CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

Object Oriented Software Engineering

LECTURE 4

Today’s Outline:

The SDLC Models

- Waterfall Model
- RAD Model
- V-Model
- Incremental Model
- Iterative Model
- BigBang Model
- Agile Model
- Prototype Model
- Spiral Model

II. RAD Model

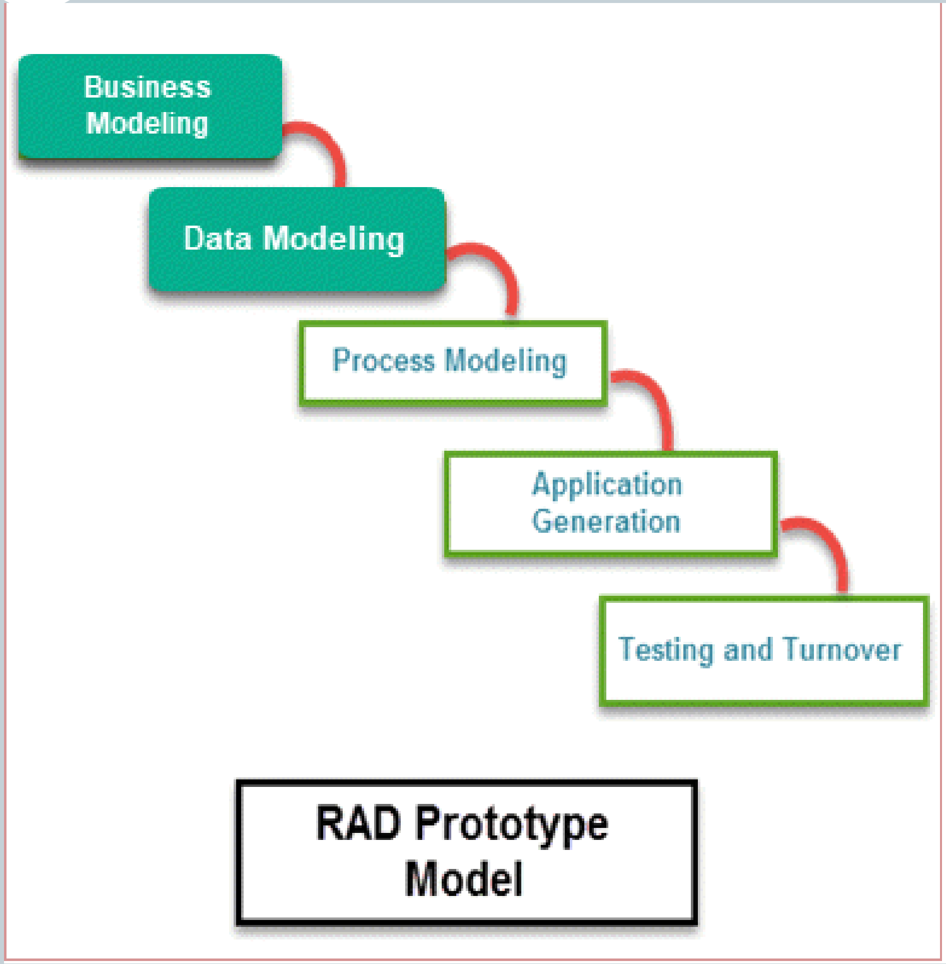
- **Rapid: happening in a short time**
- This model was first proposed by IBM in 1980’s.
- **RAD Model** or Rapid Application Development model is a software development process based on prototyping without any specific planning. In RAD model, there is *less attention paid to the planning* and more priority is given to the development tasks. It targets at developing software in a short span of time.
- Development of each module requires basics SDLC steps like waterfall steps.

CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

RAD Model

SDLC RAD modeling has following phases:

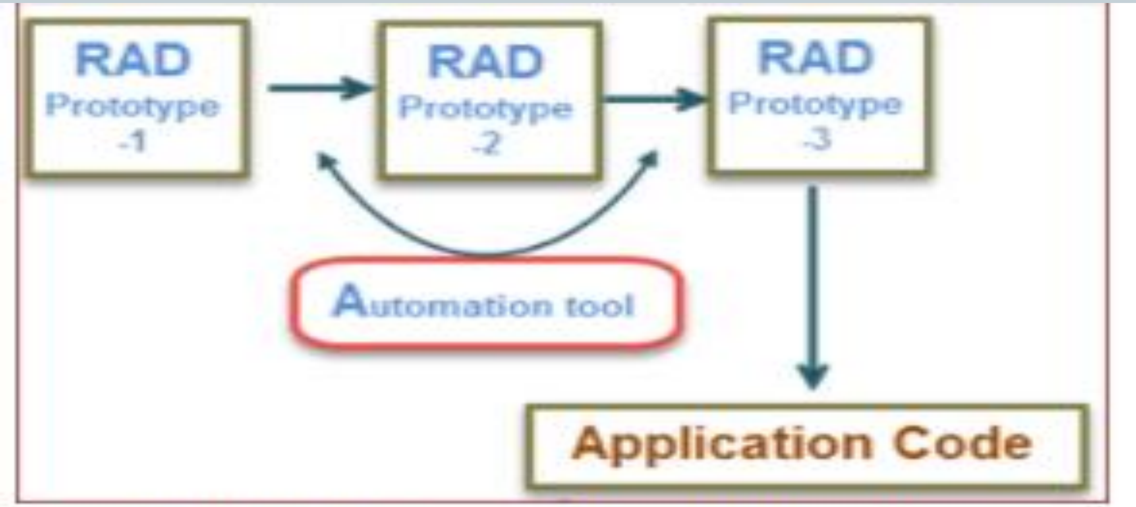
- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation
- Testing and Turnover



