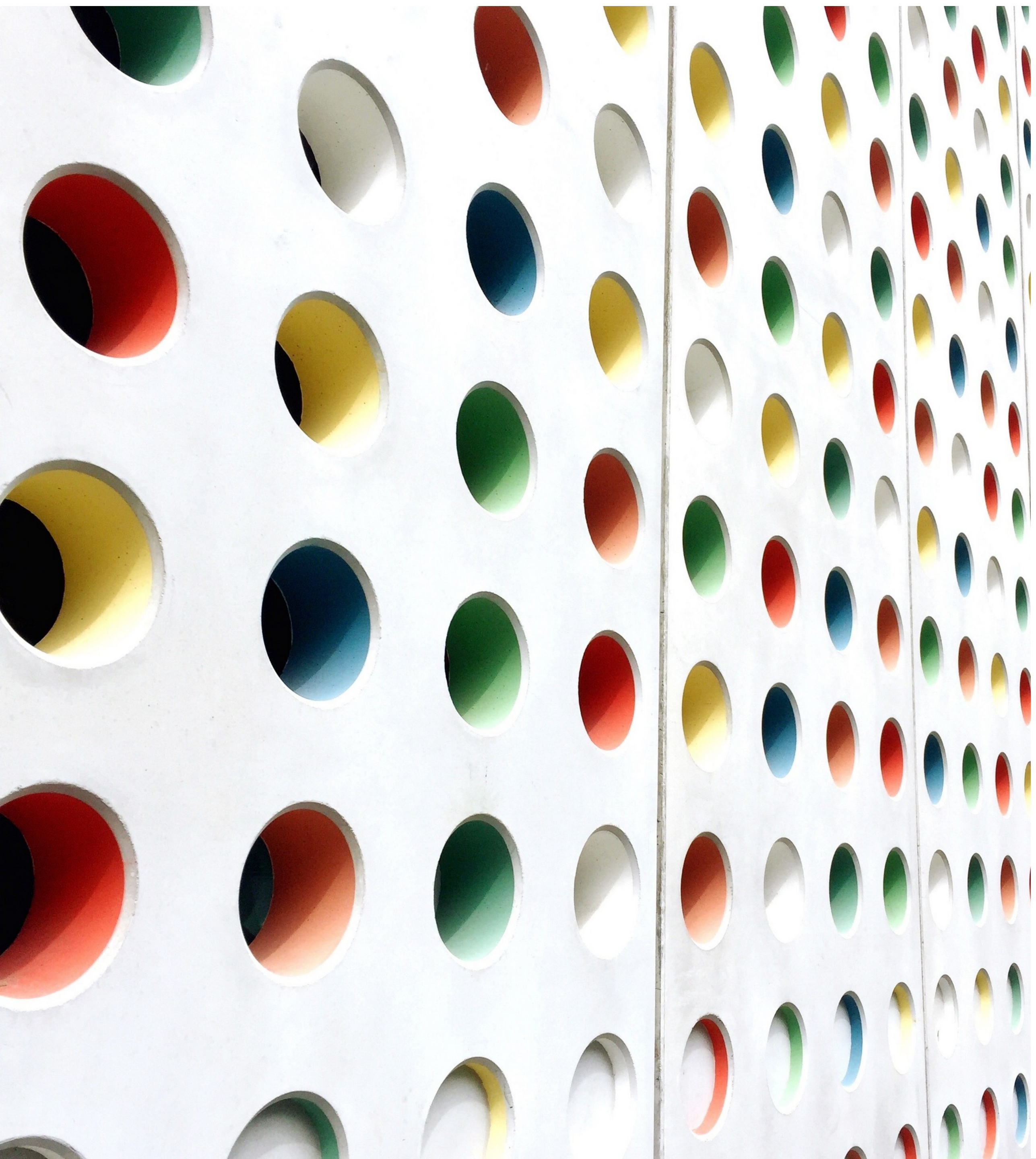


Prescriptive Process Models

Personal and Team Process Models (PSP & TSP)

PSP

- The **Personal Software Process** (**PSP**) is a structured software development process that is designed to help software engineers better understand and improve their performance by bringing discipline to the way they develop software and tracking their predicted and actual development of the code.
- PSP has two review phases : Design review , Code review
- It works on 2 stages : PSP0 – and progresses in process maturity to the final level – PSP2.1.

