**What is SDLC?**

SDLC or software development life cycle is a process that helps to design, develop, test high quality software and maintaining of the software. It is also a framework that defines task that is done at every step in the whole development of the software process. These steps include planning, defining, designing, building, testing and deployment. The first step which is planning and requirement analysis is the most important part of SDLC, it is to gather requirements from the client, normally by the most experienced and senior members of the organization, after which it is gathered, a scope of the hole project will be done, this scope includes what is needed for the project and what benefits risks that comes with proceeding with this project.

The second step is to define requirements, which is to define the document the product requirements and getting approval from the customer. Software requirement specification(SRS) which is a document containing requirements of the product is used to do this step.

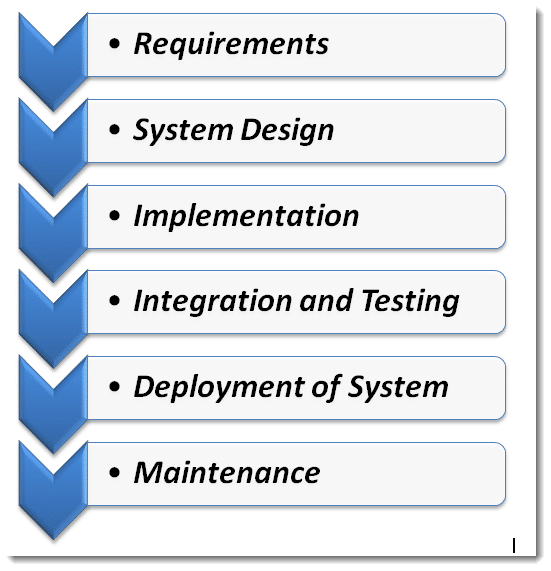
The third step is designing the product architecture. Basing on the SRS in the second step, a few design approaches is usually proposed and documented in a design document specification(DDS). The stakeholders will review the DDS and select the best approach for the product.

The fourth step is building or developing the product. Basing on the DDS in the third step, developer will begin to build the product, and following coding guidelines set by the organization.

The fifth step is to test the product, where all the defects of the product are reported, tracked, fixed and retested until all the requirements in the SRS from second step is met.

The final step is deployment in the market and maintenance, once enough testing is done, the product is release into the market and after which maintenance will be done to either improve the product more or fix bugs that appears.

