



# DIFFERENT SDLC MODELS

# WATERFALL MODEL

- Introduced in 1970 by Winston Royce
- Steps goes in order and must be completed before beginning of the next phase
  - 5 Phases

## WHEN TO USE

- Phase 1: Requirement Analysis and Specification phase
  - Phase 2: Design Phase
  - Phase 3: Implementation and Unit testing
  - Phase 4: Integration and System Testing
  - Phase 5: Operation and Maintenance Phase
- When the requirements are constant and not changed regularly.
  - A project is short
  - The situation is calm
  - Where the tools and technology used is consistent and is not changing
  - When resources are well prepared and are available to use.





# ADVANTAGES OF WATERFALL MODEL



Simple to Implement.



Requirements are simple and explicitly declared



Start and end points for each phase is fixed



Release date can be determined before development.



Gives control and clarity for the customer due to a strict reporting system.



# DISADVANTAGES OF WATERFALL MODEL

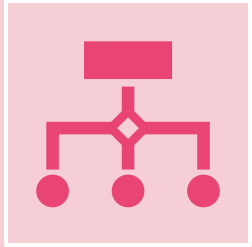
Risk Factor is very High

Cannot accept changes in requirements

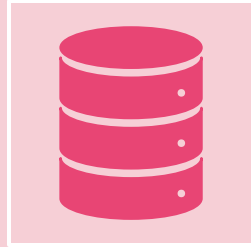
Tough to go back to the phase