TSP

- The **Team software process** (**TSP**) provides a defined operational process framework that is designed to help teams of managers and engineers organize projects and produce software for products that range in size from small projects of several thousand lines of code to very large projects greater than half a million lines of code.
- TSP has two principal components: team-building and team-working
-

Prescriptive Process Models

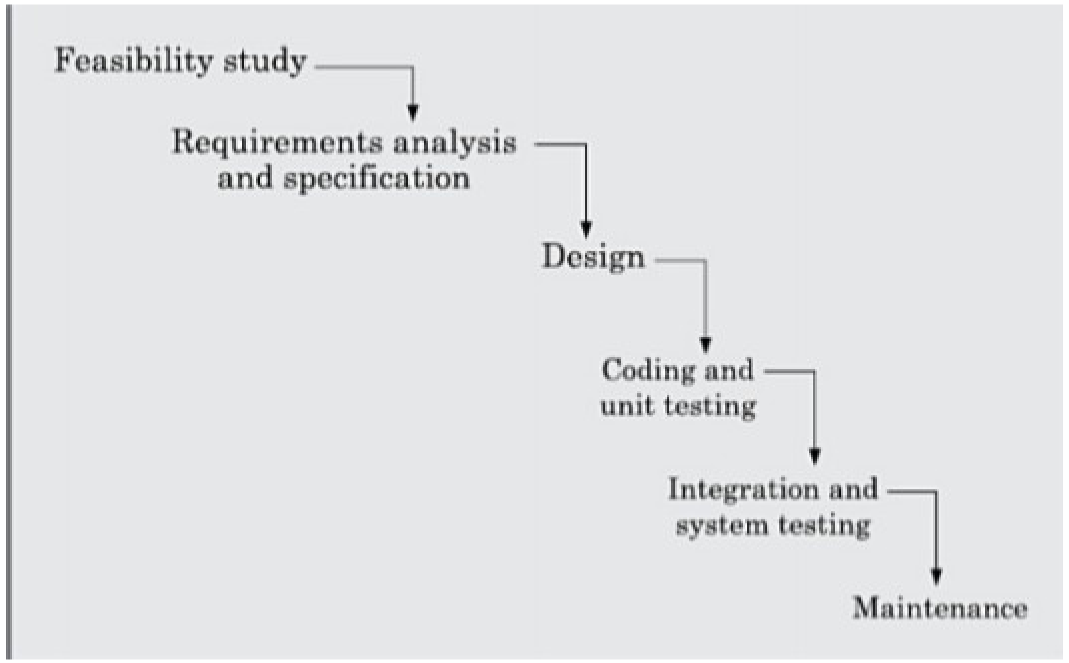
- **The Waterfall Model :**

i. It is simple but idealistic.

ii. In fact, it is hard to put this model into use in any non-trivial software development project. One might wonder if this model is hard to use in practical development projects, then why study it at all? The reason is that all other life cycle models can be thought of as being extensions of the classical waterfall model.

iii. Therefore, to first understand the classical waterfall model, in order to be able to develop a proper understanding of other life cycle models. Besides, we shall see later in this text that this model though not used for software development; is implicitly used while documenting software.

- **Phases of Waterfall Model –**


Feasibility study
Requirements analysis and specification
Design
Coding and unit testing
Integration and system testing
Maintenance

i. Feasibility Study :

The main focus of the feasibility study stage is to determine whether it would be financially and technically feasible to develop the software.

feasibility study involves carrying out several activities such as collection of basic information relating to the software such as the different data items that would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system, as well as various constraints on the development.

ii. Requirement Analysis and Specification :

The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly.

This phase consists of two distinct activities, requirements gathering and analysis, and requirements specification.

