**SOFTWARE DEVELOPMENT LIFE CYCLE PROCESS.**

**Software Project Management.**

**Nkuraija Gary Matthew BSCS3.**

**S19B23/537.**

**SOFTWARE DEVELOPMENT LIFE CYCLE PROCESS**

Software Development Life Cycle Process is a process that defines the various stages involved in the development of software for delivering a high-quality product. SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.

Purpose of SDLC is to deliver a high-quality product which is as per the customer’s requirement.

*SDLC has defined its phases as, Requirement gathering, Designing, Coding, Testing, and Maintenance. It is important to adhere to the phases to provide the Product in a systematic manner.*

**For Example,** software has to be developed and a team is divided to work on a feature of the product and is allowed to work as they want. One of the developers decides to design first whereas the other decides to code first and the other on the documentation part. This will lead to project failure because of which it is necessary to have a good knowledge and understanding among the team members to deliver an expected product.

**Software Development Life Cycle Models**

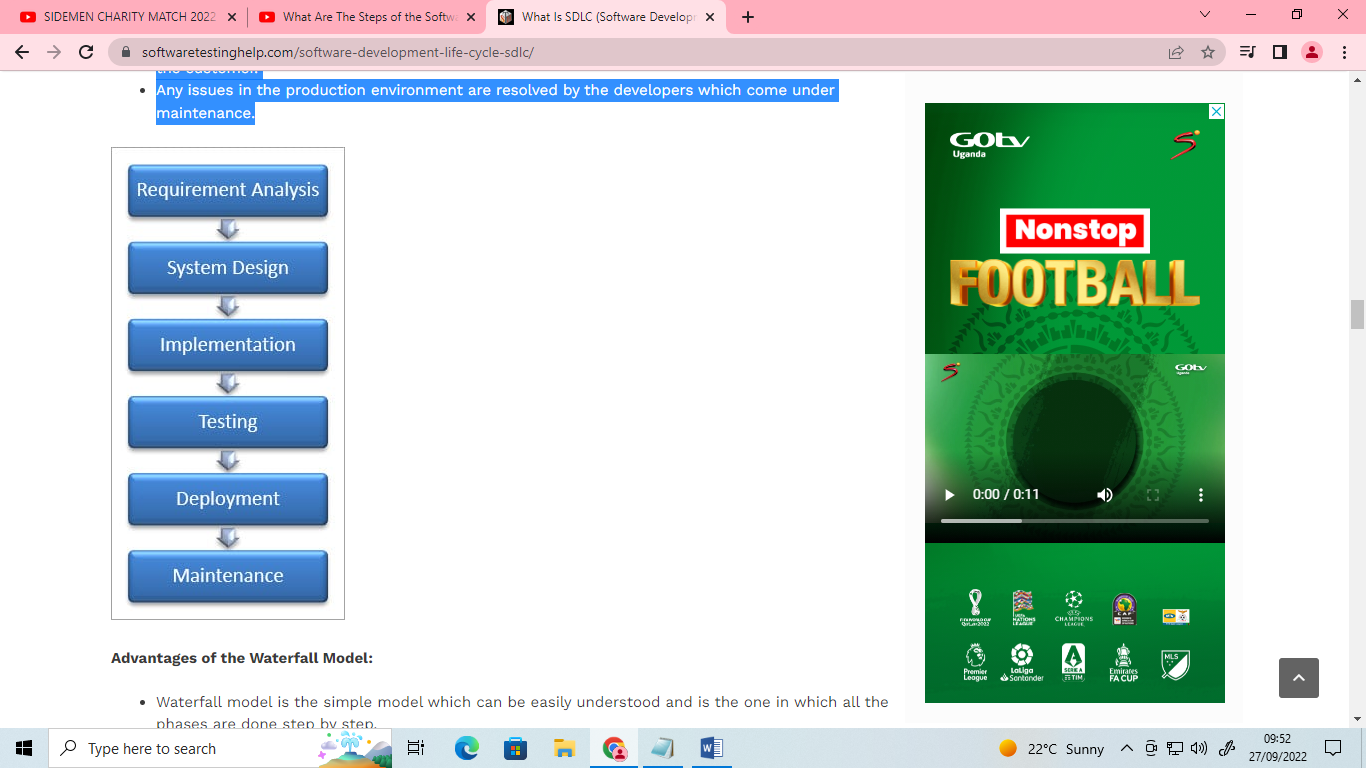
A software life cycle model is a descriptive representation of the software development cycle. SDLC models might have a different approach but the basic phases and activity remain the same for all the models.

**Examples of life cycle models**.

1. **Waterfall model.**

Waterfall model is the very first model that is used in SDLC. It is also known as the linear sequential model. In this model, the outcome of one phase is the input for the next phase. Development of the next phase starts only when the previous phase is complete.

