**ASSIGNMENT – 4**

**Q1.Explain the various phases of SDLC.**

Ans. There are following six phases in every Software development life cycle model:

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

**1)Requirement gathering and analysis:-** Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system?

**2)** **Design**:- In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

In this phase the testers comes up with the [**Test strategy**](http://istqbexamcertification.com/what-are-the-test-approaches-or-strategies-in-software-testing/), where they mention what to test, how to test.

**3)** **Implementation and coding:-** On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer.

**4)** **Testing**:- After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.

**5)** **Deployment**:-  After successful testing the product is delivered / deployed to the customer for their use.

**6)** **Maintenance**:- Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as

**Q2.Explain the various phases of STLC.**

Ans. Below are the phases of STLC:

1. Requirements phase
2. Planning Phase
3. Analysis phase
4. Design Phase
5. Implementation Phase
6. Execution Phase
7. Conclusion Phase
8. Closure Phase

**#1.**[**Requirement Phase**](http://www.softwaretestinghelp.com/rview-srs-document-and-create-test-scenarios-software-testing-training-course-day-2/)**:-** During this phase of STLC, analyze and study the requirements. During this phase of STLC, analyze and study the requirements.

**#2. Planning Phase:-** Test planning is the first step of the testing process. In this phase we identify the activities and resources which would help to meet the testing objectives.

**#3. Analysis Phase:-** This STLC phase defines “WHAT” to be tested. We basically identify the test conditions through the requirements document, product risks and other test basis. There are various factors which effect the identification of test conditions:-

– Levels and depth of testing  
– Complexity of the product  
– Product and project risks  
– Software development life cycle involved.  
– Test management  
– Skills and knowledge of the team.  
– Availability of the stakeholders.

**#4. Design Phase:-** This phase defines “HOW” to test. This phase involves the following tasks:

– Detail the test condition. Break down the test conditions into multiple sub conditions to increase coverage.  
– Identify and get the test data  
– Identify and set up the test environment.  
– Create the requirement traceability metrics  
– Create the test coverage metrics.

**#5. Implementation Phase**:- The major task in this STLC phase is of creation of the detailed test cases. Before finalizing the test case, it is important to carry out the review to ensure the correctness of the test cases.

**#6. Execution Phase:- :-**  This is the Software Testing Life Cycle phase where the actual execution takes place. But before you start your execution, make sure that your entry criterion is met.

**#7. Conclusion Phase:-** This STLC phase concentrates on the exit criteria and reporting. Depending on your project and stakeholders choice, you can decide on reporting whether you want to send out a daily report of weekly report etc.

**#8. Closure Phase:-** Tasks for the closure activities include the following:

– Check for the completion of the test. Whether all the test cases are executed or mitigated deliberately. Check there are no severity 1 defects opened.  
– Do lessons learnt meeting and create lessons learnt document. ( Include what went well, where are the scope of improvements and what can be improved)

**Q3.Explain Traditional Software Development Model.**

Ans. **1)** **Waterfall Model:-** The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**.  It is very simple to understand and use.

**Advantages of waterfall model:**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are very well understood.

**Disadvantages of waterfall model:**

* Once an application is in the [testing](http://istqbexamcertification.com/what-is-a-software-testing/) stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

**2) V-Model:-** V- model means Verification and Validation model. Just like the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/), the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. **V-Model**is one of the [many software development models](http://istqbexamcertification.com/what-are-the-software-development-models/).

Testing of the product is planned in parallel with a corresponding phase of development in **V-model**.

**Advantages of V-model:**

* Simple and easy to use.
* Testing activities like planning, [test designing](http://istqbexamcertification.com/what-is-test-design-or-how-to-specify-test-cases/) happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
* Proactive defect tracking – that is defects are found at early stage.
* Avoids the downward flow of the defects.
* Works well for small projects where requirements are easily understood.

**Disadvantages of V-model:**

* Very rigid and least flexible.
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* If any changes happen in midway, then the test documents along with requirement documents has to be updated.

**3) RAD Model:-** RAD model is Rapid Application Development model. It is a type of [**i**ncremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype.

**Advantages of the RAD model:**

* Reduced development time.
* Increases reusability of components
* Quick initial reviews occur
* Encourages customer feedback
* Integration from very beginning solves a lot of [integration issues](http://istqbexamcertification.com/what-is-system-integration-testing/).

**Disadvantages of RAD model:**

* Depends on strong team and individual performances for identifying business requirements.
* Only system that can be modularized can be built using RAD
* Requires highly skilled developers/designers.
* High dependency on modeling skills

**4) Prototype Model:-** The basic idea in **Prototype model** is that instead of freezing the requirements before a design or coding can proceed. This prototype is developed based on the currently known requirements. Prototype model is a [software development model](http://istqbexamcertification.com/what-are-the-software-development-models/).

**Advantages of Prototype model:**

* Users are actively involved in the development
* Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
* Errors can be detected much earlier.
* Quicker user feedback is available leading to better solutions.
* Missing functionality can be identified easily

**Disadvantages of Prototype model:**

* Leads to implementing and then repairing way of building systems.
* Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

**5) Spiral Model:-** The spiral model is similar to the [incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/), with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.

 A software project repeatedly passes through these phases in iterations (called Spirals in this model).

**Advantages of Spiral model:**

* High amount of risk analysis hence, avoidance of Risk is enhanced.
* Good for large and mission-critical projects.
* Strong approval and documentation control.
* Additional Functionality can be added at a later date.
* Software is produced early in the [software life cycle](http://istqbexamcertification.com/what-are-the-software-development-life-cycle-phases/).

