

### A complete guide to Defect Triage Process and Effective ways to handle Defect Triage Meeting:

In today's article, we will learn about Defect Triage meeting and how to handle a triage meeting in an easier and effective way.

Before proceeding further with this article, I wish that everyone knows what is meant by a Defect, Defect Life Cycle, and how to set Priority & Severity for each defect. And it is necessary to understand these basic concepts related to a defect or bug.

You can also go through my earlier article "Defect Life Cycle and Defect Management Process" to understand these concepts quickly.



What You Will Learn: [\[show\]](#)

#### Overview

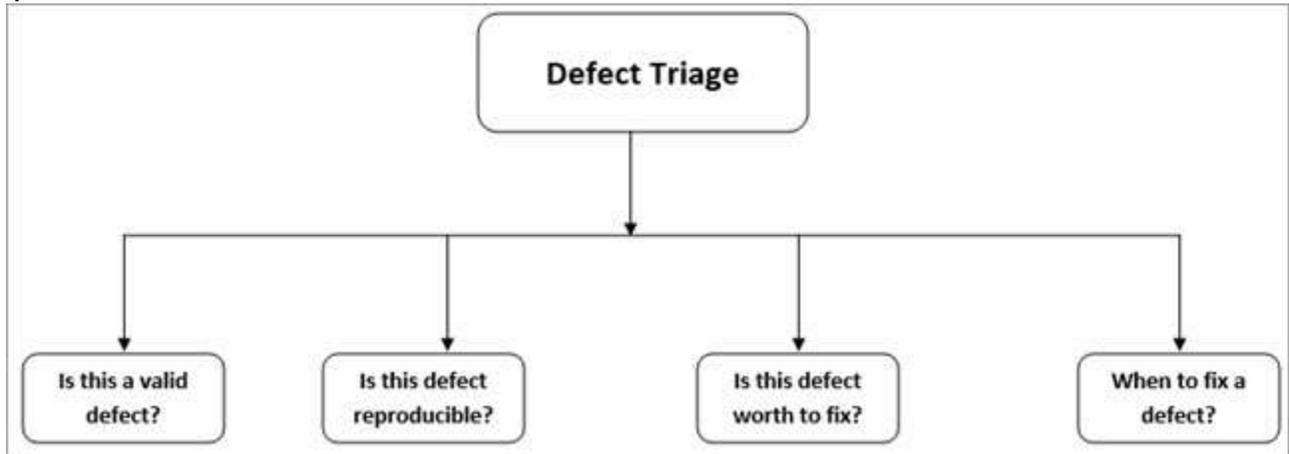
The word "**Triage**" is basically used in the medical field. Actually, it used to decide the order in which the patients should be treated. Usually, in big hospitals, where there are thousands of patient's approaches for consultation or actual treatment on a daily basis. But not all the patients are admitted or treated immediately.

The severity of the illness or the injury is the main criteria for consultation and based on this all the patients are categorized accordingly. If the injury or health of any patient is very critical then the doctors usually treat such patients as a priority and get admitted if required.

Normal diseases or non-critical injuries are considered at a lower priority and such patients are treated later.

Similarly, the term Triage is introduced in software testing for defects in the application or a project. Usually, the Defect Triage process is implemented in large projects and in many cases, it is not applicable for small-scale projects. There are chances to identify a huge number of defects in bigger projects than medium or small projects. Also in bigger projects, the frequency of defect identification is quite higher.

Take a look at the below image which shows the outcome of Defect triage meeting and gives answers to specific questions like:



### Defect Triage Meeting

The main objective of a triage meeting is to track all the defects and ensure the correct resolution in a timely manner.

During the test execution phase, the testers start reporting defects in the Defect Management tool like [HP ALM](#), QC etc. Then **Defect Triage Meeting** is held in which the developers and testers are required to be present as these people will discuss all the defects and take the necessary further course of action.

**Mainly the presence of the below participants is required mandatorily:**

- Project Manager
- Test Lead
- Development Lead or Developer
- Tester
- Test Manager
- Business Analyst
- Environment Manager

Although I have given an exhaustive list of all the participants in the meeting, it is not necessary to involve all of them like Business Analyst, Environment Manager, Test Manager, etc in the daily meeting. Whenever necessary the Test Lead or Project Manager invite them and they can share their valuable feedback and opinion regarding a specific defect.

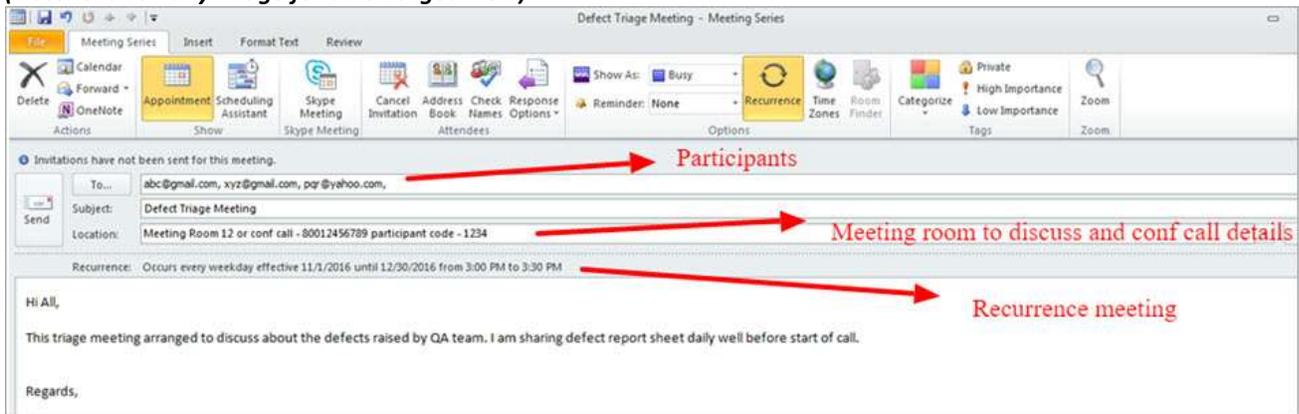
And the entire team is known as a **Triage Team**. Now, I'm going to explain the exact process of triage meeting and how this meeting is set-up.

**Consider one hypothetical Example:** We have one project related to the Banking application, size is very large and the frequency of identifying and reporting the defect is high. Hence the Test Lead decides to set-up a Defect Triage Meeting with the required participants.

For setting up a meeting the Test Lead sends a meeting invite via email to everyone and sets a particular timing for Triage Meeting. The below given hypothetical image shows the meeting invite sent by a Test Lead via outlook to all the participants.

Here everything is imaginary in the below image like – the participant names, meeting room, conference call details, date, time etc.