Iii. Design :

The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language.

Two approaches are present –procedural design approach and Object oriented approach.

- Procedural design approach -This traditional design technique is based on the data flow-oriented design approach. It includes – structured analysis and structured design.
- Object oriented approach – It has lower development time and effort, and better maintainability of the software.

iv . Coding and Unit testing :

The purpose of the coding and unit testing phase is to translate a software design into source code and to ensure that individually each function is working correctly. The coding phase is also sometimes called the implementation phase, The main objective of unit testing is to determine the correct working of the individual modules. The specific activities carried out during unit testing include designing test cases, testing, debugging to fix problems, and management of test cases.

v . Integration and System testing –

Integration of different modules is after they have been coded and unit tested. During the integration and system testing phase, the different modules are integrated in a planned manner. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained. System testing is carried out on this fully working system.

System testing usually consists of three different kinds of testing activities: -

Testing, testing , Acceptance testing:

vi . Maintenance :

The total effort spent on maintenance of a typical software during its operation phase is much more than that required for developing the software itself.

Maintenance is required in the following three types of situations:

- Corrective maintenance:
- Perfective maintenance:
- Adaptive maintenance

- **Weakness :**

- Requirements must be known prior
- Inhibits flexibility.
- No problem solving nature
- Little opportunity for customer to preview the system
- Integration at the end is a major bang
- Can give false impression of progress.

- Example of waterfall model :

Automobile industry , Car or bike production – Requirements are drafted. Once they are finalized then design phase starts. Now requirements do not change unless the car or bike is completely ready .

➢ **Strengths :**

- Easy to understand
- Provides basic structure
- Sets requirement stability
- Good for management control
- Works well when quality is more important than cost or schedule

- **V Model**

i. V-model is a variant of the waterfall model. This model gets its name from its visual appearance .

ii. In this model verification and validation activities are carried out throughout the development life cycle, and therefore the chances bugs in the work products considerably reduce.

iii. This model is therefore generally considered to be suitable for use in projects concerned with development of safety-critical software that are required to have high reliability.

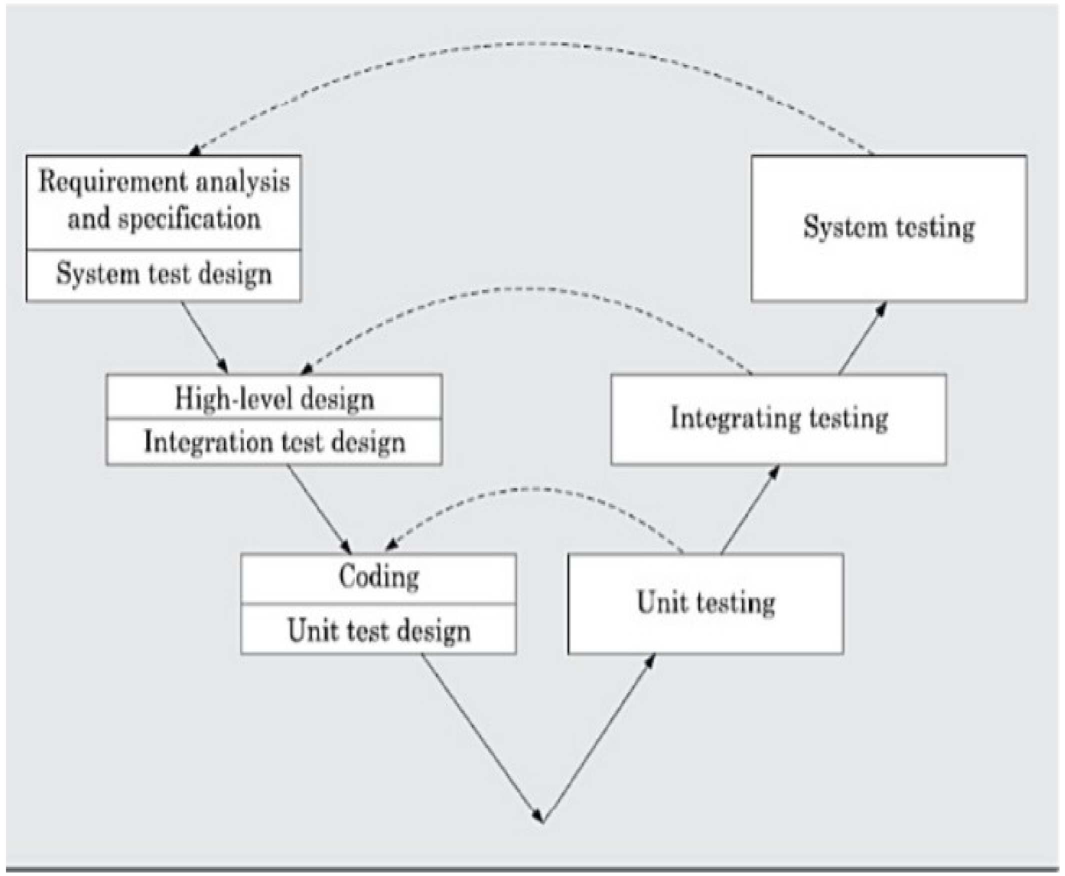
- There are two main phases—development and validation phases. The left half of the model comprises the development phases and the right half comprises the validation phases.
- Development phase :

