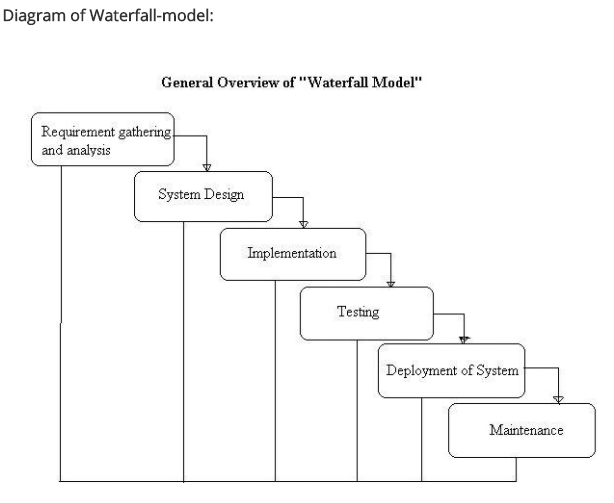
**Kenji Leong**

**Software Development Life Cycle (SDLC) + 3 development models**

SDLC also known as application development life-cycle is a process to ensure good software is build and typically it includes 5 phases starting with the analysis and requirements gathering and ending with the implementation. Firstly, requirement gathering is a phase that is critical to the success of the project as expectation is set by the sponsor or your team. Secondly, analysis and design phase is when the team choose the software that they want to use. Thirdly, Coding is the longest phase due to enhancement needed from the sponsor. Fourthly, testing phase includes user acceptance testing to ensure the system meets their expectation. Lastly, deployment phase is when the team roll out the system to the public.

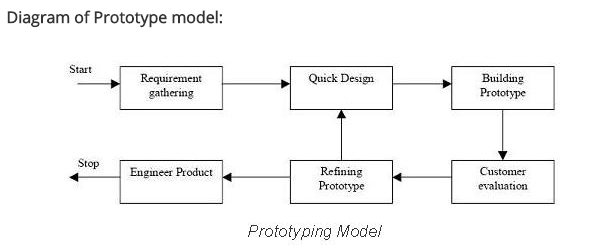
Besides, there are many different models that can be used to achieve different required objectives. Firstly, Waterfall model also known as **linear-sequential life cycle model is a traditional model that is well structured. It uses a sequential model as it flows downwards where each phase must be completed fully before the next phase can begin. This model is usually used for** small project that have no uncertain requirements.



As shown above there are 6 steps required for a waterfall model. For the advantages, it is easy to manage because of the rigidity of the model as each phase has a specific deliverable and a review process. Also, since the steps are completed one at a time. As a result, phases does not overlap with each other. However, the disadvantages include where once it reaches the testing stage, it is very troublesome and difficult to change something that is in the concept stage and it not a suitable for long term project due to many changes that need to be made during the process.

Thus, a waterfall model is best used only when the requirements are very clear and fixed where no changes will be made and also the size of the project is small.

Besides, Prototype model is where instead of freezing the requirements before any coding can be done. A throwaway prototype is built so that the developers can better understand the requirements that are needed for the required systems. By doing so, this can help verify requirement that are unclear.

