1. **what is SDLC and different phases in SDLC?**

Ans: Software development life cycle (SDLC) is a process to develop the application

**Different phases like:**

**Requirement Analysis and planning :** Senior team members analyze the requirements/input given by customers/business users. They will check whether the requirement is feasible or not (can be done or not). They also identify the risks associated with project.

Note: this high level requirements will be written in BRD (Business Requirement document) by Business Analyst

Define/Design : in the define stage Business Analyst define more details about requirements (which are in BRD) in the form of SRS (software requirement specification) or Use Case diagram.

As part of design,

Senior Developers write High Level Design Document (HLD)

Developers write Low Level Design Document (LLD)

Seniors Tester write Test Planning document

Implementation/Development: Developers write the code for the requirements

Testers write test cases as per SRS

Testing : Execute the test cases what we prepared in previous stage

Deployment : Release the tested code to production

Maintenance : Support team monitoring the system that is running in production

**SDLC :**

* *The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process.*
* *SDLC is a structure followed by a development team within the software organization.*
* *It consists of a detailed plan describing how to develop, maintain and replace specific software.*
* *The life cycle defines a methodology for improving the quality of software and the overall development process.*

***SDLC consists of following activities:***

***Planning:*** *The most important parts of software development, requirement gathering or requirement analysis are usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.*

***Implementation:*** *The software engineers start writing the code according to the client's requirements.*

***Testing:*** *This is the process of finding defects or bugs in the created software.*

***Documentation:*** *Every step in the project is documented for future reference and for the improvement of the software in the development process. The design documentation may include writing the application programming interface (API).*

***Deployment and maintenance:*** *The software is deployed after it has been approved for release.*

***Maintaining:*** *Software maintenance is done for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to create the initial development of the software.*

1. **what is waterfal in SDLC?**
2. **what is the process in agile model**

[**http://www.tutorialspoint.com//sdlc/sdlc\_agile\_model.htm**](http://www.tutorialspoint.com//sdlc/sdlc_agile_model.htm)

1. **what is scrum methodology**

[**http://www.guru99.com/agile-scrum-extreme-testing.html**](http://www.guru99.com/agile-scrum-extreme-testing.html)

1. **what is daily standup meeting and what we discuss**

[**http://searchsoftwarequality.techtarget.com/definition/daily-stand-up-meeting**](http://searchsoftwarequality.techtarget.com/definition/daily-stand-up-meeting)

1. **what is user story and tasks in user story**

[**https://www.mountaingoatsoftware.com/agile/user-stories**](https://www.mountaingoatsoftware.com/agile/user-stories)

[**http://searchsoftwarequality.techtarget.com/definition/user-story**](http://searchsoftwarequality.techtarget.com/definition/user-story)

[**https://www.mountaingoatsoftware.com/blog/the-difference-between-a-story-and-a-task**](https://www.mountaingoatsoftware.com/blog/the-difference-between-a-story-and-a-task)

1. **what is sprint planning and spring retro**

[**http://www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/**](http://www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/)

**timebox ;** [**https://www.agilealliance.org/glossary/timebox/**](https://www.agilealliance.org/glossary/timebox/)

**sprint retrospective :** [**https://www.scrumalliance.org/community/articles/2014/april/key-elements-of-sprint-retrospective**](https://www.scrumalliance.org/community/articles/2014/april/key-elements-of-sprint-retrospective)

1. **what is burndown chart and velocity**

[**https://www.mountaingoatsoftware.com/agile/scrum/release-burndown**](https://www.mountaingoatsoftware.com/agile/scrum/release-burndown)

[**http://whatis.techtarget.com/definition/Agile-velocity**](http://whatis.techtarget.com/definition/Agile-velocity)

1. **what is product backlog item and sprint backlog items**

[**http://www.scrum-institute.org/The\_Scrum\_Product\_Backlog.php**](http://www.scrum-institute.org/The_Scrum_Product_Backlog.php)

[**https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog**](https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog)

[**https://www.quora.com/Agile-Software-Development-What-is-the-difference-between-the-Product-Backlog-and-Sprint-Backlog**](https://www.quora.com/Agile-Software-Development-What-is-the-difference-between-the-Product-Backlog-and-Sprint-Backlog)

[**https://www.scruminc.com/product-backlog-item-pbi/**](https://www.scruminc.com/product-backlog-item-pbi/)

1. **what is user acceptance criteria test cases**
2. **what is v model?**

[**http://www.tutorialspoint.com/sdlc/sdlc\_v\_model.htm**](http://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)

1. **what is STLC?**

[**http://www.softwaretestinghelp.com/what-is-software-testing-life-cycle-stlc/**](http://www.softwaretestinghelp.com/what-is-software-testing-life-cycle-stlc/)

[**http://softwaretestingfundamentals.com/software-testing-life-cycle/**](http://softwaretestingfundamentals.com/software-testing-life-cycle/)

1. **what is defect?**

[**http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/)

1. **how to arise a defect and what we specify while logging defect?**

[**http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/)

[**http://istqbexamcertification.com/what-is-incident-logging-or-how-to-log-an-incident-in-software-testing/**](http://istqbexamcertification.com/what-is-incident-logging-or-how-to-log-an-incident-in-software-testing/)

1. **defect lifecycle**

[**http://www.tutorialspoint.com/software\_testing\_dictionary/defect\_life\_cycle.htm**](http://www.tutorialspoint.com/software_testing_dictionary/defect_life_cycle.htm)

1. **Different types of testing?**

[**http://www.softwaretestinghelp.com/types-of-software-testing/**](http://www.softwaretestinghelp.com/types-of-software-testing/)

1. **when do we use regression testing?**

[**http://www.softwaretestinghelp.com/regression-testing-tools-and-methods/**](http://www.softwaretestinghelp.com/regression-testing-tools-and-methods/)

1. **when do we use integration testing?**

[**http://searchsoftwarequality.techtarget.com/definition/integration-testing**](http://searchsoftwarequality.techtarget.com/definition/integration-testing)

1. **when do we use smoke testing and sanity testing?**

[**http://www.guru99.com/smoke-sanity-testing.html**](http://www.guru99.com/smoke-sanity-testing.html)

[**http://testingbasicinterviewquestions.blogspot.com/2012/09/smoke-testing-example-easy-and-simple.html**](http://testingbasicinterviewquestions.blogspot.com/2012/09/smoke-testing-example-easy-and-simple.html)

1. **what is unit testing?**

[**http://searchsoftwarequality.techtarget.com/definition/unit-testing**](http://searchsoftwarequality.techtarget.com/definition/unit-testing)

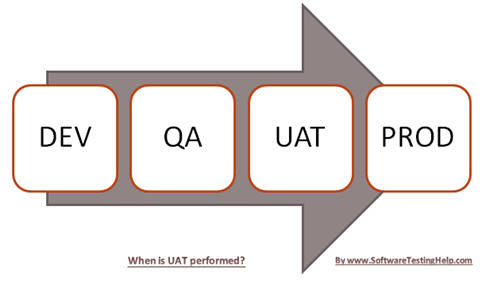
1. **what is UAT?**

**UAT – User Acceptance Testing**

UAT means testing a software by the user/client to determine whether it can be accepted or not – the definition.

**When is it performed?**

This is typically the last step before the product goes live or before the delivery of the product is accepted. UAT is after the product itself is thoroughly tested (i.e [after system testing](http://www.softwaretestinghelp.com/system-testing/)).

**[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/when-UAT-performed.jpg)**

**Who performs UAT?**

Users or client – This could be either someone who is buying a product (in the case of commercial software) or someone who has had a software custom built through a software service provider or the end user if the software is made available to them ahead of time and when their feedback is sought.

http://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/

1. **what is alpha and beta testing?**



**Alpha testing** is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public.

The focus of this testing is to simulate real users by using blackbox and whitebox techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually the testers are internal employees of the organization.

To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing.

**Beta Testing** of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external user acceptance testing.

 Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in real time environment.

**http://www.guru99.com/alpha-beta-testing-demystified.html**

1. **when do we use white box testing and block box testing?**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Black Box Testing** | **White Box Testing** |
| Definition | Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester | White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. |
| Levels Applicable To | Mainly applicable to higher levels of testing:[Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/)  [System Testing](http://softwaretestingfundamentals.com/system-testing/) | Mainly applicable to lower levels of testing:[Unit Testing](http://softwaretestingfundamentals.com/unit-testing/)  [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/) |

Seriously, white-box testing (i.e. testing the internals of code) should ideally be done with unit tests by the developer who wrote the code. Unit tests would be built up over time, and part of the build process so we don't waste the poor tester's time with code we know doesn't work as it should. Unit testing becomes more important the smaller your team is--particularly because you don't have an army of testers to shake problems out.