Testing is done at later stage

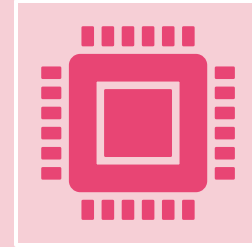
# PROTOTYPE MODEL



BEFORE CARRYING OUT THE DEVELOPMENT OF ACTUAL SOFTWARE, A WORKING PROTOTYPE SYSTEM SHOULD BE BUILT

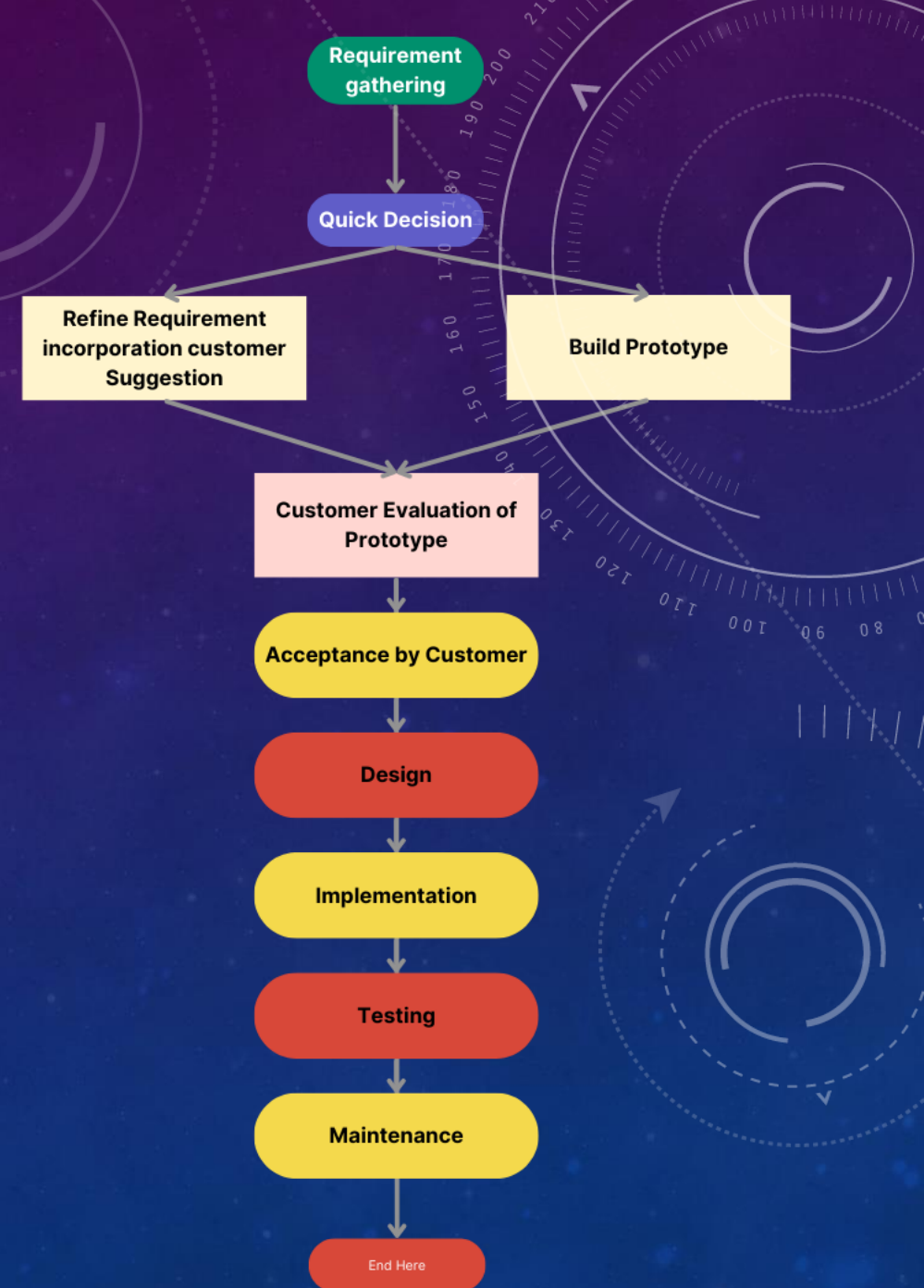


IT'S A VERY RAW AND ROUGH VERSION OF ACTUAL SOFTWARE WHEREIN A CLIENT WILL HAVE ONLY AN OVERVIEW OF ACTUAL SOFTWARE



IT IS USED WHEN THERE IS AN ABSENCE OF DETAILED INFORMATION REGARDING THE INPUT, THE PROCESSING NEEDS AND WHAT OUTPUT IS EXPECTED

# STEPS OF PROTOTYPE MODEL





# ADVANTAGES OF PROTOTYPE MODEL

---

Reduce the risk of incorrect user requirement

---

Good where requirement are changing/uncommitted

---

Regular visible process aids management

---

Support early product marketing

---

Reduce Maintenance cost.

---

Errors can be detected much earlier as the system is made side by side.

# DISADVANTAGES OF PROTOTYPE MODEL



An unstable/badly implemented prototype often becomes the final product.



Require extensive customer collaboration



Difficult to know how long the project will last.



Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.



Prototyping tools are expensive.

