|  |
| --- |
| 1. What is software testing?  * Testing the functionality of the software, to find any defect in the software, it's called software testing. |
| 1. What are the principles of software testing?  * it is 7 principles of software testing.  1. Start software testing at early stages.   Means from the beginning when you get the requirements.  2. Test the software in order to find the defects.  3. Highly impossible to give bug-free software to the customer.    4. Should not do Exhaustive testing.  This means we should not use the same type of data for testing every time.  5. Testing is context-based.  This means deciding what types of testing should be conducted based on the type of  applications.  6. We should follow the concept of the pesticide paradox.  This means, if you are executing the cases for a longer run, they won't find any defects.  We have to keep updating test cases in every cycle/release to find more defects.  7. We should follow defect clustering.  This means some of the modules contain most of the defects.  By experience, we can identify such risky modules. 80% of the problems are found in 20%  Of the modules. |
| 1. What is a project and product?  * If a software application is developed for a specific customer based on the requirement, then it is called a project. * If a software application is developed for multiple customers based on market requirements, then it is called a product. |
| 1. What is a defect?  * The developer during testing finds the difference between actual behavior and accepted behavior that leads to defect. |
| 1. What is a bug?  * Tester during testing finds mismatch application between actual result and accepted result that leads to bug. |
| 1. What is an error?  * Mistake in the program while writing code because developers are not able to either run or compile the code that is called error. |
| 1. What is a failure?  * Once the software is ready to verify by the tester and finally end-user or customer are phasing some issue in the production that is called failure. |
| 1. What is SDLC (software development life cycle)?  * It’s a step-by-step procedure to develop any software it’s called SDLC.      1. Requirements Collection:  * Done by business analysts and product analysts * Gathering requirements * Translates business language into software language  1. Feasibility study:  * Done by software team consisting of project managers, business analysts, architects, finance, HR, developers but not testers. * Architect – is the person who tells whether the product can be developed and if yes, then which technology is best suited to develop it. * Here we check for, - technical feasibility   Financial feasibility  Resource feasibility   1. Design:  * There are 2 stages in design, - HLD – High-level design   LLD – Low-level design   * HLD – gives the architecture of the software product to be developed and is done to be developed and is done by architects and senior developers. * LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.  1. Coding / Programming:  * Done by all developers – seniors, juniors, freshers * This is the process where we start building the software and start writing the code for the product.  1. Testing:  * Done by test engineers * It is the process of checking for all defects and rectifying them.  1. Installation:  * Done by installation engineers * To install the product at a client’s place for use after the software has been developed and tested.  1. Maintenance:  * Here as the customer uses the product, he finds certain bugs and defects and sends the product back for error correction and bug fixing. * Bug fixing takes place * Minor changes like adding, deleting, or modifying any small feature in the software product. |
| 1. What are service-based companies and product-based companies?  * Service-based companies: * They provide service and develop software for other companies * They provide software that is specified as per the client company’s requirement and never keep the code of the developed product and does not provide the software to any other company other than the client company. * Ex – Wipro, Infosys, TCS * Product-based companies: * The develop software products and sell them to many companies which may need the software and make profits for themselves. * They are the sole owners of the product they develop and the code used and sell it to other companies which may need the software. * Ex – Oracle, Microsoft |
| 1. What is the waterfall model?  * It is a traditional or basic model. * It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards. * Requirement’s collection – feasibility study – design – coding – testing – installation – maintenance |
| 1. What are the drawbacks of the waterfall model?  * In the waterfall model, backtracking is not possible. We cannot back and change requirements once the design stage is reached. Thus, the requirements are freezed once the design of the software product is started. * The major drawback of the waterfall model – testing is a small phase that is done after coding. The requirement is not tested, design is not tested, if there is a bug in the requirement, it goes on till the end and leads to a lot of reworks. |
| 1. What is the agile model / agile process / agile metrology?  * The agile model is an iterative and incremental approach where requirements are kept on changes as a company, we should be flexible to handle those requirements and develop those requirements, test those requirements and finally give the working software to the customer within short periods of time. |
| 1. What are the advantages of the agile model?  * Requirement changes are allowed in any stage of development. * Releases will be very fast. * Customer no need to wait for a long time. * Good communication between the team. * It is very easy to learn. |
| 1. What are the disadvantages of the waterfall model?  * Less focus on design and documentation. |
| 1. What is manual testing?  * Testing the functionality of the application repeatedly or again and again manually in order to find the defect in the software according to the customer requirement is called manual testing. |
| 1. Why do we do manual testing / what is the importance of manual testing?  * In order to give quality software to the customer when we perform manually. * Easy to learn for the tester. * It does not require programming knowledge while using manual testing. * The stability of the application can be checked only from manual testing. * Tester interacts with software as a real user so that they are able to discover usability and user interface issue. |
| 1. What are the disadvantages of manual testing? |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |