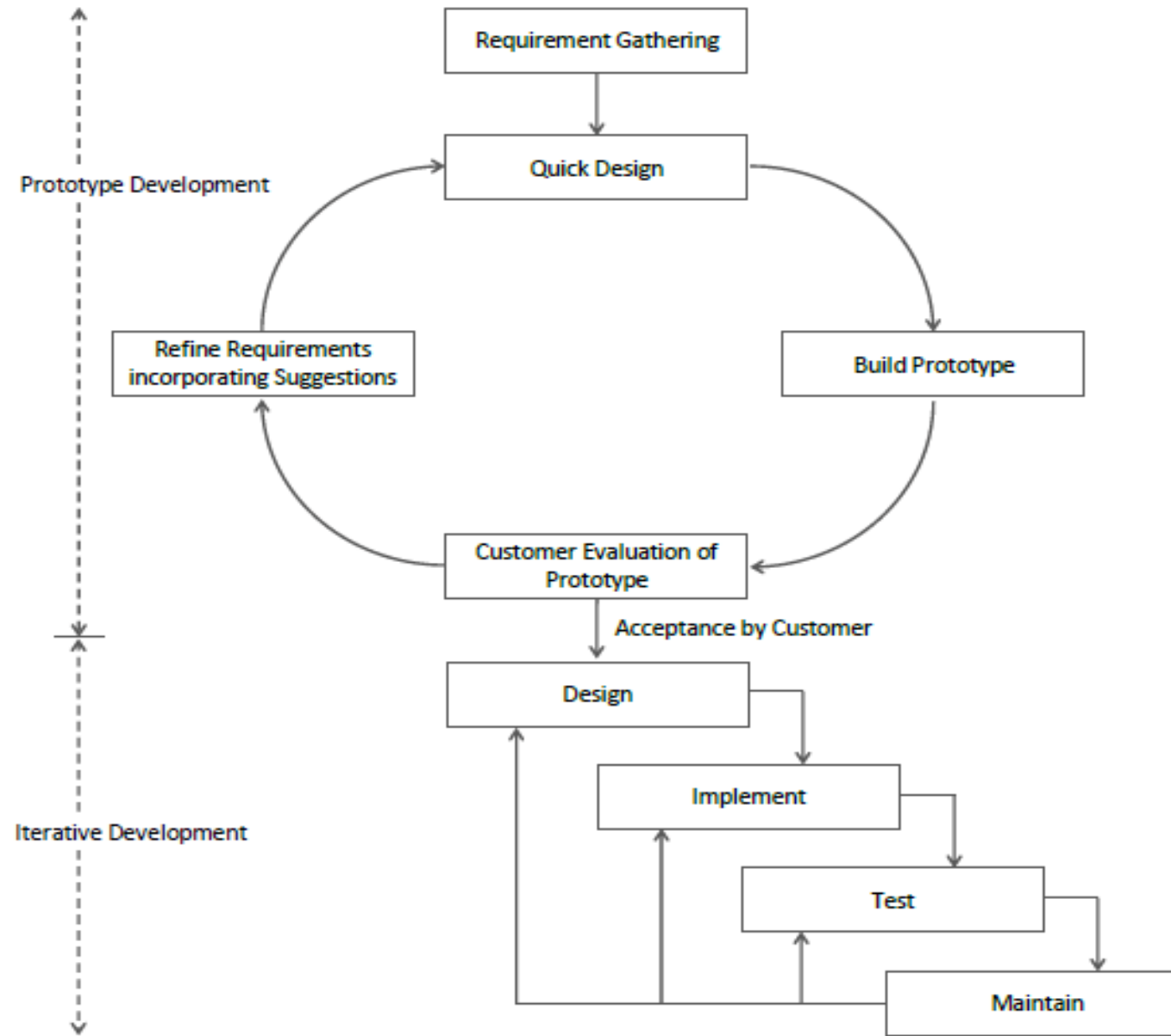


Figure 3: Prototype Model of Software Development

# Types of Prototypes

- **Rapid throwaway prototype**- In this method, a developed prototype need not necessarily be a part of the ultimately accepted prototype. i.e initial prototype is discarded after getting customer feedback. Customer feedback helps in preventing unnecessary design faults and hence, the final prototype developed is of better quality.
- **Evolutionary Prototyping** – In this method, the prototype developed initially is incrementally refined on the basis of customer feedback till it finally gets accepted.
- In comparison to Rapid Throwaway Prototyping, it offers a better approach which saves time as well as effort. This is because developing a prototype from scratch for every iteration of the process can sometimes be very frustrating for the developers.

# Cont.

- **Incremental Prototyping** – In this type of incremental Prototyping, the final expected product is broken into different small pieces of prototypes and each piece is developed individually.
- In the end, when all individual pieces are properly developed, then the different prototypes are collectively merged into a single final product in a predefined order.
- It's a very efficient approach which reduces the complexity of the development process, where the goal is divided into sub-parts and each sub-part is developed individually.



# Advantages of Prototyping

- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
- New requirements can be easily accommodated as there is scope for refinement.
- Missing functionalities can be easily figured out.
- Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
- The developed prototype can be reused by the developer for more complicated projects in the future.
- Flexibility in design.