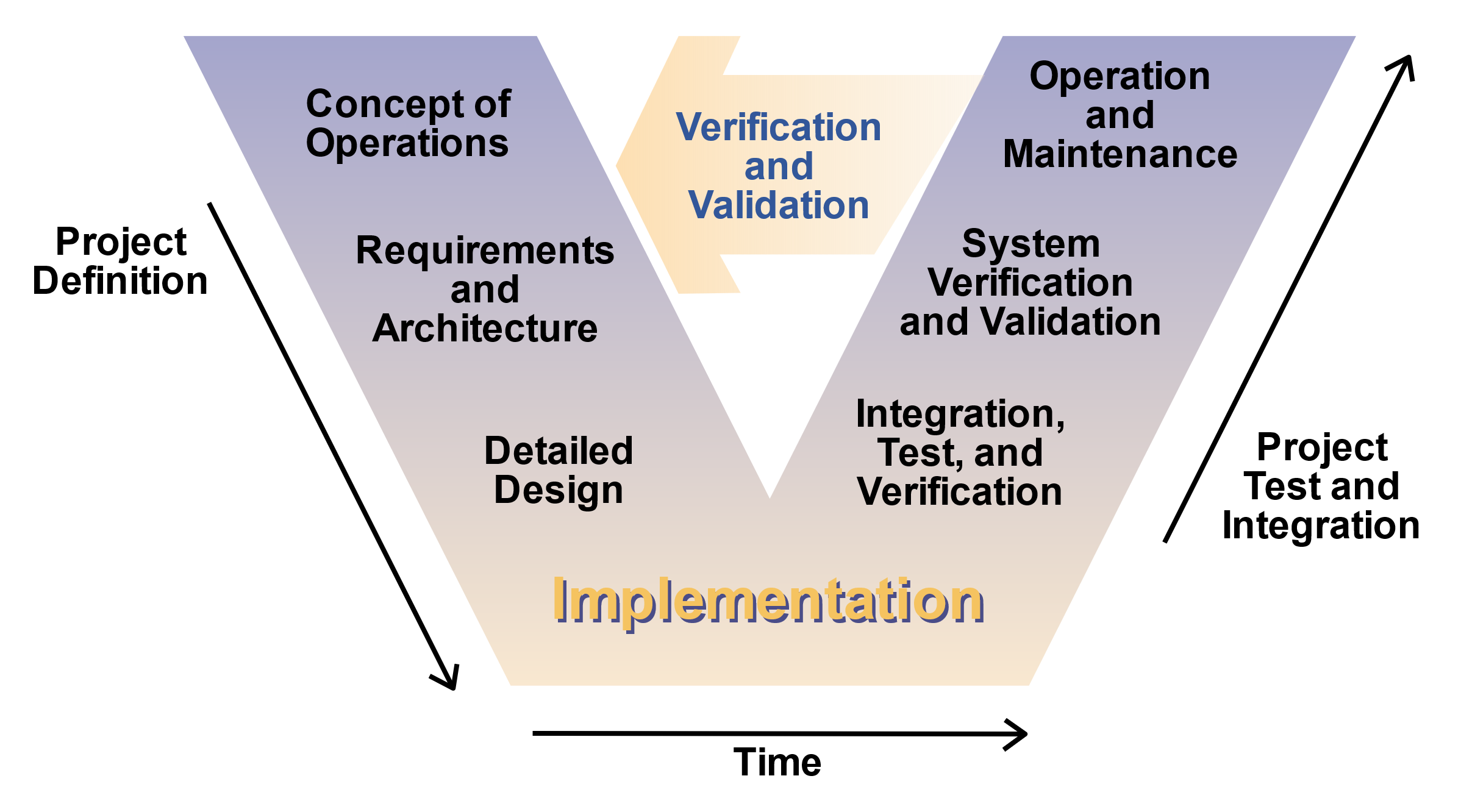
**3 models**

****

Waterfall Model. In this model, each phrase must be completed before the next phrase can begin, this means that this model is sequential, as each phrase “flows downward”, hence waterfall. The phrase are requirements, system design, implementation, integration and testing, deployment of system and maintenance. Requirements is to gather understanding on what need be to design and its function. System design is to design an product from the requirements gathered, this includes knowing which hardware and system requirements needed. Implementation is to develop units which are smaller programs. Integration and testing is to integrate all the smaller units together and finding error or flaws to fix. Deployment of system is to deploy the product once enough testing is done. Maintenance is to makes additional features or changes and to fix any bug discovered by the users.

The advantages of this model are that it allows for control as the project can be set by schedule and proceed phrase by phrase, easier to manage die to the rigidity of the model, each phrase do not overlap one another. The disadvantages are that it is hard to estimate time and cost for each of the phrases, when the application in the phrase of being tested, it is very difficult to make changes to the application, not meant to be used for complex and object-oriented projects and not suitable to project where the requirements have a high chance to be changed.

The hotel management application could use this model by knowing exactly what is needed through comprehensive research on the current market, having good knowledge to plan out when each phrase of the project starts and ends.

