SOFTWARE CONSTRUCTION MODELS

Software construction models are models that show the ways to navigate through the complex and demanding process of software building.

There are more than 50 recognized models in use but none is perfect and each brings its favorable aspects and disadvantages for a specific software development project or team.

Here, is the outline of the popular software construction models

1. **Waterfall**

Waterfall model is a continuous software development model in which development is seen as flowing steadily downwards like a waterfall through the steps of requirements analysis, design, implementation, testing, integration, and maintenance. In this method, the process of software development is divided into various phases. First, to identify the end of a phase and the beginning of the next, some certification techniques have to be employed at the end of each step.

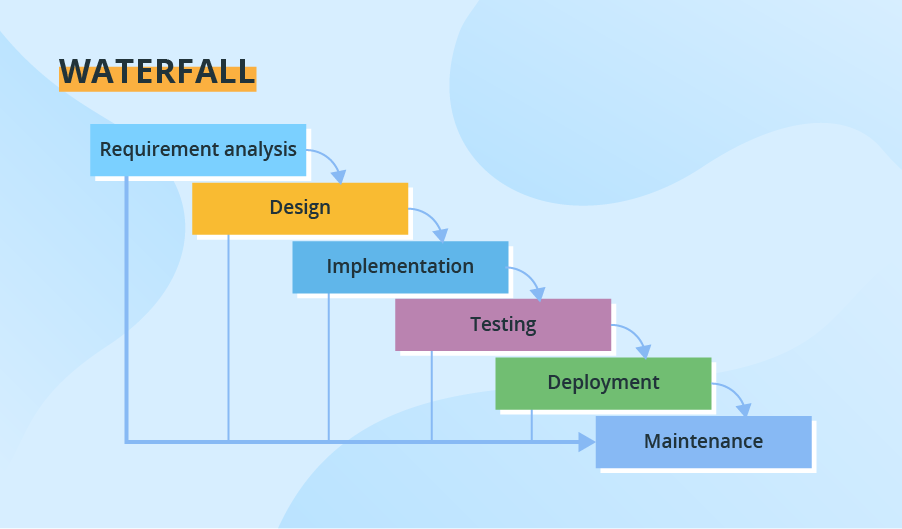
Reflective of its name, the model’s process flows steadily downwards through the phases of software implementation[[1]](#footnote-1).

This model relies on the completion of the previous phase for the following to begin. Because the model does not support going back to previously completed phases, the Waterfall Model should be used with projects that do not anticipate unforeseen changes mid-development.

Advantages and Disadvantages of Waterfall

**Advantages**: Simple and understandable, the Waterfall Model is a manageable method ideal for lifecycle management of smaller projects where the requirements are established and finalized upfront.

**Disadvantages**: Because of its rigid structure, the Waterfall Model does not work well for complex projects where there is a chance of a change in requirements and/or significant impromptu testing throughout the software development stage.



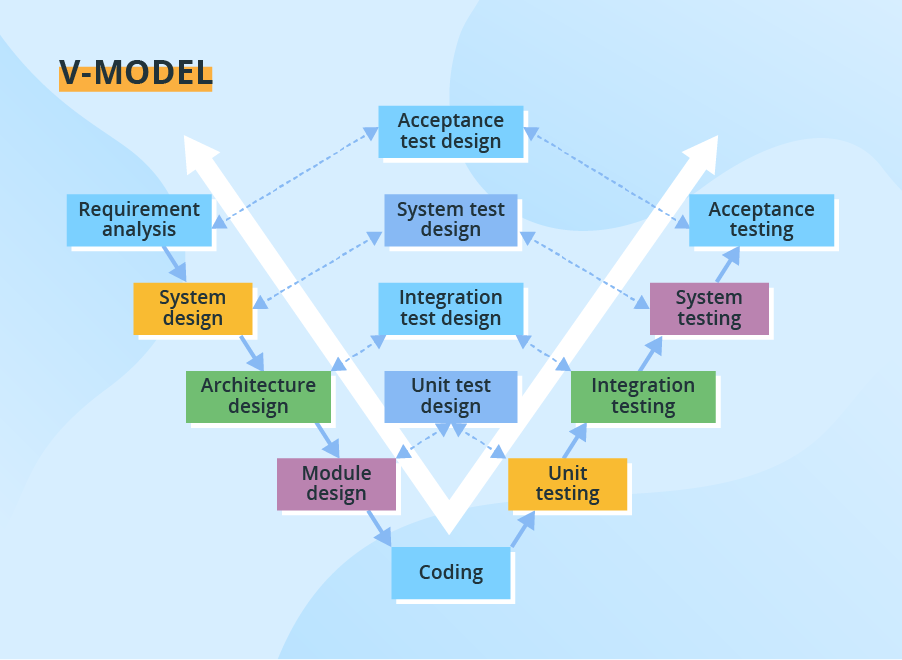
1. **V-MODEL** (VALIDATION AND VERIFICATION MODEL).