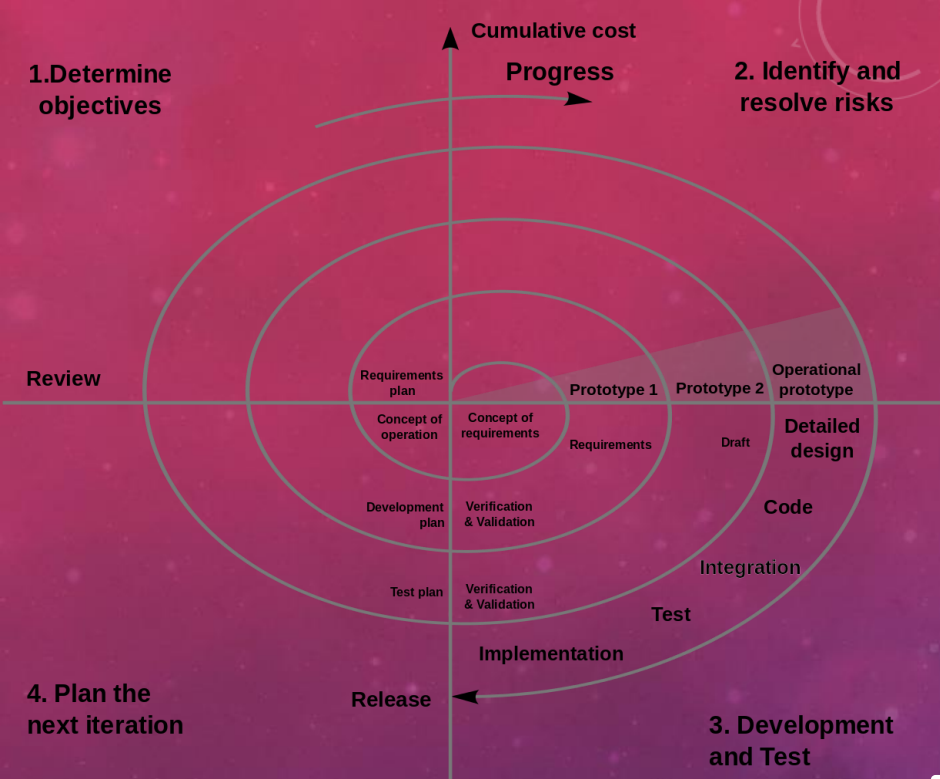


Special tools & techniques are required to build a prototype.



It is a time-consuming process.





# SPIRAL MODEL

- Initially proposed by Boehm
- Implements the potential for Rapid development
- The software is developed in a series of incremental release
- Early iteration: Paper or Prototype model
- Later iteration: Complete version of Engineered System

## FOUR PARTS OF SPIRAL MODEL

- Determine objectives
- Identify and resolve risks
- Development and Test
- Plan the next iteration

# WHEN TO USE SPIRAL MODEL



WHEN DELIVERANCE IS  
REQUIRED TO BE  
FREQUENT.



WHEN THE PROJECT IS  
LARGE



WHEN REQUIREMENTS  
ARE UNCLEAR AND  
COMPLEX



WHEN CHANGES MAY  
REQUIRE AT ANY TIME



LARGE AND HIGH  
BUDGET PROJECTS



# ADVANTAGES OF SPIRAL MODEL



HIGH AMOUNT OF RISK  
ANALYSIS



USEFUL FOR LARGE AND  
MISSION-CRITICAL PROJECTS.

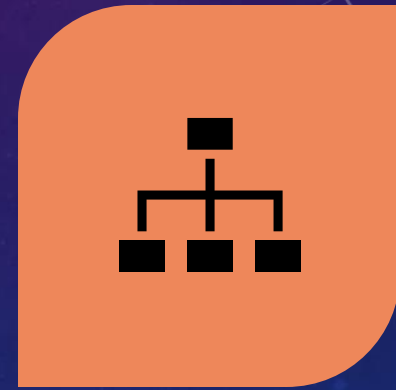
# DISADVANTAGES OF SPIRAL MODEL



CAN BE A COSTLY MODEL TO  
USE.



RISK ANALYSIS NEEDED HIGHLY  
PARTICULAR EXPERTISE



DOESN'T WORK WELL FOR  
SMALLER PROJECTS.



# INCREMENTAL MODEL

- Process where requirements are divided into multiple standalone modules
- Each module goes through requirements to implementation phase
- The process continues till the complete system is achieved
- Various phases are Requirement analysis, Design and Development, Testing and Implementation

# WHEN TO USE INCREMENTAL MODEL



---

When the requirements are superior.

---

A project has a lengthy development schedule.

---

When Software team are not very well skilled or trained.

---

When the customer demands a quick release of the product.

---

You can develop prioritized requirements first.



# ADVANTAGES OF INCREMENTAL MODEL



Errors are easy to be recognized.



Easier to test and debug



More flexible.



Simple to manage risk because it handled during its iteration.



The Client gets important functionality early.

