CSE- 321
Software Engineering

Lecture : 07

Software Processes

Fahad Ahmed

Lecturer, Dept. of CSE

E-mail: fahadahmed@uap-bd.edu

Lecture Outlines

- ❖Software Processes
- ❖Software Process Model

Software process

What is a software process?

A software process is **a sequence of activities** that leads to the production of a software product.

Generic activities in all software processes are:

- **Software specifications:**
  The functionality of the software and constraints on its operation must be defined.
- **Software development:**
  The software to meet the requirement must be produced.
- **Software validation:**
  The software must be validated to ensure that it does what the customer wants.
- **Software evolution:**
  The software must evolve to meet changing client needs.

Software Process Model

What is a software **process model**?

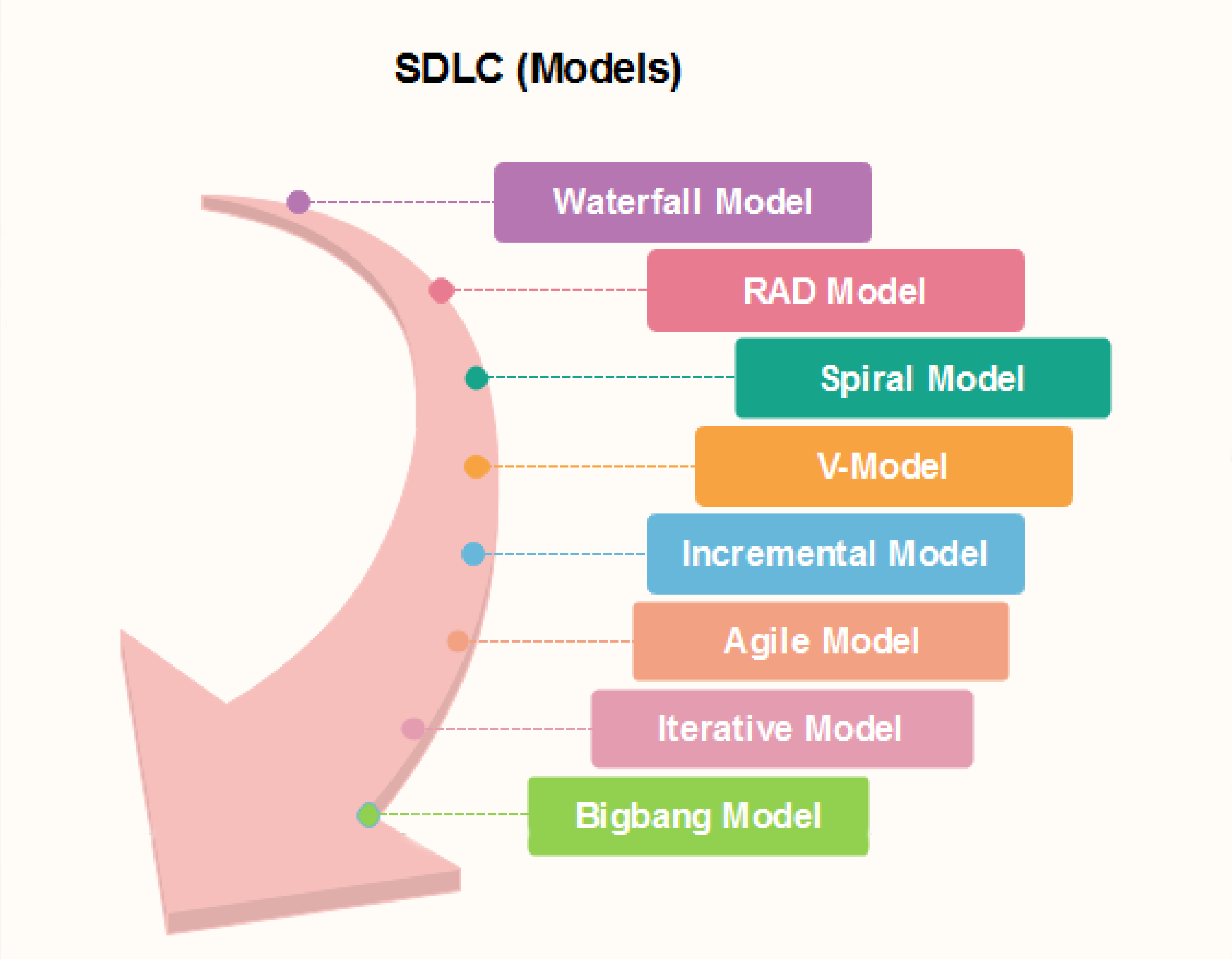
A software process model is **an abstract representation of a process**. It presents a **description of a process** from some **particular perspective**.

Process descriptions may also include:

- **Products,** which are the outcomes of a process activity;
- **Roles,** which reflect the responsibilities of the people involved in the process;
- **Pre- and post-conditions,** which are statements that are true before and after a process activity has been enacted or a product produced.

In the **Cleanroom process** each software increment is formally specified and this specification is transformed into an implementation. The Cleanroom process, which was originally developed by **IBM**.

Popular SDLC models



SDLC (Models)

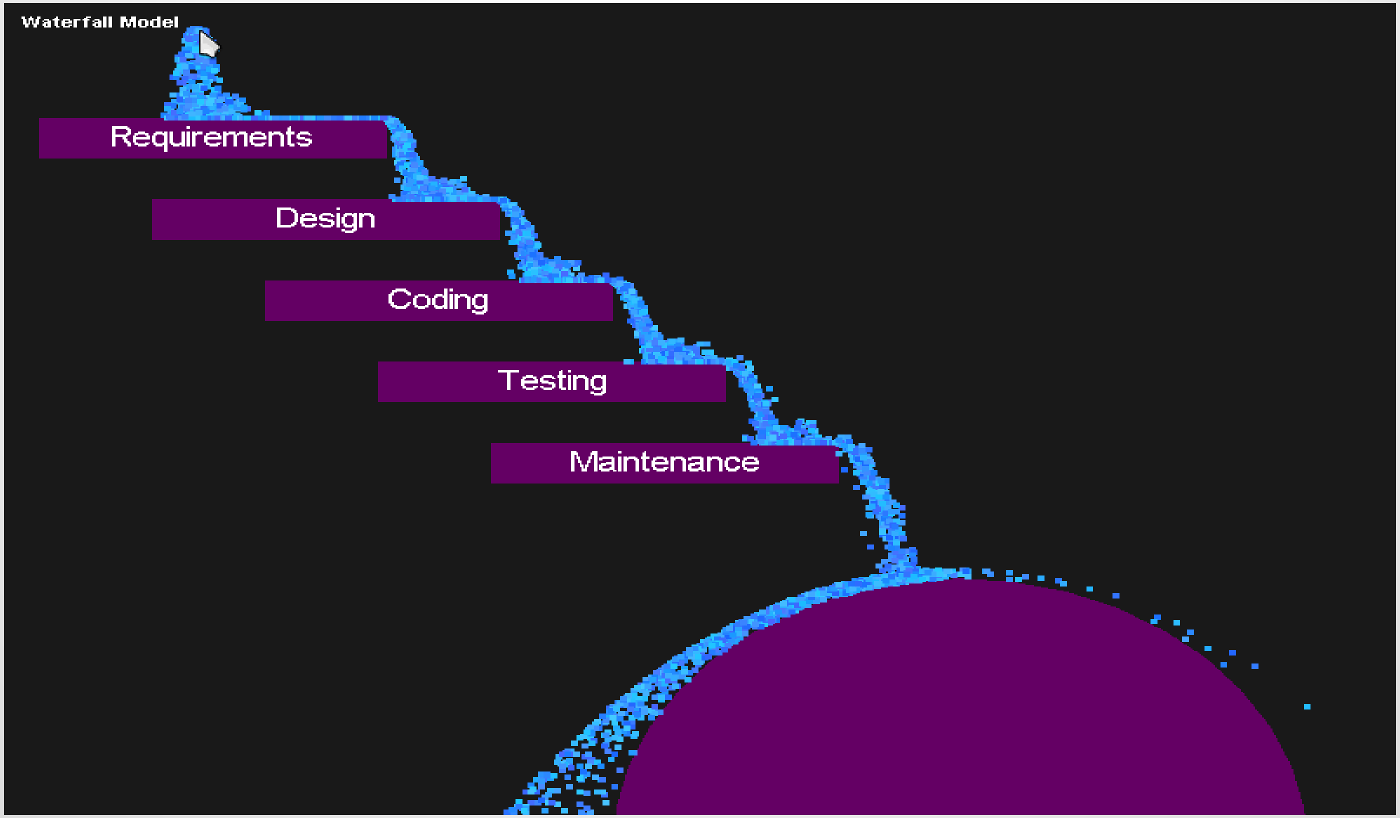
- Waterfall Model
- RAD Model
- Spiral Model
- V-Model
- Incremental Model
- Agile Model
- Iterative Model
- Bigbang Model

Waterfall != Waterfall Model

Waterfall Model

What is The Waterfall Model? (Iterative)

