

Lecture 3-4

Process Models

Outline

- Introduction to process
- Generic Process model
- Waterfall model
- Incremental model
- V model
- Prototyping model
- Spiral model

Software Process

- When you work to build a product or system, it's important to go through a series of predictable steps a road map that helps you create a timely, high-quality result.
- **A software Process is a framework of the activities, actions and tasks that are required to build high quality software.**
- A software process is a collection of activities, actions, and tasks that are performed when some work product is to be created.

Software Process...

1. Activity (Big Process)

Example: Developing a mobile app.

This includes planning, designing, coding, testing, and releasing the app.

2. Action (Major Step in the Process)

Example: Designing the app's architecture.

This involves creating diagrams and models to show how different parts of the app will work together.

3. Task (Small, Specific Job)

Example: Writing code for the login feature.

This includes coding the username and password input fields and testing if the login works correctly.

Umbrella activities



Umbrella activities

Umbrella activities are **high level activities** that encompass and support various stage of software development life cycle.

1. Software project tracking and control

Check the progress of software development. such as **defining project goals, estimating resources and schedules, managing risks.**

2. Risk management

Understand and manage uncertainty. **such as risk identification, risk assessment, risk mitigation planning, and monitoring and controlling** risks throughout the project lifecycle.

3. Software Quality Assurance (SQA)

Defines and conducts the activities required to ensure software quality.

4. Formal Technical Review

Uncover and remove errors before they are circulated to the next activity.

Umbrella activities

5. Measurement

Define and compile process, project, and product metrics to help the team deliver software that meets the needs of stakeholders

6. Software Configuration Management (SCM)

Software Configuration Management (SCM) is the practice of **tracking, controlling, and organizing all the pieces** of a software project so that when changes happen, nothing breaks or gets lost.)

7. Reusability management

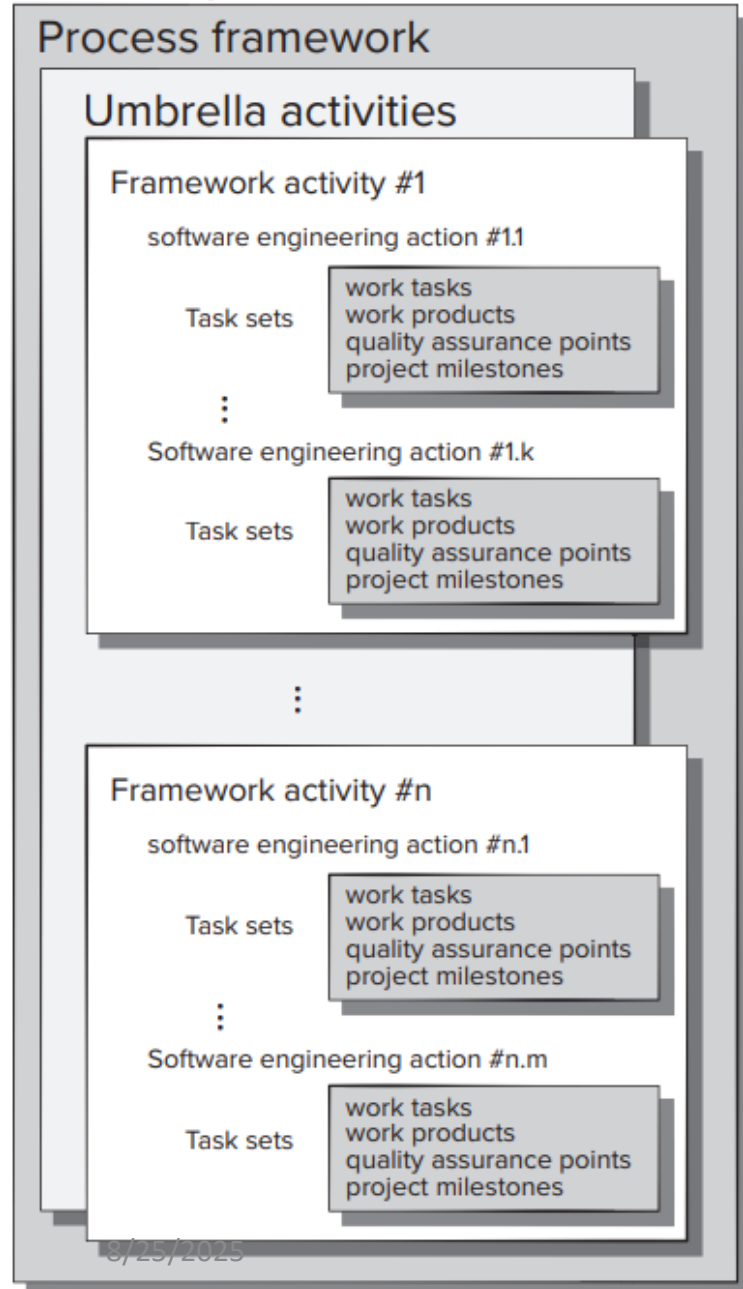
Develop mechanisms to implement reusable components.

Instead of writing new code from scratch every time, we **reuse existing code, modules, or designs.**

8. Work product preparation and production

Create work products such as models, documents, logs, forms, and lists.

Software Process...



Each **framework** activity is populated by a **set of software engineering Actions.**

Each software engineering action is defined by a **task set** that identifies the:

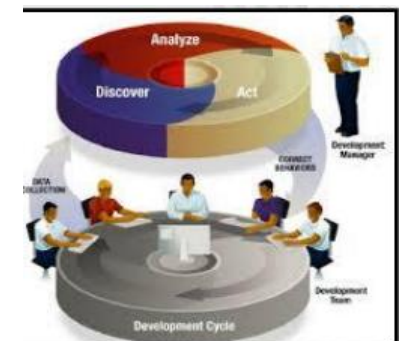
- Work tasks that are to be completed,
- Work products that will be produced,
- Quality assurance points that will be required,
- Project milestones that will be used to indicate progress.

- A Generic Model has five basic activities

- Communication
- Planning
- Modeling
- Construction
- Deployment

A Generic Model has five basic activities

- **Communication:**
 - Before any technical work can commence, it is critically important to communicate and collaborate with the customer or client.(Stakeholders)
 - The objective is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.
 - Stakeholder : A stakeholder is anyone who has a stake in the successful outcome of the project business managers, end users, software engineers, support people, etc
 - Includes requirements gathering and related activities



A Generic Model has five basic activities

- **Planning:**
 - It defines the software engineering work by
 - Describing the technical tasks to be conducted,
 - The risks that are likely,
 - The resources that will be required,
 - The work products to be produced, and a work schedule.
 - **It is similar like a map which guide us about our journey.**



A Generic Model has five basic activities

- **Modeling**

- It creates a “sketch” of the thing so that you’ll understand the real picture.
- A software engineer does the same thing by creating **models** to better understand software requirements and the **design** that will achieve those requirements.
- Help developer and customer to understand requirements (Analysis of requirements) & Design of software.



A Generic Model has five basic activities

- **Construction**
 - **Code generation**
 - **Testing – to uncover error in the code.**



A Generic Model has five basic activities

- **Deployment:**

- The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.
- Delivery to the customer for evaluation
- Customer provide feedback
- Software engineering process framework activities are complemented by a number of umbrella activities.
- In general, umbrella activities are applied throughout a software project and help a software team manage and control progress, quality, change, and risk.



A Generic Model has five basic activities

These five generic framework activities can be used during the development of small, simple programs, the creation of large Web applications, and for the engineering of large, complex computer-based systems. The details of the software process will be quite different in each case, but the framework activities remain the same .

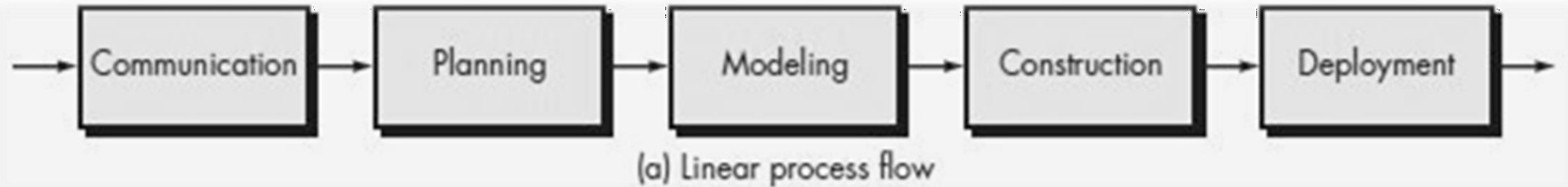
Process Flow

- Each Software process has a process flow.
- Describes how the framework activities , the actions and tasks within each framework are organized with respect to sequence and time.
- How can we execute these five numbers activities which is called process flow.
- There are four ways to execute all the five activities.
 - *linear process flow*
 - *iterative process flow*
 - *evolutionary process flow*
 - *parallel process flow*

Process Flow Cont..

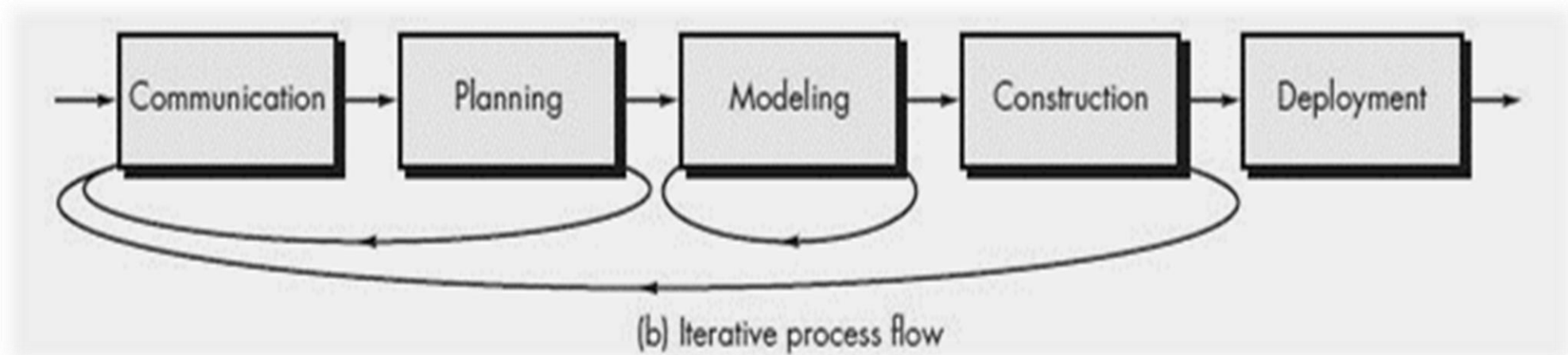
linear process flow: executes each of the five framework activities in sequence, beginning with communication and ending with deployment.

We are not changing the order.



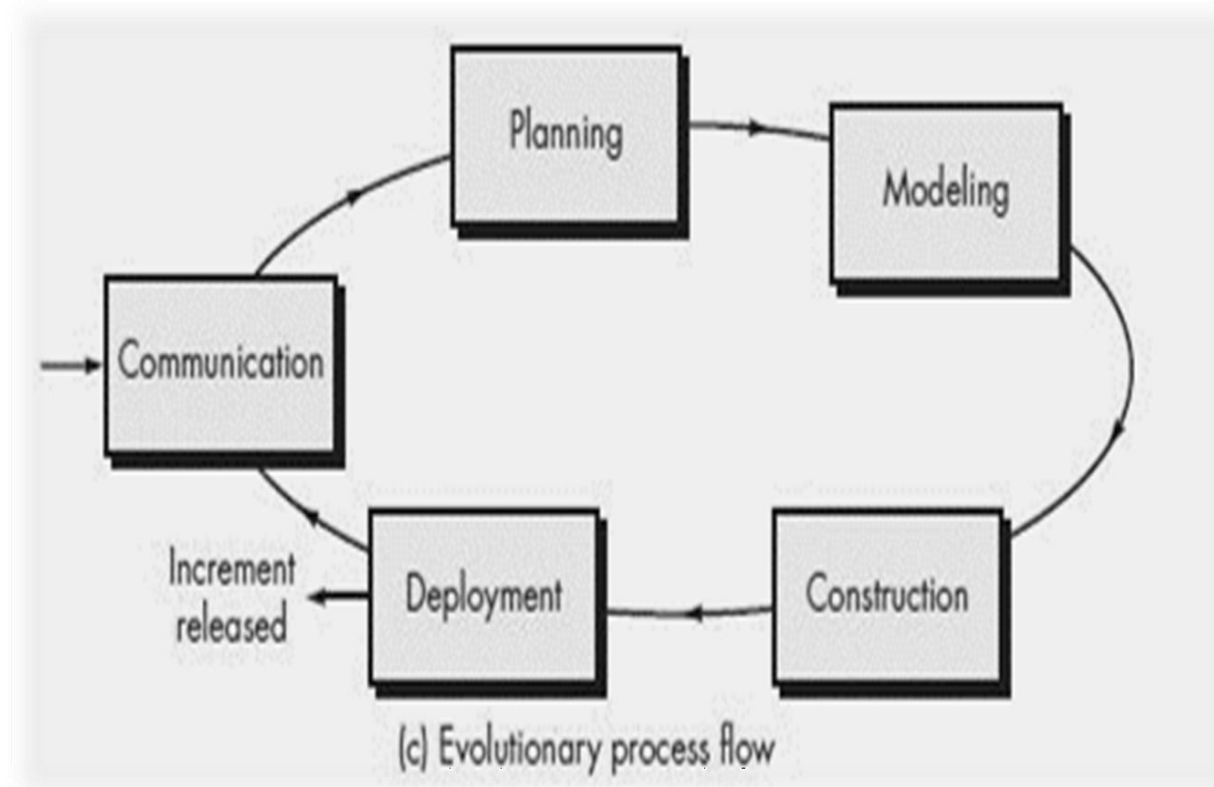
Process Flow Cont..