Black-box testing (i.e. testing through the user/system interface) is typically what most testers do.

All testing needs to be prioritized on how critical a function is for the finished product. If the mission is to provide a tool to do X and the product doesn't do X, that's a big problem.

[**http://programmers.stackexchange.com/questions/27491/black-box-or-white-box-testing-which-do-you-do-first**](http://programmers.stackexchange.com/questions/27491/black-box-or-white-box-testing-which-do-you-do-first)

[**http://neurontesting.blogspot.com/2012/05/functional-black-box-testing-and.html**](http://neurontesting.blogspot.com/2012/05/functional-black-box-testing-and.html)

1. **what we will do if we don’t have a time to test all stories?**
2. **what we will do if come across any severity issue before release day?**

In such situation, Firstly, I with my team members, will try to find out the cause for it. And with discussion with developers also, we will try to fix that out. And will work extra hours to get it fixed, and do testing again.  
  
And if the defect will take time to be resolved, then we will talk to Project Manager, that we can't send this release, as it has this defect, and we are working on it.  
  
Its better to send the release late, but it should be bug free. Whats the fun, if we send the release on time, but it has many defects, that the customer is not satisfied.

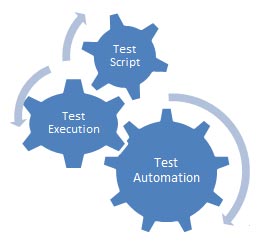
One thing we have to do is measure the severity and frequency of that defect .  
  
If it is having High Severity and low frequency of occurance then we can make the bug as known issue and move it to maintainance phase.  
  
If it is having high severity and high occurence then we have to fix the bug immediately and test.

1. **when do we use automation testing?**

#### Automated testing is the preferred option in the following areas/scenarios:

* **Regression Testing:** Here, automated testing is suitable because of frequent code changes and the ability to run the regressions in a timely manner.
* **Load Testing:** Automated testing is also the best way to complete the testing efficiently when it comes to load testing.
* **Repeated Execution:** Testing which requires the repeated execution of a task is best automated.
* **Performance Testing:** Similarly, testing which requires the simulation of thousands of concurrent users requires automation.

Keeping these factors in mind, you can find the best approach in any given testing situation and achieve quality output well within your budget and timeline.



**https://www.apicasystem.com/blog/automated-testing-vs-manual-testing/**

1. **what tester will do in each phase of SDLC?**

Testing Phase will start from Requirement stage itself

1. Requirement stage - PM,Tech.Lead, Testlead will review the document

2. Design Stage - PM or Test lease will review the document

3. coding Stage - Testlead will prepare Testplan, Tester will prepare testcases

4. Testing Stage - Tester will execute the testcase

The Role of a Tester in SDLC  
  
1. Tester prepares the Test cases, Test Scenarios  from the SRS  
2.  Using the script the tester performs different kinds of testing (Regression, Function)  
3. Tester Notes the results(pass/Fail)  
4. If Result=Fail then the scenario is raised in the Test director   
5. Once its fixed by the developer the tester performs a regression testing

[**http://www.geekinterview.com/question\_details/24591**](http://www.geekinterview.com/question_details/24591)

1. **difference between load and performance testing?**

Basically Load, Stress and Performance Testing are the same. However, Load testing is the test to check the users’ response time of number of users of any one scenario of the application whereas Performance Testing is the test to check the user response time for multiple scenario of the same application.

[**http://www.qaquestions.org/qa-interview-questions/**](http://www.qaquestions.org/qa-interview-questions/)

Performance testing is the testing, which is performed, to ascertain how the components of a system are performing, given a particular situation. Resource usage, scalability and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of software product.

**Performance Testing Goal:**

The primary goal of performance testing includes establishing the benchmark behaviour of the system. There are a number of industry-defined benchmarks, which should be met during performance testing.