**(Note: Click on any image for an enlarged view)**



Every day before the start of the triage meeting, the Test Lead sends a list of all the “Open” defects in a spreadsheet format to all the participants so that they can go through all the defects before the meeting and understand what exactly the defect is and what kind of fix is required for it.

**Before the start of every triage meeting, ensure that each defect:**

- Has enough information to understand the defect for all the participants in the meeting.
- Has reported under correct project and category.
- Has mentioned the priority and severity of the defects.
- All the detailed information provided in the defect to understand it correctly to all the participants.

**Recommended Read => A Complete Guide to Defect Management Process**

### Defect Triage Template

Before the kickstart of every Defect Triage Meeting, the Test Lead shares the defect report to all the participants in a specific format and the report pulled out from the Defect Management Tool like HP ALM, HP QC etc. I am showing one sample format in the below image which will give a high-level idea of which fields are mentioned in the Defect Report Template.

	A	B	C	D	E	F	G	H	I	J	K
1	Defect ID	Description	Detected By Name	Assigned To	Severity	Priority	Status	Detected on Date	Root Cause	Closing Date	Application Name
2	1	User is unable to transfer funds to other party	XYZ	DEF	1-Critical	High	Open	14-Nov-16			Application1
3	2	User is able to logged in with invalid credentials	PQR	LMI	1-Critical	High	Open	14-Nov-16			Application1
4	3	Spelling is incorrect for "Submit" it is displayed as "Sbumit" on login page	XYZ	LMI	1-Critical	Low	Open	15-Nov-16			Application1
5	4	On Account Summary page, balance displayed in pound instead of rupees	PQR	NOP	2-High	Medium	Open	16-Nov-16			Application1
6	<b>Defect Report</b>										

Usually, the fields included in the defect report are:

- Defect ID

- Description
- Priority
- Severity
- Detected Date
- Detected By
- Status

The list is not exhaustive but as per the project need, the other fields in the defect report template can be included.

Usually, the spreadsheet format is used as a template for defect reporting, hence I have given the hypothetical defect details in the spreadsheet format. **Please note that all the information provided in the above defect report is only imaginary and not related to any project or actual application.**

### Defect Triage Process

A commonly heard and experienced situation in test teams is limited availability of resources. Defect triage is a process which tries to do some re-balancing as a result of this phenomenon. So when there are many defects and limited Developers/testers to fix/verify them, defect triage helps to get as many defects resolved as possible by balancing the technical personnel based on defect parameters like priority and severity.

Typically, a defect triage session is attended by the Product Manager, a development lead, a test lead and sometimes business analysts. In some cases, certain other members may also be invited to give their opinions and perspectives regarding certain defects. These are collectively called a triage team.



Most systems use priority as the main criteria to assess the defect, however, a good triage process considers the severity as well.

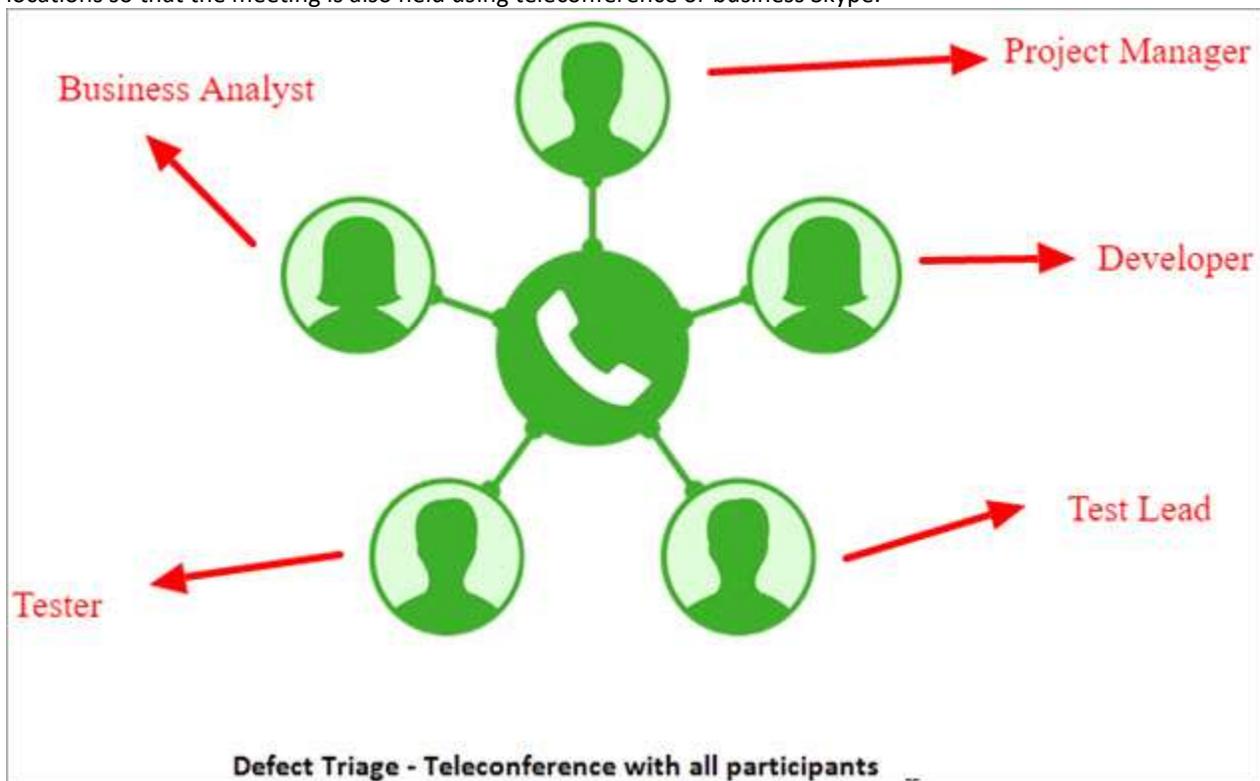
Let's take a closer look at the triage process with two examples that we've talked about in the previous section. In both the examples above, it would actually be the first defect that would be given a very high priority. Despite it being only a cosmetic defect, the impact of not fixing would be huge.

The second one, on the other hand, is a surely functionality defect, however, its occurrence is in only certain conditions which are seldom practiced customer scenarios. Fixing it may need more time and people, which could be better utilized for other defects. Hence it would deem lower priority than that of the first and maybe deferral candidate to another release.

Thus the triage process involves triage team sitting down together, reviewing all the defects including rejected defects. They draw an initial assessment of the defects based on its content, their respective priority, and severity settings; with each person in the triage team presenting their perspective on how to prioritize the defects. The product manager then sets the priority based on all the inputs and assigns the defect to the correct release i.e. in the current release or any future release. He also redirects the defect to the correct owner/team for further action. Rejected defects also are put through a similar analysis. Based on the reason for rejection, the futuristic action of whether it needs to be deferred or canceled is determined.

In the triage meeting, each and every defect should be discussed including the defects which are categorized as a lower priority one. The triage team review evaluates all the defects and takes necessary action on each defect. If a defect is running short of information then the developer assigns back such defects to the testers and requests for necessary information.

The triage meeting can be held in the meeting room if all the participants are at the same location. But in many organizations, the work is carried out from a different location and all the teams are spread across various locations so that the meeting is also held using teleconference or business Skype.



[image source]

**Step by step process of the defect triage meeting:**

- Test Lead kicks off the meeting with the defect report which was sent earlier on the day.
- The discussion starts with the actions pending from the previous triage meeting. The necessary updates or action that was taken on any defect is discussed initially.

- If there are new defects in the defect report then these defects are reviewed and evaluated. It also verifies if the priority and severity are assigned properly, if not, then these are corrected in the meeting.
- All the defects are discussed in the meeting and the development team also discusses the complexity of fixing the defect. The risk associated with the defect is also discussed by the triage team.
- Triage team comes to a conclusion on, which defect should require immediate attention & fix and which defect needs to wait for some time and if required those defects can be postponed to future releases.
- All the defects are assigned to the respective team in QC or ALM simultaneously during the meeting. Appropriate comments are also added in the QC/ALM.
- All the essential updates and action items are noted down and the Test Lead calls out for the end of the meeting.
- After triage meeting completion, Test Lead sends out minutes of meeting to all the participants.

### **Roles and Responsibilities**

**Roles and responsibilities based on each category are explained below:**

#### **Test Lead**

- Test Lead schedules a defect triage meeting and sends out a formal meeting invite to the required team.
- Sends the defect report before every triage meeting.
- Kicks off the meeting with the pending action items from the previous triage meeting.
- Discuss each defect and impact on the schedule if any functionalities are blocked due to the defect.
- Helps in assigning priority and severity of each defect if it was not assigned correctly earlier.
- Update the QC/ALM with appropriate comments.
- Note down all the updates, action items, risk related to a defect, etc.
- Sends minutes of meeting to all the participants.

#### **Development Lead/Developer**

- Share updates on the action items pending from the last triage meeting.
- Discuss all the defects from a technical perspective.
- Identify how much time it will require for fixing based on the complexity of the defect and functionality.
- Discuss the complexity of the defect and risk associated with the defect if any.
- Development Lead assigns defect to the appropriate developer after validating all the available detailed information.
- Updates the defect with the expected resolution date.
- Assists in identifying the root cause of the defect.

#### **Project Manager**

- Ensure that if all the representative from every area is available for the meeting.
- If necessary, project manager invites Business Analyst in the meeting for their opinion on a specific defect.
- If the defects are not moving or if there is any major blocker then escalates with the escalation process.

- If required, acts as a mediator if any dispute or conflict happens between the teams and takes the necessary decision.
- Take the confirmation from the development team for the next release date for fixed defects.
- Make aware of the updated schedule and release date of the project to all the teams.

At times, it is also a good idea to involve the other team members in the triage call so that they can also understand and contribute to the meeting and if required they can also provide their feedback.

### **Conclusion**

Every defect logged should be discussed at the triage meeting.

Even if a defect is rejected, the testing team should know the reason for rejection. Also if any of the defects is not reproducible then during the triage meeting the developer can ask the testers for real-time details and they can try to reproduce the defect.

Defect Triage is important as everyone will know when the defect will get fixed and be available for re-test. If any of the defects is non-critical and in order to fix the defect, it requires huge efforts from the development team and the decision will be taken by the project manager.

### **What is Regression Testing? Test Cases, Tools & Examples**

#### **What is Regression Testing?**

Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

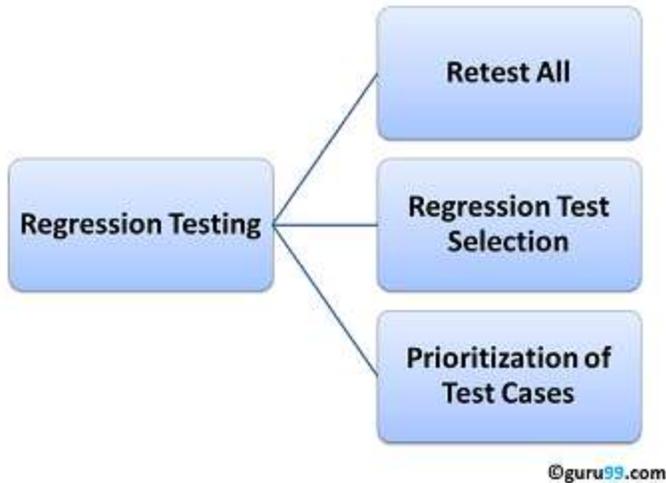
#### **Need of Regression Testing**

Regression Testing is required when there is a

- Change in requirements and code is modified according to the requirement
- New feature is added to the software
- Defect fixing
- Performance issue fix

#### **Regression Testing Techniques**

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary. Regression Testing can be carried out using following techniques:



#### Retest All

- This is one of the methods for Regression Testing in which all the tests in the existing test bucket or suite should be re-executed. This is very expensive as it requires huge time and resources.

#### Regression Test Selection

- Instead of re-executing the entire test suite, it is better to select part of test suite to be run
- Test cases selected can be categorized as 1) Reusable Test Cases 2) Obsolete Test Cases.
- Re-usable Test cases can be used in succeeding regression cycles.
- Obsolete Test Cases can't be used in succeeding cycles.

#### Prioritization of Test Cases

- Prioritize the test cases depending on business impact, critical & frequently used functionalities. Selection of test cases based on priority will greatly reduce the regression test suite.

#### Selecting test cases for regression testing

It was found from industry data that good number of the defects reported by customers were due to last minute bug fixes creating side effects and hence selecting the Test Case for regression testing is an art and not that easy. Effective Regression Tests can be done by selecting following test cases -

- Test cases which have frequent defects
- Functionalities which are more visible to the users
- Test cases which verify core features of the product
- Test cases of Functionalities which has undergone more and recent changes
- All Integration Test Cases
- All Complex Test Cases
- Boundary value test cases
- Sample of Successful test cases
- Sample of Failure test cases

#### Regression Testing Tools

If your software undergoes frequent changes, regression testing costs will escalate.

In such cases, Manual execution of test cases increases test execution time as well as costs.

Automation of regression test cases is the smart choice in such cases.

Extent of automation depends on the number of test cases that remain re-usable for successive regression cycles.

Following are most important tools used for both functional and regression testing:

**Selenium:** This is an open source tool used for automating web applications. Selenium can be used for browser based regression testing.