In this model testing and the development, the step is planned in parallel. So, there are verification phases on the side and the validation phase on the other side. V-Model joins by Coding phase. An extension of the Waterfall Model, the V-model also functions as a sequential flow[[2]](#footnote-2). However, instead of only moving linearly downward, the software development lifecycle bends upwards after the coding.

For every phase in the downward sequence, there is a corresponding testing phase in the following upwards sequence. This model is used for projects where software requirements and tools are known upfront.

#### Advantages and Disadvantages of V-Model

**Advantages**: The V-Model is a simple process that’s great for smaller projects. Using the V-Model can yield a higher chance of success due to the test plans of the development stage and regularly schedule updates throughout its lifecycle.

**Disadvantages**: Similar to the Waterfall Model, the V-Model is very rigid in nature so it isn’t ideal for applications or systems software that may require unforeseen changes/updates throughout the software lifecycle.

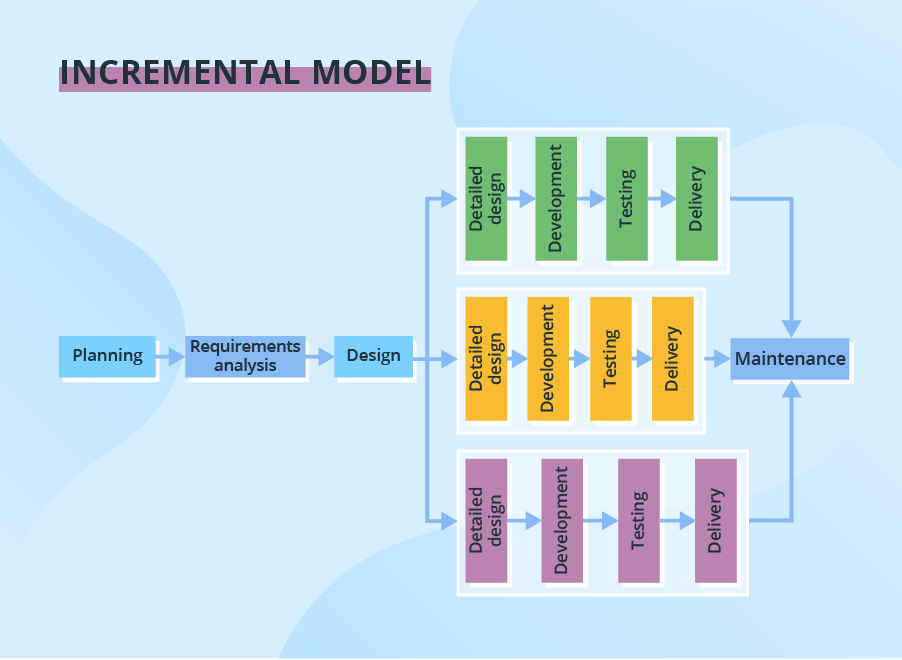
1. **INCREMENTAL MODEL**

consists of iterative and incremental development stages. The Incremental Model is essentially comprised of several mini-Waterfall cycles. So, it mediates the short comings of the waterfall model.

This model divides the development into small sections/stages and it can allow the software developers to take advantage of learnings and insights gleaned from earlier development stages.

#### Advantages and Disadvantages of Incremental Model

**Advantages**: The Incremental Model is a great solution for projects that need accommodation for some change requests between increments. This model also yields the benefit of being able to detect problems earlier in the software development for better lifecycle management planning.

**Disadvantages**: A potential disadvantage to the Incremental Model is the need for strategic planning and documentation. This method also tends to require more resources, staff and monetary, behind the project. This model isn’t ideal for ongoing development as the next sequence cannot begin until the previous stage has fully completed.