Performance testing does not aim to find defects in the application, it address a little more critical task of testing the benchmark and standard set for the application. Accuracy and close monitoring of the performance and results of the test is the primary characteristic of performance testing.

**Example:**

For instance, you can test the application network performance on Connection Speed vs. Latency chart. Latency is the time difference between the data to reach from source to destination. Thus, a 70kb page would take not more than 15 seconds to load for a worst connection of 28.8kbps modem (latency=1000 milliseconds), while the page of same size would appear within 5 seconds, for the average connection of 256kbps DSL (latency=100 milliseconds). 1.5mbps T1 connection (latency=50 milliseconds) would have the performance benchmark set within 1 second to achieve this target.

For example, the time difference between the generation of request and acknowledgement of response should be in the range of x ms (milliseconds) and y ms, where x and y are standard digits. A successful performance testing should project most of the performance issues, which could be related to database, network, software, hardware etc…

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2011/07/Performance-Load-and-stress-testing.jpg)

### **2) Load Testing:**

Load testing is meant to test the system by constantly and steadily increasing the load on the system till the time it reaches the threshold limit. It is the simplest form of testing which employs the use of automation tools such as LoadRunner or any other good tools, which are available. Load testing is also famous by the names like**volume testing** and **endurance testing**.

The sole purpose of load testing is to assign the system the largest job it could possible handle to test the endurance and monitoring the results. An interesting fact is that sometimes the system is fed with empty task to determine the behaviour of system in zero-load situation.

**Load Testing Goal:**

The goals of load testing are to expose the defects in application related to buffer overflow, memory leaks and mismanagement of memory. Another target of load testing is to determine the upper limit of all the components of application like database, hardware and network etc… so that it could manage the anticipated load in future. The issues that would eventually come out as the result of load testing may include load balancing problems, bandwidth issues, capacity of the existing system etc…

**Example:**

For example, to check the email functionality of an application, it could be flooded with 1000 users at a time. Now, 1000 users can fire the email transactions (read, send, delete, forward, reply) in many different ways. If we take one transaction per user per hour, then it would be 1000 transactions per hour. By simulating 10 transactions/user, we could load test the email server by occupying it with 10000 transactions/hour.

**http://www.softwaretestinghelp.com/what-is-performance-testing-load-testing-stress-testing/**

1. **different types of non-functional testing types?**

In non-functional testing the quality characteristics of the component or system is tested. Non-functional refers to aspects of the software that may not be related to a specific function or user action such as scalability or security. Eg. How many people can log in at once? Non-functional testing is also performed at all levels like functional testing.

Non-functional testing includes:

* Reliability testing
* Usability testing
* Efficiency testing
* Maintainability testing
* Portability testing
* Baseline testing
* Compliance testing
* Documentation testing
* Endurance testing
* Load testing
* Performance testing
* Compatibility testing
* Security testing
* Scalability testing
* Volume testing
* Stress testing
* Recovery testing
* Internationalization testing and Localization testing

**http://istqbexamcertification.com/what-is-non-functional-testing-testing-of-software-product-characteristics/**

1. **what is test case?**

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

[**http://softwaretestingfundamentals.com/test-case/**](http://softwaretestingfundamentals.com/test-case/)

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

[**http://www.tutorialspoint.com/software\_testing\_dictionary/test\_case.htm**](http://www.tutorialspoint.com/software_testing_dictionary/test_case.htm)

1. **what is test plan/test strategy document**

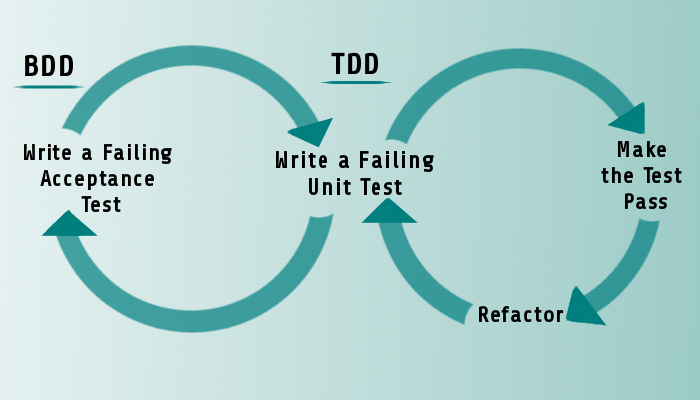
A **Test Plan Documents** the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from [test engineers](http://en.wikipedia.org/wiki/Test_engineer). There are three major elements that should be described in the test plan: Test Coverage, Test Methods, and Test Responsibilities. These are also used in a formal [test strategy](http://en.wikipedia.org/wiki/Test_strategy).  
The **Test Strategy Document** is a living document that is created in the project’s Requirements Definition phase, after the Requirements have been specified. The Test Strategy document describes the scope, approach, resources and schedule for the testing activities of the project. This includes defining what will be tested, who will perform testing, how testing will be managed, and the associated risks and contingencies. The Test Strategy document is maintained throughout the life of a project.