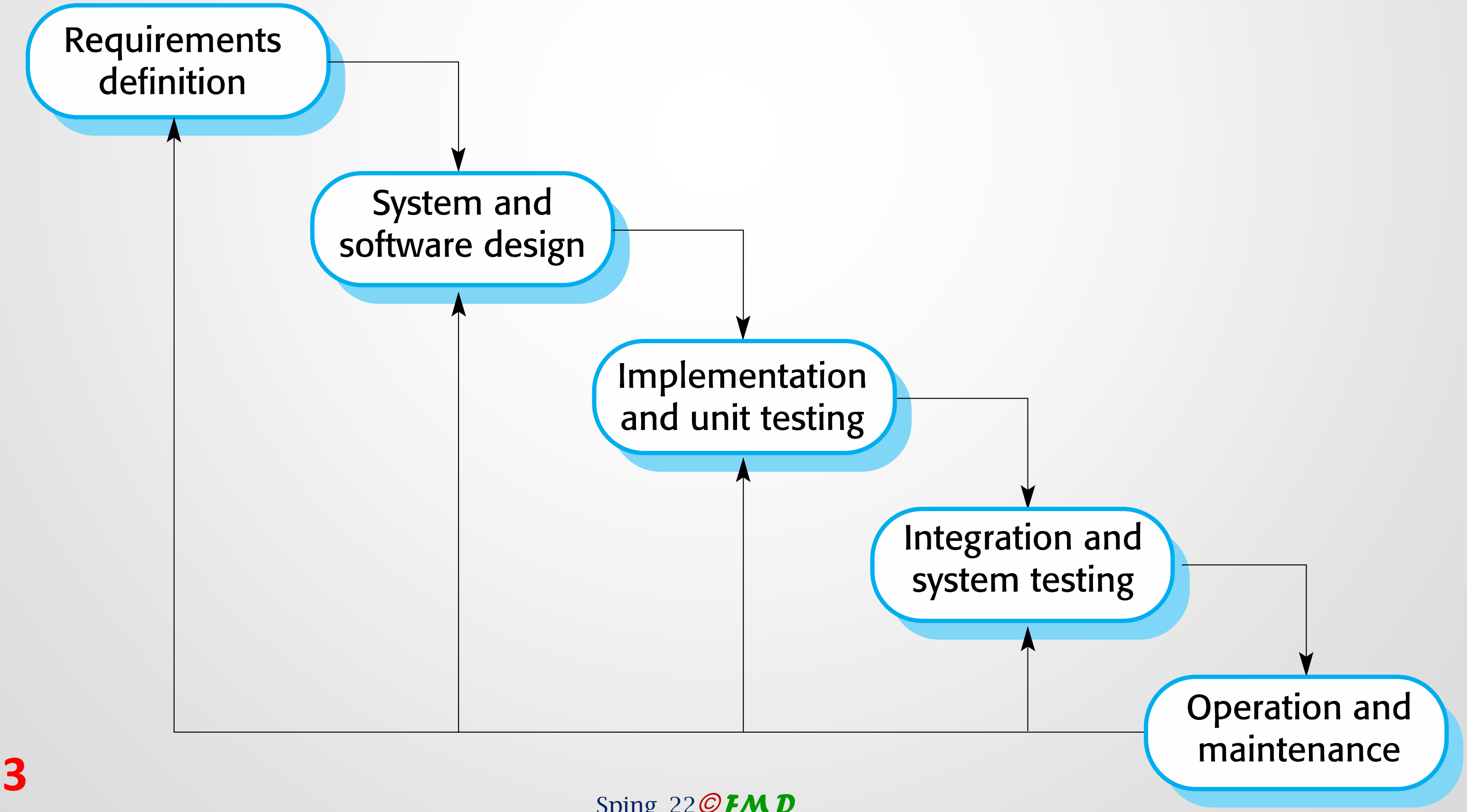
- Also called **classic software life cycle** or **sequential model or linear-sequential life cycle model**
- Process activities (phases/stages) are **clearly separated**
- After a number of iterations, phases of the life cycle (such as specification and design) are “frozen”
- It was introduced in 1970 by Winston Royce.



Waterfall Model

What is The Waterfall Model? (Iterative)

- Also called **classic software life cycle** or **sequential model or linear-sequential life cycle model**
- Process activities (phases/stages) are **clearly separated**
- After a number of iterations, phases of the life cycle (such as specification and design) are “frozen”
- It was introduced in 1970 by Winston Royce.


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Waterfall Model

When to use SDLC Waterfall Model

Waterfall model can be used when

- Requirements are not changing frequently
- Can be used also for parts of larger software systems
- Requirement is clear
- Environment is stable
- Technology and tools used are not dynamic and is stable
- Resources are available and trained

Drawback of the waterfall model

- Difficult to incorporate change requests.
- Incremental delivery not supported.
- Overlapping of phases not supported.
- Risk handling not supported.
- Limited customer interactions.

Examples of Waterfall Model

Examples of Waterfall Model

In the olden days, Waterfall model was used to develop enterprise applications like

- **Customer Relationship Management (CRM)** systems,
- **Human Resource Management Systems** (HRMS),
- **Supply Chain Management Systems**,
- **Inventory Management Systems,**
- **Point of Sales (POS)**

systems for Retail chains etc.

Waterfall model was used significantly in the development of software till the year 2000. Even after the **Agile manifesto** was published in 2001, Waterfall model continued to be used by many organization till the last decade.

Examples of Waterfall Model

Examples of Waterfall Model

However, there are some areas where Waterfall model was continued to be preferred.

Consider a system where **human life** is on the line, where a system failure could result in one or more deaths.

In some countries, such mishaps could lead to imprisonment for those who are accountable.

Consider a system where time and money were secondary considerations and **human safety was first**.

Rapid application development model (RAD)

Rapid Application Development, or RAD for short, is a methodology for fast application production using an iterative software development process.

Rapid application development model (RAD) is a linear sequential software development process model that emphasizes a concise development cycle using an element based construction approach.

Rapid Application Development process is an adoption of the waterfall model, it targets at developing software in a short span of time.

The Rapid Application Development Model was first proposed by **IBM in 1980's**. The critical feature of this model is the use of **powerful development tools and techniques**.

Rapid application development model (RAD)

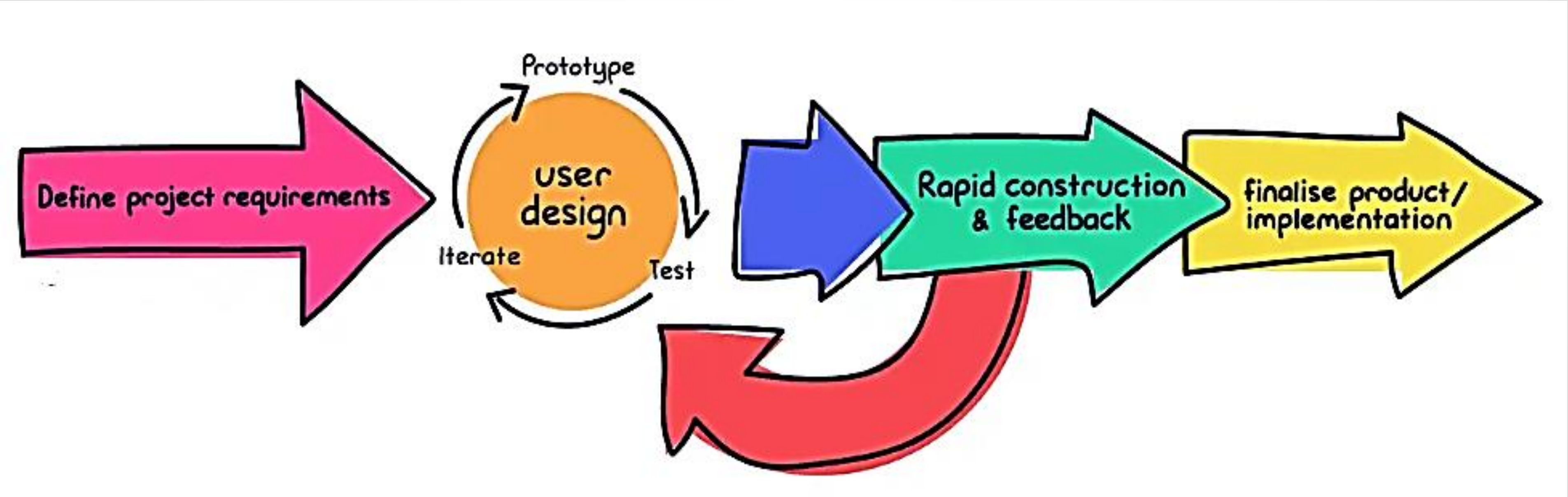
A rapid application development cycle consists of four steps:

1.Define project requirements;

2.Prototype;

3.Rapid construction & feedback gathering; and

4.Finalize product / implementation.