CHITKARA UNIVERSITY
HIMACHAL PRADESH - NAAC Accredited

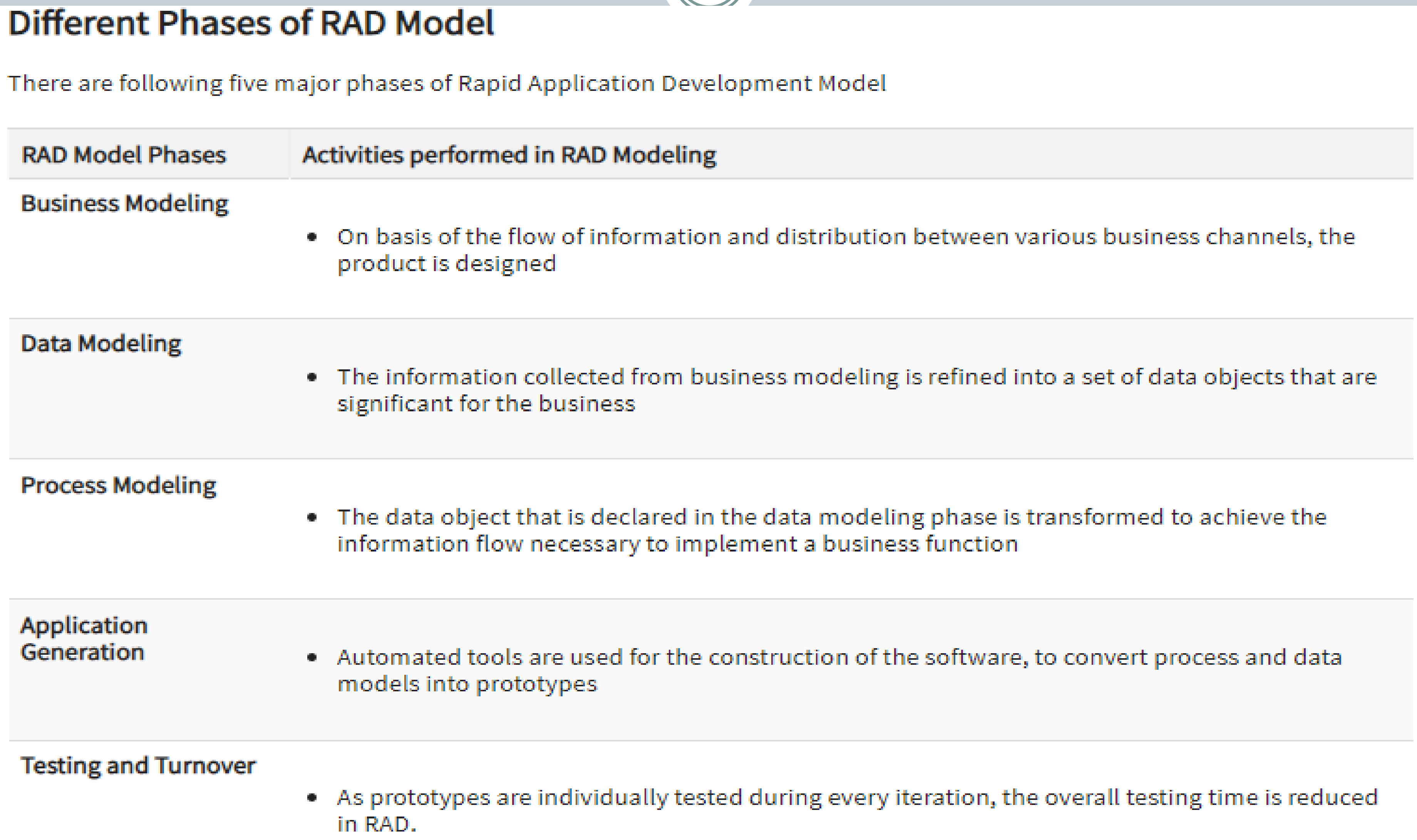
RAD Model

- It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD modeling are that it focuses on the reuse of templates, tools, processes, and code.



RAD Model in Software Engineering

RAD Model



Different Phases of RAD Model

There are following five major phases of Rapid Application Development Model

RAD Model Phases	Activities performed in RAD Modeling
Business Modeling	• On basis of the flow of information and distribution between various business channels, the product is designed
Data Modeling	• The information collected from business modeling is refined into a set of data objects that are significant for the business
Process Modeling	• The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function
Application Generation	• Automated tools are used for the construction of the software, to convert process and data models into prototypes
Testing and Turnover	• As prototypes are individually tested during every iteration, the overall testing time is reduced in RAD.

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.

RAD Model

Advantages:

- Reduce Development time.
- Encourage customer feedback
- Quick initial reviews
- Easy to accommodate changing the requirements

Disadvantage:

- Need of highly skilled team
- Customer involvements is required throughout the RAD model.
- Implementable to only to those projects which can be modularized.

III. V-model

- The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape.
- It is also known as ***Verification and Validation model***.
- The V-model overcomes a drawback of waterfall model which you can not go back to the previous stage for correction.
- The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

What is verification and Validation

- Verification: It is the process of checking whether the software meets the specification.

- Validation: Process of checking whether the specifications meets the customer needs.

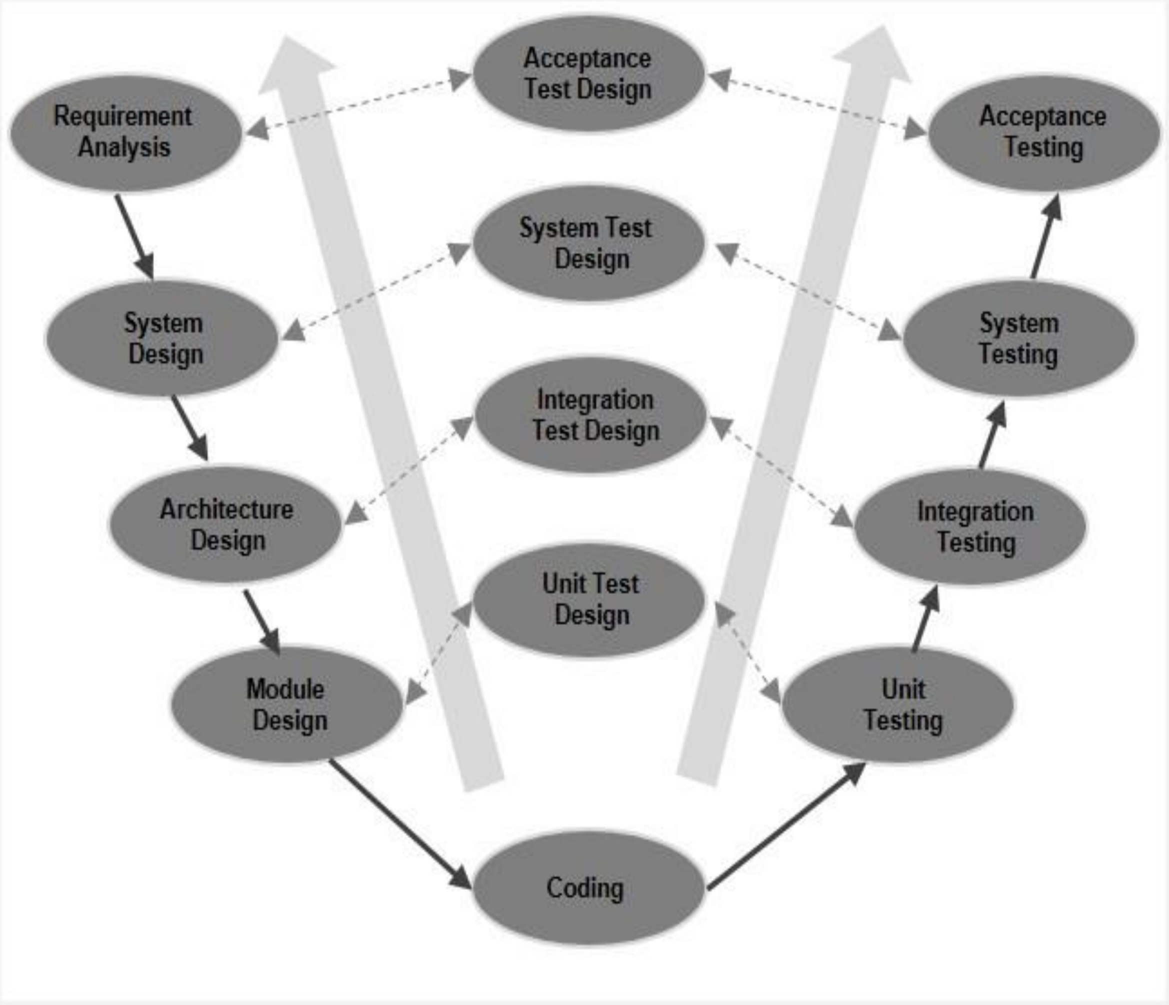
- V-model allow you to verify and validate in parallel.