# DISADVANTAGES OF INCREMENTAL MODEL



Need for good planning



Total Cost is high.



Well defined module  
interfaces are needed.

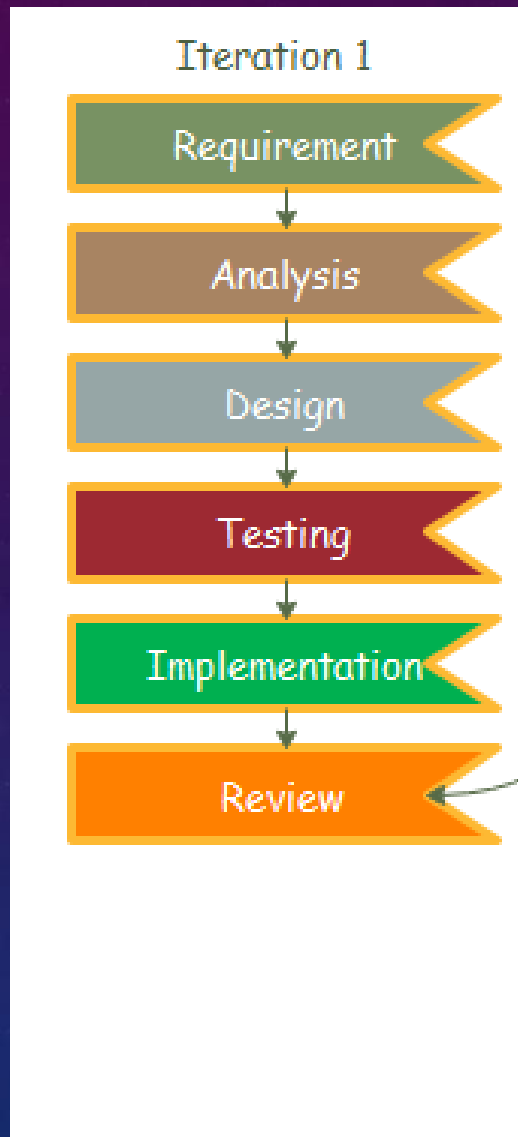