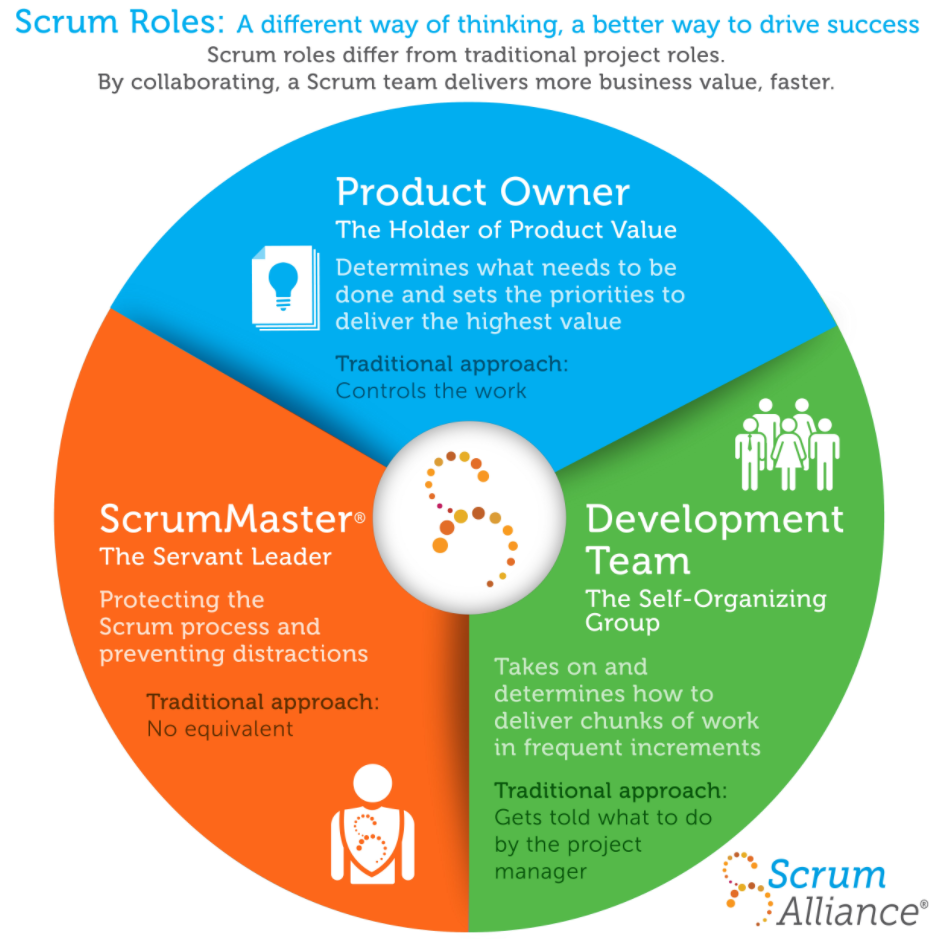


As shown above, a throwaway model is always being used to get feedback from the customer or sponsor and the team will then refine the product based on the feedback that was given. The advantage of using a prototype model includes the users are actively involved in the development since they have to provide feedback and errors such as bug can be discovered earlier so that the development team will have more time to solve these problems.

On the other hand, the disadvantages is that it can increase the complexity of the system. For example, as more feedback is given, there will be more requirements needed to satisfy the demand. As a result, the scope may expand beyond the original plan.

Thus, a prototype model should only be used when the system has a very high amount of interaction with end users such as online system where reviews and feedback will be consistently done by customers.

Lastly, agile model is a type of incremental model where for each phases a software is being build and tested. Overtime, more functionality will then slowly be added into the software until the system is completed. For example, SCRUM is one of the most popular agile framework.



There are three roles in this framework. Firstly, the product owner determine what need to be done by creating a wish list called a product backlog. Next, the development team then choose some of things that need to be done on the wish list and decides on how to implement these things. After that, the scrum master will ensure to keep the development team focused on their goal. Lastly, at the end of each sprint which is one round of work, the team will have to complete the task they chosen from the wish list and repeat the cycle again.

The advantage of using an agile model include is that the working software is developed frequently and the team is flexible to any late changes at any time of time period of the project. However, one of the disadvantage is that only senior programmers can handle the development process because this would require many experience as decision has to made fast. As a result, it is not recommended for newbie to be in charge of this model.

Thus, an agile model should only be used when the sponsor or clients are always changing their requirement based on the feedback given from the customer and also changes can be easily made since it is flexible due to the frequency of new functionality being produced.

**References**

**Kenji Leong**

<http://istqbexamcertification.com/what-are-the-software-development-models/>

<http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/>

<http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/>

<https://www.scrumalliance.org/why-scrum#why-its-called-scrum>

**Jia Cheng**

**Caleb**