**Quick Test Professional (QTP):** HP Quick Test Professional is automated software designed to automate functional and regression test cases. It uses VBScript language for automation. It is a Data driven, Keyword based tool.

**Rational Functional Tester (RFT):** IBM's rational functional tester is a Java tool used to automate the test cases of software applications. This is primarily used for automating regression test cases and it also integrates with Rational Test Manager.

#### Regression Testing and Configuration Management

Configuration Management during Regression Testing becomes imperative in Agile Environments where code is being continuously modified. To ensure effective regression tests, observe the following :

- Code being regression tested should be under a configuration management tool
- No changes must be allowed to code, during the regression test phase. Regression test code must be kept immune to developer changes.
- The database used for regression testing must be isolated. No database changes must be allowed

Difference between Re-Testing and Regression Testing:

Retesting means testing the functionality or bug again to ensure the code is fixed. If it is not fixed, Defect needs to be re-opened. If fixed, Defect is closed.

Regression testing means testing your software application when it undergoes a code change to ensure that the new code has not affected other parts of the software.

Also, Check out the complete list of differences over here .

Challenges in Regression Testing:



Following are the major testing problems for doing regression testing:

- With successive regression runs, test suites become fairly large. Due to time and budget constraints, the entire regression test suite cannot be executed
- Minimizing test suite while achieving maximum Test coverage remains a challenge
- Determination of frequency of Regression Tests, i.e., after every modification or every build update or after a bunch of bug fixes, is a challenge.

Practical Application of Regression Testing with a Video

Please be patient. The Video will load in some time. If you still face issue viewing video click here

#### 1) Explain what is JIRA?

JIRA is an issue tracking product or a software tool developed by Atlassian, commonly used for bug tracking, project management and issue tracking; it is entirely based on this three aspects.

#### 2) Explain what is a workflow?

Workflow is defined as a movement of the bug/issue through various stages during its life-cycle

- Created/Open

- WIP ( Work In Progress)
- Completed/Closed

### 3) What can be referred as an issue in JIRA?

In JIRA, an issue can be anything like a

- Software bug
- The project task
- A help-desk ticket
- The leave request form

### 4) List out the source control programs with which it integrates?

It integrates with source control programs such as CVS, Git, Subversion, Clearcase, Visual SourceSafe, Mercurial, and Perforce.

### 5) Why use JIRA?

The reason behind using JIRA is

- Upfront and fair licensing policy
- Features that is not available elsewhere
- Get latest update on the progress of projects
- It run anywhere and recognized with many famous companies
- Easily extensible and customizable



### 6) Is it possible to access JIRA cloud site via a mobile device?

Yes, it is possible to access JIRA cloud site via a mobile device. You have to just use the URL of the JIRA cloud site in your mobile web browser.

### 7) Can you disable JIRA mobile for the site?

You can disable JIRA mobile for the site, so that users can be unable to operate the desktop view of JIRA on their mobile device. JIRA mobile comes as a system add-on and can be disabled any time.

### 8) Explain labelling and linking issue in JIRA?

- **Labelling Issue:** It enables you to categorize an issue in a more informal way than assigning it to a component or version. You can then search issues according to label.
- **Linking Issue:** This feature enables you to link an association between two issues on either on the same or different JIRA servers.

### 9) Mention the types of reports generated in JIRA?

JIRA offer reports that show statistics for projects, versions, people or other fields within issues. Various reports included with JIRA are

- Average Age Report
- Pie Chart Report
- Resolution Time Report
- Recently Created Issues Report
- Resolved vs. Created Issues Report
- Single Level Group by Report
- Time Tracking Report
- User Workload Report

- Workload Pie Chart Report, etc.

**10) Explain what is Cloning an Issue?**

Cloning an issue allows you to create a duplicate of the original issue so that many employees can work on a single issue within a single project. The clone issue can be connected to the original issue. A clone issue holds following the information

- Summary
- Description
- Assignee
- Environment
- Priority
- Issue Type
- Security
- Reporter
- Components, etc.

**11) Mention what things are not included in cloned issue in JIRA?**

- Time tracking
- Issue history
- Comments

**12) Explain what is the use of “Move Issue” wizard in JIRA?**

The move issue wizard enables you to specify another project in your JIRA instance. Move wizard permit you to change certain attributes of an issue like

- **Issue Type:** If your issue is a custom issue type and does not occur in your target project, you must choose a new issue type for your issue
- **Issue Status:** If you have assigned your issue as a custom issue status and it does not exist in your project, you must select a new issue status for your issue
- **Custom Fields:** If you have determined required custom fields for your issue, which do not occur in the target project, you must set values for them.

**13) How security setting is helpful in JIRA?**

JIRA’S security setting restricts the access to the issue to only those person who is allowed to work on the issue or a member of the chosen security level. Security level of an issue can be set either when the issue is created or when the issue is being edited

**14) Explain how you can share an issue with other users?**

You can email an issue by using the share option in JIRA. You can also email other JIRA users a link to the issue by sharing the issue with them or by mentioning them in an issue’s Description or Comment field.

**15) Explain how you can modify multiple bulk issues?**

To modify multiple bulk issues, you can use **Bulk Change** option from the “Tools” menu of the navigator. All the issues on the current page can be selected for the bulk operation. The following list details the available bulk operations like

- Workflow Transition
- Delete
- Move
- Edit

**16) Explain how you can disable mail notification for Bulk Operations?**

To disable mail notification for a particular Bulk Operations, you have to de-select the “Send Notification” checkbox in the bulk operation wizard.

**17) What does an issue change history include?**

Issue change history includes

- Deletion of a comment
- Deletion of a worklog
- Creation or deletion of an issue link
- Attachment of a file

- Changes to an issue field

**18) Explain what does the three color indicates tracking times or duration for an issue?**

Three color will be displayed representing the amount of time spent behind the issue

- Original Estimate (Blue): The amount of time originally estimated to resolve the issue
- Remaining Estimate(Orange): The remaining amount of time left to resolve the issue
- Time Spen or Logged (Green): The amount of time spent so far while resolving the issue

**19) Mention some of the popular add-ons for JIRA?**

Some popular add-ons for JIRA include,

- Suites utilities for JIRA
- ScriptRunner for JIRA
- Zephyr for JIRA – Test Management
- JIRA Toolkit Plugin
- Atlassian REST API Browser
- Portfolio for JIRA
- JIRA Misc Workflow Extensions
- Tempo Timesheets for JIRA
- JIRA Charting Plugin

**20) Mention what is Schemes in JIRA?**

Schemes are a major part of JIRA configuration. It is a collection of configured values that can be used by one or more JIRA project. For instance, Notification Schemes, Permission Scheme, Issue Type Scheme, and so on. There are total seven types of schemes.

**21) Mention what can be configured for JIRA project and issue type?**

You can configure following things for each pair of an issue type and JIRA project.

- The order of custom fields appears on an issue screen
- Workflow of an issue including the statuses
- Which custom fields and system an issue can use
- Project accessibility
- Permissions for what a user can do with an issue
- Versions and components available for an issue

**22) Mention is it possible to get back up your JIRA cloud data?**

In JIRA, you can take backup of your JIRA cloud data using Backup Manager. But only one backup file is stored at a time. The existing backup is overwritten by new ones.

**23) Mention what data can be backed up?**

The backup data includes,

- Attachments if selected
- Users and their group settings
- Avatars
- Issues

**24) Mention some useful tips on JIRA Workflow?**

- As such Statuses are global objects in JIRA. Changing the name of the status on one workflow will change the status on all workflows that use that status
- Hover over a status or transition to see the relevant transition labels
- One cannot clone transitions in the workflow designer
- In the workflow designer, one cannot create annotations
- Directly you cannot set the issue.editable property.

**25) Mention what are the limitations when editing an active workflow?**

- If a workflow is active, you cannot edit the workflow name (only the description)
- You cannot delete the workflow steps
- A step associated status cannot be edited

- You cannot add any new outgoing transition if a step has no outgoing transitions (Global transitions are not considered).
- A step's Step ID cannot be changed.

**26) In JIRA workflow, is it possible to transition an issue back to its previous status?**

Practically, it is not possible to transition an issue back to its previous status. However, you can use "onhold" feature to transition an issue back to its previous status. Here are the steps,

- In workflow, Create a global transition to the 'On Hold' status.
- Now from 'On Hold' status create another transition to every other status you want to come back to
- Since the transition names cannot be the same, just add a blank space at the end of it.
- Now you don't want the status transition from the 'On Hold' and 'Done' to 'On Hold' So you will hide the other status "On Hold" by adding the value field condition on the global transition.

**27) Mention what is the role of Validators in JIRA?**

The Validators in JIRA checks that any input made to the transition is valid before the transition is performed. If a validator fails, the issue will not progress to the destination status of the transition.

**28) Mention what types of Post functions are carried out after the transition is executed?**

Types of Post functions carried out after transition is executed includes

- Adding a comment to an issue
- Generating change history for an issue
- Updating an issue's fields
- Generating an event to trigger email notifications

**29) What is an event in JIRA?**

The events are classified in two a System event (JIRA defined events) and Custom event (User defined events). An event describes the status, the default template and the notification scheme and workflow transition post function associations for the event.

**30) What is Audit Log?**

Under Audit Log, you can see all the details about the issue created, and the changes made in the issues.

**31) For a Agile project, how user stories in JIRA are created?**

For Agile project to create user stories in JIRA, follow below steps.

- Issue type -Epic and Issue type – Story linked to it. In order to do so, in the 'Create Issue' page, go to "Configure Fields" and select "Epic link" field to be included in the issue creation screen.
- Or you can have a product backlog by creating a main User story and having various sub-tasks under it.

**32) Mention what is an "issue collector"?**

An "issue collector" enables you to easily embed a JIRA feedback form into your own web site. This helps website visitors to log issues into JIRA through our website. To use JIRA feedback form, visitors to our website do not need a user account in JIRA.

**33) Mention the difference between Bugzilla and JIRA?**

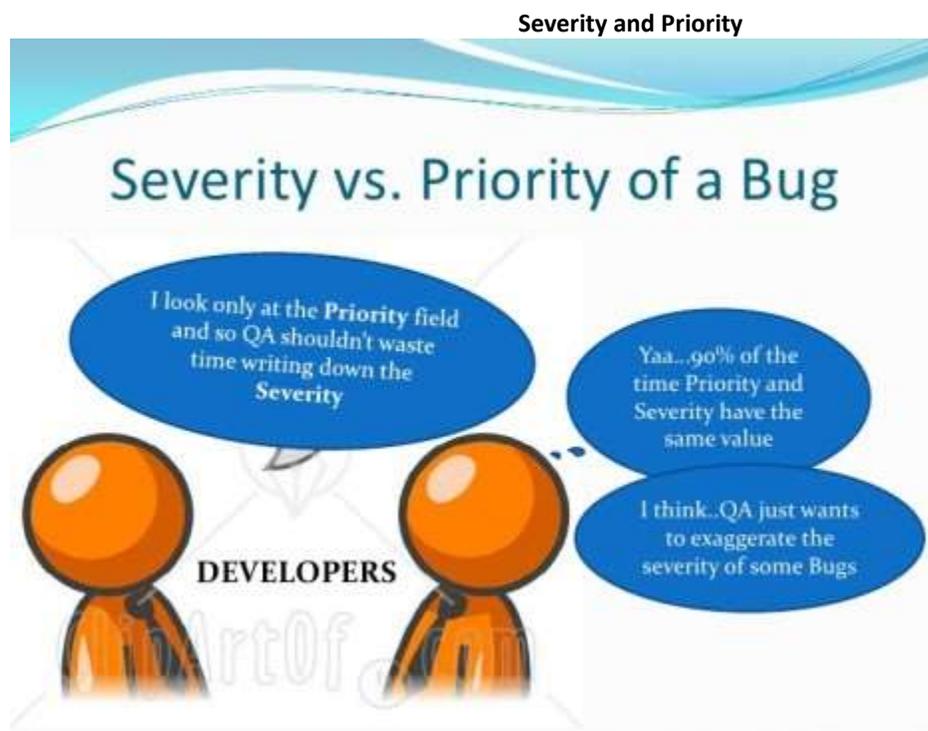
Bugzilla	JIRA
<ul style="list-style-type: none"> <li>• It is an Open Source</li> </ul>	<ul style="list-style-type: none"> <li>• It is a commercial tool</li> </ul>
<ul style="list-style-type: none"> <li>• Using Bugzilla might be little complicated for few due to grouping users and granting permissions</li> </ul>	<ul style="list-style-type: none"> <li>• For some using JIRA would be more convenient than</li> </ul>
<ul style="list-style-type: none"> <li>• Bugzilla allows you to show/hide the whole custom field or specific values based on the value of some other field</li> </ul>	<ul style="list-style-type: none"> <li>• JIRA enables conditional configuration based only on Project.</li> </ul>
<ul style="list-style-type: none"> <li>• Bugzilla's has a powerful advanced search option</li> </ul>	<ul style="list-style-type: none"> <li>• JIRA lacks advance-level search options. JIRA has flex (JIRA Query Language). It enables you to build arbitrary b expressions.</li> </ul>

<ul style="list-style-type: none"> <li>• Unlike JIRA, Bugzilla allows users select the initial status of a new issue.</li> </ul>	<ul style="list-style-type: none"> <li>• Unlike Bugzilla, JIRA enables you to define multiple filters that are applied based on the issue's Project and Type.</li> </ul>
<ul style="list-style-type: none"> <li>• Bugzilla has only one link type: Blocks/depends and a Bug ID custom field</li> </ul>	<ul style="list-style-type: none"> <li>• JIRA has configurable link types with user-defined search filters that enables to link an issue to any other entity outside JIRA.</li> </ul>

**34) Explain how you can modify multiple bulk issues?**

You can modify multiple bulk issues by using option "Bulk Change" option.

