Fundamental notions of programming (NFP)

Class 8

The software development life cycle

Before anything else, a program starts as an idea. In order for the idea to become a program ready to use by the end-user, it goes through a series of tasks divided into multiple stages. These tasks are organized in a well structured manner we call development process.

There are no standards on what these tasks are and how to organize and group them, especially because each project is unique and requires different steps from a unique group of people, the development team.

The Basic Stages

Even though every project is unique, there are still five or six usual steps in the program development process: planning (or analysis), design, development, implementation, testing and deployment. Maintenance can also be considered as one of these steps, when required by the client.

Step 1: Planning (or analysis)

First and probably most important stage of the development process, this stage usually starts with a meeting (or rather a few) between the most senior members of a team, sales department, the customers and some industry specialists to define and understand the mandate, the customer's needs.

Of course, the technical aspects are considered, but also industry-related and market requirements. At the end of this stage, a document is created (Software Requirement Specification) which details the scope and requirements of the project. It will be used as basis for the next stages.

Step 2: Design

The design stage is sometimes also referred to as *algorithm Development stage*. Based on the *software Requirement specification* document, the architects team comes out with the best possible architecture for the program to be developed. This is where the algorithms are created and tested in order to solve every requirement priorly established.

Step 3: Development (production)

The development stage is the one where the actual code is written by the programmers. It is the longest phase of the process along with implementation. Based on the algorithms, the team produces the program that was planned and structured.

Step 4: Testing

Here, quality control specialists check if the program is working as it should according to the *Software Requirement Specification* document. Logical and syntax errors will be logged and sent to the programmers for correction.

Step 5: Implementation (deployment)

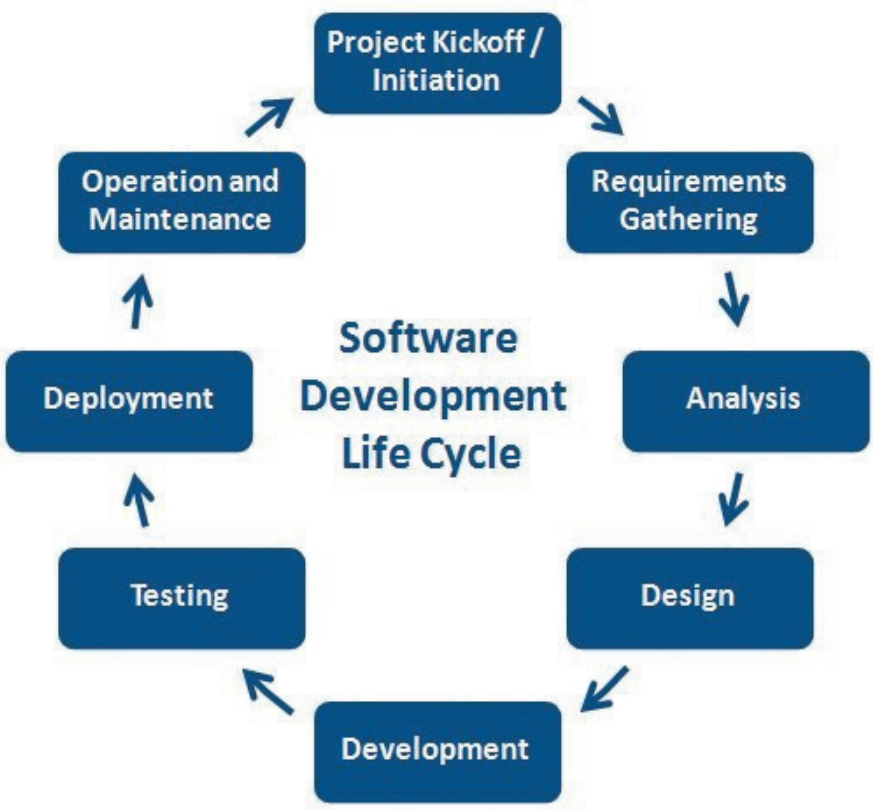
When the tests and corrections are completed, the program is deployed. When it replaces and older program, all the data and components of the old program are transfered to the new one.

This may be quite a simple stage for smaller projects, but for large applications, it can be considered as a project in itself. The architects need to make sure the hardware and third party requirements are fulfilled.

Also, starting at the development stage, software documentation starts to be created and updated as needed along the process in order to help future programmers who will work on the product. A well documented program will prevent a lot of headache and waste of future resources.

Step 6: Maintenance

Programs can't (or shouldn't) simply be deployed and forgotten about. After deployment, a team should ensure that the product is maintained, updated as needed and, in some cases, adapted to changes in the market or industry.

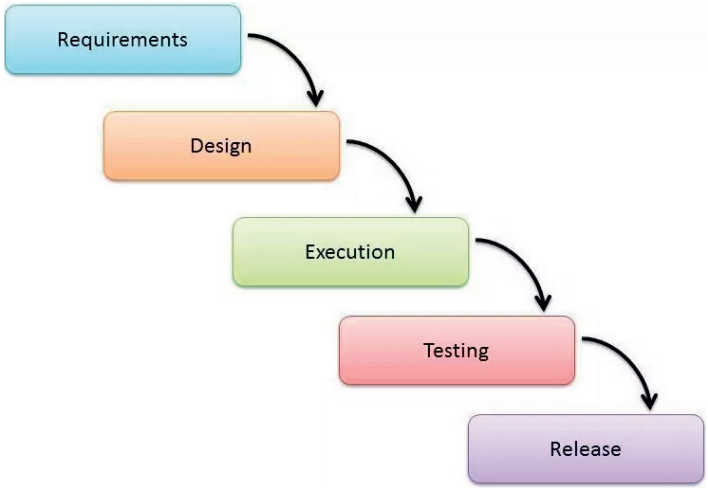


Popular Software Development Life Cycle Models

There are several well-known models used across multiple companies, NGOs and government agencies. A single institution may use several models depending on the team, project size and requirements. Following are the most well known models and their basic characteristics:

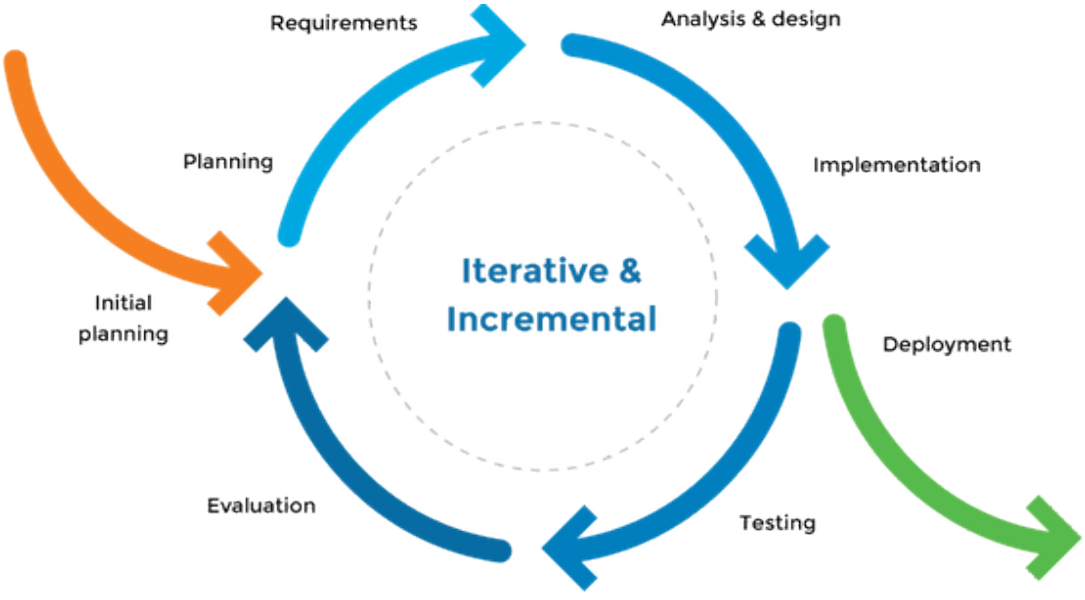
Waterfall Model

Considered to be the first major (widely known) model, each phase in this model must be completed before the project may move to the next one, and no phases overlapping in possible. This is a sequential design process consisting in the following phases: conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.