V-model

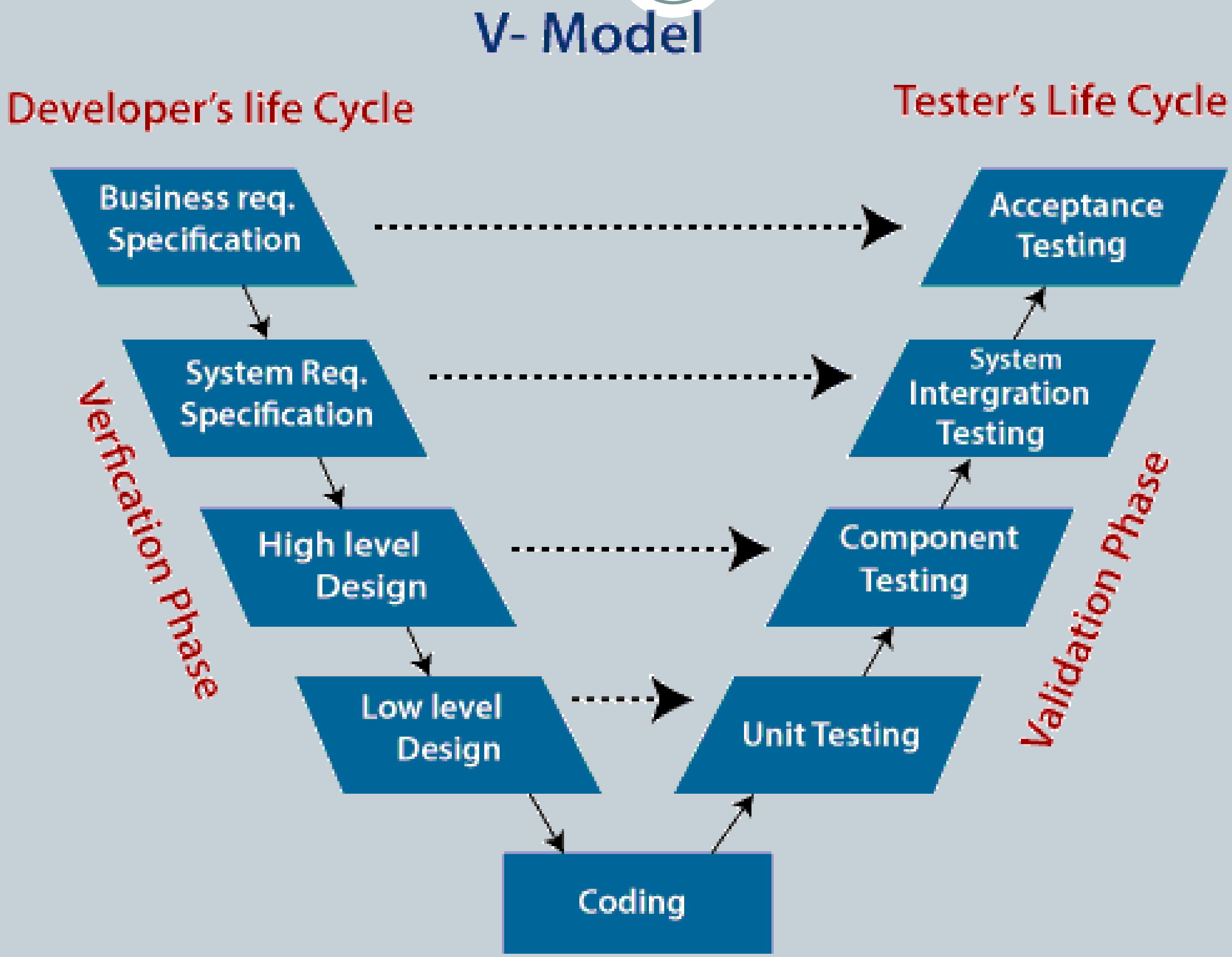
The left side of the model is Software Development Life Cycle - **SDLC**

The right side of the model is Software Test Life Cycle - **STLC**

The entire figure looks like a V, hence the name **V - model**



Phases of Verification Phase of V-model:



Phases of Verification Phase of V-model:

- **There are the various phases of Verification Phase of V-model:**
- **Business requirement analysis:** This is the first step where product requirements understood from the customer's side. This phase contains detailed communication to understand customer's expectations and exact requirements.
- **System Design:** In this stage system engineers analyze and interpret the business of the proposed system by studying the user requirements document.
- **Architecture Design:** The baseline in selecting the architecture is that it should understand all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology detail, etc. The integration testing model is carried out in a particular phase.
- **Module Design:** In the module design phase, the system breaks down into small modules. The detailed design of the modules is specified, which is known as Low-Level Design
- **Coding Phase:** After designing, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are some guidelines and standards for coding. Before checking in the repository, the final build is optimized for better performance, and the code goes through many code reviews to check the performance.

Phases of Validation Phase of V-model:

- **There are the various phases of Validation Phase of V-model:**
- **Unit Testing:** In the V-Model, Unit Test Plans (UTPs) are developed during the module design phase. These UTPs are executed to eliminate errors at code level or unit level. A unit is the smallest entity which can independently exist, e.g., a program module. Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/ units.
- **Integration Testing:** Integration Test Plans are developed during the Architectural Design Phase. These tests verify that groups created and tested independently can coexist and communicate among themselves.
- **System Testing:** System Tests Plans are developed during System Design Phase. Unlike Unit and Integration Test Plans, System Tests Plans are composed by the client?s business team. System Test ensures that expectations from an application developer are met.
- **Acceptance Testing:** Acceptance testing is related to the business requirement analysis part. It includes testing the software product in user atmosphere. Acceptance tests reveal the compatibility problems with the different systems, which is available within the user atmosphere. It conjointly discovers the non-functional problems like load and performance defects within the real user atmosphere.