**Disadvantages of Spiral model:**

* Can be a costly model to use.
* Risk analysis requires highly specific expertise.
* Project’s success is highly dependent on the risk analysis phase.
* Doesn’t work well for smaller projects.

**Q4.Explain Prototype Software Development Model.**

Ans. The basic idea in **Prototype model** is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. Prototype model is a [software development model](http://istqbexamcertification.com/what-are-the-software-development-models/).

Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.



**Advantages of Prototype model:**

* Users are actively involved in the development
* Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
* Errors can be detected much earlier.
* Quicker user feedback is available leading to better solutions.
* Missing functionality can be identified easily

**Disadvantages of Prototype model:**

* Leads to implementing and then repairing way of building systems.
* Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

**When to use Prototype model:**

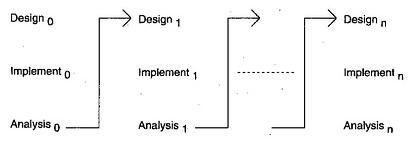
* Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
* Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

**Q5.Explain Iterative Enhancement life cycle Model.**

Ans. An iterative [life cycle model](http://istqbexamcertification.com/what-are-the-software-development-models/) does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements.

This process is then repeated, producing a new version of the software for each cycle of the model.

**Diagram of the iterative model**:-



**Advantages of Iterative model:**

* In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
* In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
* In iterative model less time is spent on documenting and more time is given for designing.

**Disadvantages of Iterative model:**

* Each phase of an iteration is rigid with no overlaps
* Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

**When to use iterative model:**

* Requirements of the complete system are clearly defined and understood.
* When the project is big.
* Major requirements must be defined; however, some details can evolve with time.

**Q6.Explain the various types of Maintenance Phase.**

Ans. There are four types of maintenance namely, corrective, adaptive, perfective, and preventive.

**Corrective Maintenance**:- Corrective maintenance is concerned with fixing errors that are observed when the software is in use.

**Adaptive Maintenance:-** Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system.

**Perfective Maintenance :-**Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults.

**Preventive Maintenance:-**Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability.

**Q7.What is the difference between High Level design an Low Level design?**

Ans.1) HLD is nothing but, heirarchy of all possible functionality   
ass module in the functionality. eg. flowgraph, flowchart, data structure.

LLD is nothing but, static logic bet. every sub module.  
 eg. ER Diagram, Algorithm

2) HLD-level Design gives the overall System Design in terms   
of Functional Architecture and Database design.it will show   
system using use case, class diagram, sequence deployment   
diagram.

LLD-During the detailed phase, the view of the application   
developed during the high level design is broken down into   
modules and programs.

3) High level Design: to test the function modules ,operators  
etc.

Low level design means to test the functions, operatus, sub modules this is called the LLd.

4) The entry criteria for this will be the HLD document.

The exit criteria will the program specification and unit test plan (LLD).

**Q8.Explain V-Shaped Model.**

Ans**.** The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model.

V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase.

**V- Model design:-**

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the .V. and Validation phases on the other side. Coding phase joins the two sides of the V-Model.

## Verification Phases:-

Following are the Verification phases in V-Model:

* **Business Requirement Analysis:** This is the first phase in the development cycle where the product requirements are understood from the customer perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement.
* **System Design:** Once you have the clear and detailed product requirements, its time to design the complete system. System design would comprise of understanding and detailing the complete hardware and communication setup for the product under development.
* **Architectural Design:** Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken.
* **Module Design:** In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems.

## Coding Phase:-

## The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements. The coding is performed based on the coding guidelines and standards.

## Validation Phases:-

Following are the Validation phases in V-Model:

* **Unit Testing:** Unit tests designed in the module design phase are executed on the code during this validation phase.
* **Integration Testing:** Integration testing is associated with the architectural design phase.
* **System Testing:** System testing is directly associated with the System design phase.
* **Acceptance Testing:** Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment.

## V- Model Application:-

V- Model application is almost same as waterfall model, as both the models are of sequential type.

* Requirements are well defined, clearly documented and fixed.
* Product definition is stable.
* Technology is not dynamic and is well understood by the project team.
* There are no ambiguous or undefined requirements
* The project is short.

## V- Model Pros and Cons:-

The following table lists out the pros and cons of V-Model:

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * This is a highly disciplined model and Phases are completed one at a time. * Works well for smaller projects where requirements are very well understood. * Simple and easy to understand and use. * Easy to manage due to the rigidity of the model . each phase has specific deliverables and a review process. | * High risk and uncertainty. * Not a good model for complex and object-oriented projects. * Poor model for long and ongoing projects. * Not suitable for the projects where requirements are at a moderate to high risk of changing. |

**Q9**.**What is the difference between functional specification and Business requirement specification.**

### **Ans. 1)** Functional requirements are prepared by the system analyst. Its all about the functionality for the software prepared by the system analyst. Business Requirement specification is prepared by the business analysts. It contains the business logic which is going to execute into the software.

### **2)** Functional specification (FRS - Functional Requirement Specification) is one which is prepared by Project Analyst by using Business Requirement Specifications. Main purpose of this is to understand the functionality of the project.

### Business Requirement Specification is given by client, which describes the Requirements for Business (Project).

**Q10.What is the difference between Testing and debugging.**

Ans. Main Differences between Testing and Debugging:- 

|  |  |
| --- | --- |
| **Testing** | **Debugging** |
| Testing is the process of detecting an error in software. | Debugging is the process of removal of that error from the software. |
| Testing is the proactive process to catch the bugs earlier. | Debugging is always done after bug is found due to testing activity. |
| Testing is done by tester in testing phase. | Debugging is done by developer in development phase. |