Basic stages of software development life cycle.

1. Planning and requirement analysis.

Each software development life cycle model always starts with analysis and discuss the requirements for the final product in order to have a detailed definition of the system requirements. At the same time to make sure that all the process participants have clearly understood the tasks and how every requirement is going to be implemented.

1. Designing project architecture.

The second phase of the software development life cycle, the developers will be designing the architecture. There will be technical question that may appear on this stage that are discussed by the participants including the customers. The developers have to share the team load, there will be some limitations due to the budget and lastly developers have to plan the time frames in order to make it before the project is due.

1. Development and programming

After the requirements have approved, it’s time to proceed to next stage which is the actual development. Programmers will start with the coding while keeping in mind of the defined requirements.

1. Testing

The testing phase includes debugging process. If the developers have any flaws during development will be detected here, documented and passed back to the developers to fix it. This process will repeat until all the critical issues are fixed and software workflow is stable.

1. Deployment

When the program is finalized and there’s no critical issues, it’s time to launch. After the new program version have release, the tech support team joins.

**Waterfall model**

Waterfall model was the first process model to be introduced and it’s also referred as linear-sequential life cycle model. In waterfall model each stage must be completed in order to proceed to the next stage. This type of software development model is mostly used for project that is small and there aren’t any uncertain requirements. At the end of each stage, a review will take place to determine if the project is on the right path and whether to continue or discontinue the project. In this model software testing only starts after the development is complete so that the stages do not overlap.

When to use the waterfall model?

This model can only be use when the requirement is very well known, clear and fixed at the same time tt does not require a lot of interaction with customer during the development of the product but once the product is ready then it can be demoed to the end users. Once the product is fully developed and if any failure occurs then the cost of fixing any issues will be very high.

The advantage of waterfall model is easy to manage due to rigidity of the model in each stage has specific deliverables and a review process.

The disadvantage of waterfall model is not suitable for project that requirements that have the high risk of changing as once the application is in the testing stage, it’s very difficult to go back and change something that was not well-thought out in the concept stage.

This model isn’t suitable for the hotel as Mr and Mrs Wang expect to have an update on the project progress from time to time.

V-model

V-model stands for Verification and Validation model. It’s the same as waterfall model but it’s in the V-shaped life cycle is a sequential path of execution of processes and each stage must be completed before moving on to the new stage. But in this model before the development start a system test plan is created. The test plan focuses more on meeting the function of the requirements.

The high-level design stages focus more on the architecture and design. It also provides an overview of solution, platform, system, product and service. At this stage, an integration test plan is will be created, to test the pieces of the software system if it’s able to work together.

The low-level design stages are the actual software components are designed, it defines the actual logic for the whole components of the system. Class diagram and all other methods including relation between classes come under low-level design and at this stage, component test is created.

The implementation stage is all coding take place. Once coding is done, execution continues up the right side of the V where the test plan developed earlier are apply.

As for coding is the bottom part of the V-shape model. The model design is converted into code by developers. Unit testing is function by the developers on the code as it’s written by them.

When to use the V-model?

When the project is small to medium sized where requirements are clearly defined and fixed.

When customer have a very high confidence will choose the V-shaped model as there’s no prototyped and it’s a high risk involved in meeting customer expectations.

The advantages of using V-model as it can found the defects at the early stages hence, avoids the downward flow of the defects. Testing actives like planning, test designing happens before coding. This saves a lot of time hence, there’s a higher chance of success over the waterfall model.

As for the disadvantage of V-model is if there’s any adjustment while doing the project halfway, the test documents along with requirement documents must updated. Software is developed during the implementation stage hence, there’s no early prototypes of software that are produced.

This model is suitable for the hotel as Mr and Mrs Wang expect to have an update on the project progress from time to time.

Incremental model

Incremental model is requirement will be branched out into different types of builds. Multiple development will be making the life cycle a “multi-watefall” cycle. Cycles are divided into smaller so that it will be easier to manage modules. In this model, each module pass through the requirements, design, implementation and testing stages. Developers have to start working on the software early during the software life cycle so that the application can be demo to the customers. It will release the add on function of the module to the previous release and this process will continue till the complete system is achieved.

When to use the incremental model?

When there’s a need to get the product to the market early but there’re some high-risk features and goals. Hence, resources with needed skill set are not available.

The advantage of incremental model is this model is more flexible in a way that is less costly to change scope and requirements. At the same time, it’s also easier to test, debug and manages risk as the risk will be identified and handle during iteration.

For the disadvantage of incremental model, the total cost is higher than waterfall hence, it needs a good planning and design not only that it also needs a clear and complete definition of the system before it can be break down and built incrementally.

This model isn’t suitable for the hotel as Mr and Mrs Wang expect to have an update on the project progress from time to time.