Assignment-4

Ques- what are various phases of SDLC?

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

1Requirement gathering and analysis

2Design

3Implementation or coding

4Testing

5Deployment

6Maintenance

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? How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

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. The system design specifications serve as input for the next phase of the model.

In this phase the testers comes up with the test strategy , where they mention what to test, how to test.

3)  Implementation / 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. This is the longest phase of the software development life cycle.

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.

As soon as the product is given to the customers they will first do beta testing. If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the bugs are fixed then the final deployment will happen.

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 maintenance.

Ques- what are 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/" \o "SRS planning):

During this phase of STLC, analyze and study the requirements. Have brain storming sessions with other teams and try to find out whether the requirements are testable or not. This phase helps to identify the scope of the testing. If any feature is not testable, communicate it during this phase so that the mitigation strategy can be planned.

#2. [Planning Phase](http://www.softwaretestinghelp.com/how-to-write-test-plan-document-software-testing-training-day3/" \o "Test planning phase):

In practical scenarios, 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. During planning we also try to identify the metrics, the method of gathering and tracking those metrics.

**#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. The test condition should be traceable back to the requirement. 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. Prioritize the test cases also identify which test case will become part of the regression suite. Before finalizing the test case, It is important to carry out the review to ensure the correctness of the test cases. Also don’t forget to take the sign off of the test cases before actual execution starts. If your project involves automation, identify the candidate test cases for automation and proceed for scripting the test cases. Don’t forget to review them!

#6. [Execution Phase](http://www.softwaretestinghelp.com/test-execution-software-testing-qa-training-on-a-live-project-day-5/" \o "Test execution phase):

As the name suggests, 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. Execute the test cases, log defects in case of any discrepancy. Simultaneously fill your traceability metrics to track your progress.

#7. [Conclusion Phase](http://www.softwaretestinghelp.com/bug-tracking-test-metrics-and-test-sign-off-free-qa-training-day-6/" \o "Test 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. There are different types of reports ( DSR – Daily status report, WSR – Weekly status reports) which you can send, but the important point is, the content of the report changes and depends upon whom you are sending your reports. If Project managers belong to testing background then they are more interested in the technical aspect of the project, so include the technical things in your report ( number of test cases passed, failed, defects raised, severity 1 defects etc.). But if you are reporting to upper stakeholders, they might not be interested in the technical things so report them about the risks that have been mitigated through the testing.

#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.

Ques- Explain traditional software development model?

Ans- Traditional methodologies are characterized by a sequential series of steps like requirement definition, planning, building, testing and deployment. First, the client requirements are carefully documented to the fullest extent. Then, the general architecture of the software is visualized and the actual coding commences. Then comes the various types of testing and the final deployment. The basic idea here is the detailed visualization of the finished project before the building starts, and working one’s way through to the visualized finished structure.

Ques- 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/). By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.  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.

The prototype are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

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
* Confusing or difficult functions can be identified  
  Requirements validation, Quick implementation of, incomplete, but  
  functional, application.

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.
* Incomplete application may cause application not to be used as the  
  full system was designed  
  Incomplete or inadequate problem analysis.

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.

Ques- Explain iterative enhancement life cycle model?

Ans- An iterative [life cycle model](http://istqbexamcertification.com/what-are-the-software-development-models/" \o "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.

Advantages of Iterative model:

* In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
* 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.

Ques- Explain various type of maintenance phases?

Ans- There are four types of maintenance, namely, corrective, adaptive, perfective, and preventive. Corrective maintenance is concerned with fixing errors that are observed when the software is in use. Adaptive maintenance is concerned with the change in the software that takes place to make the software adaptable to new environment such as to run the software on a new [operating system](http://ecomputernotes.com/fundamental/disk-operating-system/what-is-operating-system" \t "_self" \o "Operating System is software that works as an interface between a user and the computer hardware.). Perfective maintenance is concerned with the change in the software that occurs while adding new functionalities in the software. Preventive maintenance involves implementing changes to prevent the occurrence of errors. The distribution of types of maintenance by type and by percentage of time consumed.

Ques- what is difference between high level design and low level design?

Ans-High level says what … what does the solution look like; components, capabilities, information flows, use cases, and how do they all interrelate; clearly define the abstract solution.

Low level says how … define specific solutions; how do I assemble a system of real components that implements all the whats and optimally satisfies all the whys.

High – level Design gives the overall System Design in terms of Functional Architecture and Database design. It designs the over all architecture of the entire system from main module to all sub module. This is very useful for the developers to understand the flow of the system. In this phase design team, review team (testers) and customers plays a major role. For this the entry criteria are the requirement document that is SRS. And the exit criteria will be HLD, projects standards, the functional design documents, and the database design document. Further, High level deign gives the overview of the development of product. In other words how the program is going to be divided into functions, modules, subdivision etc.  
  
Low – Level Design (LLD): During the detailed phase, the view of the application developed during the high level design is broken down into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created. The entry criteria for this will be the HLD document. And the exit criteria will the program specification and unit test plan (LLD).

Ques- Explain V shaped model?

Ans- 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/" \o "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/" \o "What is Test design?) 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.

**When to use the V-model:**

* The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
* The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

Ques- What is difference between testing and debugging?

Ans- Testing activity is carried down by a team of testers, in order to find the defect in the software. Test engineers run their tests on the piece of software and if they encounter any defect (i.e. actual results don't match expected results), they report it to the development team. Along with the nature of defect, testers also have to report at what point the defect occurred and what happened due the occurrence of that defect. All this information will be used by development team to DEBUG the defect. (Read more about [Defect Life Cycle](http://www.ianswer4u.com/2016/01/what-is-software-bug-defect-life-cycle.html))

|  |
| --- |
|  |
| Testing V/s Debugging |

Debugging is the activity which is carried out by the development team (or developer), after getting the test report from the testing team about defect(s) (you may note defects can also be reports by the client). The developer then tries to find the cause of the defect, in this quest he may need to go through lines of code and find which part of code in causing that defect. After finding out the bug, he tries to modify that portion of code and then he rechecks if the defect has been finally removed.