Iterative process flow: repeats one or more of the activities before proceeding to the next.



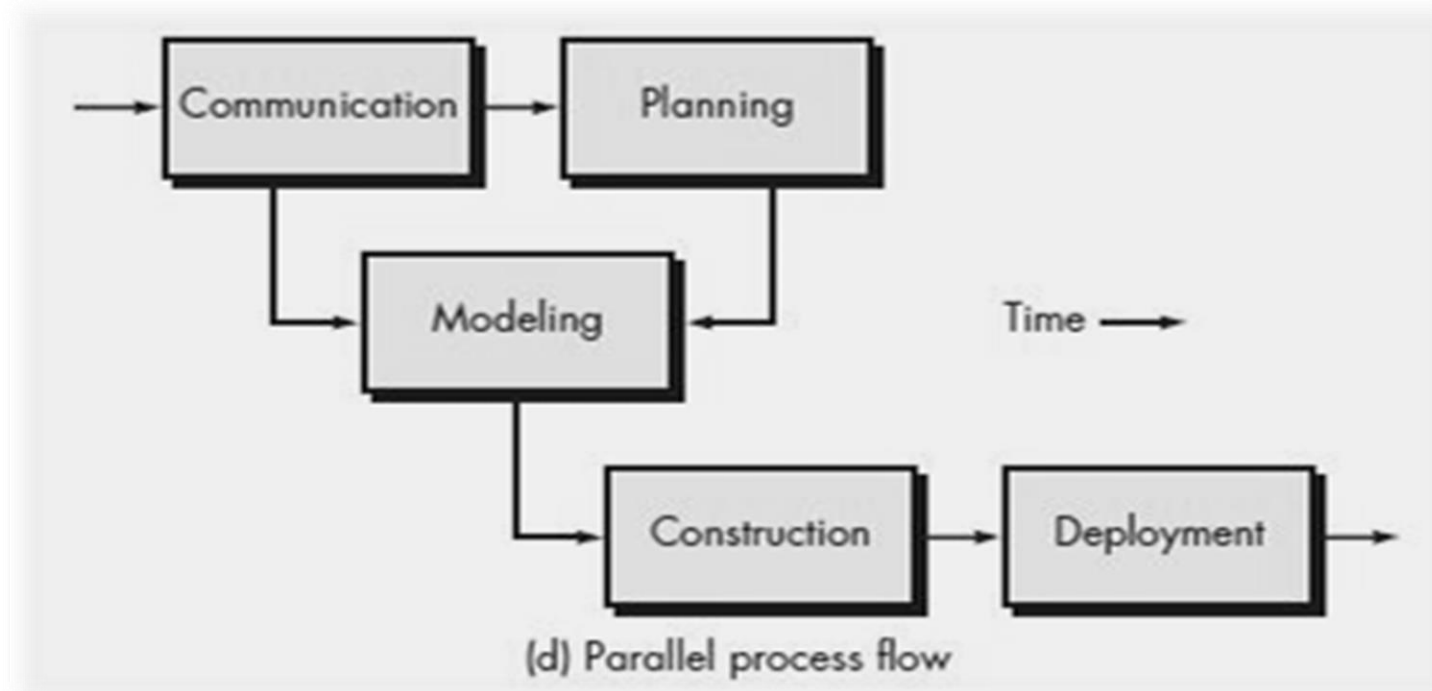
Process Flow Cont..

Evolutionary process flow: executes the activities in a “circular” manner. Each circuit through the five activities leads to a more complete version of the software.



Process Flow Cont..

Parallel process flow: executes one or more activities in parallel with other activities (e.g., modeling for one aspect of the software might be executed in parallel with construction of another aspect of the software).

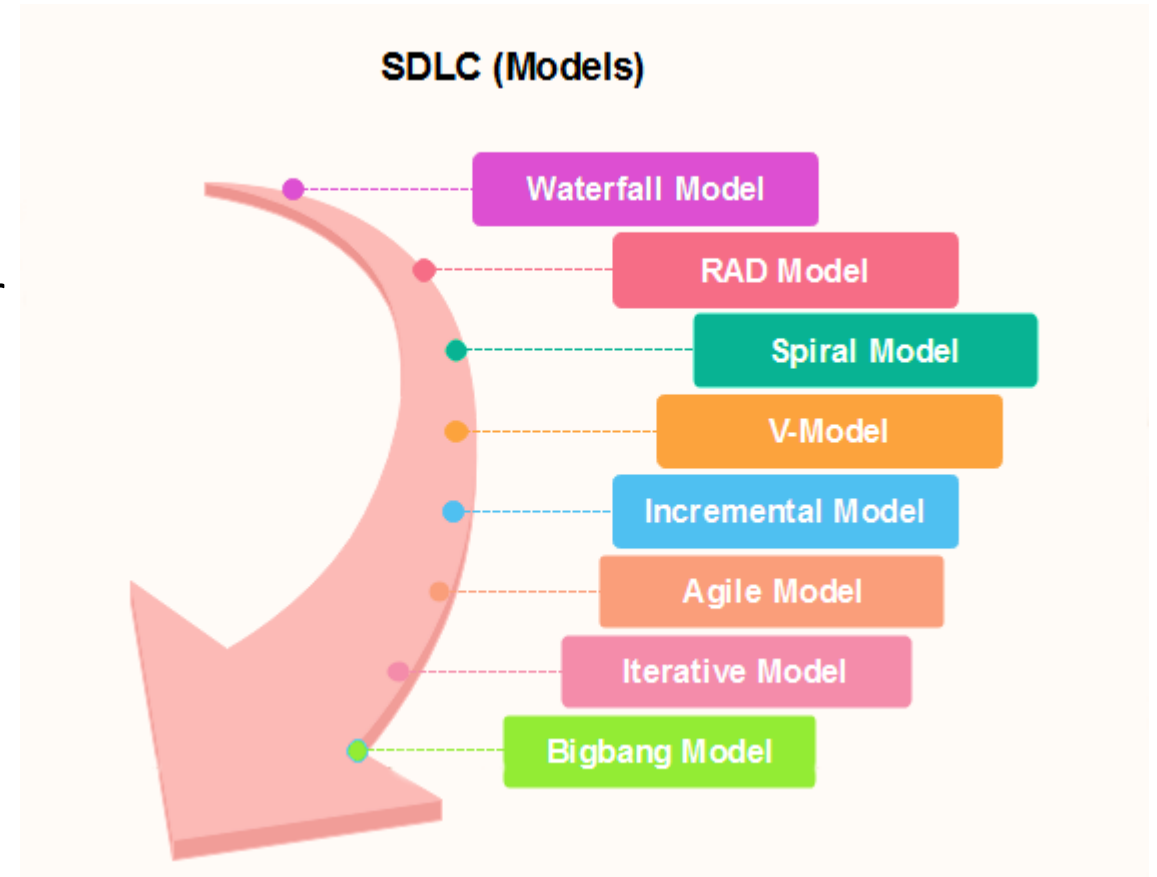


Software Development Life Cycle (SDLC)