Define project requirements
Prototype
user design
Iterate
Test
Rapid construction & feedback
finalise product/ implementation

The 5 Essential Stages of a RAD Model

1.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.

2. 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.

3. 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.

4. Application Generation: Automated tools are used to facilitate construction of the software; even they use the fourth-generation language (4GL) technologies.

5. 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.

https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html

Rapid application development model (RAD)

When to use RAD Model?

- ❖ RAD was the solution for projects with unclear or evolving requirements. It takes the pressure off the customer to have a complete and unambiguous specification before a development project could begin.
- ❖ RAD should only be used when a system can be modulated to be delivered incrementally. If you need to build an internal business tool or a customer facing portal, RAD can assist you to deliver better experience to your end users. If software is mission critical however, and technical risk is high, i.e. outcomes affect people’s lives, a RAD approach is inappropriate.
- ❖ When a prototype is expected by the customer within two to three months;

Rapid application development model (RAD)

Advantage of RAD Model

- Use of reusable components helps to **reduce the cycle time** of the project.
- **Feedback from the customer** is available at initial stages.
- Reduced costs as fewer developers are required.
- Use of powerful development tools results in better quality products in comparatively shorter time spans.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.

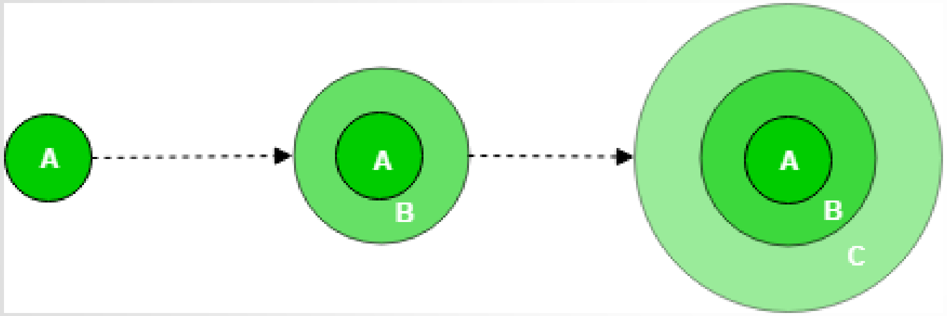
Disadvantage of RAD Model

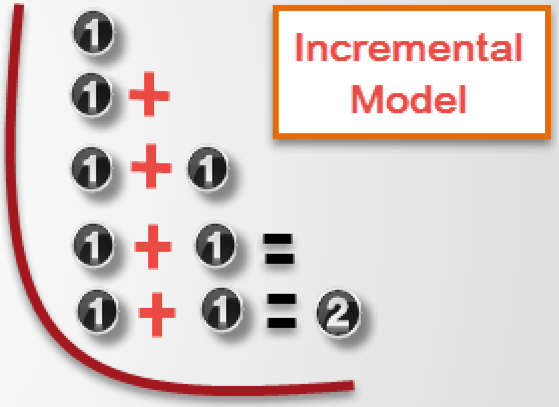
- It required **highly skilled designers**.
- All application is not compatible with RAD.
- For smaller projects, we cannot use the RAD model.
- On the high technical risk, it's not suitable.
- Required user involvement.

Incremental Model

Incremental process model is also know as **Successive version model**.

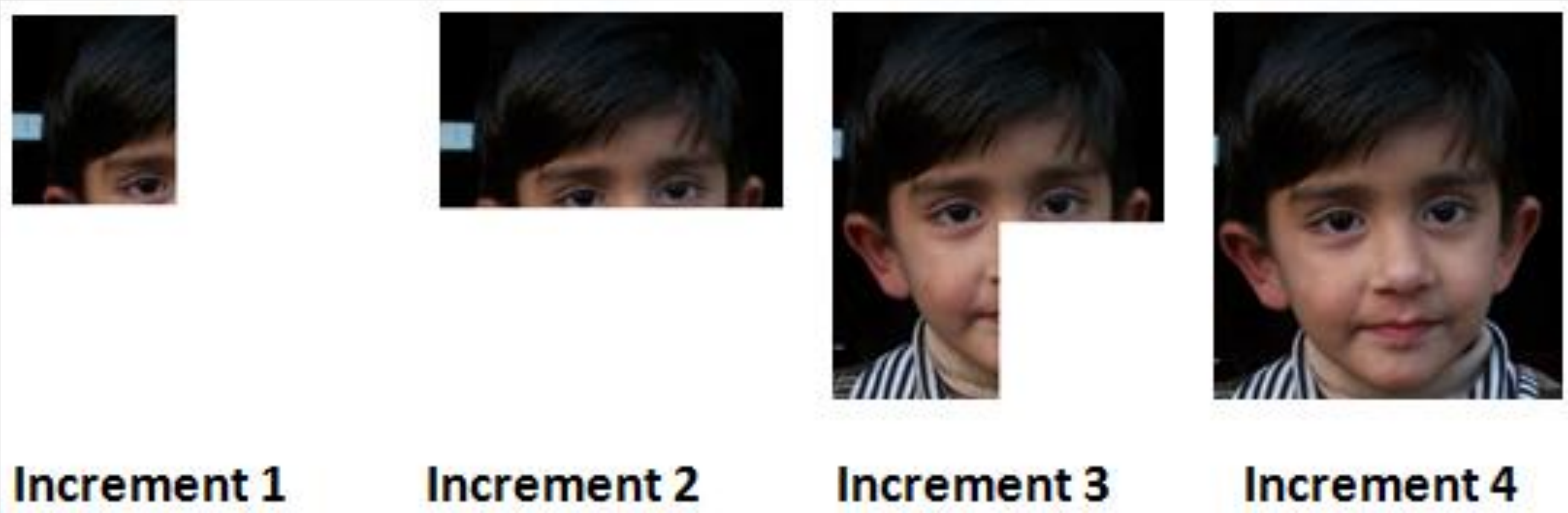
First, a simple working system implementing **only a few basic features is built** and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.





Incremental Model is a process of software development where requirements are **broken down into multiple standalone** modules of software development cycle.

Example of the incremental model (Just for understanding)



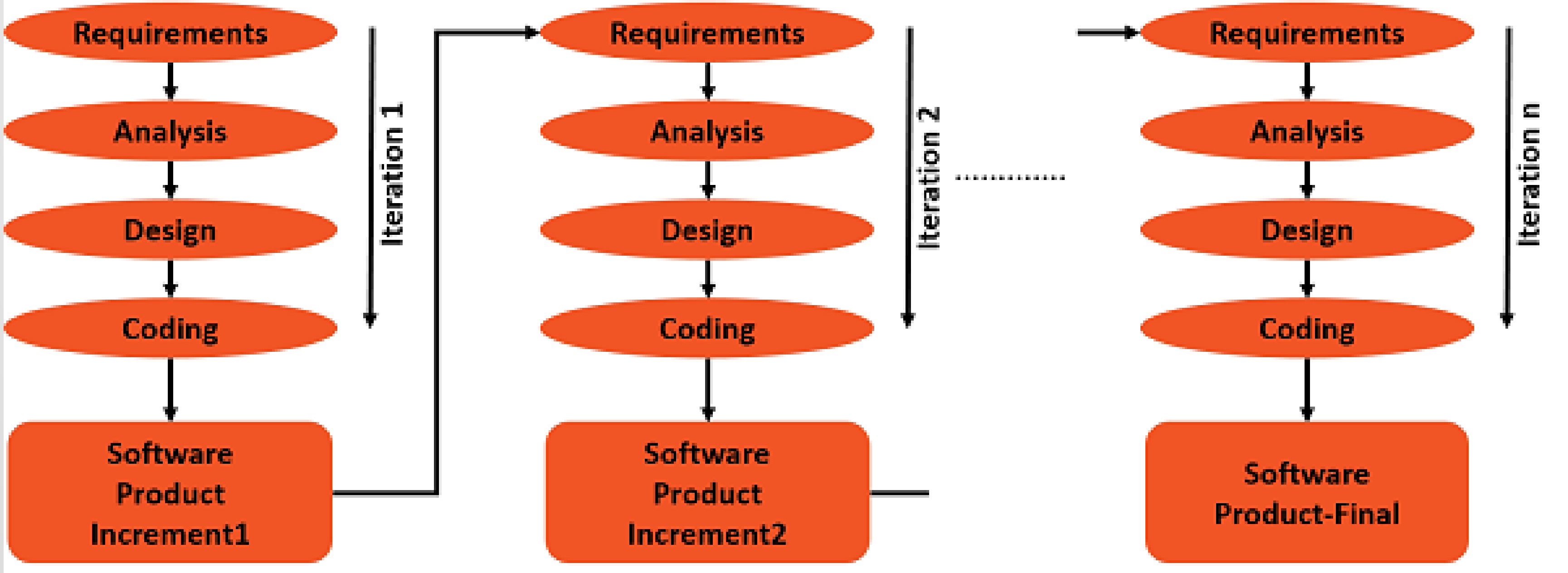
In this daily life example, we want to draw a picture, first, we draw the first part of the picture as shown in increment 1, Similarly after completing the first part of the picture we have to add one another part of the picture labelled as increment 2 in the above picture. Similarly, we complete this picture in four increments.

https://t4tutorials.com/incremental-model-in-software-engineering-advantages-disadvantages-examples-case-study-of-incremental-model/

Incremental Model

Each incremental version is usually developed using an iterative waterfall model of development.