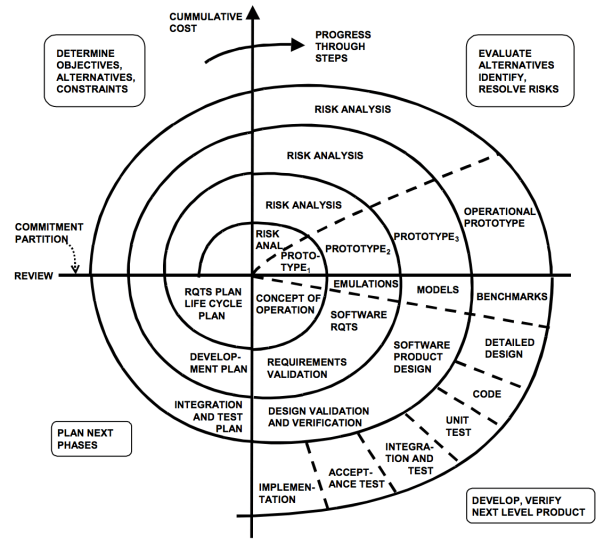


V Model, this model follows waterfall model in the sense that the phrases are very linear, following one another. Requirements and analysis are done to determine what is need, during each design stage, corresponding tests are also implemented later during the testing stages, thus acceptance tests are designed. System design is then created from the requirements gathered and a specification document which shows all technical components like data layers and business logic, at this phrase system tests are also developed. Architecture design, an high level design which is how the product will link all its components, integration tests are also develop at this phrase. Module design which is how all the functional, coded business logic are implemented, unit tests are also created during this phrase. In implementation, the actual coding and implementation begins, this phrase need as much time as possible.

Unit testing, this phrase starts testing the unit tests developed during the module phrase, making sure most of the bugs are gone. Integration testing, the tests developed during integration testing is done here. System testing, tests developed in system design are done here. Acceptance testing is to implement all tests developed during the initial requirement phrase and ensure that the system is ready for deployment.

Advantages of this model are that it is suited for restricted projects where everything is well defined, technology used is stable and all the documentation are clear. It is also ideal for time management where a strict deadline is needed. The disadvantages are that it lacks adaptability that waterfall model also faces, that any changes will be very hard to make, timeline restriction as all the focus on testing at the end will be rushed to meet the deadline. Lengthy life cycle projects are not suited for this model as the V-model is linear and changes are not easily able to be made.

The hotel management project can use this model as the deadlines are strict as part of the project constraint, it can be an advantage to use this model to force the project to be finish by the required dateline.



Spiral Model, this model is primary risk-driven which in turn ensures that the approach taken to the project is highly flexible. This model has the best usage when the stakes are high and major setbacks are not an option. First, define artefacts concurrently. Artefact is anything that produced by people in the software development process, this planning of every single artefact allows much less potential problems. The four-essential spiral task are to consider critical stakeholder objective and constraints, elaborate and evaluate alternative for achieving the objective, identifying and resolving rock attendant on the choices of alternative solutions and for stakeholder to review and agree to proceed based on satisfaction of their critical objective and constraints. By focusing on the wishes of the stakeholders, the team will be sure not to do option that do not meets the core need of the projects.

Risk determines level of efforts, if the amount of time taken to test something is too long, it is often cut short due to the risk of losing market share which is more important for this model. Risk also determines degree of details. It is best to gather more data and make the change more carefully. If the additional feature added is going to mess up badly if done wrongly, it is often best not to do it.

The three anchor point milestones are used, life cycle objectives(LCO), life cycle architecture(LCA) and initial operational capability(IOC). LCO checks if the technical part of the project is define well enough to proceed and if stakeholder requirements are met. LCA milestone checks if the best approach has been define well enough and all the big risks are accounted for and plans has been made for it. ICO milestone checks if enough preparations have been done to make the stakeholder satisfied, this includes software, site, users, operators and maintainers.

The hotel management project can use this project to define very well what is worth and what is not worth it to implement in the application and that the hotel owners are very satisfied with the final product.

**References:**

https://www.tutorialspoint.com/sdlc/sdlc\_overview.htm

https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc

http://toolsqa.com/software-testing/waterfall-model/

https://airbrake.io/blog/sdlc/v-model

https://airbrake.io/blog/sdlc/spiral-model