Severity and Priority – What is the Difference?



severity-vs-priority-testing

Both Severity and Priority are attributes of a defect and should be provided in the bug report. This information is used to determine how quickly a bug should be fixed.

Severity of a defect is related to how severe a bug is. Usually the severity is defined in terms of financial loss, damage to environment, company's reputation and loss of life.

Priority of a defect is related to how quickly a bug should be fixed and deployed to live servers. When a defect is of high severity, most likely it will also have a high priority. Likewise, a low severity defect will normally have a low priority as well.

Although it is recommended to provide both Severity and Priority when submitting a defect report, many companies will use just one, normally priority.

In the bug report, Severity and Priority are normally filled in by the person writing the bug report, but should be reviewed by the whole team.

#### **High Severity – High Priority bug**

This is when major path through the application is broken, for example, on an eCommerce website, every customer gets error message on the booking form and cannot place orders, or the product page throws a 500 response.

#### **High Severity – Low Priority bug**

This happens when the bug causes major problems, but it only happens in very rare conditions or situations, for example, customers who use very old browsers cannot continue with their purchase of a product. Because the number of customers with very old browsers is very low, it is not a high priority to fix the issue.

#### **High Priority – Low Severity bug**

This could happen when, for example, the logo or name of the company is not displayed on the website. It is important to fix the issue as soon as possible, although it may not cause a lot of damage.

#### **Low Priority – Low Severity bug**

For cases where the bug doesn't cause disaster and only affects very small number of customers, both Severity and Priority are assigned low, for example, the privacy policy page takes a long time to load. Not many people view the privacy policy page and slow loading doesn't affect the customers much.

The above are just examples. It is the team who should decide the Severity and Priority for each bug.

#### **Q #1) What do you understand by the term 'Functional testing'?**

**Answer:** A black box testing technique, where the functionality of an application is tested to generate the desired output by providing certain input is called 'Functional testing'.

The role of functional testing is not only to validate the behavior of the application as per the requirement document specification but is also to verify whether the application is ready to be released into the live environment or not.

**Given below are few functional testing techniques that are commonly used:**

- Unit testing

- Smoke testing
- Integration testing
- System Testing
- Usability testing
- Regression testing
- User Acceptance testing

**Q #2) What are the important steps that are covered in Functional testing?**

**Answer: Following are the steps that should be covered as a part of functional testing:**

- Understanding the Requirement document specification and clearing the doubts and queries in the form of review comments.
- Writing the test cases with respect to the requirement specification by keeping in mind all the scenarios that should be considered for all the cases.
- Identifying the test inputs and requesting the test data that is required to execute the test cases as well as to check the functionality of the application.
- Determine the actual outcomes as per the input values to be tested.
- Execute the test cases that determine whether application behavior is as expected or any defect has occurred.
- Compare the actual result and the computed result to find out the actual outcome.



**Q #3) Explain the difference between Functional testing and Non-Functional testing.**

**Answer:** The difference between Functional testing and Non-functional testing can be found in the table given below:

Functional Testing	NonFunctional Testing
Functional testing is performed to determine the system behaviour as per the client functional requirements.	Non-functional testing is the process to determine the system performance as per client expectations
Functional testing is performed first with the help of Manual and Automation testing tools.	Non-functional testing is performed after functional testing with the effective tools required.
It is easy to perform manual testing as client requirements are the input in functional testing.	It is difficult to perform manual testing as scalability, reliability, speed and other performance parameters are input in non functional testing.
Functional testing is of following types: <ul style="list-style-type: none"> <li>• Unit Testing</li> <li>• Smoke Testing</li> <li>• Sanity Testing</li> <li>• Integration testing</li> </ul>	Non-functional testing is of following types: <ul style="list-style-type: none"> <li>• Performance testing</li> <li>• Load, Stress, Volume Testing</li> <li>• Security testing</li> <li>• Compatibility testing</li> </ul>

## Functional Testing

## NonFunctional Testing

- User Acceptance testing
- Regression testing

### Q #4) How is 'Build' different from 'Release'?

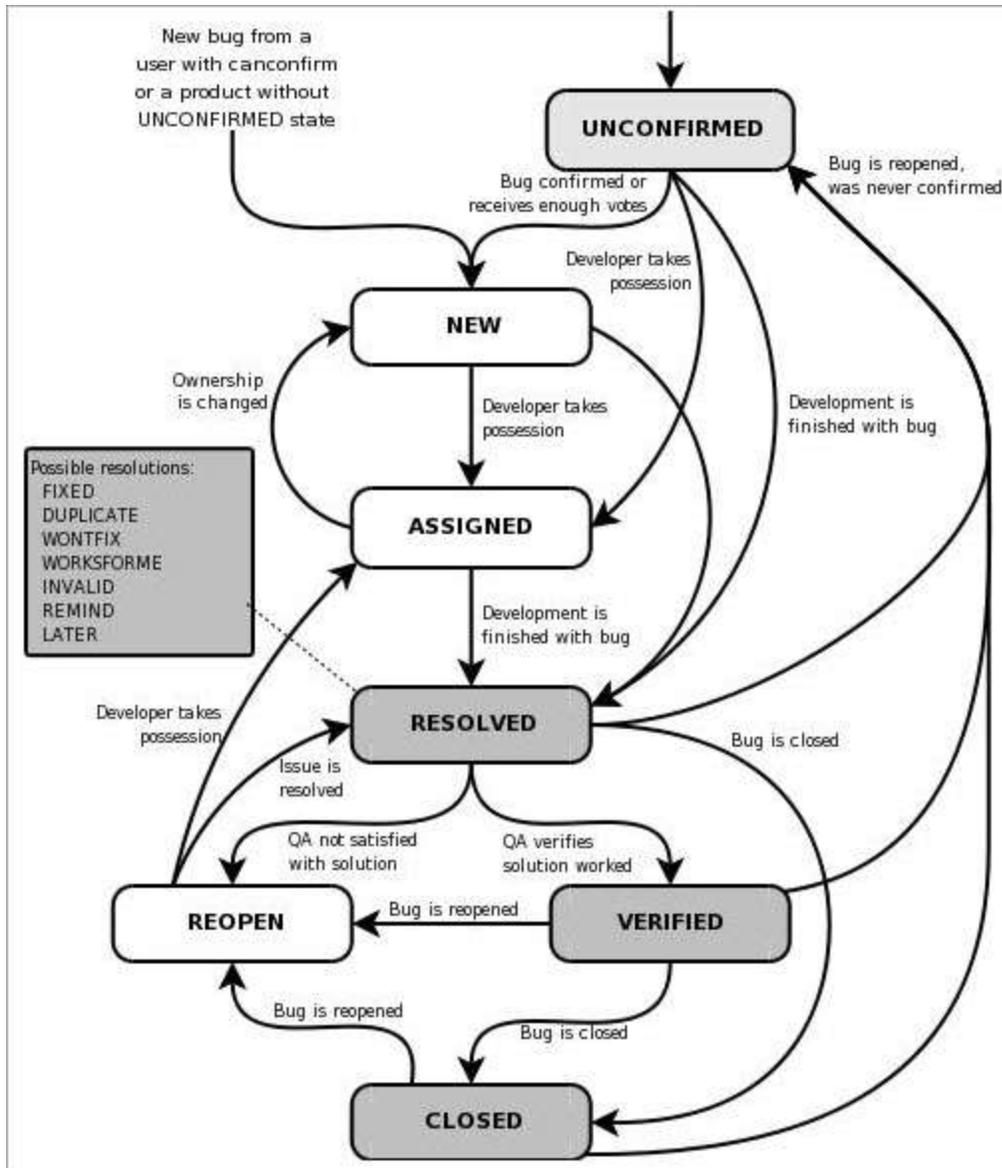
**Answer: Build** is an executable file which refers to that part of an application which is handed over to a tester to test the implemented functionality of the application along with some bug fixes. The build can be rejected by the testing team if it does not pass the critical checklist which contains the major functionality of the application. There can be multiple builds in the testing cycle of an application.