* First, Requirement gathering and analysis is done. Once the requirement is freeze then only the System Design can start. Herein, the SRS document created is the output for the Requirement phase and it acts as an input for the System Design.
* In System Design Software architecture and Design, documents which act as an input for the next phase are created i.e. Implementation and coding.
* In the Implementation phase, coding is done and the software developed is the input for the next phase i.e. testing.
* In the testing phase, the developed code is tested thoroughly to detect the defects in the software. Defects are logged into the defect tracking tool and are retested once fixed. Bug logging, Retest, Regression testing goes on until the time the software is in go-live state.
* In the Deployment phase, the developed code is moved into production after the sign off is given by the customer.
* Any issues in the production environment are resolved by the developers which come under maintenance.



**Advantages of the Waterfall Model:**

* Waterfall model is the simple model which can be easily understood and is the one in which all the phases are done step by step.
* Deliverables of each phase are well defined, and this leads to no complexity and makes the project easily manageable.

**Disadvantages of the Waterfall Model:**

* Waterfall model is time-consuming & cannot be used in the short duration projects as in this model a new phase cannot be started until the ongoing phase is completed.

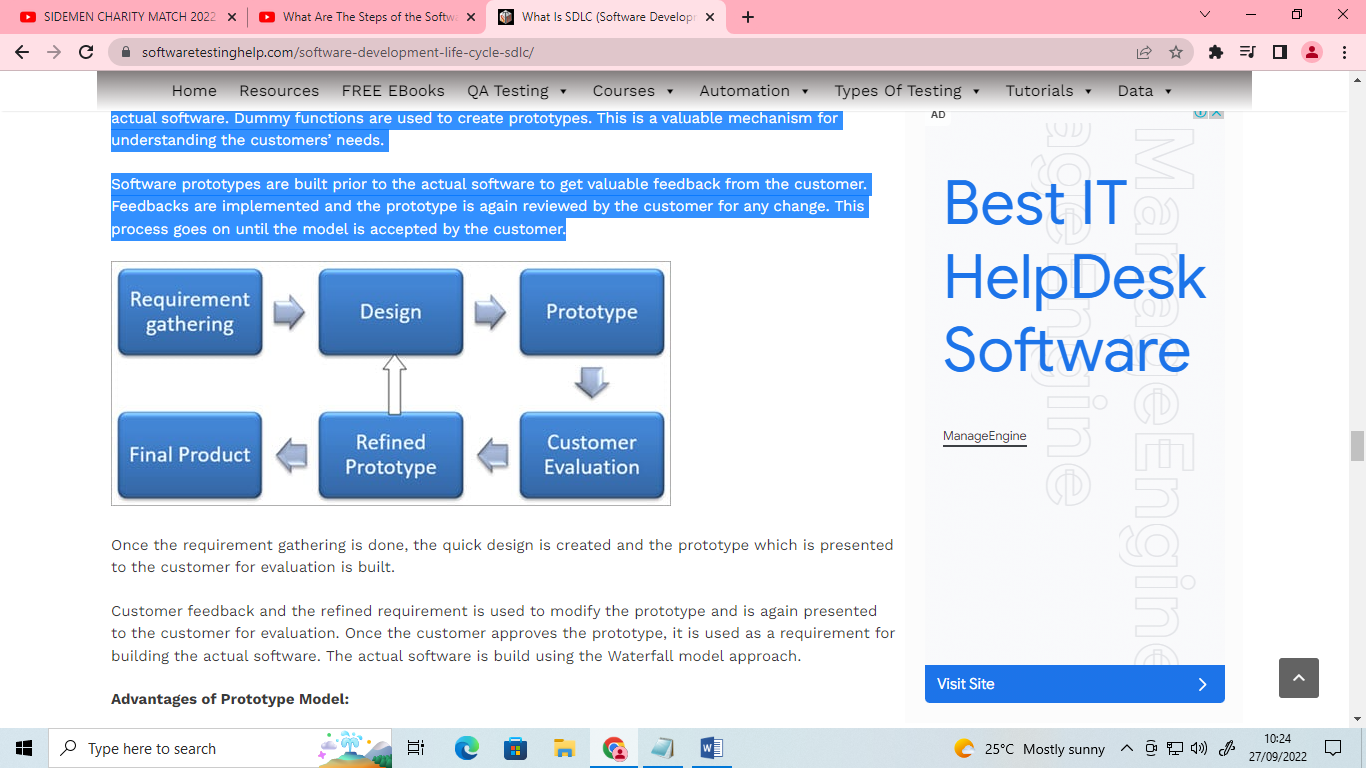
Waterfall model cannot be used for the projects which have uncertain requirement or wherein the requirement keeps on changing as this model expects the requirement to be clear in the requirement gathering and analysis phase itself and any change in the later stages would lead to cost higher as the changes would be required in all the phases.

1. **Prototype Model**

The prototype model is a model in which the prototype is developed prior to the actual software.

Prototype models have limited functional capabilities and inefficient performance when compared to the actual software. Dummy functions are used to create prototypes. This is a valuable mechanism for understanding the customers’ needs.

Software prototypes are built prior to the actual software to get valuable feedback from the customer. Feedbacks are implemented and the prototype is again reviewed by the customer for any change. This process goes on until the model is accepted by the customer.



Once the requirement gathering is done, the quick design is created and the prototype which is presented to the customer for evaluation is built.