When to use V-Model?

- When the requirement is well defined and not ambiguous.
- The V-shaped model should be used for small to medium-sized projects where requirements are clearly defined and fixed.
- The V-shaped model should be chosen when sample technical resources are available with essential technical expertise.

V-model

- **Advantages of V-model:**
- Simple and easy to use.
- Testing activities like planning, **test designing** happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

V-model

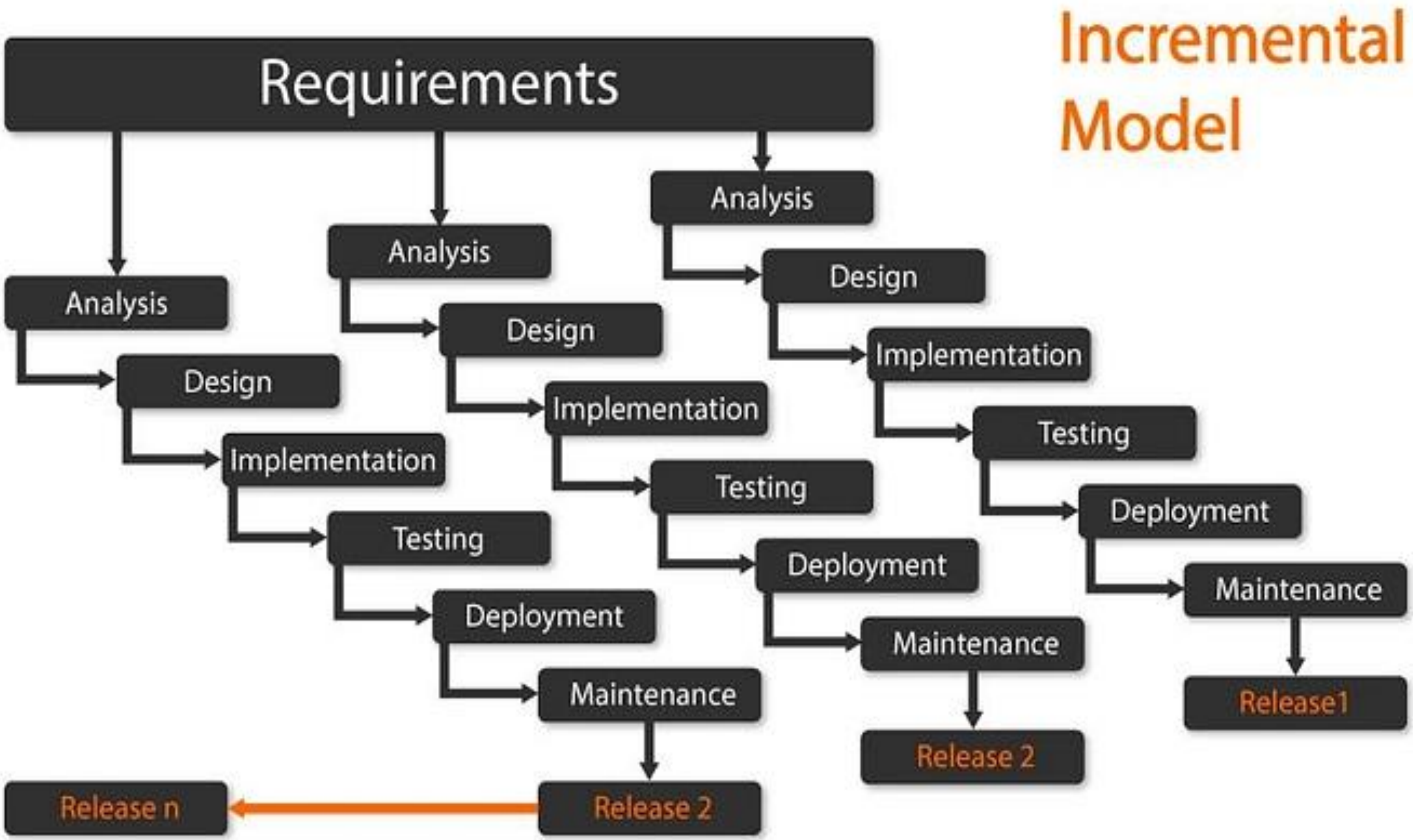
Disadvantages of V-model:

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

IV. Incremental Model

- In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Incremental model is a type of software development model like V-model, Agile model etc.
- In this model, each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

Incremental Model



The various phases of incremental model are as follows:

- **1. Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
- **2. Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
- **3. Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

The various phases of incremental model are as follows:

- **3. Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

- **4. Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

When we use the 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.

Incremental Model

- **Advantages of Incremental model:**
- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Incremental Model

Disadvantages of Incremental model:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than **waterfall**.