Along with the development of a work product, test case design and the plan for testing the work product are carried out, whereas the actual testing is carried out in the validation phase.

- Validation phase :

Testing is carried out in three steps—unit, integration, and system testing. The purpose of these three different steps of testing during the validation phase is to detect defects that arise in the corresponding phases of software development requirements analysis and specification, design, and coding respectively.

➢ **Phases of V model :**



➢ **Strengths :**

- this model usually leads to a shorter testing phase and an overall faster product development as compared to the iterative model.
- quality of the test cases are usually better.
- the test team is associated with the project from the beginning. Therefore they build up a good understanding of the development artifacts, and this in turn, helps them to carry out effective testing of the software.

- **Weakness :**

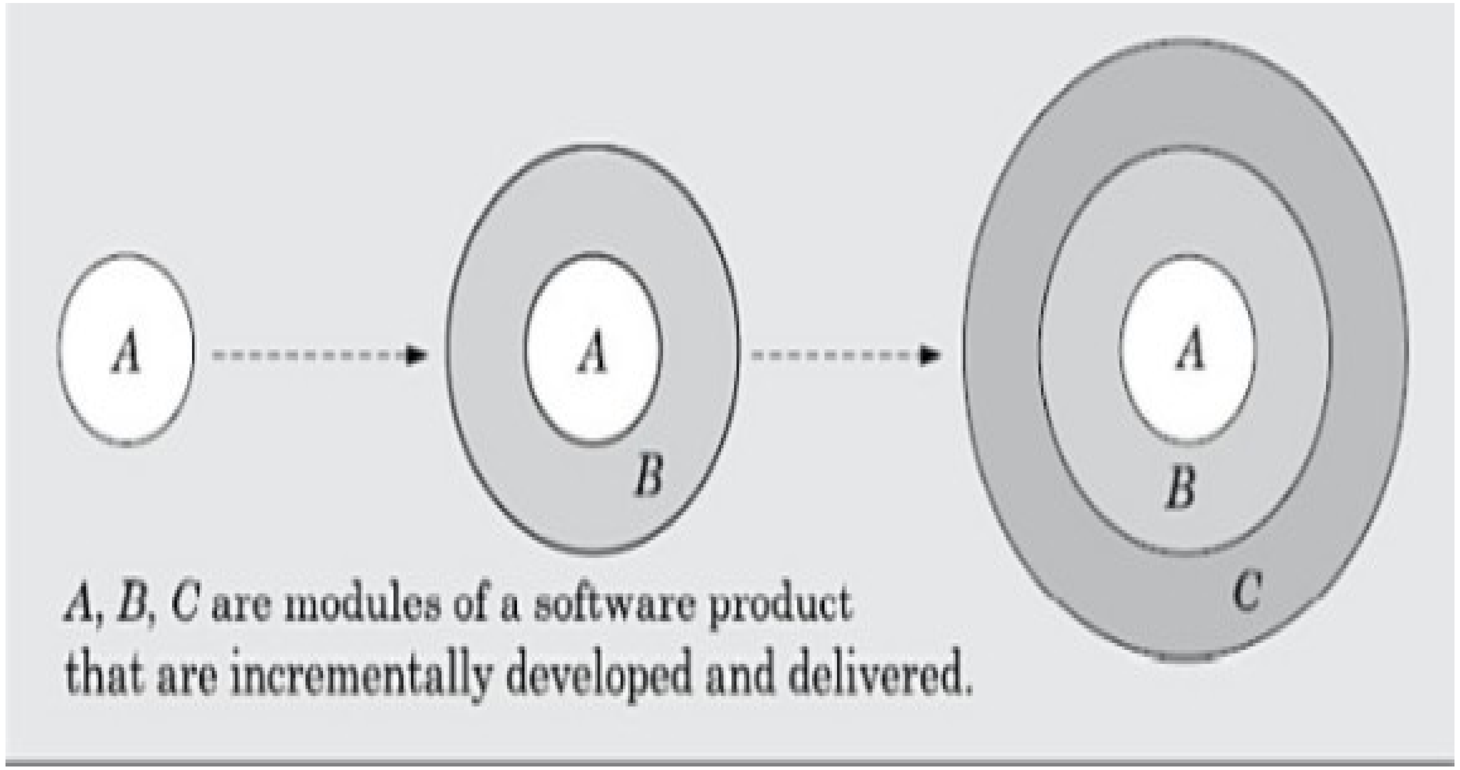
- Requirements must be known prior
- Inhibits flexibility.
- No problem solving nature