As each successive version of the software is constructed and delivered, now the feedback of the Customer is to be taken and these were then incorporated into the next version. Each version of the software has more additional features than the previous ones.



A case study of incremental model

Suppose we want to develop a web-based social network with the following functionalities;

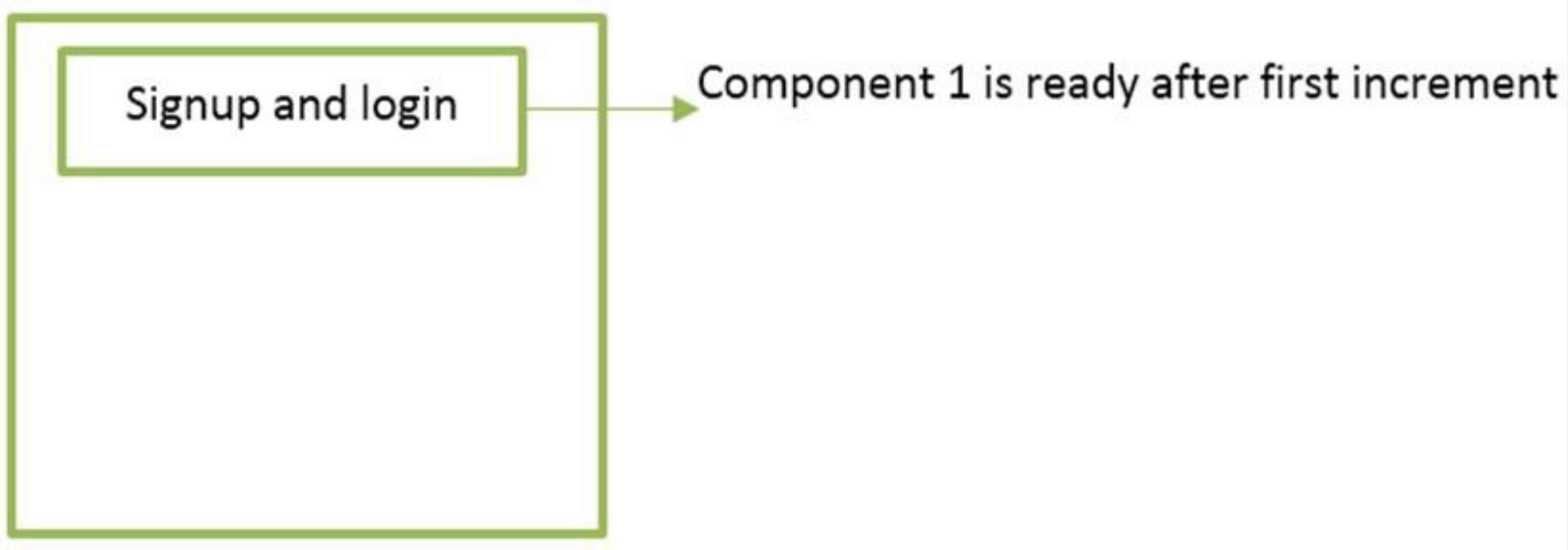
The user should signup for the system.
The user should log into the system and can send or accept the friend request.

How can use the incremental model in this scenario;

Solution:
We need to convert this system into separate components;
Component 1: Sign up and log in
Component 2: Send Friend request
Component 3: Accept friend request

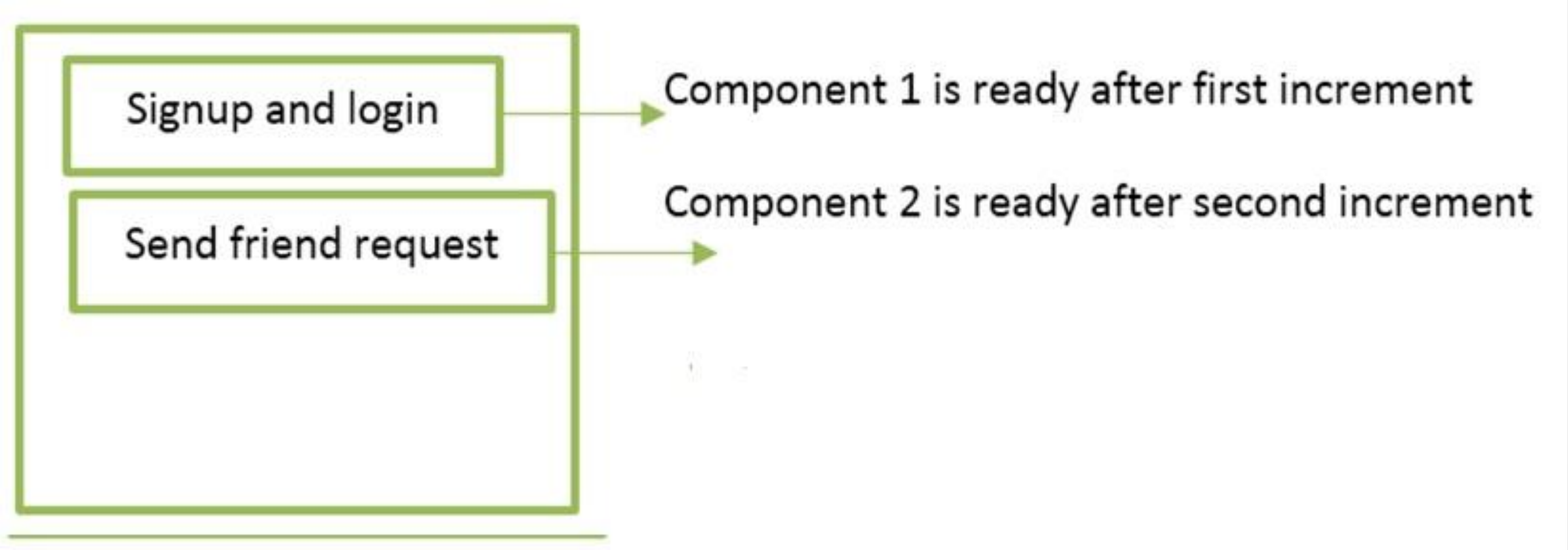
A case study of incremental model

Now, when we start our activities, then we can start with component 1(signup and login). This component undergoes the phases of requirements gathering and analysis, design, implementation, deployment, and maintenance. When this component is ready, we deliver this one component to the customer.



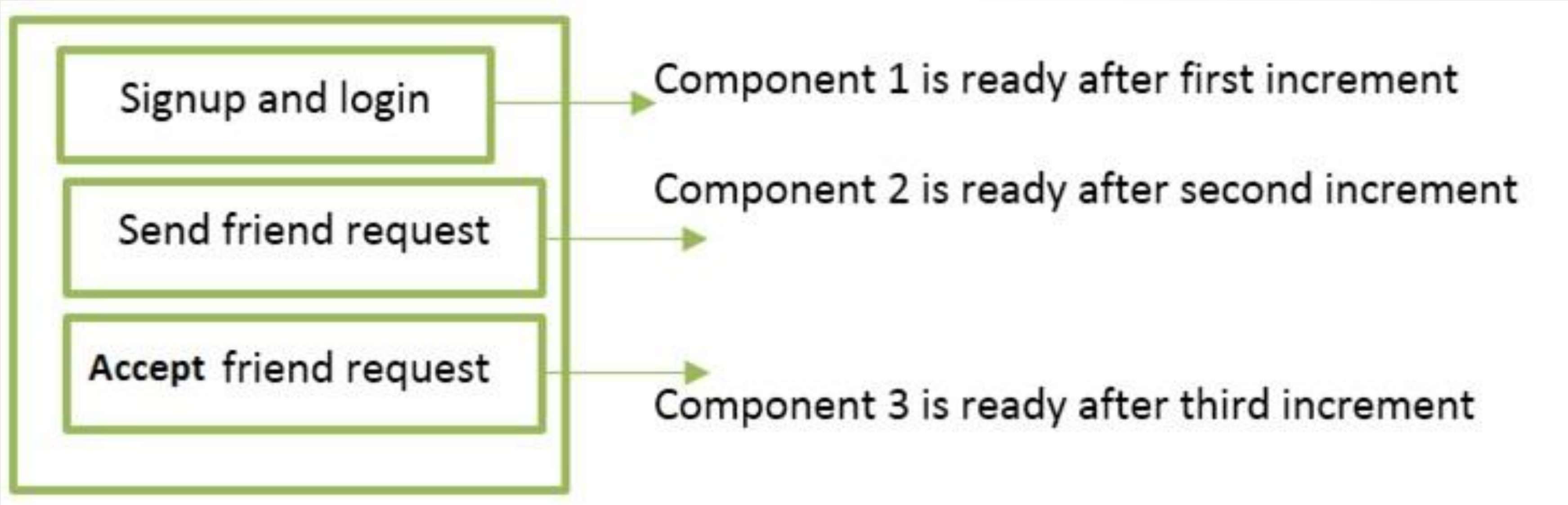
A case study of incremental model

After that, we add or increment another component 2 that sends the friend request. This component undergoes the phases of requirements gathering and analysis, design, implementation, deployment, and maintenance. When this component is ready, we deliver this one component to the customer.