- SDLC stands for Software Development Life Cycle. SDLC is a process that consists of a series of planned activities to develop or alter the Software Products.
- A “life-cycle” or a software process is the **organizational framework** for a project.

Software Development Life Cycle (SDLC)

- There are various software development life cycle models defined and designed which are followed during the software development process.
- Following are the most important and popular SDLC models followed in the industry;
 - Build and Fix model
 - Waterfall Model
 - Incremental Model
 - Spiral Model
 - V-Model
 - Prototyping Model



Software Process Models

- Process Model

- A Process Model is a strategy to combine the **PROCESS, METHODS, AND TOOLS** together to effectively manage and deliver a software product.

- Selection of a Process Model

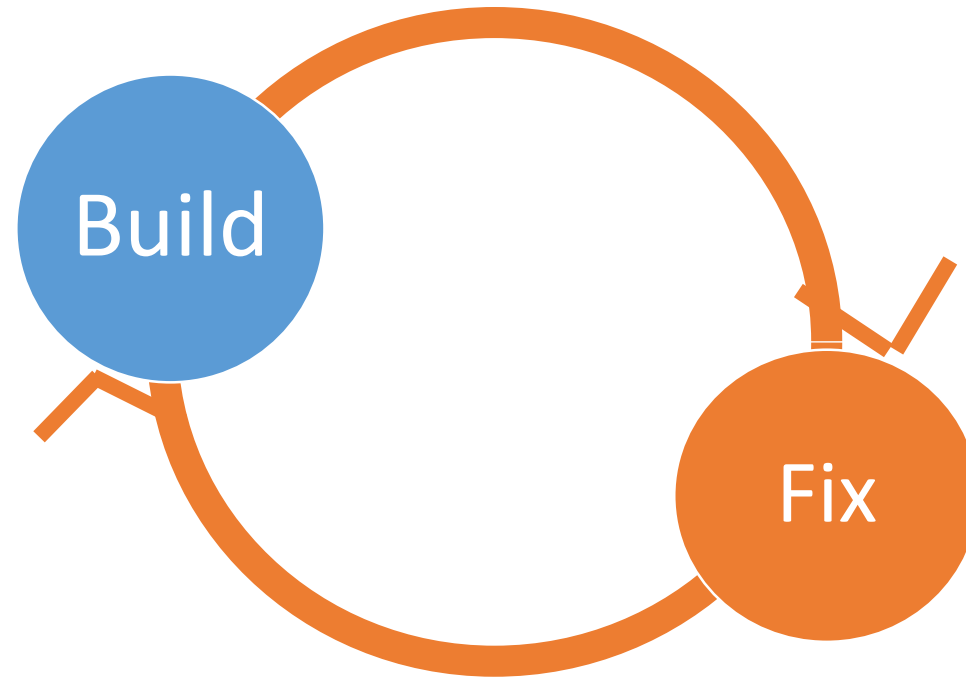
- Following is the selection criteria for a process model
 - The **NATURE** of the project and application
 - The **METHODS** and **tools** to be used
 - The **DELIVERABLES** to be produced

Build & Fix Model

- Build and Fix Process Model

- Product is **constructed** without any **specifications** or any attempt to **design**.
 - The developer simply **builds the product** that is reworked as many times as necessary to satisfy the client.
 - This is an **ad-hoc approach** and not well defined.
- Build and Fix Model is a simple two phase process model:
 - The first phase is to write the **code**
 - The next phase is to **fix** it.
 - Fixing may be error correction or addition of further functionality.