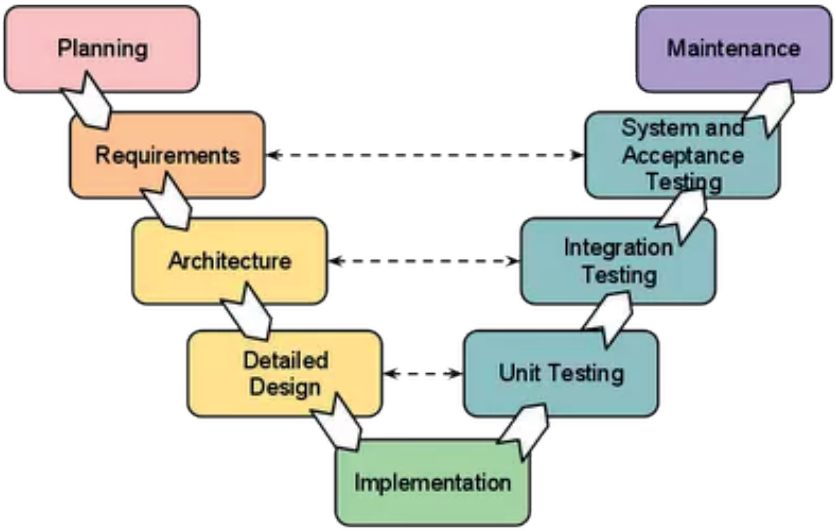
Iterative Model

The Iterative Model (or Incremental Model) is a system development life cycle model focusing on an initial simplified implementation which then progressively gains complexity by adding features through a series of planning, analysis, implementation, testing and evaluation phases happening several times.



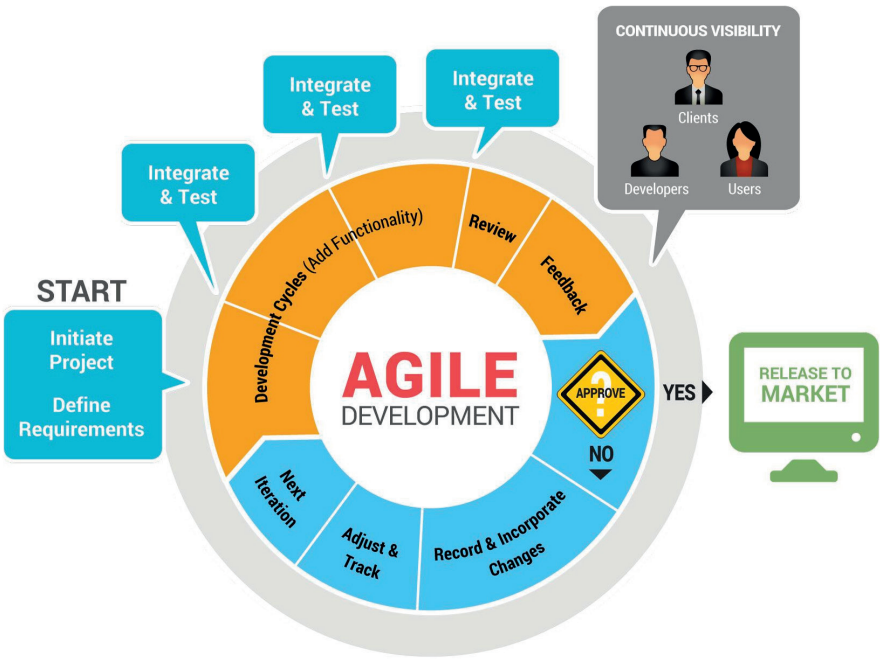
V-Shaped Model

Sometimes referred to as *validation model* or *verification model*, this is an expansion of the classic *waterfall model*. It is a very strict model where the next stage starts only after the previous one is finished. The difference here being that every development stage has an associated test stage in order to ensure that the conversion to the next stage is possible.



Agile Model

Based on iterative and incremental development, this model requires a greater level of engagement from both the customer and the team because the requirements for the project are not completely clear from the very beginning. Collaboration between cross-functional teams is also a characteristic of this model.



Assignment 6:
Popular Software Development Life Cycle Models

Create a simple HTML/CSS web page where you will briefly explain and represent visually the most popular software development life cycle models.

You can explain each of the ones you choose to talk about or even create a comparative table. You may include as many image and external links if needed.

You must write the article yourself, in your own words to describe the advantages and characteristics of each chosen models.

(Minimum 500 words)