V. 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

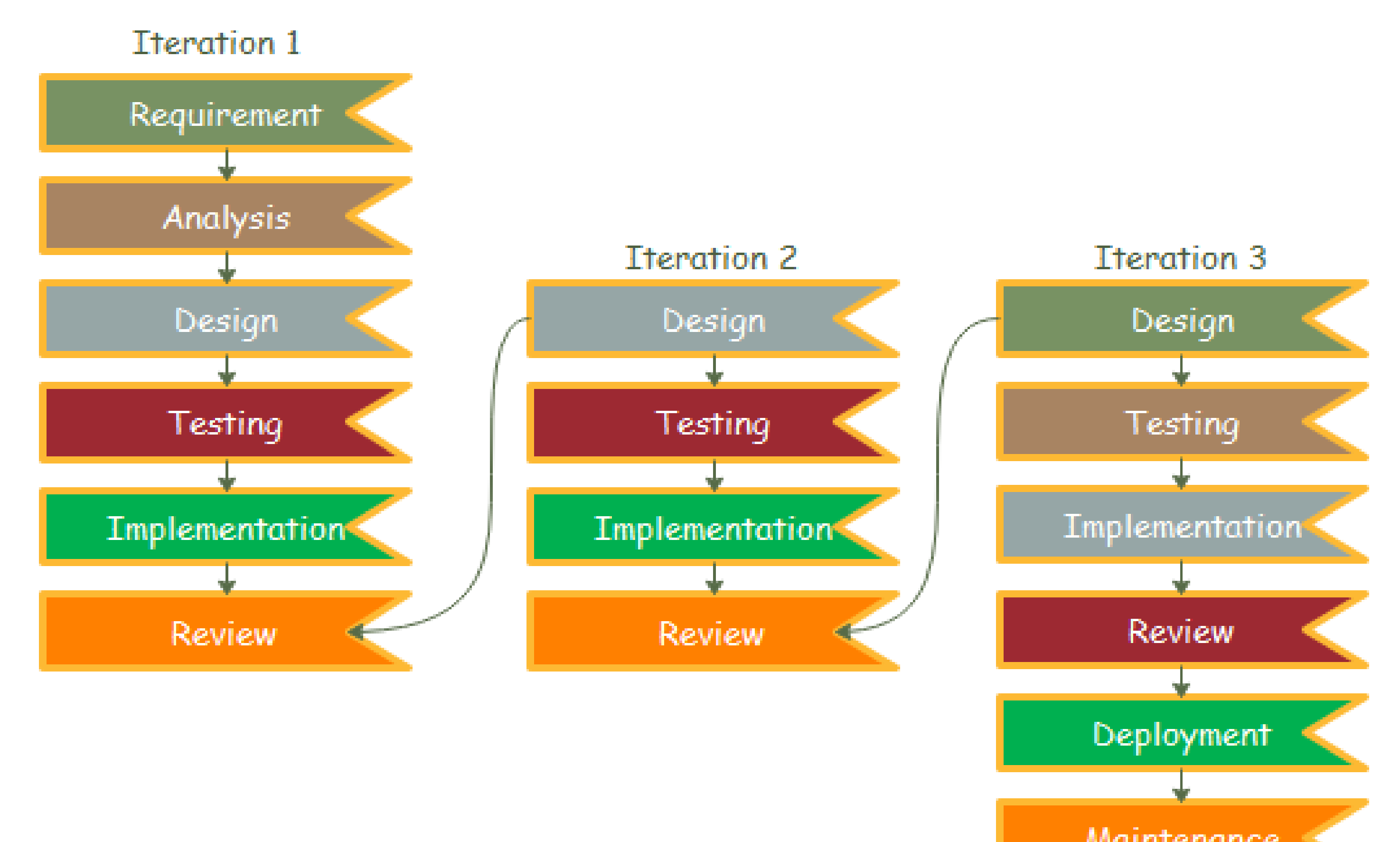


Fig. Iterative Model

Iterative Model

- **1. 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.
- **2. 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.
- **3. Implementation:** In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

Iterative Model

- **4. 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.
- **5. Deployment:** After completing all the phases, software is deployed to its work environment.
- **6. 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.
- **7. 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.

Iterative Model

- **Advantages of Iterative model:**
- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

Iterative Model

- **Disadvantages of Iterative model:**
- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

When to use iterative model:

- **When to use iterative model:**
- Requirements of the complete system are clearly defined and understood.
- When the project is big.
- Major requirements must be defined; however, some details can evolve with time.

VI. BigBang Model

- According to a theory known as big bang theory the existence of this world and other stars and all those things found in the space is the result of a big blast.

- The name of Big Bang model in software engineering is inspiration from the Big bang theory.

BigBang Model

In this model, developers do not follow any specific process. Development begins with the necessary funds and efforts in the form of inputs. And the result may or may not be as per the customer's requirement, because in this model, even the customer requirements are not defined.

This model is ideal for small projects like academic projects or practical projects. One or two developers can work together on this model.

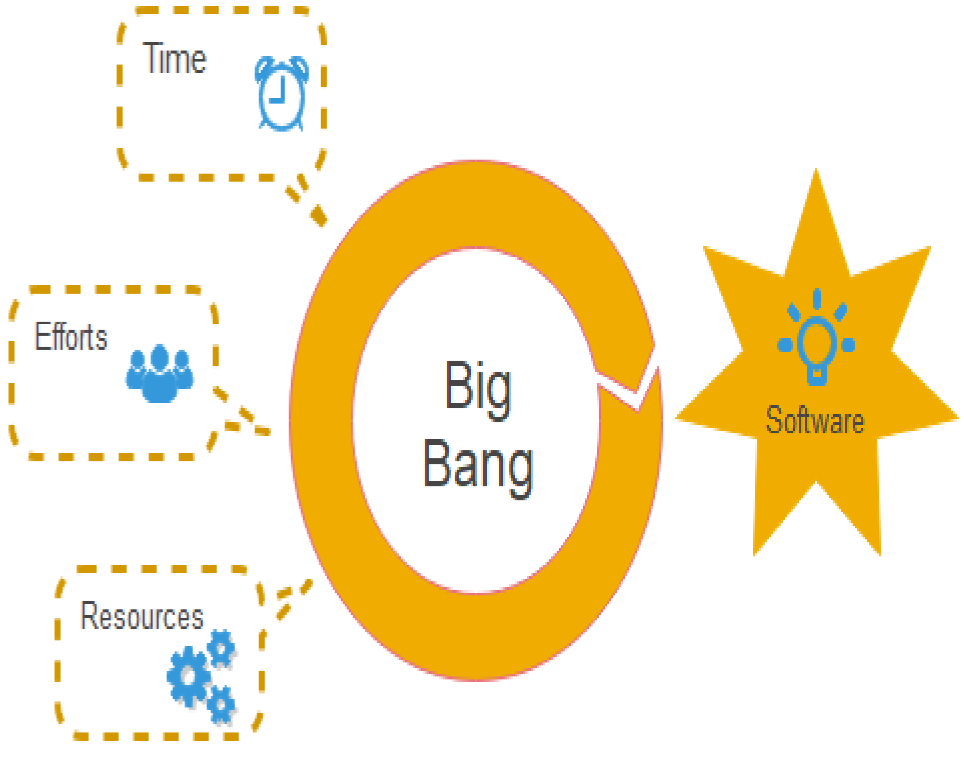


Fig. Big Bang Model

BigBang Model

- **Advantages of Big Bang Model:**
- The advantage of Big Bang is that it's very simple and easy to implement. This model requires very little or no planning. There is no formal procedure are required before starting of any project so this model is easy to manage. It is ideal for repetitive or small projects with minimum risks.

BigBang Model

- **Disadvantages of Big Bang Model:**
- Due to there is no pre planning required before starting the project hence the Big Bang model is a very high risky model. In addition if changes in the requirements or misunderstood requirements may even lead to complete reversal or scraping of the project.

BigBang Model

- **When to use Big Bang Model?**

This model is required when this project is small like an academic project or a practical project. This method is also used when the size of the developer team is small and when requirements are not defined, and the release date is not confirmed or given by the customer.

VII. Agile Model

- 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.