Build & Fix Model

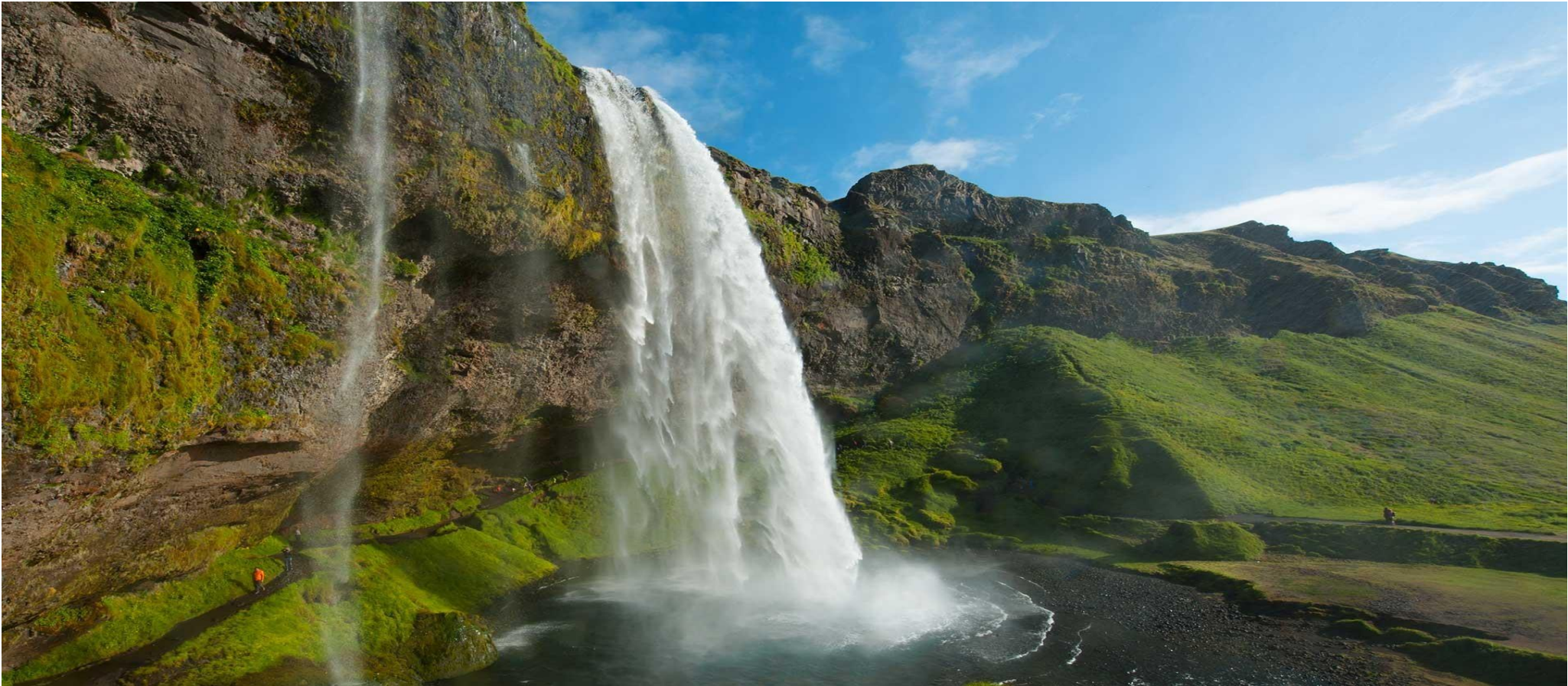


Build and Fixed Model Strength and weaknesses

- **Strength and Weaknesses**

- Following are some of the strength and problems in build and fix process model
 - Works well on problems having about 100 to 200 lines of code
 - This model doesn't work well on problems with reasonable size
 - This model is costly
 - Maintenance is always difficult
 - Chances of errors are high

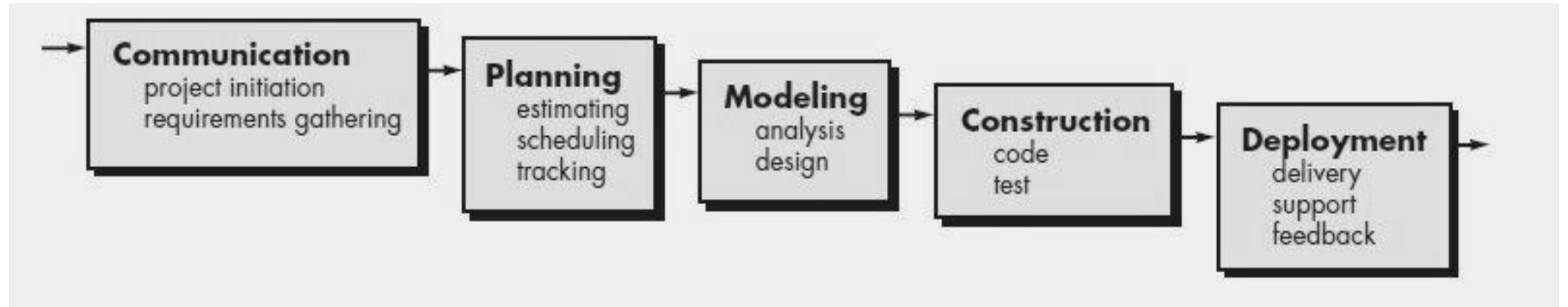
The Waterfall Life-Cycle



The Waterfall Life-Cycle

- There are times when the requirements for a problem are well understood When work flows from **communication** through **deployment** in a reasonably linear fashion.
- The waterfall Model shows the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete.
- Invented in 1970. The waterfall model, sometimes called the classic life cycle

Waterfall Model or Classic Life Cycle



When to use the waterfall model

- This model is used only when the requirements are very well known, clear and fixed.
- Technology is understood.
- There are no ambiguous requirements
- Very less customer interaction is involved during the development of the product. Once the product is ready then only it can be give to the end users.
- Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the logic.

Advantages of waterfall model

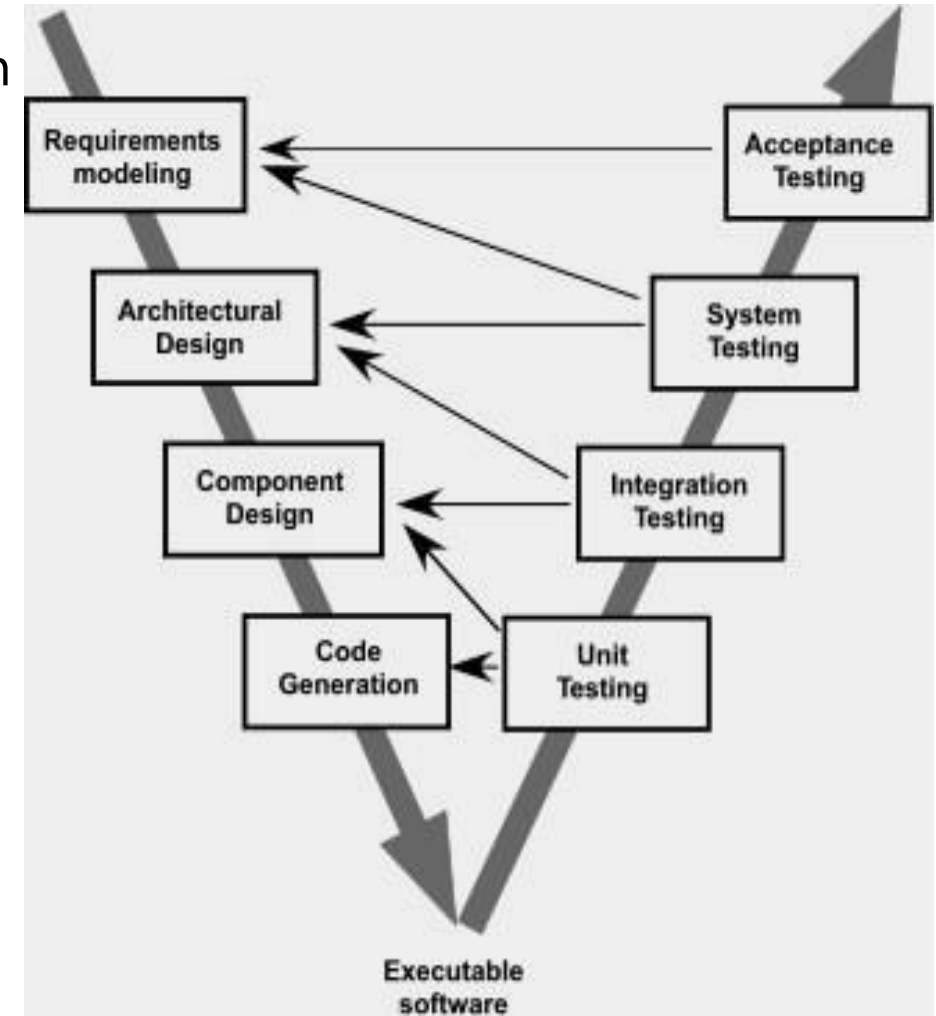
- This model is simple and easy to understand and use.
- In this model phases are processed and completed one at a time.
- Waterfall model works well for smaller projects where requirements are very well understood.
- Easy to arrange tasks.
- Process and results are well documented.