- **Incremental Process Model**

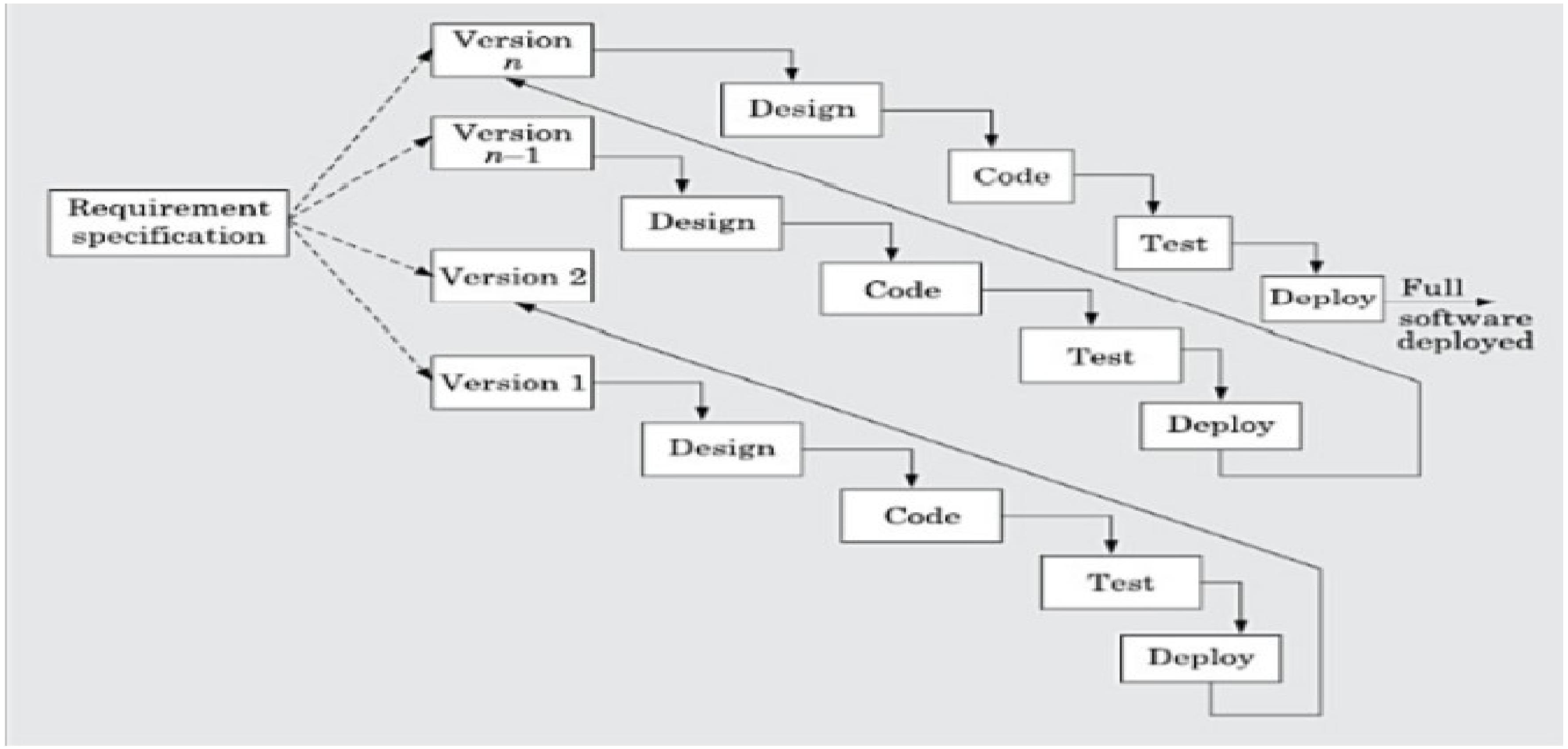
i. Sometimes also referred to as successive versions model.