Agile Model

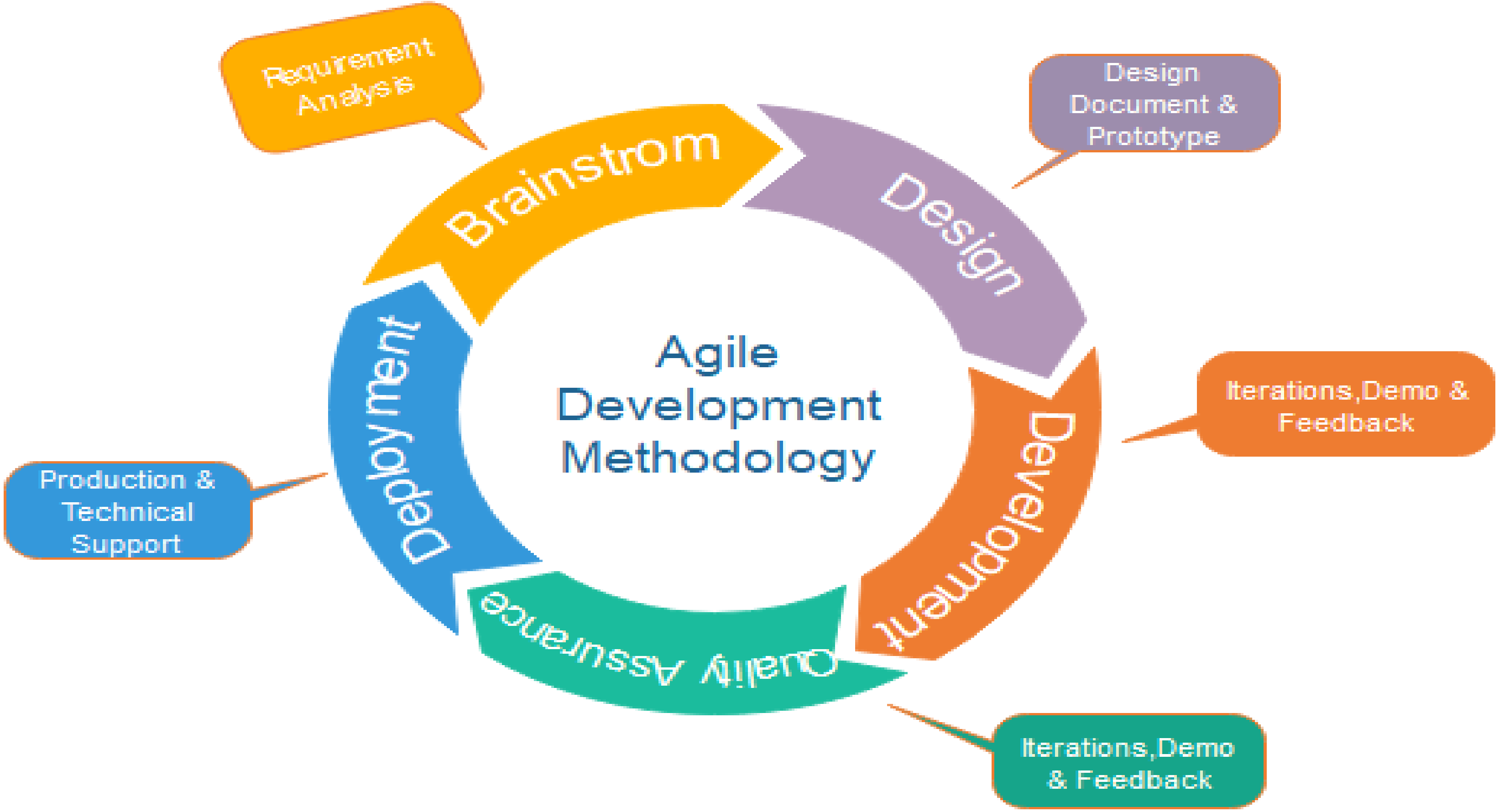

Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Development
Iterations,Demo & Feedback
Quality Assurance
Iterations,Demo & Feedback
Deployment
Production & Technical Support
Agile Development Methodology

Fig. Agile Model

Evolutionary Process Models

- **Evolutionary model** is a combination of Iterative and Incremental model of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model. Some initial requirements and architecture envisioning need to be done.
- It is better for software products that have their feature sets redefined during development because of user feedback and other factors.

Evolutionary Process Models

- Evolutionary Process Models
- Evolutionary models are iterative type models.
- They allow to develop more complete versions of the software.
- **Following are the evolutionary process models.**

1. The prototyping model
2. The spiral model

Prototype Model

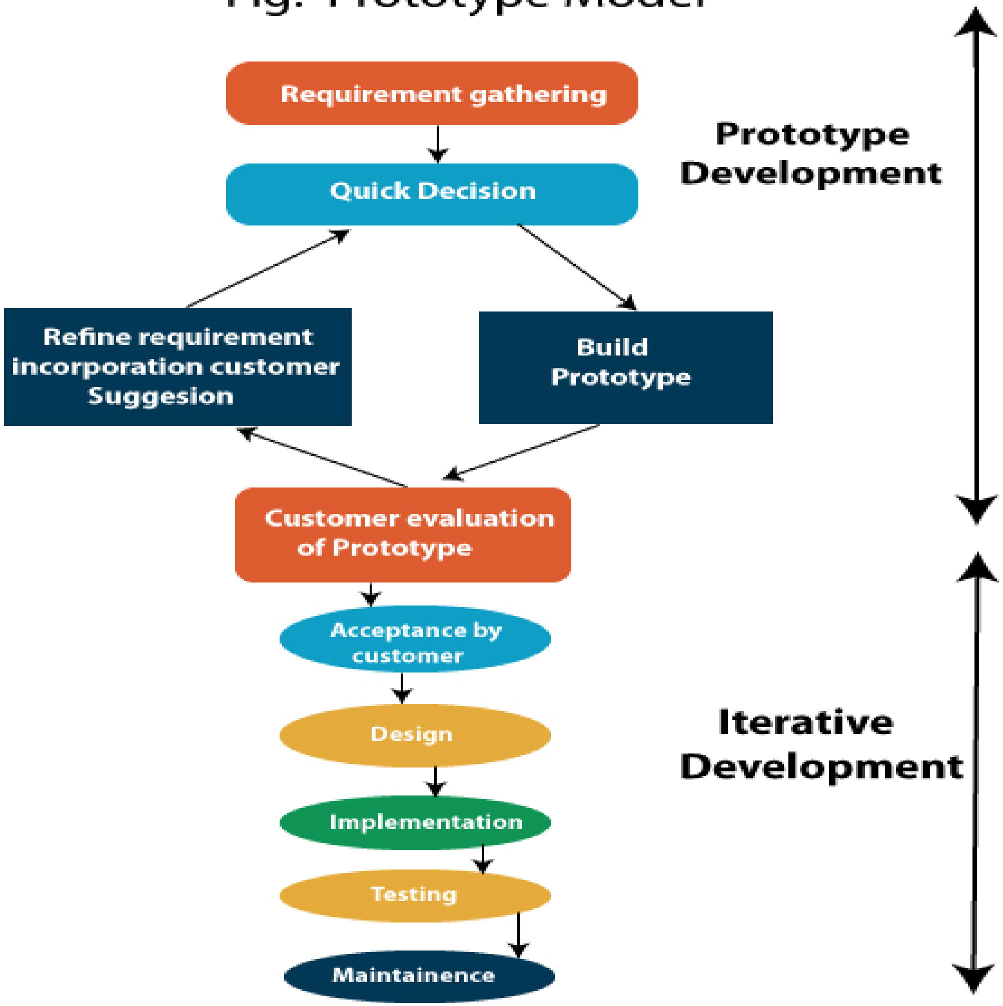
- Prototype is defined as first or preliminary form using which other forms are copied or derived.
- Prototype model is a set of general objectives for software.
- It does not identify the requirements like detailed input, output.
- It is software working model of limited functionality.
- In this model, working programs are quickly produced.