# Disadvantages of Prototyping

- Costly with respect to time as well as money.
- There may be too much variation in requirements each time the prototype is evaluated by the customer.
- Poor Documentation due to continuously changing customer requirements.
- It is very difficult for developers to accommodate all the changes demanded by the customer.
- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
- Developers in a hurry to build prototypes may end up with sub-optimal solutions.
- The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

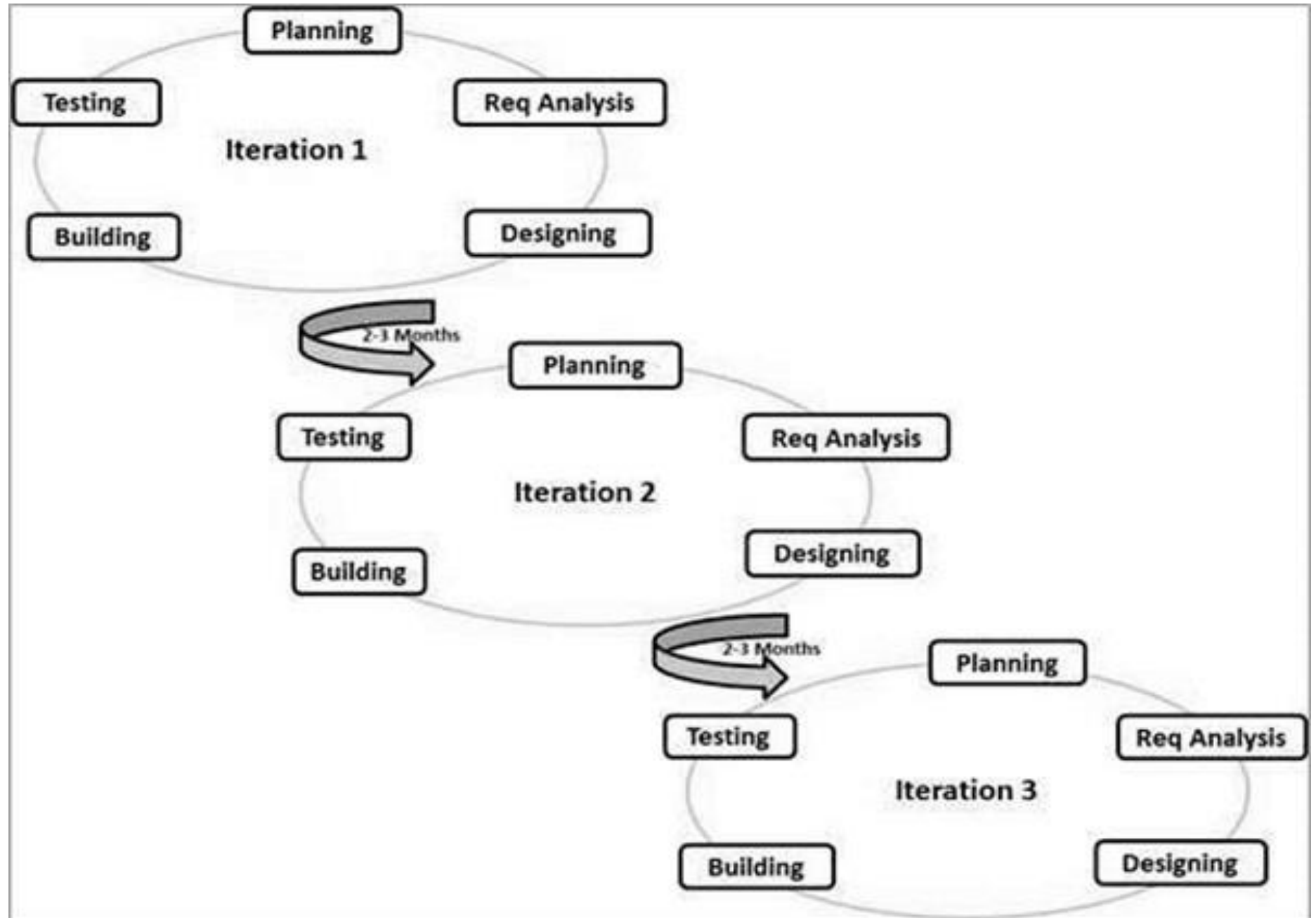
# Agile Software Development Methodology

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds. These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like –
  - Planning
  - Requirements Analysis
  - Design
  - Coding
  - Unit Testing and
  - Acceptance Testing.
- At the end of the iteration, a working product is displayed to the customer and important stakeholders.

# Cont.

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.
- In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iterative approach is taken and working software build is delivered after each iteration.
- Each build is incremental in terms of features; the final build holds all the features required by the customer.

Cont.



# Agile Principles

Agile has 12 key principals in its manifesto. The principals are:

- 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.

# Agile Principles Cont.

- 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.
- 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.

# Agile Principles Cont.

- 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.



# Advantages of Agile Model

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Gives flexibility to developers.

# Disadvantages of Agile model

- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile project manager practice is a must without which it will not work.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

# Assignment

- Discuss Spiral model.
- The assignment should be handwritten and submitted on or before 15<sup>th</sup> October 2022.
- Ensure you discuss also the advantages and disadvantages.

# The End !

**Everybody in this country should  
learn to program a computer...  
because it teaches you how to think**

Steve Jobs, co-founder and CEO of Apple Inc. (1955 - 2011)