ii. In this life cycle model, first a simple working system implementing only a few basic features is built and delivered to the customer. Over many successive iterations successive versions are implemented and delivered to the customer until the desired system is released.

iii. When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features (some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed review). As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.



A, B, C are modules of a software product that are incrementally developed and delivered.


Requirement specification
Version n
Version n–1
Version 2
Version 1
Design
Code
Test
Deploy
Full software deployed

- Strengths

- Reduces chances of errors in the core modules of the final product, leading to greater reliability of the software.
- This model obviates the need for the customer to commit large resources at one go for development of the system. It also saves the developing organisation from deploying large resources and manpower for a project in one go.

- Weakness

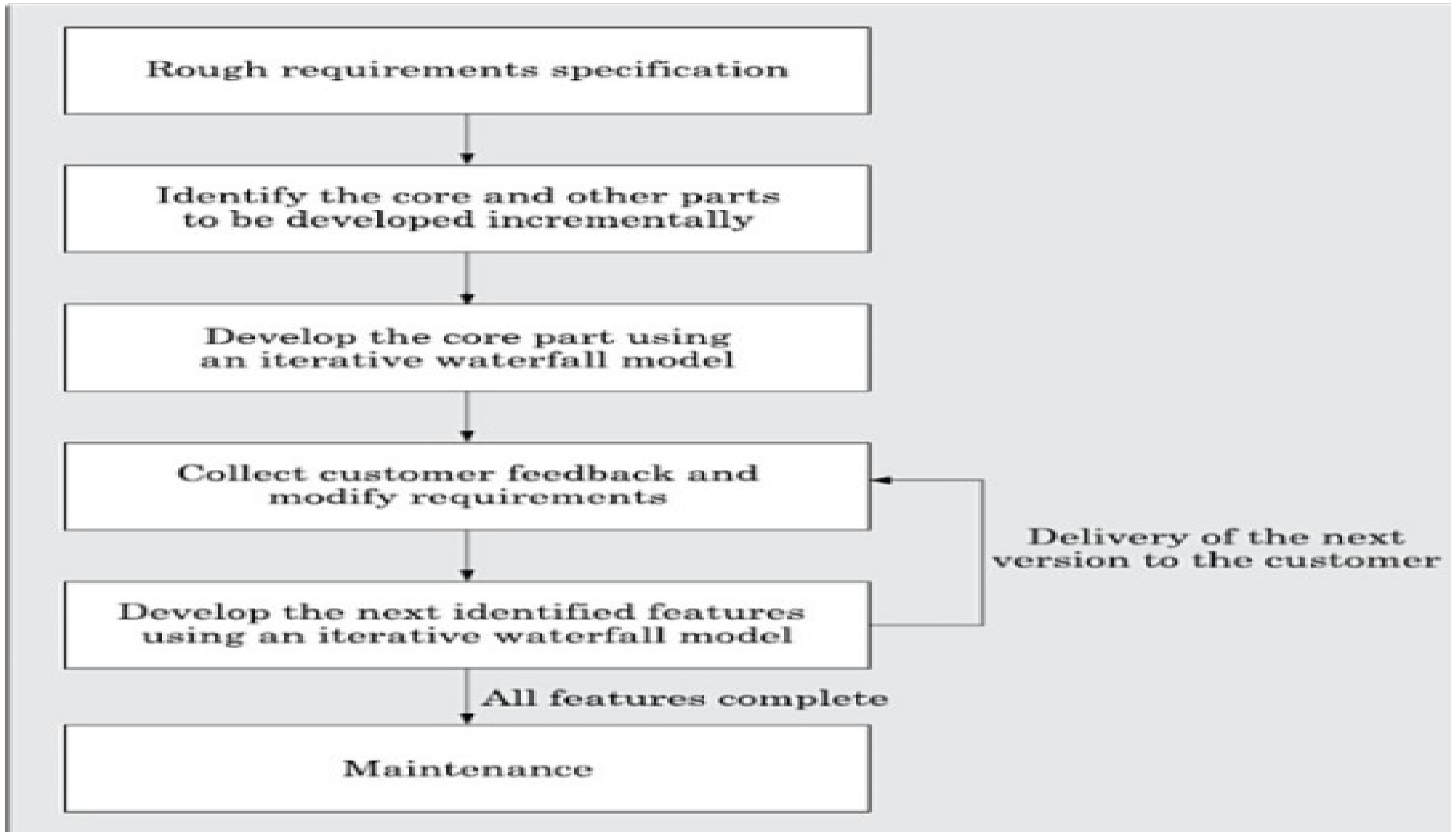
- Total cost is high
- Needs clear view of the entire system before it can be broken down and built incrementally
- Needs good planning and design

- Evolutionary model : Prototyping

i. The prototyping paradigm begins with requirements gathering.

ii. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A "quick design" then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g., input approaches and output formats).

iii. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.


Rough requirements specification
Identify the core and other parts to be developed incrementally
Develop the core part using an iterative waterfall model
Collect customer feedback and modify requirements
Develop the next identified features using an iterative waterfall model
Delivery of the next version to the customer
All features complete
Maintenance

- Strengths

- User gets a chance to experiment with a partially developed software much before the complete requirements are developed. As a result, the change requests after delivery of the complete software gets substantially reduced.

- In this model, handling change requests is easier as no long term plans are made. Consequently, reworks required due to change requests are normally much smaller compared to the sequential models

➢ Weakness

- It is difficult to divide the required features into several parts that can be incrementally implemented and delivered.
- Since at a time design for only the current increment is done, the design can become ad hoc without specific attention being paid to maintainability and optimality.

Selection criteria for software process model

- The software process model framework is specific to the project. Thus, it is essential to select the software process model according to the software which is to be developed.
- The software project is considered efficient if the process model is selected according to the requirements.
- It is also essential to consider time and cost while choosing a process model as cost and/ or time constraints play an important role in software development.
- The basic characteristics required to select the process model are project type and associated risks, requirements of the project, and the users.

- One of the key features of selecting a process model is to understand the project in terms of size, complexity, funds available, and so on.
- In addition, the risks which are associated with the project should also be considered.
- Note that only a few process models emphasize risk assessment. Various other issues related to the project.