Disadvantages of waterfall model

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- Poor model for long and ongoing projects.
- Not a good model for complex and object-oriented projects
- Not suitable for the projects where requirements are at a moderate to high risk of changing

The V-Model

- execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation model**.
- It is used to produce rigorous development lifecycle models and project management models.
- It represents the relationships b/w the quality assurance actions (tests)

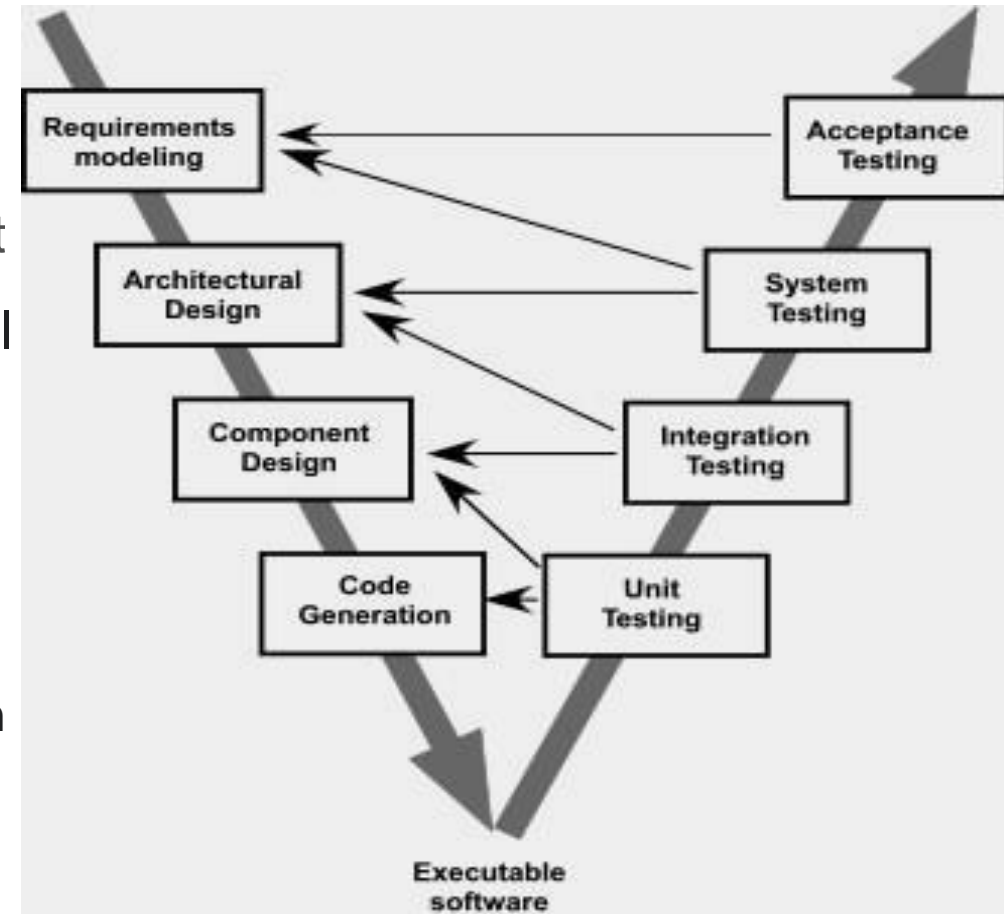


The V-Model

Validation vs. verification

Verification. verification is the process of checking that the software meets specifications.. It is often an internal process.

Validation: is the process of checking whether the specification captures the customer's requirements. It is often an external process.



The advantages of the V-Model

- This is a highly-disciplined model
- Works well for smaller projects where requirements are very well understood
- Simple and easy to understand and use

The disadvantages of the V-Model

- High risk and uncertainty
- Not a good model for complex and object-oriented projects

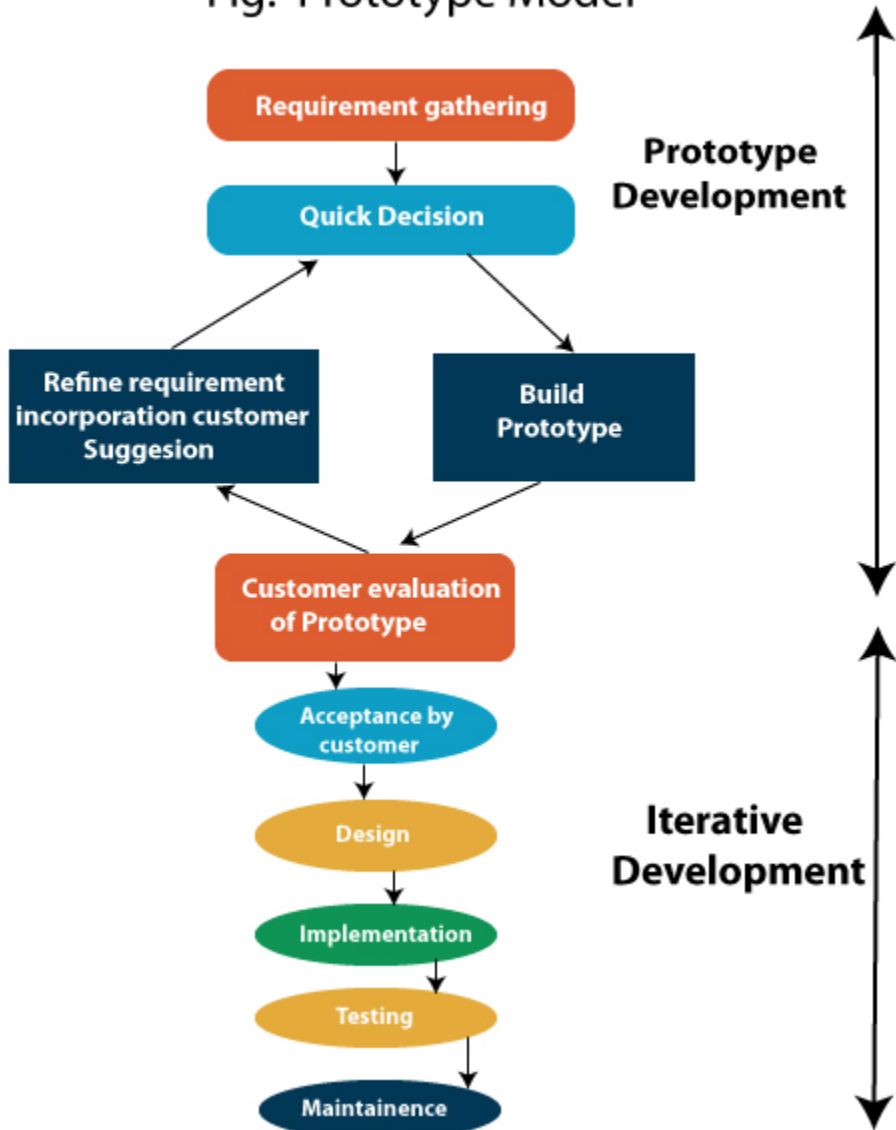
Prototyping



Prototyping

- The Software Prototyping refers to building software application prototypes which displays the functionality of the product under development, but may not actually hold the exact logic of the original software.