Prototype Model

- The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

Prototype Model

Fig: Prototype Model



Prototype Model

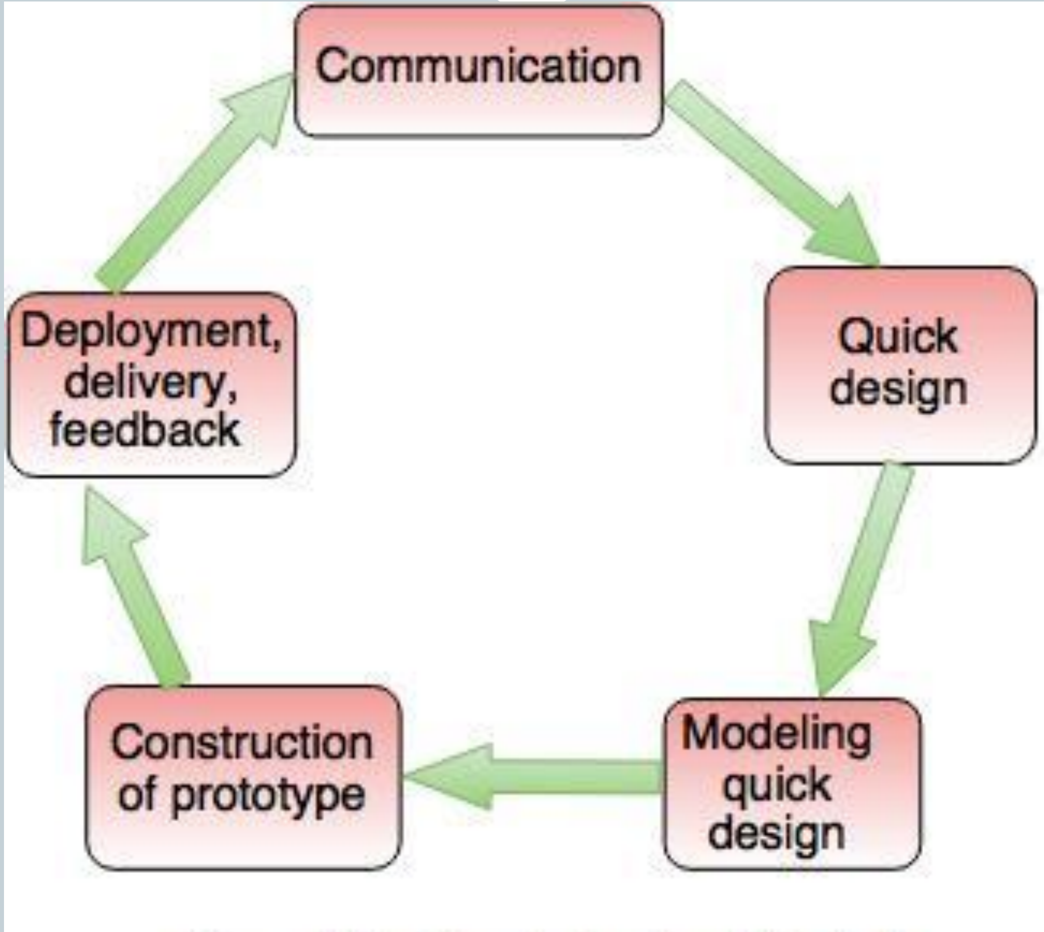


Fig. - The Prototyping Model

Prototype Model

- **The different phases of Prototyping model are:**

  1. Communication
  In this phase, developer and customer meet and discuss the overall objectives of the software.

  2. Quick design Quick design is implemented when requirements are known.
- It includes only the important aspects like input and output format of the software.
- It focuses on those aspects which are visible to the user rather than the detailed plan.
- It helps to construct a prototype.

Prototype Model

- **3. Modeling quick design**This phase gives the clear idea about the development of software because the software is now built.
- It allows the developer to better understand the exact requirements.
- **4. Construction of prototype**
  The prototype is evaluated by the customer itself.

  5. Deployment, delivery, feedback If the user is not satisfied with current prototype then it refines according to the requirements of the user.
- The process of refining the prototype is repeated until all the requirements of users are met.
- When the users are satisfied with the developed prototype then the system is developed on the basis of final prototype.

Prototype Model

- **Advantages of Prototyping Model**Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
- In the development process of this model users are actively involved.
- The development process is the best platform to understand the system by the user.
- Errors are detected much earlier.
- Gives quick user feedback for better solutions.
- It identifies the missing functionality easily. It also identifies the confusing or difficult functions.

Prototype Model

Disadvantages of Prototyping Model:

- The client involvement is more and it is not always considered by the developer.
- It is a slow process because it takes more time for development.
- Many changes can disturb the rhythm of the development team.
- It is a thrown away prototype when the users are confused with it.