**Release** refers to the software application which is no longer in the testing phase and after completion of testing and development, the application is handed over to the client. One release has several builds associated with it.

### Q #5) Explain Bug cycle.

**Answer:** Bug is said to be an unwanted error, flaw, mistake, etc that has occurred within the application and prevents it from delivering the desired output. When any defect or bug is encountered in an application while testing, then from logging a defect till its resolution, a bug moves through a definite life cycle known as Bug Lifecycle.

**Below figure will give you an idea of Bug lifecycle:**



**image source: Bugzilla bug life cycle**

The whole process goes as and when an issue or bug is encountered. It is reported /logged in bug tracking tool following a considerable format. These bugs are assigned to the developer and its status is made as 'Open'. Developer can now review the bug, reproduce it at its end and start working on it.

If the bug is fixed, the developer changes its status to 'Fixed' or the status can be moved to 'need more information', 'won't fix', 'cannot reproduce' etc., in other cases. QA then performs regression i.e. re-verify the bugs with a specific action and responds accordingly.

If the issues/bug is now behaving as expected then its status is changed to Verified /Closed else Reopen.

**Q #6) Enlist some Bug status along with its description.**

**Answer: Enlisted below are few bug statuses along with their descriptions:**

- **New:** When the defect or bug is logged for the first time it is said as New.

- **Assigned:** After the tester has logged a bug, his bug is being reviewed by the tester lead and then it is assigned to the corresponding developer team.
- **Open:** Tester logs a bug in the Open state and it remains in the open state until the developer has performed some task on that bug.
- **Resolved/Fixed:** When a developer has resolved the bug, i.e. now the application is producing the desired output for a particular issue, then the developer changes its status to resolved/fixed.
- **Verified/Closed:** When a developer has changed the status to resolved/fixed then the tester now tests the issue at its end and if it's fixed then he changes the status of the bug to 'Verified/Close'.
- **Reopen:** If a tester is able to reproduce the bug again i.e. the bug still exists even after fixing by the developer, it's status is marked as reopen.
- **Not a bug/Invalid:** A bug can be marked as invalid or not a bug by the developer when the reported issue is as per the functionality but is logged due to misinterpretation.
- **Deferred:** Usually when the bug is of minimal priority for the release and if there is lack of time, in that case, those minimal priority bugs are deferred to the next release.
- **Cannot Reproduce:** If the developer is unable to reproduce the bug at its end by following the steps as mentioned in the issue.

**Q #7) What is known as Data-driven testing?**

**Answer:** Data-driven testing is the methodology where a series of test script containing test cases are executed repeatedly using data sources like Excel spreadsheet, XML file, CSV file, SQL database for input values and the actual output is compared to the expected one in the verification process.

**For Example:** Test studio is used for data-driven testing.

**Some advantages of data-driven testing are:**

- Reusability.
- Repeatability.
- Test data separation from test logic.
- The number of test cases is reduced.

**Q #8) What are the important points that should be considered while writing Test Cases?**

**Answer:** Writing a test case is said to be the most important activity of the test execution process which requires writing skills as well as in-depth knowledge of the application to make effective and reusable test cases.

**Few important points that should be considered while writing test cases includes:**

- There should be a clear understanding of the client's requirements before beginning to write the test cases. Nothing should be assumed and every doubt regarding the requirements should be cleared.
- Every requirement should be included in the form of test cases and nothing should be left out. Usually Traceability matrix is maintained to keep a check on every requirement implementation and testing completion.
- As per the requirement document specifications, every functional and non-functional requirement including UI interface, compatibility should be covered.
- Test cases should be checked from time to time for no repetition or redundancy.
- Priority is an important factor which should be set for the test cases while writing. This priority helps tester to test the application first with the high priority tests cases which includes basic functionality, then the medium and later the low priority test cases.
- For a particular release, test cases can also be built Sprint wise so that the tester, as well as the developer, can analyze the quality of the product based on test case execution.
- Structure of test cases should be easily understood and must be in a simple language. The input data values for test cases should be valid as well as in a wide range.

**Q #9) What is Automation testing?**

**Answer:** Automation testing is a testing methodology where automation tool is used to execute the test cases suite in order to increase test coverage as well speed to test execution. Automation testing does not require any human intervention as it executes pre-scripted tests and is capable of reporting and comparing outcomes with previous test runs.

Repeatability, ease of use, accuracy, and greater consistency are some of the advantages of Automation testing.

**Some automation testing tools are listed below:**

- Selenium
- Tellurium
- Watir
- SoapUI

**Q #10) Explain the term Stress Testing and Load testing.**

**Answer: Stress Testing** is a form of performance testing where the application is bound to go through exertion or stress i.e. execution of application above the threshold of the break to determine the point where the application crashes. This condition usually arises when there are too many users and too much of data.

Stress testing also verifies the application recovery when the work load is reduced.