Signup and login
Send friend request
Component 1 is ready after first increment
Component 2 is ready after second increment

A case study of incremental model

After that, we add or increment another component 3 that accepts a friend request. This component undergoes the phases of requirements gathering and analysis, design, implementation, deployment, and maintenance. When this component is ready, we deliver this one component to the customer.



Signup and login

Send friend request

Accept friend request

Component 1 is ready after first increment

Component 2 is ready after second increment

Component 3 is ready after third increment

Incremental Model

Types of Incremental model –

Staged Delivery Model –

Construction of **only one part of the project at a time**.

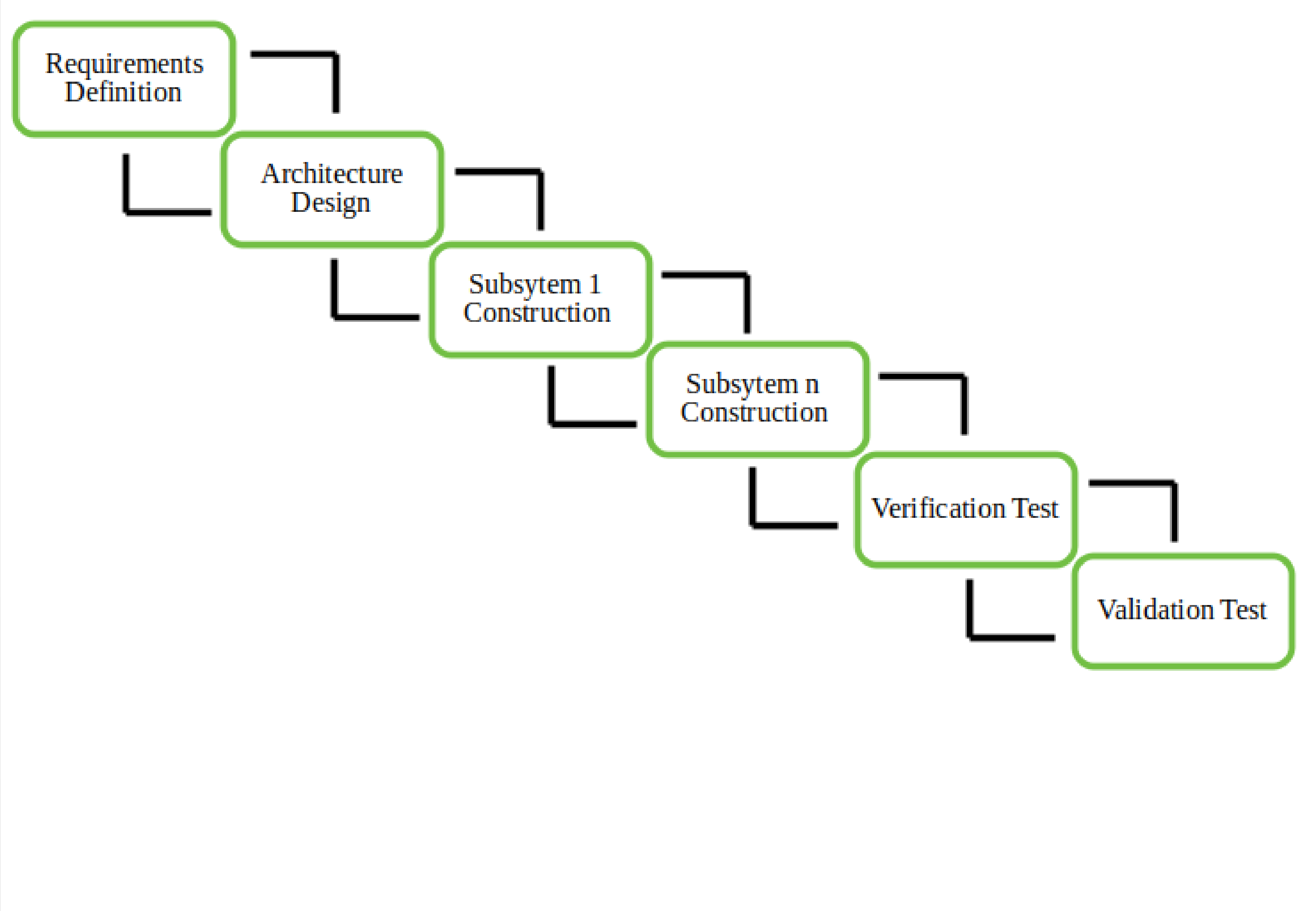
Parallel Development Model –

Different subsystems are developed at the same time. It can decrease the calendar time needed for the development, i.e. TTM (Time to Market), if enough Resources are available.

https://www.geeksforgeeks.org/software-engineering-incremental-process-model/

Incremental Model

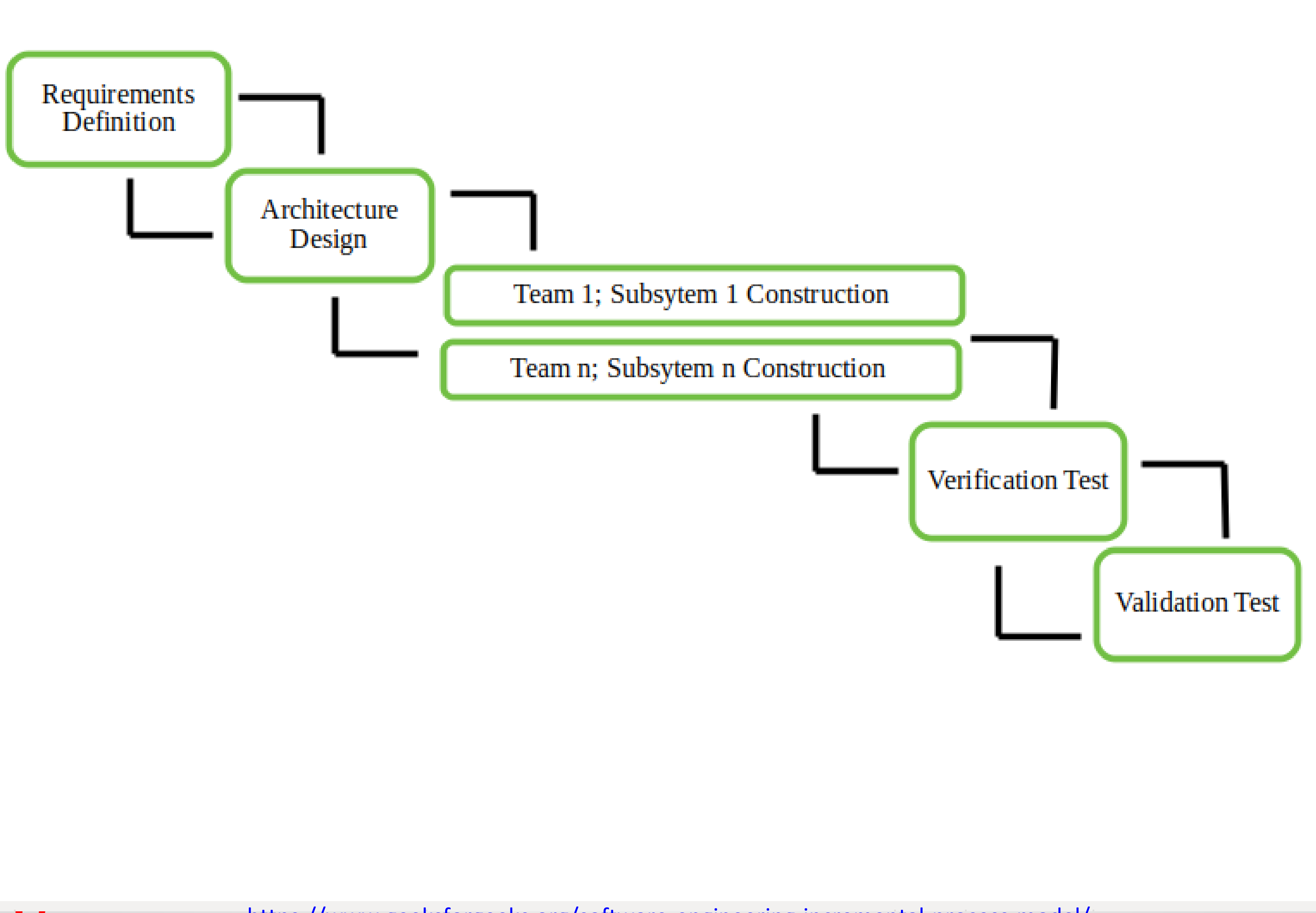
Staged Delivery Model –



https://www.geeksforgeeks.org/software-engineering-incremental-process-model/

Incremental Model

Parallel Development Model –



https://www.geeksforgeeks.org/software-engineering-incremental-process-model/

Incremental Model

When we use the Incremental Model?