[**https://www.quora.com/What-is-the-difference-between-Test-Plan-document-and-Test-Strategy-document**](https://www.quora.com/What-is-the-difference-between-Test-Plan-document-and-Test-Strategy-document)

1. **what is TDD and BDD (cucumber framework)**

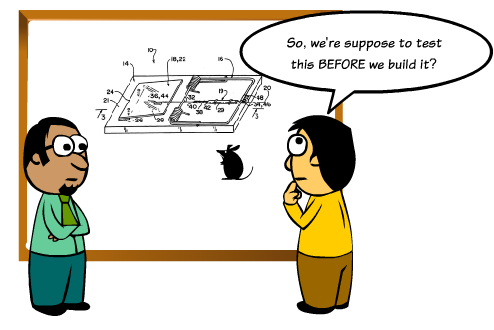
### **2. Behavior Driven Development**

In software engineering, **behavior-driven development** (abbreviated **BDD**) is a software development process based on test-driven development (TDD). Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design to provide software development and management teams with shared tools and a shared process to collaborate on software development.

[](http://www.seleniumframework.com/wp-content/uploads/2014/10/BDD.jpg)

### **3. Test Driven Development**

**Test-driven development** (**TDD**) is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards. Kent Beck, who is credited with having developed or ‘rediscovered’ the technique, stated in 2003 that TDD encourages simple designs and inspires confidence

[](http://www.seleniumframework.com/wp-content/uploads/2014/10/TestDrivenDevelopment_155FE5992.png)

[**http://www.seleniumframework.com/cucumber-2/make-a-case/atdd-tdd-bdd/**](http://www.seleniumframework.com/cucumber-2/make-a-case/atdd-tdd-bdd/)

1. **what is priority and severity in defect?**

There are two key things in defects of the [**software testing**](http://istqbexamcertification.com/what-is-a-software-testing/). They are:

1)     Severity

2)     Priority

What is the difference between Severity and Priority?

**1)  Severity**:

It is the extent to which the [**defect**](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) can affect the software. In other words it defines the impact that a given defect has on the system.**For example:** If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of  application crashing is severe. So the severity is high but priority is low.

Severity can be of following types:

* **Critical:**The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.
* **Major:**The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.
* **Moderate:**The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.
* **Minor:**The defect that does not result in the termination and does not damage the [**usability**](http://istqbexamcertification.com/what-is-usability-testing-in-software-and-its-benifits-to-end-user/)of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.
* **Cosmetic:**The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

**2)  Priority**:

Priority defines the order in which we should resolve a defect. Should   we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.**For example:**If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

Priority can be of following types:

* **Low:**The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect have been fixed.
* **Medium:**The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
* **High:**The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the  repair has been done.

**Few very important scenarios related to the severity and priority which are asked during the interview:**

**High Priority & High Severity**: An error which occurs on the basic functionality of the application and will not allow the user to use the system. (Eg. A site maintaining the student details, on saving record if it, doesn’t allow to save the record then this is high priority and high severity bug.)

**High Priority & Low Severity:** The spelling mistakes that happens on the cover page or heading or title of an application.