# ITERATIVE MODEL

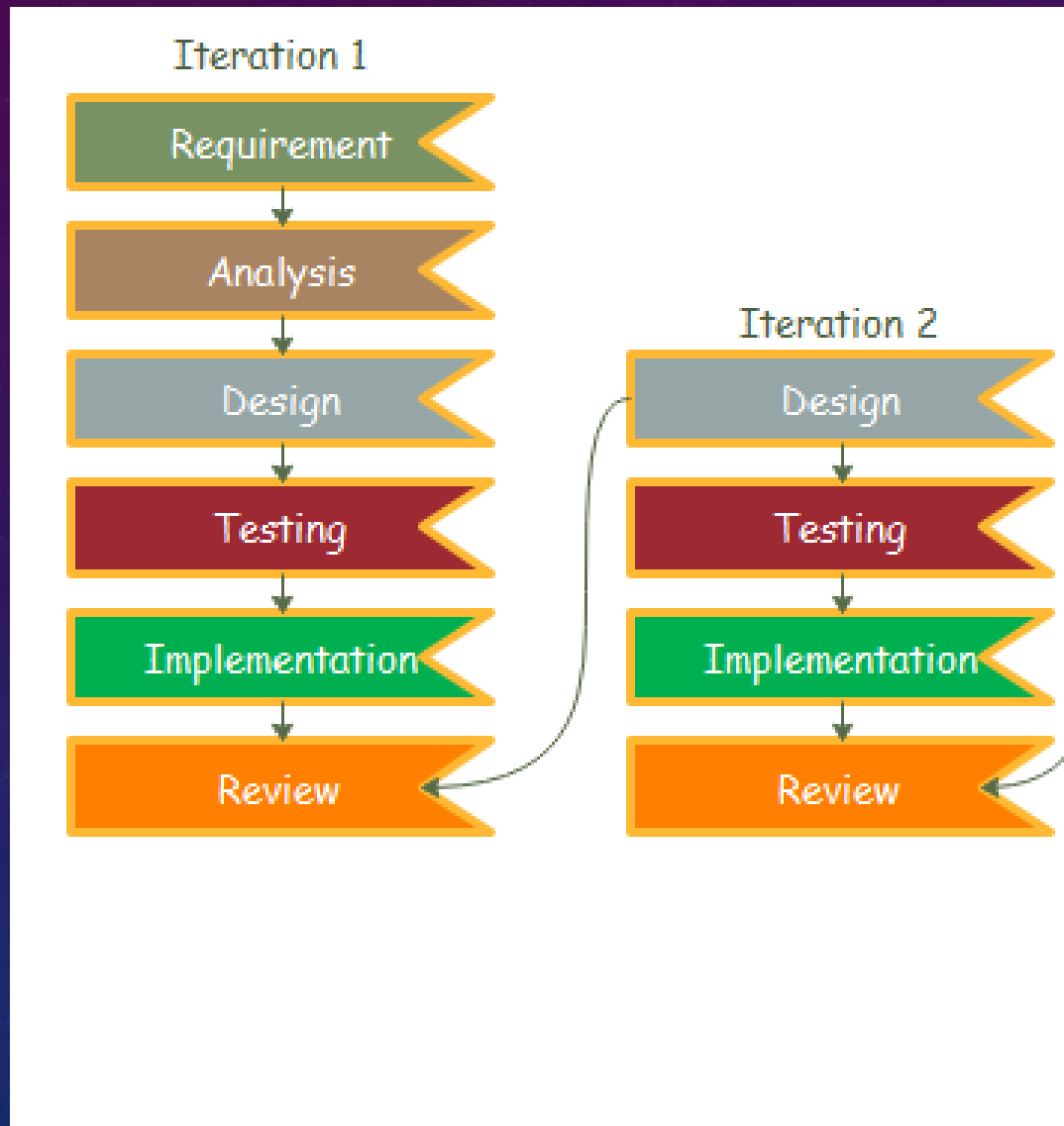
- In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.
- The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.