- Funding Schedule, Risk, Program Complexity, or need for early realization of benefits.
- When Requirements are known up-front.
- When Projects having lengthy developments schedules.
- Projects with new Technology.

Advantage

- 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.

Disadvantage

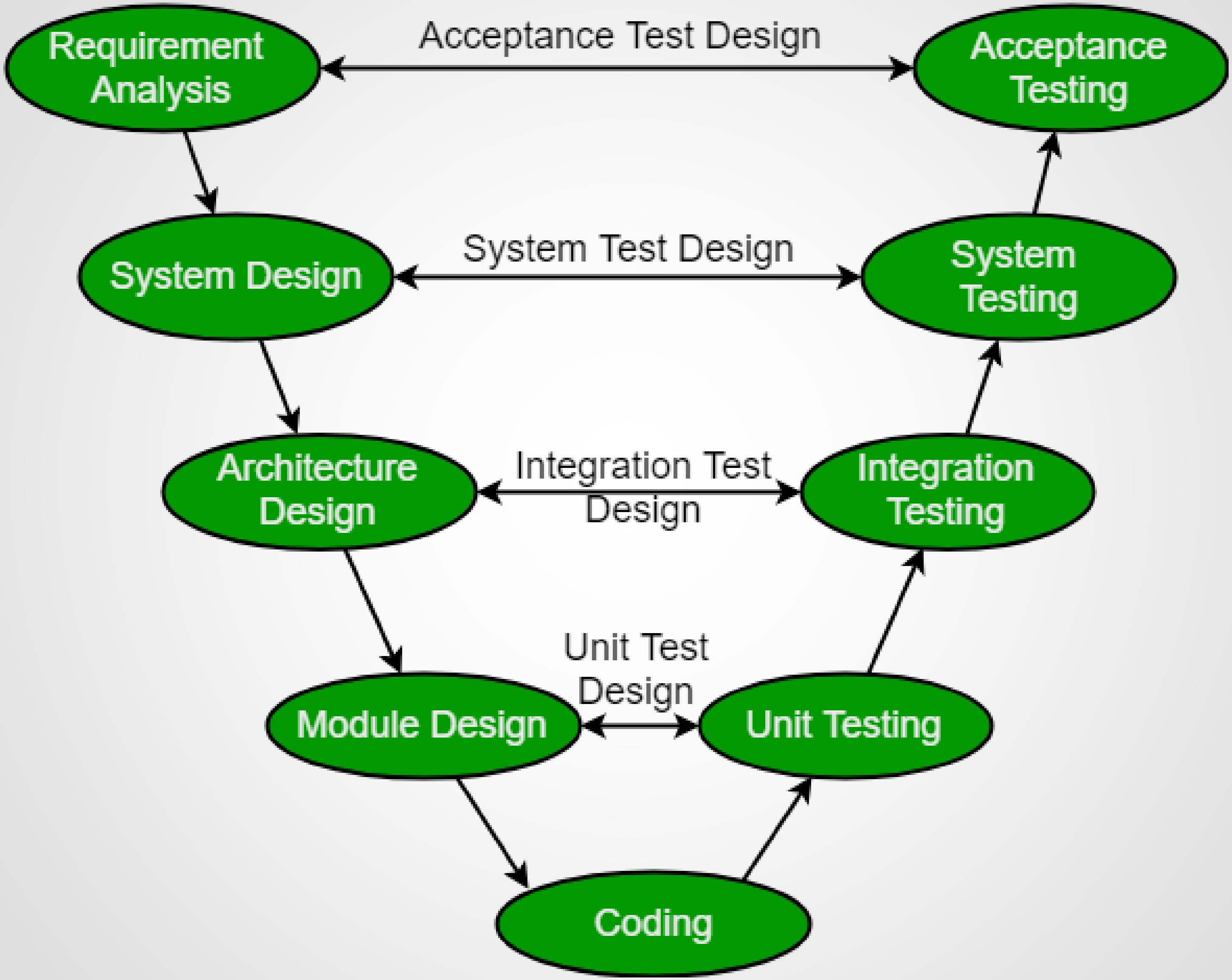
- Need for good planning
- Total Cost is high.
- Well defined module interfaces are needed.

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**
- V-Model referred to as the **Verification and Validation Model**.

- Process executes in a sequential manner in V-shape.
- It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase.
- The next phase starts only after completion of the previous phase.

V-Model



https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

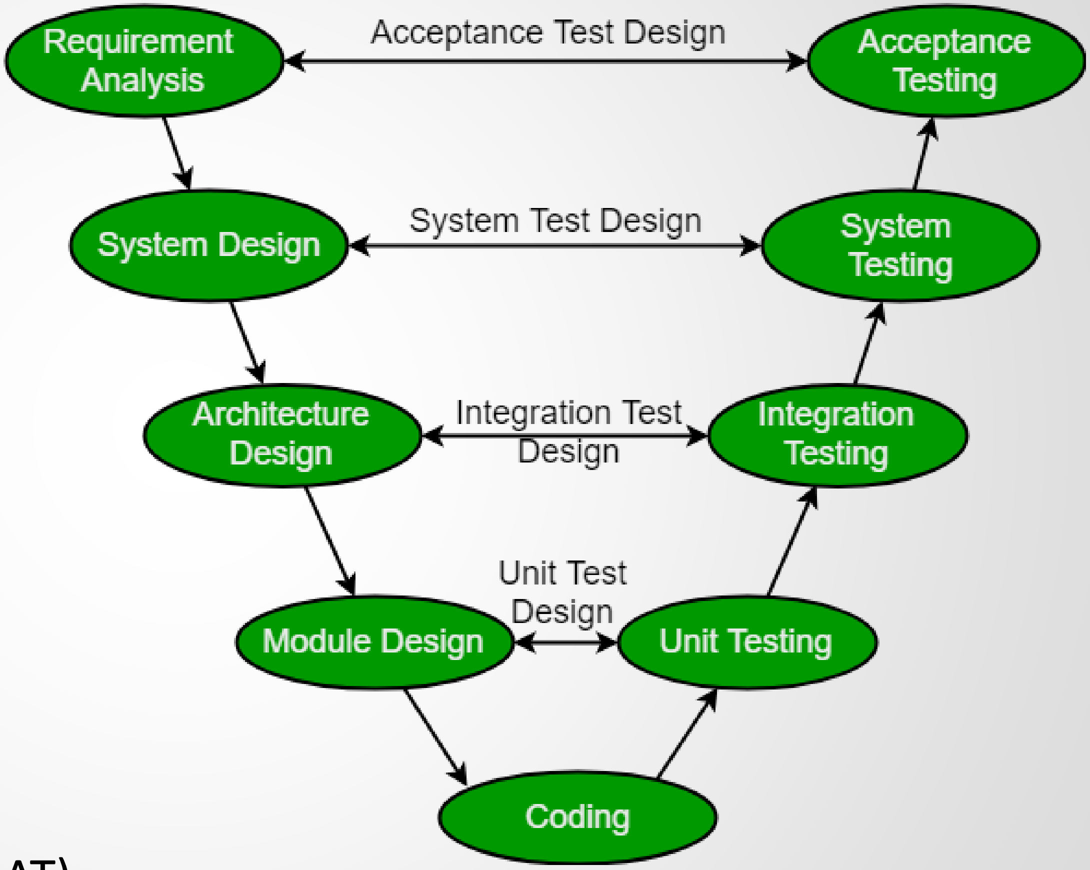
V-Model

Design Phase:

- ❖ Requirement Analysis
- ❖ System Design
- ❖ Architectural Design
- ❖ Module Design

Testing Phases

- ❖ Unit Testing
- ❖ Integration testing
- ❖ System Testing
- ❖ User Acceptance Testing (UAT)



V-Model

Design Phase:

- **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.

- **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.

- **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.

- **Module Design:** In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

V-Model

Testing Phases

- **Unit Testing:** Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.

- **Integration testing:** After **completion of unit tes**ting Integration testing is performed. In integration testing, the modules are integrated and the system is tested. Integration testing is performed on the **Architecture design phase**. **This test verifies the communication of modules among themselves.**

- **System Testing:** System testing test the **complete application** with its functionality, inter dependency, and communication. It tests the functional and non-functional requirements of the developed application.

- **User Acceptance Testing (UAT):** UAT is performed in a **user environment** that resembles the production environment. UAT verifies that the delivered system **meets user's requirement** and system is ready for use in real world.

V-Model

When to use?

- Where requirements are clearly defined and fixed.
- The V-Model is used when ample technical resources are available with technical expertise.
- Proactive defect tracking – that is defects are found at early stage.