**High Severity & Low Priority:** An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.

**Low Priority and Low Severity:** Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title).

[**http://istqbexamcertification.com/what-is-the-difference-between-severity-and-priority/**](http://istqbexamcertification.com/what-is-the-difference-between-severity-and-priority/)

1. **how to estimate test cases?**

[**http://www.softwaretestingclass.com/software-estimation-techniques/**](http://www.softwaretestingclass.com/software-estimation-techniques/)

1. **what is most challenge defect u came across?**

**Q63. What is the biggest bug you have ever found?**

Ans: Well, there are many big defects I have found in various projects. For example, in the last project, on a page, there was a button called “More Information”. Once the user clicked that button, the system would open a new window (pop up). We could close the new window in 3 ways: -By clicking X at the top right corner of the page -By clicking “Close” button on the page -By pressing combination keys (Alt+F4) on the key board Although the combination key (Alt+F4) was not mentioned in the test case, I just wanted to try how the application reacts when Alt+F4 is pressed. Then I pressed Alt+F4. The result was a disaster-the application crashed (broke). The application disappeared from the computer monitor. Since it was the last day of testing for us, it brought chaos in our Managers, Leads and the whole teams. Finally, the developers disabled Alt+F4 as a temporary solution and the application went into production.

[**http://www.qaquestions.org/qa-interview-questions/**](http://www.qaquestions.org/qa-interview-questions/)

1. **how to deal the production defects?**

Ans: normally end user will report this issue.

we need to talk to them and reproduce the issue with test logins

Create defect in defect tool under the production release version

developers will fix the issue

we (QA) test the issue on production version code and release the fix to proudction after we verify

we have to create a defect on current sprint/release so that developer will add this code to the current sprint/release

---------------------------------------------------------------------------------------------------------------------------

<https://leantesting.com/resources/handle-software-defects/>

Test design review steps

1. **if we dont have time to test call test cases what we will do**
2. **how we learn the functionality of system?**
3. **what are the tools to manage defects/stories?**

[**http://www.softwaretestinghelp.com/popular-bug-tracking-software/**](http://www.softwaretestinghelp.com/popular-bug-tracking-software/)

1. **who will assign the work?**

**What do you do on your  first day of the work?**

Ans: (Note:  The person who is asking this question probably wants to know how the real scenario of a working person at work. It is a hard question for those who has never worked in a work place as a Software Tester.) On the first day, normally, we will be given a computer and support people will set up the User Name and Password for the computer.  If that is done already, then the QA Lead or QA Manager will give me a brief walk through of the documents (which documents are where), introduce to different team members (normally to the ones you will be working with).  Then your boss will ask you to step into work what needs to be done.  However, the first thing normally is, they will ask you to read the documents available for that project.

[**http://www.qaquestions.org/qa-interview-questions/**](http://www.qaquestions.org/qa-interview-questions/)

1. **types of test metrics we use normally**

[**http://www.guru99.com/software-testing-metrics-complete-tutorial.html**](http://www.guru99.com/software-testing-metrics-complete-tutorial.html)

1. **what is traceability matrix?**

[**http://www.guru99.com/traceability-matrix.html**](http://www.guru99.com/traceability-matrix.html)

1. **what are typical environments we have in projects**

[**http://www.softwaretestinghelp.com/what-is-actual-testing-process-in-practical-or-company-environment/**](http://www.softwaretestinghelp.com/what-is-actual-testing-process-in-practical-or-company-environment/)

[**http://sqa.stackexchange.com/questions/1532/list-and-role-of-different-types-of-testing-environments**](http://sqa.stackexchange.com/questions/1532/list-and-role-of-different-types-of-testing-environments)

1. **what is development environment**

**what is QA environment**

**what is production environment**

[**https://docs.oracle.com/cd/E19225-01/821-0763/ahxbb/index.html**](https://docs.oracle.com/cd/E19225-01/821-0763/ahxbb/index.html)

1. **what are different defect metrics and measurements we prepare**

[**http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/**](http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/)**\***

1. **what are weakness and strong points**

**Q124. What is your weakness?**

Ans: I think my weakness is that whenever I am given some responsibilities and there is a deadline for it, I work day and night, 7 days a week. This is probably bad for my family life, but I can’t sleep unless I am done with my assignments. (Note: You should think of your weakness where because of your weakness (like the one above), still the employer benefits. DON’T SAY anything negative thing, like “I cannot work long hours, it is hard for me pick up things, it is difficult for me to understand requirement documents etc)

**Q123. What are your strengths?**

Ans: I am a very detailed oriented person. I have the sense of urgency. I can prioritize my job according to the deadline. I am very much dedicated towards my job. I am honest. I have the skills and expertise in QA process. These are some of my strengths.