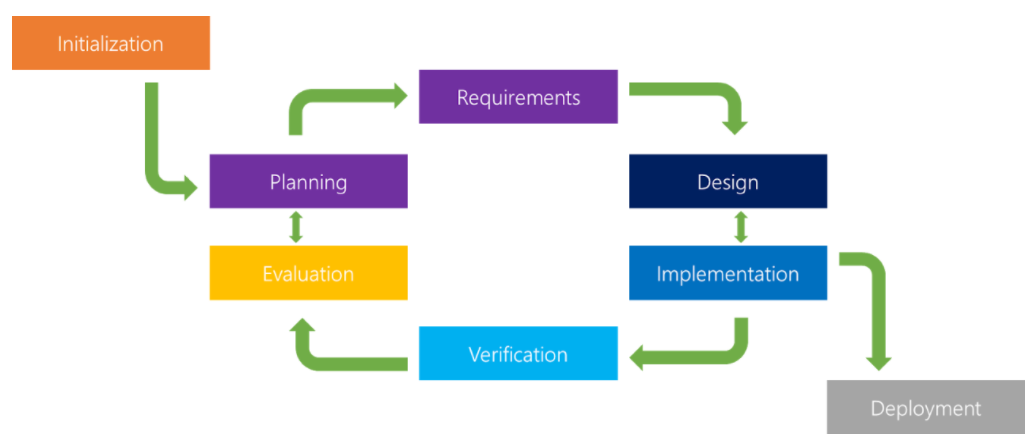
1. **ITERATIVE MODEL**

 relies on specifying and implementing individual parts of the software, rather than attempting to start with full specification requirements. Once a rough product is created within an iteration, it is then reviewed and improved in the next iteration and so on. The Iterative Model relies on the whole product being developed step-by-step (Design/Develop, Test, Implement).

#### Advantages and Disadvantages of Iterative Model

**Advantages**: Since the product is developed gradually, it’s easy to identify problems early when using this software development model.

**Disadvantages**: Because each iteration phase is rigid with no overlaps, the Iterative Model can take longer and be more costly.

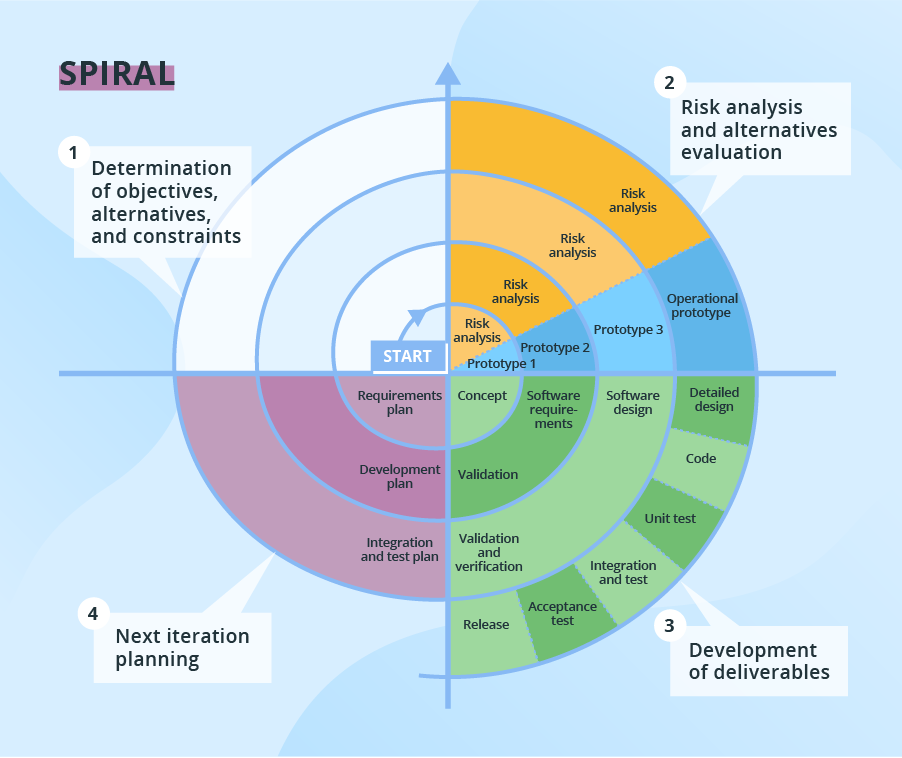
****

1. **SPIRAL MODEL.**

The spiral model is a **risk-driven process model**. This software development model helps the group to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc. The spiral technique is a combination of rapid prototyping and concurrency in design and development activities.

It combines elements of both the Iterative and Waterfall development models, in efforts to combine advantages of top-down and bottom-up production.

The Spiral Model has four phases – Identification, Design, Construct/Build, Evaluation and Risk Analysis.

The software project repeatedly passes through these phases in iterations called spirals. 