The usage of V-Model can fall under the projects which **not focus on changing the requirements**, for example:

- Military projects
- **Mission Critical projects**, for example, in a Space shuttle
- Embedded systems.
- Projects with defined and clear requirements

V-Model

Advantages:

- V-Model is used for small projects where project requirements are clear.
- Simple and easy to understand and use.
- Probability of building **an error-free** and good quality product.

Disadvantages:

- It is not a good for complex and object-oriented projects.
- It is not suitable for projects where requirements are **not clear and contains** high risk of changing.
- It does not easily handle concurrent events.

Big Bang 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.

Advantages:

1. There is no planning required.
2. Simple Model.
3. Few resources required.
4. Easy to manage.
5. Flexible for developers.

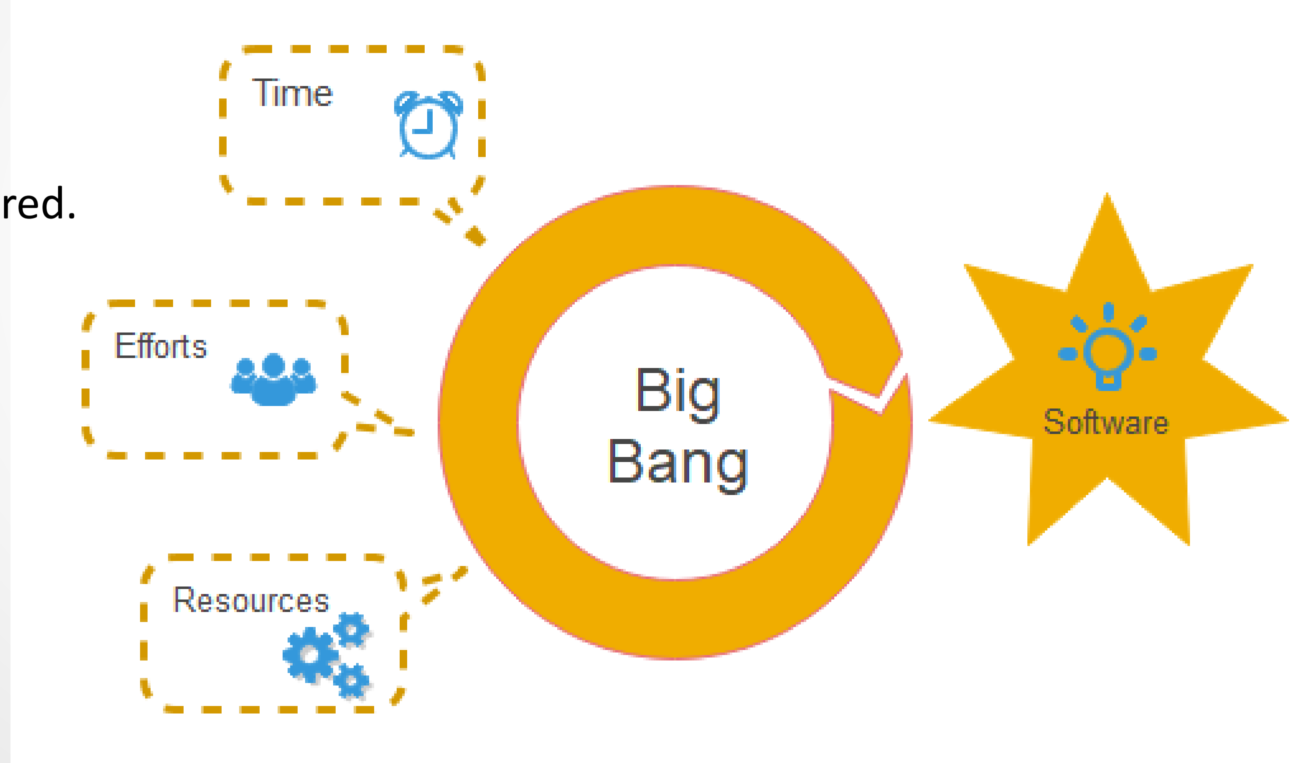
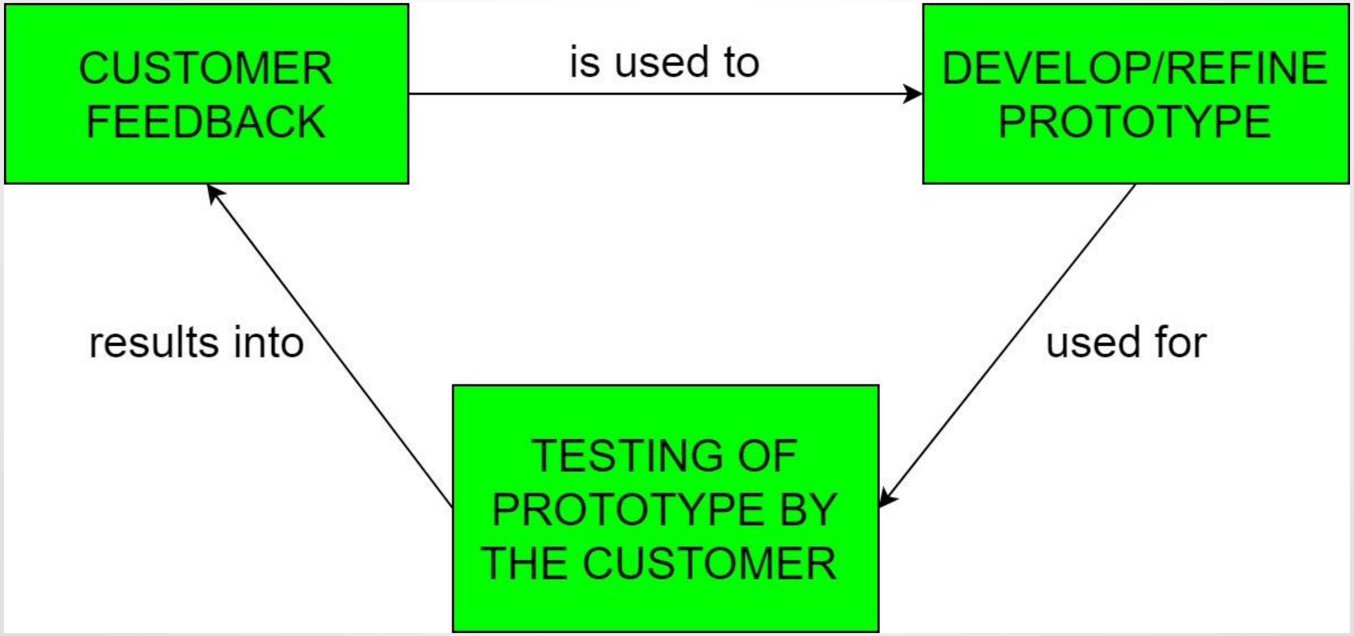


Fig. Big Bang Model

Prototyping Model

- In Software Engineering, Prototype methodology is a software development model in which a prototype is built, test and then reworked when needed until an acceptable prototype is achieved.
- Regular meetings are essential to keep the project on time and avoid costly delays in prototyping approach.
- Missing functionality can be identified
- Prototyping may encourage excessive change requests.