Customer feedback and the refined requirement is used to modify the prototype and is again presented to the customer for evaluation. Once the customer approves the prototype, it is used as a requirement for building the actual software. The actual software is build using the Waterfall model approach.

**Advantages of Prototype Model:**

* **Prototype model reduces the cost and time of development as the defects are found much earlier.**
* **Missing feature or functionality or a change in requirement can be identified in the evaluation phase and can be implemented in the refined prototype.**
* **Involvement of a customer from the initial stage reduces any confusion in the requirement or understanding of any functionality.**

**Disadvantages of Prototype Model:**

* Since the customer is involved in every phase, the customer can change the requirement of the end product which increases the complexity of the scope and may increase the delivery time of the product.

1. **Spiral Model**

The Spiral Model includes iterative and prototype approach.

Spiral model phases are followed in the iterations. The loops in the model represent the phase of the SDLC process i.e. the innermost loop is of requirement gathering & analysis which follows the Planning, Risk analysis, development, and evaluation. Next loop is Designing followed by Implementation & then testing.

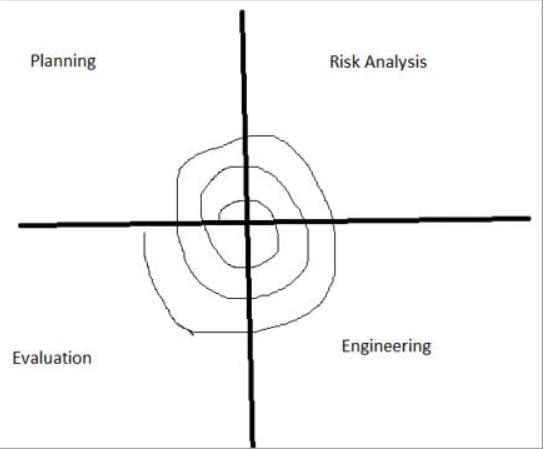
**Spiral Model has four phases:**

Planning

Risk Analysis

Engineering

Evaluation



1. **Planning:**

The planning phase includes requirement gathering wherein all the required information is gathered from the customer and is documented. Software requirement specification document is created for the next phase.

1. **Risk Analysis:**

In this phase, the best solution is selected for the risks involved and analysis is done by building the prototype.

For Example, the risk involved in accessing the data from a remote database can be that the data access rate might be too slow. The risk can be resolved by building a prototype of the data access subsystem.

1. **Engineering:**

Once the risk analysis is done, coding and testing are done.

1. **Evaluation:**

Customer evaluates the developed system and plans for the next iteration.

**Advantages of Spiral Model:**

Risk Analysis is done extensively using the prototype models.

Any enhancement or change in the functionality can be done in the next iteration.

**Disadvantages of Spiral Model:**

The spiral model is best suited for large projects only.

The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

1. **Iterative Incremental Model**

The iterative incremental model divides the product into small chunks.

For Example, Feature to be developed in the iteration is decided and implemented. Each iteration goes through the phases namely Requirement Analysis, Designing, Coding, and Testing. Detailed planning is not required in iterations.

Once the iteration is completed, a product is verified and is delivered to the customer for their evaluation and feedback. Customer’s feedback is implemented in the next iteration along with the newly added feature.

Hence, the product increments in terms of features and once the iterations are completed the final build holds all the features of the product.

**Phases of Iterative & Incremental Development Model:**

Inception phase

Elaboration Phase

Construction Phase

Transition Phase

1. **Inception Phase:**

Inception phase includes the requirement and scope of the Project.

1. **Elaboration Phase:**

In the elaboration phase, the working architecture of a product is delivered which covers the risk identified in the inception phase and also fulfills the non-functional requirements.

1. **Construction Phase:**

In the Construction phase, the architecture is filled in with the code which is ready to be deployed and is created through analysis, designing, implementation, and testing of the functional requirement.

1. **Transition Phase:**

In the Transition Phase, the product is deployed in the Production environment.

**Advantages of Iterative & Incremental Model:**

Any change in the requirement can be easily done and would not cost as there is a scope of incorporating the new requirement in the next iteration.

Risk is analyzed & identified in the iterations.

Defects are detected at an early stage.

As the product is divided into smaller chunks it is easy to manage the product.

**Disadvantages of Iterative & Incremental Model:**

Complete requirement and understanding of a product are required to break down and build incrementally.

**Conclusion**

Adherence to a suitable life cycle is very important, for the successful completion of the Project. This, in turn, makes the management easier.

Different Software Development Life Cycle models have their own Pros and Cons. The best model for any Project can be determined by the factors like Requirement (whether it is clear or unclear), System Complexity, Size of the Project, Cost, Skill limitation, etc.

Example, in case of an unclear requirement, Spiral and Agile models are best to be used as the required change can be accommodated easily at any stage.

Waterfall model is a basic model and all the other SDLC models are based on that only.

**References**

<https://www.youtube.com/watch?v=x-jqSXYE4S4&t=2s&ab_channel=Education4u>

<https://phoenixnap.com/blog/software-development-life-cycle#:~:text=Software%20Development%20Life%20Cycle%20is,%2C%20Test%2C%20Deploy%2C%20Maintain>