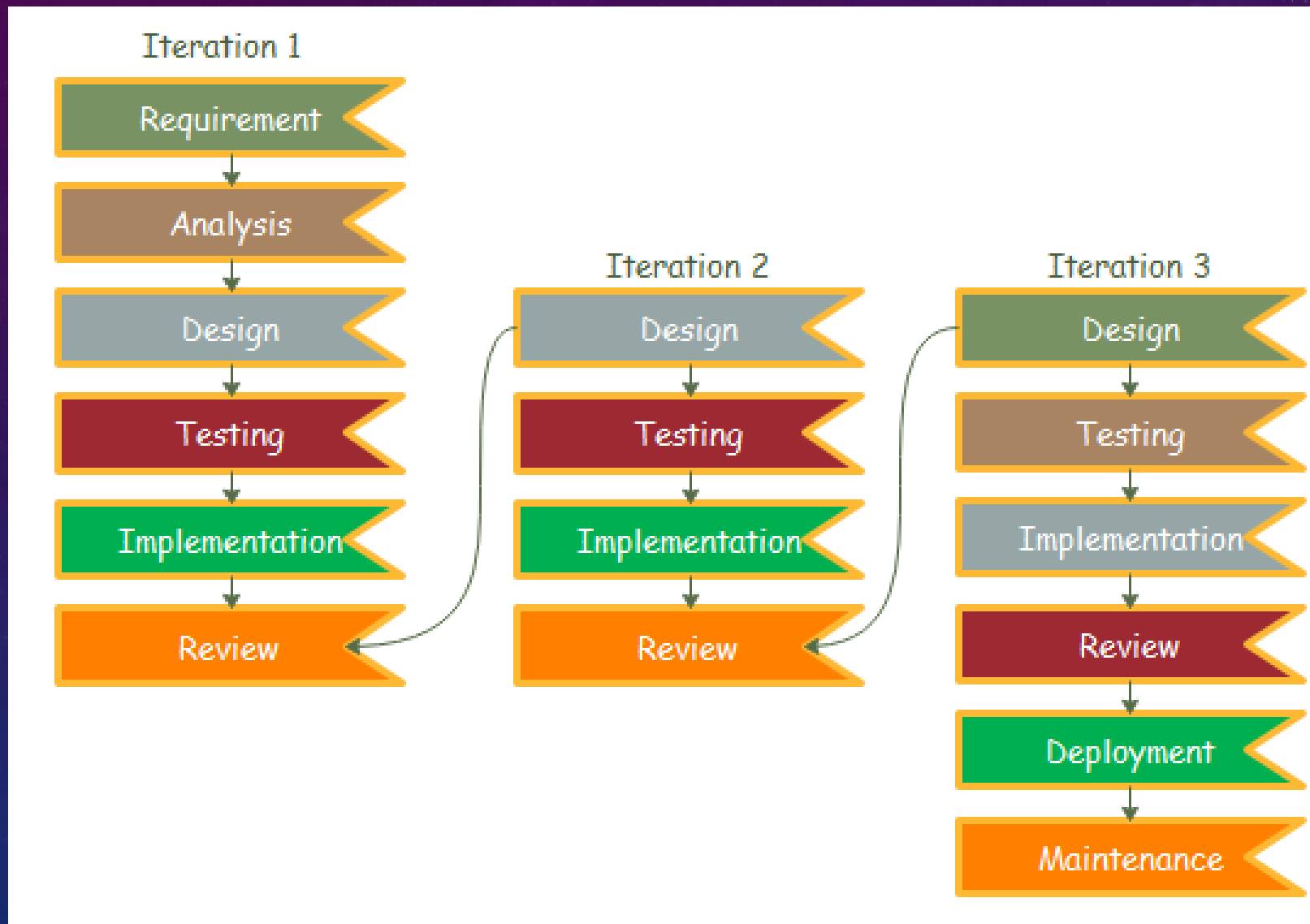
# ITERATIVE MODEL



# ITERATIVE MODEL



# ITERATIVE MODEL





# VARIOUS PHASES OF ITERATIVE MODEL

- **Requirement gathering & analysis**: In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.
- **Design**: In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.
- **Implementation**: In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.
- **Testing**: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.
- **Deployment**: After completing all the phases, software is deployed to its work environment.
- **Review**: In this phase, after the product deployment, review phase is performed to check the behaviour and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.
- **Maintenance**: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

# WHEN TO USE ITERATIVE MODEL

- When requirements are defined clearly and easy to understand.
- When the software application is large.
- When there is a requirement of changes in future.

# WHEN TO USE ITERATIVE MODEL

- When requirements are defined clearly and easy to understand.
- When the software application is large.
- When there is a requirement of changes in future.

## ADVANTAGES OF ITERATIVE MODEL

- Testing and debugging during smaller iteration is easy.
- A Parallel development can plan.
- It is easily acceptable to ever-changing needs of the project.
- Risks are identified and resolved during iteration.
- Limited time spent on documentation and extra time on designing.



# WHEN TO USE ITERATIVE MODEL

- When requirements are defined clearly and easy to understand.
- When the software application is large.
- When there is a requirement of changes in future.

## ADVANTAGES OF ITERATIVE MODEL

- Testing and debugging during smaller iteration is easy.
- A Parallel development can plan.
- It is easily acceptable to ever-changing needs of the project.
- Risks are identified and resolved during iteration.
- Limited time spent on documentation and extra time on designing.

## DISADVANTAGES OF ITERATIVE MODEL

- It is not suitable for smaller projects.
- More Resources may be required.
- Design can be changed again and again because of imperfect requirements.
- Requirement changes can cause over budget.
- Project completion date not confirmed because of changing requirements.