Fig: Prototype Model



What is Software Prototyping?

- Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.

When to use Prototype model

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.
- Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system.
- They are excellent for designing good human computer interface systems.

Advantages of Prototype model:

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily

Disadvantages of Prototype model:

- Leads to implementing and then repairing way of building systems.
- This model is costly.
- There may be too much variation in requirements.
- Incomplete or inadequate problem analysis.

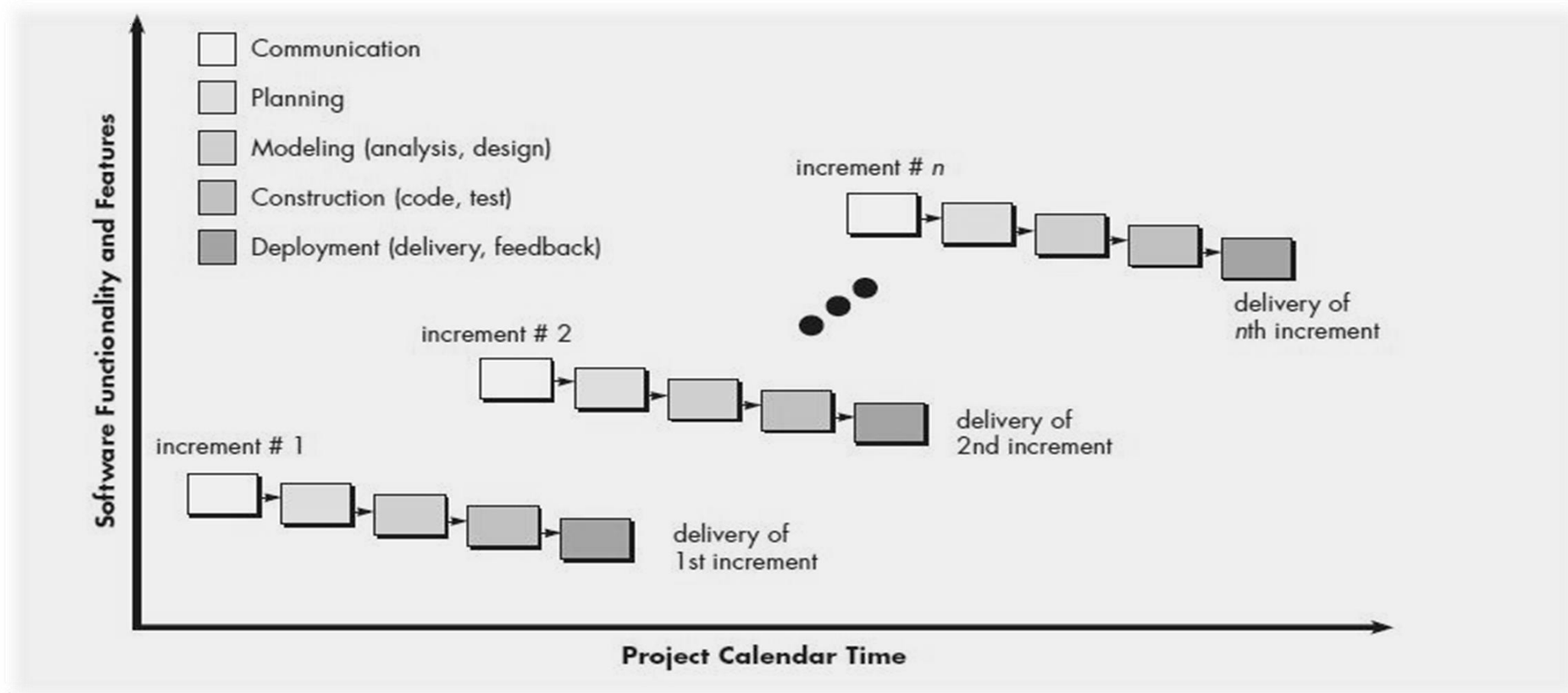
The Incremental Model

- There are many situations in which initial software requirements are reasonably well defined, but the overall scope of the development preclude a purely linear process.
- In addition, there may be a compelling need to provide a limited set of software functionality to users quickly and then refine and expand on that functionality in later software releases.
- In such cases, you can choose a process model that is designed to produce the software in increments.
- **Example:** Developing one big website in a very short time, but the developer should develop one home page with little functions and he/she can give a message which is under construction. Here the customer can be satisfy that my website is available.

The Incremental Model

- In incremental model the whole requirement is divided into various builds.
- Cycles are divided up into smaller, more easily managed modules.
- In this model, each module passes through the requirements, design, implementation and testing phases.
- Each subsequent release of the module adds function to the previous release. The process continues till the complete system is.
- It is a model by which we can develop a software in short time due to competition in the market.

The Incremental Model Cont..



When to use the Incremental model

- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- A new technology is being used
- Resources with needed skill set are not available
- There are some high risk features and goals.

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.

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](#).