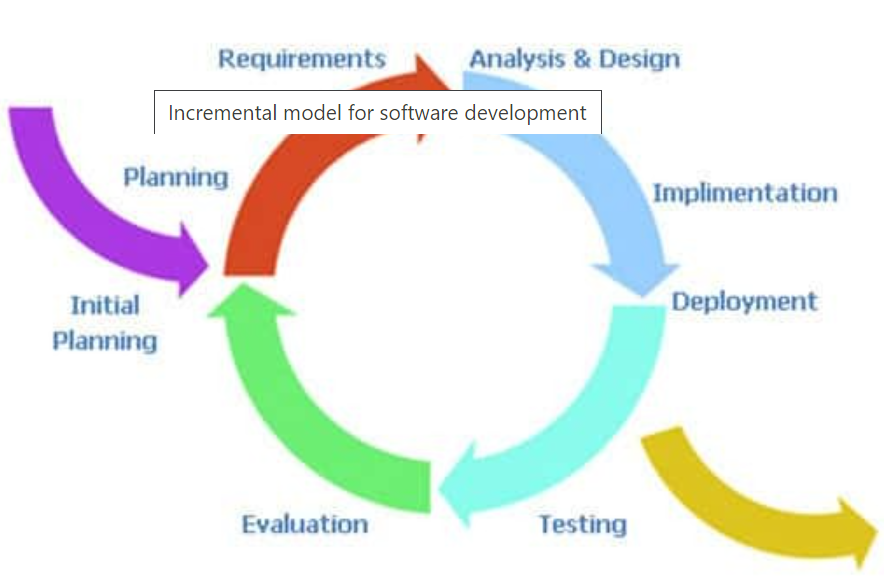
1. **THE RATIONAL UNIFIED PROCESS (RUP)**

is a modification of the Incremental Model. When implementing this model, several components are developed simultaneously as if they were smaller, individual projects. The different components are then assembled into working prototypes. A good short for Rapid application development.

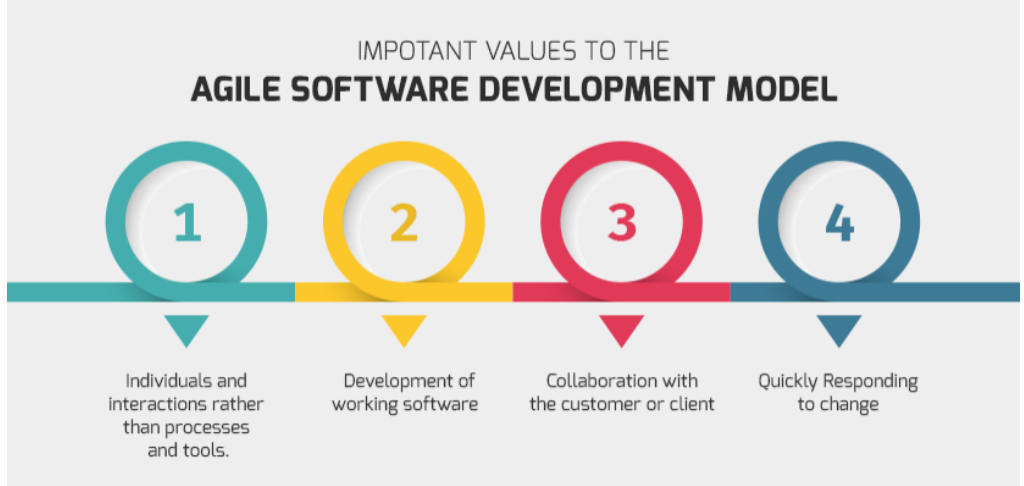
#### Advantages and Disadvantages of RAD

**Advantages:** The RAD Model allows for reduced development time and allows for more customer feedback throughout the software development.

**Disadvantages**: The applicability of the RAD Model is limited, as the project needs to be easily modularized into several increments. It also requires highly-versed developers as well as excellent modelling and planning skills. Issues with the final assembly of components could result in unforeseen setbacks and the redeveloping of components to properly fit the rest.

****

1. **THE AGILE MODEL.**

****

This model is rooted in process adaptability and user engagement with rapid delivery of functioning software components. It has both Iterative and Incremental features, as it breaks the product development into small incremental builds that are then provided in iterations.

It relies on heavy collaboration between cross functional teams and the customer/user.

#### Advantages and Disadvantages of Agile

**Advantages**: The Agile Model decreases the amount of time to yield individual system features. It also calls for a lot of communication and continuous feedback from the customer/user that can provide clear direction for the project.

**Disadvantages:** The Agile method can potentially veer-off track as it relies on end-user interaction that may or may not be clearly expressed. Documentation is also minimal for an Agile software development strategy and requires a well-versed, cross-functional team.

**Reference**

[**https://ca.insight.com/en\_CA/content-and-resources/2016/07152016-types-of-software-development-models.html**](https://ca.insight.com/en_CA/content-and-resources/2016/07152016-types-of-software-development-models.html)

[**https://www.javatpoint.com/software-engineering-sdlc-models**](https://www.javatpoint.com/software-engineering-sdlc-models)

1. https://ca.insight.com/en\_CA/content-and-resources/2016/07152016-types-of-software-development-models.html [↑](#footnote-ref-1)
2. https://ca.insight.com/en\_CA/content-and-resources/2016/07152016-types-of-software-development-models.html [↑](#footnote-ref-2)