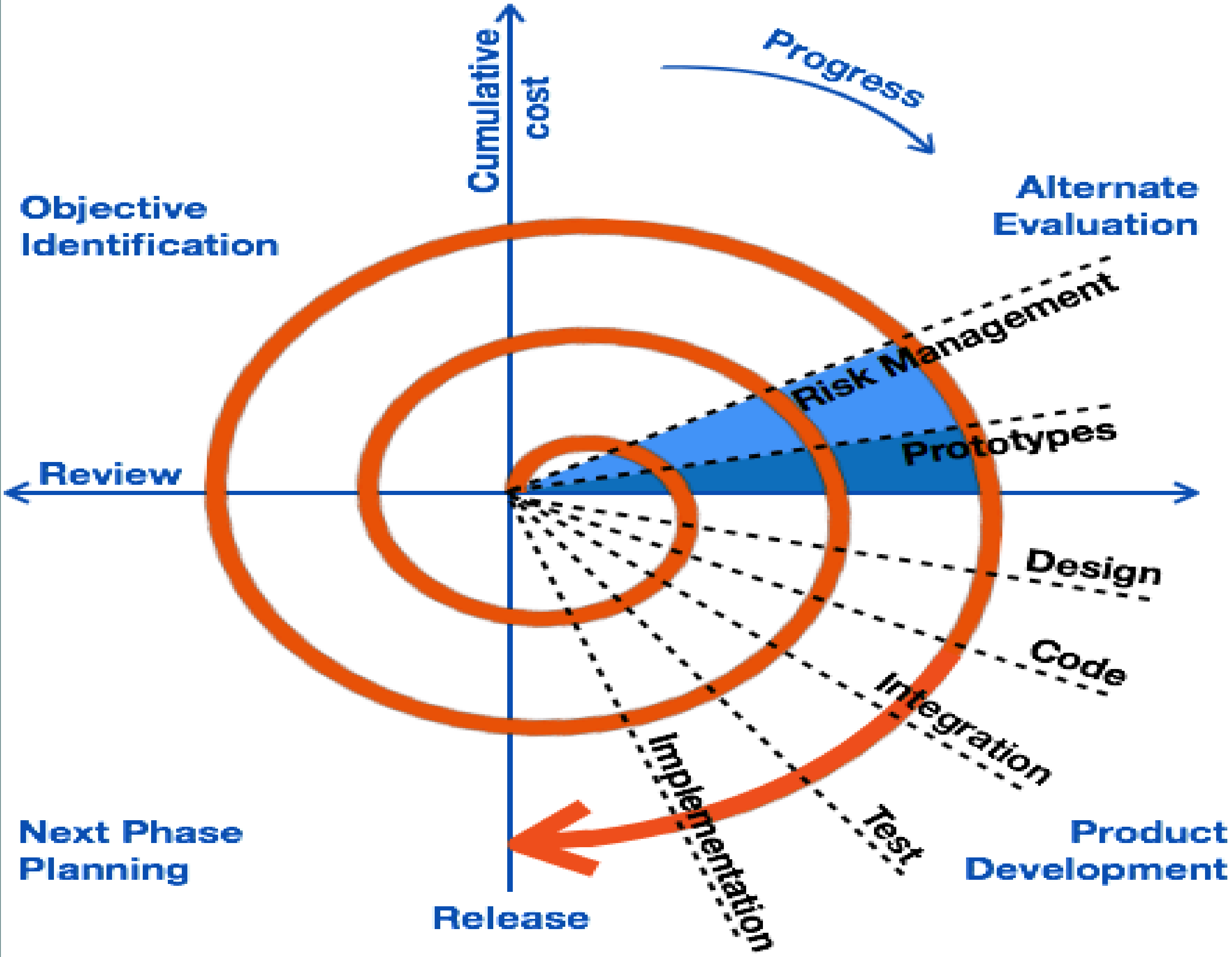
2. The Spiral model

- Barry Boehm . B Introduced Spiral Model in 1986.
- Spiral model is a risk driven process model.
- It is used for generating the software projects.
- In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented.
- It is a combination of prototype and sequential model or waterfall model.
- In one iteration all activities are done, for large project's the output is small.

The Spiral model

- **Spiral model** is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a **Phase of the software development process.** 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 the spiral model.

The Spiral model

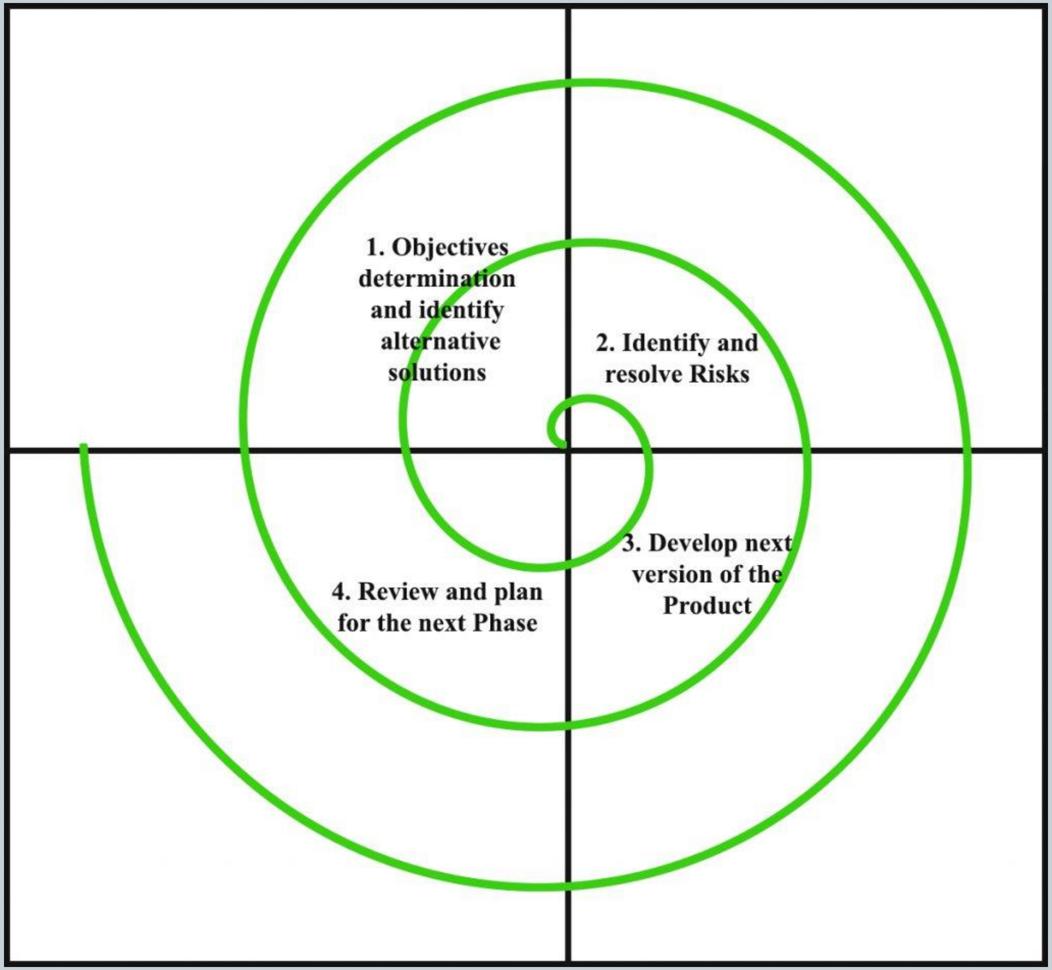


The spiral model has two dimensions namely:

The **radial** dimension of the model represents the cumulative costs and the **angular** dimension represents the progress made in completing each cycle. Each loop of the spiral from X-axis clockwise through 360^o represents one phase.

The Spiral model

The framework activities of the spiral model are as shown in the following figure.


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

The Spiral model

- **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

- **Identify and resolve Risks:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.

The Spiral model

- **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

- **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

- The **risk-driven** feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach. An essential element of the model is that each period of the spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle. The spiral model works for development as well as enhancement projects.

The Spiral model

- **Advantages of Spiral Model**It reduces high amount of risk.
- It is good for large and critical projects.
- It gives strong approval and documentation control.
- In spiral model, the software is produced early in the life cycle process.
- **Disadvantages of Spiral Model**
- It can be costly to develop a software model.
- Risk analysis needed highly particular expertise
- It is not used for small projects.

The Spiral model

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

THANK YOU