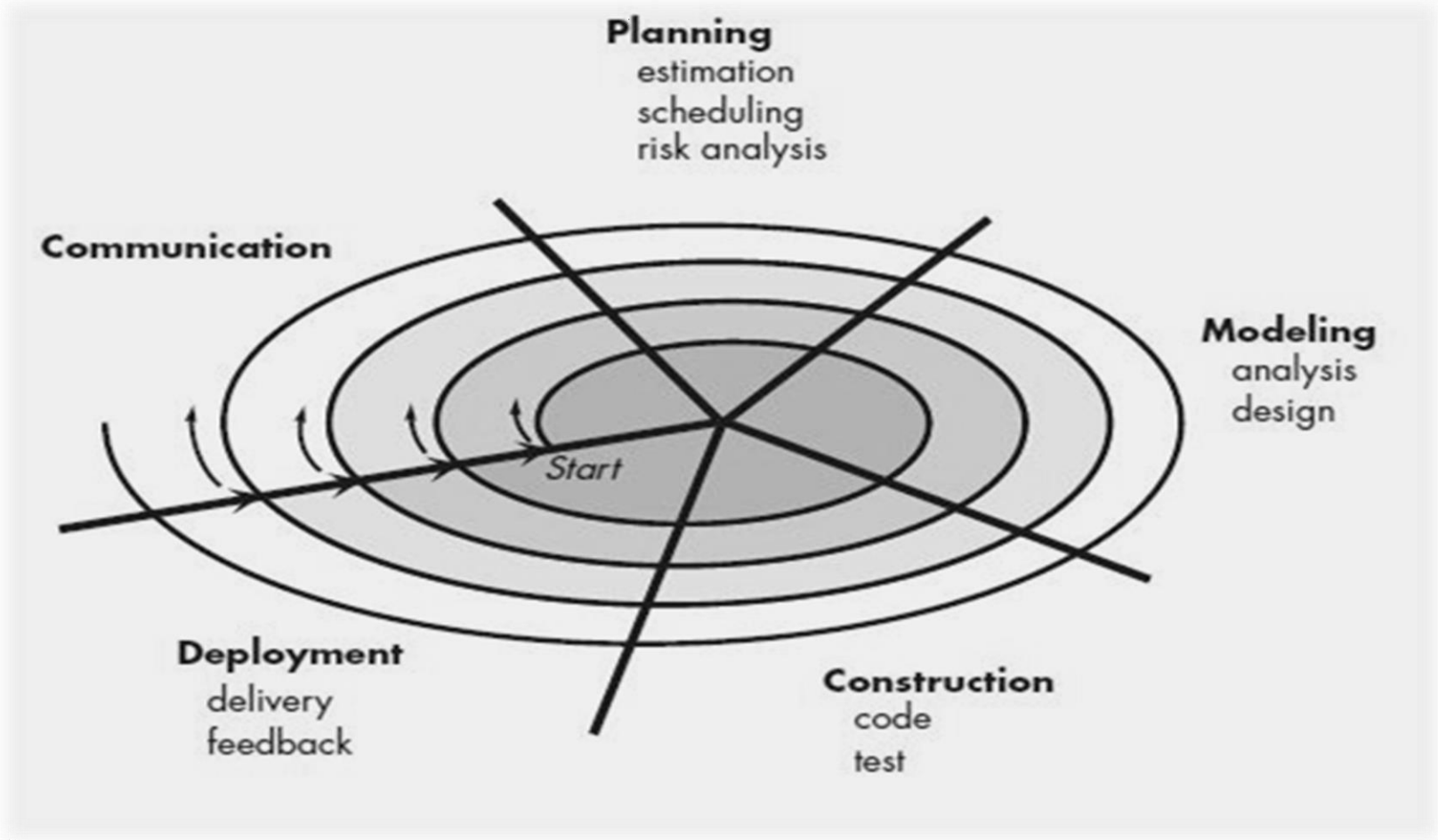
Example of a Successful Incremental Model

- The incremental model is used by leading software companies like WhatsApp, Windows, and MS Office.

The Spiral Model

- Spiral model is a combination of sequential and prototype model. This model is best used for large projects which involves continuous enhancements.
- There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software.
- The same activities are then repeated for all the spirals till the entire software is build.

The Spiral Model



When to use Spiral model

- When the project is large.
- Where the software needs continuous risk evaluation.
- Requirements are a bit complicated and require continuous clarification.
- Software requires significant changes.
- Users are unsure of their needs
- Requirements are complex

Advantages of Spiral model

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Development is fast.
- Larger projects / software are created and handled in a strategic way
- More and more features are added in a systematic way.

Disadvantages of Spiral model

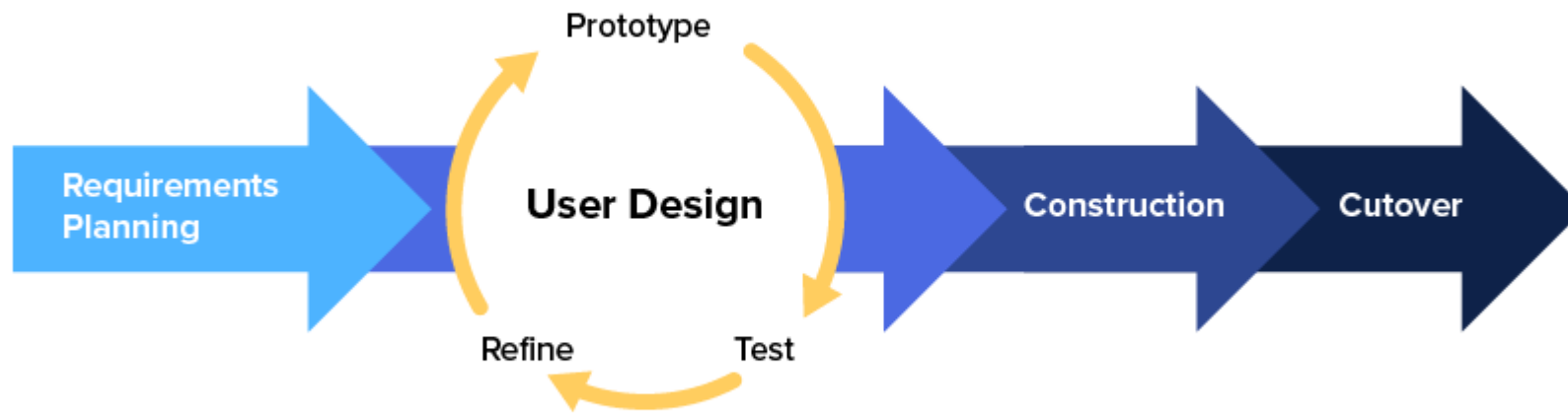
- Can be a costly model to use.
- Project's success is highly dependent on the risk analysis phase.
- Risk analysis is important phase so requires expert people.
- Is not beneficial for smaller projects.
- Budget and time can not be define in start

RAD model

- RAD model is Rapid Application Development model.
- This model was proposed by James Martin in 1991
- It is a type of incremental model.
- In RAD model the components or functions are developed in parallel as if they were mini projects.
- The developments are time boxed, delivered.
- This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

RAD model

Rapid Application Development (RAD)



Advantages of RAD model

- The process of application development and delivery are fast.
- This model is flexible, if any changes are required.
- Reviews are taken from the clients at the starting of the development hence there are lesser chances to miss the requirements.

Disadvantages of RAD model

- The feedback from the user is required at every development phase.
- This model is not a good choice for long term and large projects.
- Suitable for project requiring shorter development times.
- Requires highly skilled developers/designers.

When to use RAD model

- RAD should be used only when a system can be modularized to be delivered in an incremental manner.
- It should be used if there is a high availability of designers for modeling.
- RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.

Summary

- Introduction to process
- Generic Process model
- Waterfall model
- Incremental model
- Prototyping model
- Spiral model
- RAD Model

Assignment

1. Difference between all the process models

ANY QUESTIONS ?