Prototyping Model

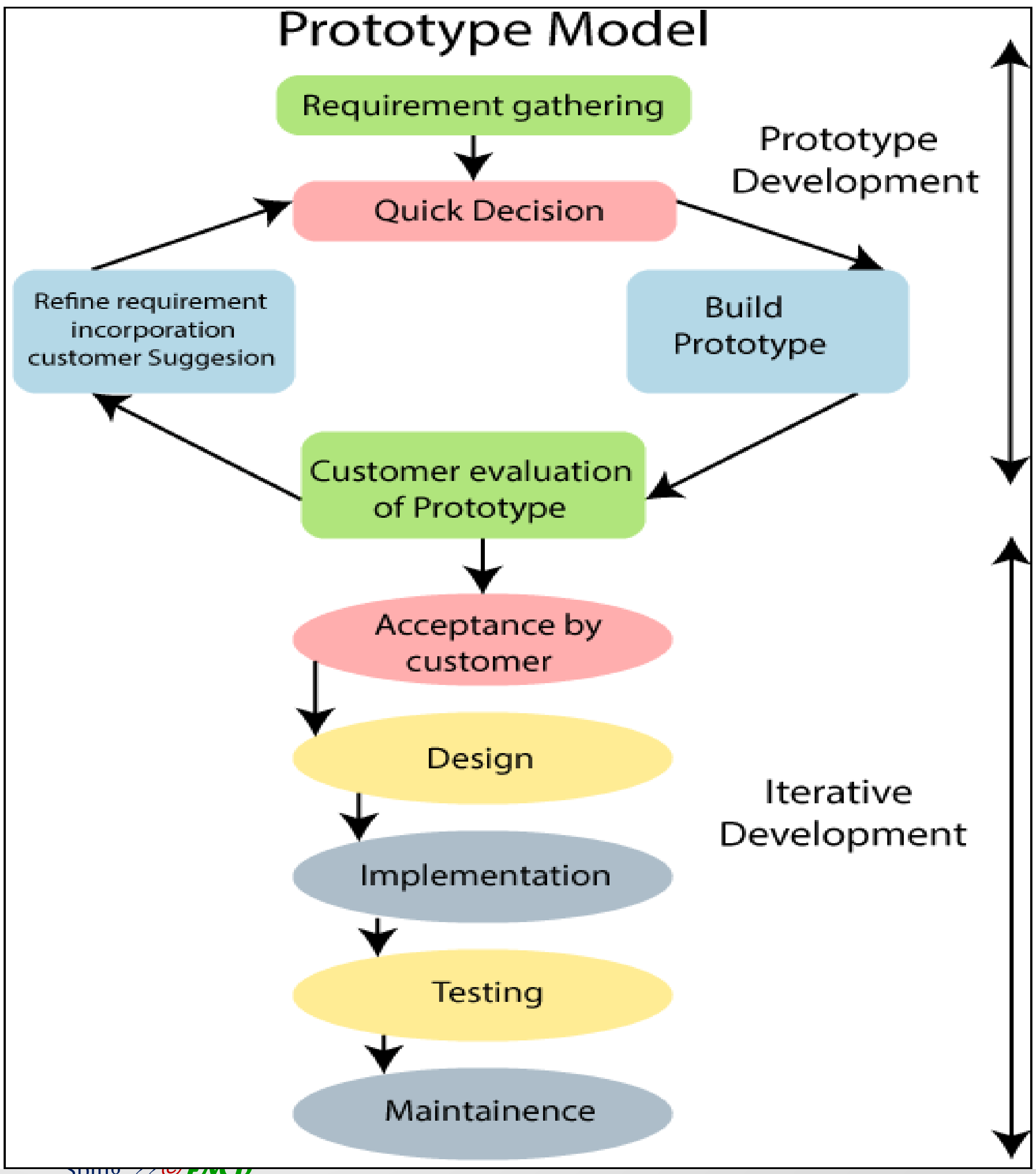
- **Remark:-** In this model client satisfaction is guaranteed

- **Basic requirement identification**

- **Developing the initial prototype**

- **Review of the prototype**

- **Revise and enhance the prototype**



Prototype Model

Requirement gathering

Quick Decision

Build Prototype

Customer evaluation of Prototype

Refine requirement incorporation customer Suggesion

Prototype Development

Acceptance by customer

Design

Implementation

Testing

Maintainence

Iterative Development

30-Jan-23

Prototyping Model

When to use

The Prototyping Model should be used when the requirements of the product are **not clearly understood or are unstable**. It can also be used if requirements are changing quickly. This model can be successfully used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces. It is also a very good choice to demonstrate the technical feasibility of the product.

Prototyping Model

Advantage 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.

Disadvantage 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

What is Spiral Model?

Each phase in spiral model begins with a **design goal (review)** and ends with the **client reviewing** the progress.

It's a **risk-driven model** which means that the overall success of a project highly depends on the risks analysis phase. The spiral model was first mentioned by Barry Boehm in his 1986 paper.

It implements the potential for rapid development of new versions of the software.

The development team in Spiral-SDLC model starts with **a small set** of requirement and goes through each development phase for those set of requirements.

The software engineering team **adds functionality** for the additional requirement in **every-increasing spirals** until the application is ready for the production phase.

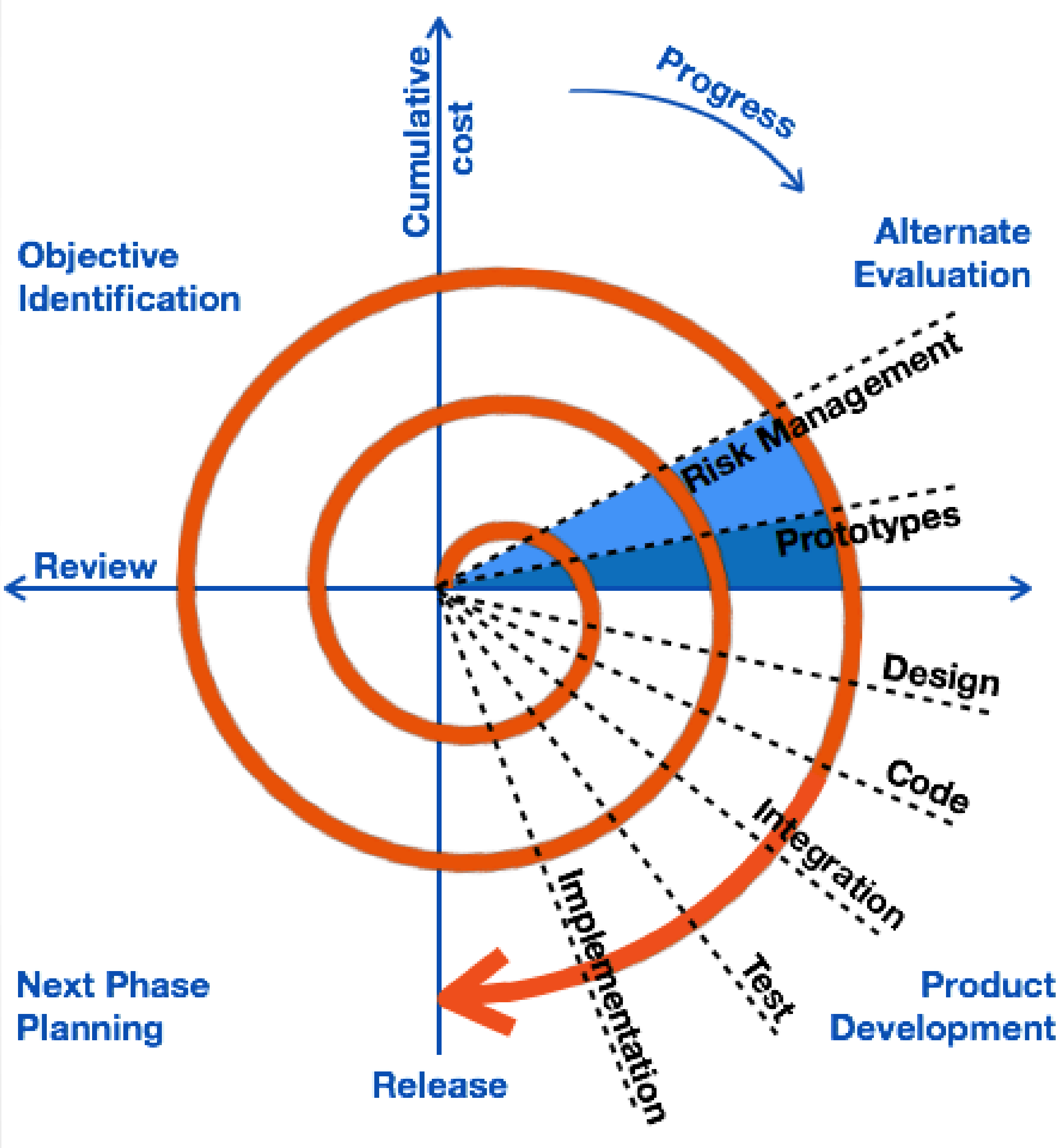
Spiral Model

1. **Determine objectives and find alternate solutions**

2. **Risk Analysis and resolving**

3. **Develop and test**

4. **Review and planning of the next phase**



Spiral Model

1.Determine objectives and find alternate solutions – This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.

2.Risk Analysis and resolving – In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.

3.Develop and test: This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.

4.Review and planning of the next phase – In this phase, the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

Spiral Model Sectors

Why Spiral Model is called Meta Model ?

The Spiral model can be viewed as a Meta-model since it subsumes all the initial models.

For example, a single loop spiral represents the **waterfall model**. The Spiral model uses a **prototyping approach** by first building a prototype before embarking on the actual product development effort. Also, Spiral model can be considered as **supporting the evolutionary model**-the iterations along the spiral model can be considered as the evolutionary levels through which the complete system is built. This enables the developer to understand and resolve the risks at each evolutionary level (i.e. iteration along the spiral). The spiral model uses prototyping as a risk reduction mechanism and also retains the systematic step wise approach of the waterfall model.

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

Advantages

- Risk Handling
- Good for large projects
- Flexibility in Requirements
- Customer Satisfaction

Disadvantages

- Complex
- Expensive
- Too much dependable on Risk Analysis
- Difficulty in time management

https://www.javatpoint.com/software-engineering-spiral-model

Spiral Model

Examples of Spiral Model

The examples of spiral model are that Microsoft used it to develop early versions of Windows. The Gantt chart software was also made using spiral model. Game development is another industry who uses spiral model to develop the games. As the gaming industry highly depend upon the early versions, at such time spiral model is a solid option. With spiral model, the game development industries can get the feedback from their customers at fast and can develop the game as per their convenient.

Quiz 20 marks

Class-Test-01 [compulsory]

Section A, B : 5-Feb-2023 (Class time)

Syllabus: Lecture 3 , 5, 6

Class-Test-02: TBA

Section A ,B

Thanks to All