# RAD MODEL – RAPID APPLICATION DEVELOPMENT

- RAD is a linear sequential software development process model that emphasizes a concise development cycle using an element-based construction approach.
- If the requirements are well understood and described, and the project scope is a constraint, the RAD process enables a development team to create a fully functional system within a concise time period.
- RAD (Rapid Application Development) is a concept that products can be developed faster and of higher quality through:
  - Gathering requirements using workshops or focus groups
  - Prototyping and early, reiterative user testing of designs
  - The re-use of software components
  - A rigidly paced schedule that refers design improvements to the next product version
  - Less formality in reviews and other team communication

# RAD MODEL

Module

1

Business  
Modelling



Data  
Modelling



Process  
Modelling



Application  
Generation



Testing &  
Turnover

2

Business  
Modelling



Data  
Modelling



Process  
Modelling



Application  
Generation



Testing &  
Turnover

3

Business  
Modelling



Data  
Modelling



Process  
Modelling



Application  
Generation



Testing &  
Turnover



# VARIOUS PHASES OF RAD MODEL

## Business Modelling

- The information flow among business functions is defined by answering questions like what data drives the business process, what data is generated, who generates it, where does the information go, who process it and so on

## Data Modelling

- The data collected from business modeling is refined into a set of data objects (entities) that are needed to support the business. The attributes (character of each entity) are identified, and the relation between these data objects (entities) is defined

## Process Modelling

- The information object defined in the data modeling phase are transformed to achieve the data flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting, or retrieving a data object.

## Application Generation

- Automated tools are used to facilitate construction of the software; even they use the 4th GL techniques.

## Testing & Turnover

- Many of the programming components have already been tested since RAD emphasis reuse. This reduces the overall testing time. But the new part must be tested, and all interfaces must be fully exercised.

# WHEN TO USE RAD MODEL

- When the system should need to create the project that modularizes in a short span time (2-3 months).
- When the requirements are well-known.
- When the technical risk is limited.
- When there's a necessity to make a system, which modularized in 2-3 months of period.
- It should be used only if the budget allows the use of automatic code generating tools.

# ADVANTAGES OF RAD MODEL

- This model is flexible for change.
- In this model, changes are adoptable.
- Each phase in RAD brings highest priority functionality to the customer.
- It reduced development time.
- It increases the reusability of features.



