Assignment of Software Engineering

By,

M. Ammar

BSCS-F19-M63

Section - B (Morning)

University of Punjab • Jhelum Campus • September 27, 2020

Difference between

Classical & Iterative Waterfall Models

Submitted to: Ma’am Aisha

Basic Definitions

Classical Waterfall Model

- Classical waterfall model is the basic **software development life cycle** model.

- It divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall

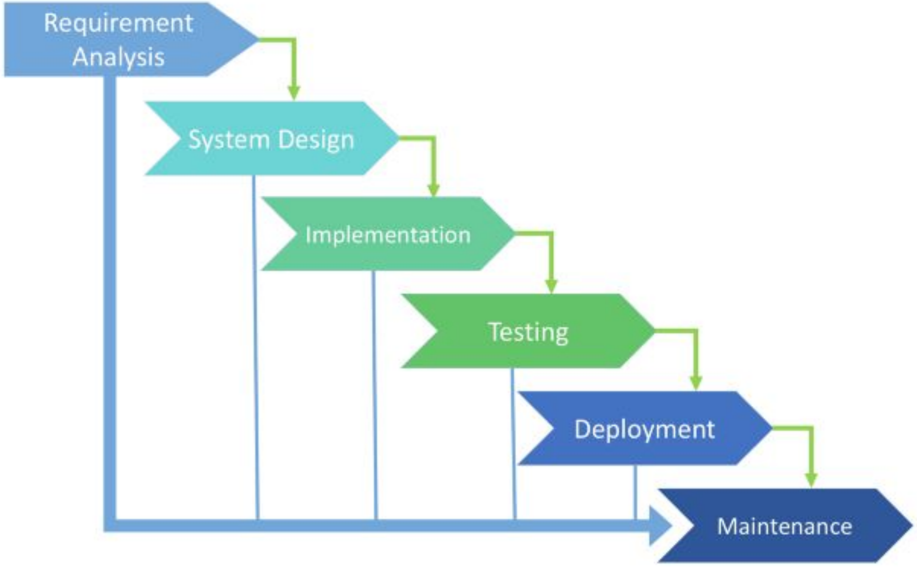
Iterative Waterfall Model

- Iterative Waterfall Model is the extension of the **Waterfall model**.

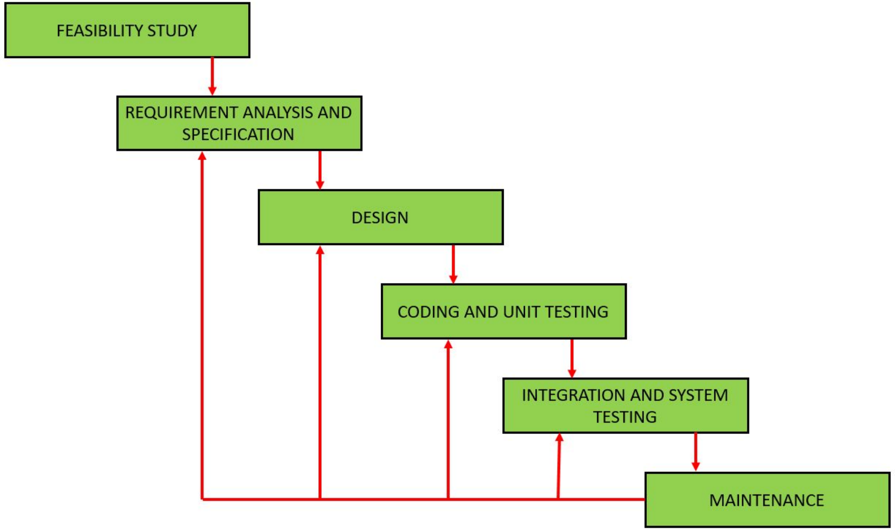
- The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.

Illustration by Figures

Classical Waterfall Model



Iterative Waterfall Model


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

Differences in Structure

Classical Waterfall

- In the **waterfall model**, the output of the previous phase will be the input to the next phase in the software development life cycle.
- In this model, we **cannot go** back to the **previous phase** to alter any thing if we come to the next phase.
- This model **requires some effort** and extra work to design each possible case.

Iterative Waterfall

- The **iterative waterfall** model provides customer's feedback paths from each phase to its previous phases.
- Iterative waterfall **allows** to **go back** on the previous phase and change the requirements and some modification can done if necessary.
- This model **reduces** the developer's effort and time required to detect and correct the errors.

Differences in Use-cases

Classical Waterfall

- In the **waterfall model**, we require a complete detail of requirements before designing a model
- **Time and Budget** is difficult to predict in each phase. So, it may cost higher
- **Waterfall model** is not suitable, when the requirements are not fixed. As, the waterfall model is a bit difficult to manage

Iterative Waterfall

- The **iterative waterfall** model, full list of requirements are not required
- A **limited budget** is enough, as errors are being identified beforehand
- Iterative waterfall is most effective and suitable, where **scope** and **requirements** are not fixed, and may change in the future

Classical Waterfall Model

Advantages

- This model is very **simple** and is **easy** to understand.
- **Phases** in this model are processed one at a time.
- **Each stage** in the model is clearly defined.
- This model has very clear and well understood **milestones**.
- **Process, actions and results** are very well documented.
- Reinforces **good habits**: define-before-design, design-before-code.
- This model works well for **smaller projects** and projects where requirements are well understood.

Classical Waterfall Model

Disadvantages

- **No feedback path:** In classical waterfall model evolution of a software from one phase to another phase is like a waterfall. It assumes that no error is ever committed by developers during any phases. Therefore, it does not incorporate any mechanism for error correction.

- **Difficult to accommodate change requests:** This model assumes that all the customer requirements can be completely and correctly defined at the beginning of the project, but actually customers' requirements keep on changing with time. It is difficult to accommodate any change requests after the requirements specification phase is complete.

- **No overlapping of phases:** This model recommends that new phase can start only after the completion of the previous phase. But in real projects, this can't be maintained. To increase the efficiency and reduce the cost, phases may overlap.

Iterative Waterfall Model

Advantages

- Iterative waterfall model is very **easy** to understand and use.
- Every phase contains **feedback path** to its previous phase.
- This is an simple to make changes or any **modifications** at any phase.
- By using this model, developer can complete project earlier.
- **Customer involvement** is not required during the software development.
- This model is suitable for **large** and **complex** projects.

Iterative Waterfall Model

Disadvantages

- There is no feedback path for **feasibility study** phase.

- This model is not suitable if **requirements** are not clear.

- It can be more **costly**.

- There is no process for **risk handling**.

- Customer can view the final project. There is **no prototype** for taking customer reviews.

- This model does not work well for **short projects**.

- If **modifications** are required repeatedly then it can be more complex projects.

Final Words

- If we want to design a model for a **Short Project**, we can use **Classical Waterfall Model** for our own ease. As it may provide more features than Iterative waterfall, in some cases.

- But if we want to design a waterfall model for a **Bigger and Complex Project**, we should consider using **Iterative Waterfall Model**, as it provides customer’s feedback, going back and forth in the model (according to certain rules), and is easy to modify according to **cost, scope** and **requirements**.

The End

By, Muhammad Ammar - From 2nd Semester - Section B (Morning) - BSCS_F19_M63