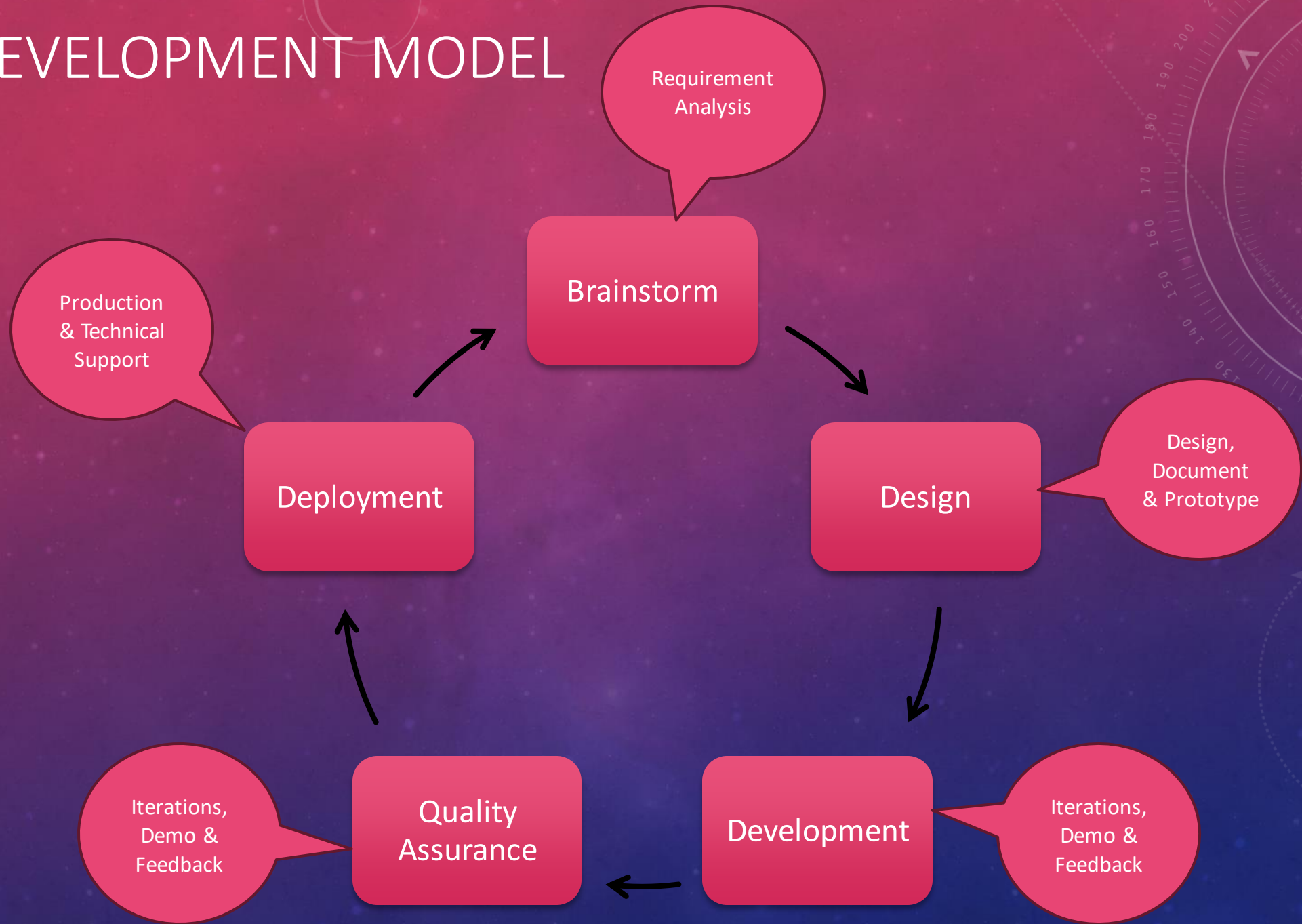
# DISADVANTAGES OF RAD MODEL

- It required highly skilled designers.
- All applications are not compatible with RAD.
- For smaller projects, we cannot use the RAD model.
- On the high technical risk, it's not suitable.
- Required user involvement.

# AGILE DEVELOPMENT MODEL

- The meaning of Agile is swift or versatile. "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.

# AGILE DEVELOPMENT MODEL





# VARIOUS PHASES OF AGILE MODEL

- **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.
- **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.
- **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.
- **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
- **Deployment:** In this phase, the team issues a product for the user's work environment.
- **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 TESTING METHODS

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

# WHEN TO USE AGILE MODEL

- When frequent changes are required
- When a highly qualified and experienced team is available
- When a customer is ready to have a meeting with a software team all the time
- When project size is small



# ADVANTAGES OF AGILE MODEL

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfils the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

# DISADVANTAGES OF AGILE MODEL

1. Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
2. Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

# SCRUM

- SCRUM is an agile development process focused primarily on ways to manage tasks in team-based development conditions.
- There are three roles in it, and their responsibilities are:
  - **Scrum Master:** The scrum can set up the master team, arrange the meeting and remove obstacles for the process
  - **Product owner:** The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
  - **Scrum Team:** The team manages its work and organizes the work to complete the sprint or cycle.





# THANK YOU!

[DEVANGTHAKAR12@GMAIL.COM](mailto:DEVANGTHAKAR12@GMAIL.COM)

9004044991