**Load Testing** is a form of performance testing where the application is executed above various load levels to monitor peak performance of the server, response time, server throughput, etc. Through load testing process stability, performance and integrity of the application are determined under concurrent system load.

**Q #11) What do you understand by Volume testing?**

**Answer:** Volume testing is a form of performance testing which determines the performance levels of the server throughput and response time when concurrent users, as well as large data load from the database, are put onto the system/application under tests.

**Q #12) What are the different Test Techniques used in Functional testing?**

**Answer:** There are two different test techniques that are used in functional testing.

**They can be defined as below:**

- **Requirement based testing:** This form of functional testing is performed prioritizing the requirements on the basis of risk criteria. This also assures that all the critical test paths have been included in the testing process.
- **Business process based testing:** This form of functional testing is performed from the business process perspective. The scenarios include knowledge of business processes for performing testing.

**Q #13) What do you understand by Exploratory Testing? When is it Performed?**

**Answer:** Exploratory testing means testing or exploring the application without following any schedules or procedures. While performing exploratory testing, testers do not follow any pattern and use their out of box thinking and diverse ideas to see how the application performs.

Following this process covers even the smallest part of the application and helps in finding more issues/bugs than in the normal test case testing process.

**Exploratory testing is usually performed in cases when:**

- There is a experienced tester in the testing team who can use their testing experience to apply all the best possible scenarios.
- All critical paths have been covered and major test cases are prepared as per the requirement specifications that have been executed.
- There is a critical application and no possible cases can be missed in any case.
- New tester has entered the team, exploring the application will help them understand better as well as they will follow their own mind while executing any scenario rather than following the path as mentioned in the requirement document.

**Q #14) For any Web Application, what are the possible login features that should be tested?**

**Answer: Enlisted below are the possible scenarios that can be performed to fully test the login feature of any application:**

- Check the input fields i.e. Username and password with both valid and invalid values.
- Try entering valid email id with an incorrect password and also enter an invalid email and valid password. Check for the proper error message displayed.

- Enter valid credentials and get logged in to the application. Close and reopen the browser to check if still logged in.
- Enter the application after logging in and then again navigate back to the login page to check whether the user is asked again to login or not.
- Sign in from one browser and open the application from another browser to verify whether you are logged into another browser also or not.
- Change password after logging into the application and then try to login with that old password.

There are few other possible scenarios as well which can be tested.

**Q #15) Explain Accessibility testing and its importance in the present scenario.**

**Answer:** Accessibility testing is a form of usability testing where testing is performed to ensure that the application can be easily handled by people with disabilities like hearing, colour blindness, low visibility etc. In today's scenario, the web has acquired the major place in our life in the form of e-commerce sites, e-learning, e-payments, etc.

Thus in order to grow better in life, everyone should be able to be a part of technology especially people with some disabilities.

**Enlisted below are few types of software which helps and assist people with disabilities to use technology:**

- Speech recognition software
- Screen reader software
- Screen magnification software
- Special keyboard

**Q #16) What is Adhoc testing?**

**Answer:** Adhoc testing, usually known as random testing is a form of testing which does not follow any test case or requirement of the application. Adhoc testing is basically an unplanned activity where any part of the application is randomly checked to find defects.

In such cases, the defects encountered are very difficult to reproduce as no planned test cases are followed. Adhoc testing is usually performed when there is a limited time to perform elaborative testing.

**Q #17) What is Equivalence Partitioning?**

**Answer:** Equivalence partitioning also known as equivalence class partitioning is a form of black box testing where input data is being divided into data classes. This process is done in order to reduce the number of test cases, but still covering the maximum requirement.

Equivalence partitioning technique is applied where input data values can be divided into ranges. The range of the input values is defined in such a way that only one condition from each range partition is to be tested assuming that all the other conditions of the same partition will behave the same for the software.

**For Example:** To identify the rate of interest as per the balance in the account, we can identify the range of balance amount in the account that earn a different rate of interest.

**Q #18) Explain Boundary Value Analysis.**

**Answer:** Boundary value analysis method checks the boundary values of Equivalence class partitions. Boundary value analysis is basically a testing technique which identifies the errors at the boundaries rather than within the range values.

**For Example,** An input field can allow a minimum of 8 characters and maximum 12 characters then 8-12 is considered as the valid range and <7 and >13 are considered as the invalid range. Accordingly, the test cases are written for valid partition value, exact boundary value, and invalid partition value.

**Q #19) Explain the difference between Severity and Priority.**

**Answer: Defect Severity** is defined by the level or the degree of impact by the defect on the application under test. Higher the severity of the defect, the more is the impact on the application.

**Following are the 4 classes in which a defect severity is categorized:**

- Critical
- Major
- Medium
- Low

**Defect priority** defines the order in which the defect should be resolved first i.e. the higher the priority of the defect implies that the application is unusable or stuck at some point and the defect should be resolved as soon as possible.

**Following are the 3 classes in which a defect priority is defined:**

- High
- Medium
- Low

**Q #20) When do we perform Smoke testing?**

**Answer:** Smoke testing is performed on the application after receiving the build. Tester usually tests for the critical path and not the functionality in deep to make sure, whether the build is to be accepted for further testing or to be rejected in case of broken application.

A smoke checklist usually contains the critical path of the application without which an application is blocked.

**Q #21) What do you understand by Sanity testing?**

**Answer:** Sanity testing is performed after receiving the build to check the new functionality/defects to be fixed. In this form of testing the goal is to check the functionality roughly as expected and determine whether the bug is fixed and also the effect of the fixed bug on the application under test.

There is no point in accepting the build by the tester and wasting time if Sanity testing fails.

**Q #22) What do you understand by Requirement Traceability Matrix?**

**Answer:** Requirement Traceability matrix (RTM) is a tool to keep a track of requirement coverage over the process of testing.

In RTM, all requirements are categorized as their development in course of sprint and their respective ids (new feature implementation/ enhancement/ previous issues, etc) are maintained for keeping a track that everything mentioned in the requirement document has been implemented before the release of the product.

RTM is created as soon as the requirement document is received and is maintained till the release of the product.

**Q #23) What are the factors to be considered in Risk-based testing?**

**Answer:** By Risk-based testing of a project, it is not just to deliver a project risk-free but the main aim of risk-based testing is to achieve the project outcome by carrying out best practices of risk management.

The major factors to be considered in Risk-based testing are as follows:

- To identify when and how to implement risk-based testing on an appropriate application.
- To identify the measures that act well in finding as well as handling risk in critical areas of the application.
- To achieve the project outcome that balances risk with the quality and feature of the application.

**Q #24) Differentiate between Regression testing and Re-testing.**

**Answer: Difference between Regression testing and Re-testing can be explained as follows:**

Regression testing	Retesting
Regression testing is the form of testing which is carried out to make sure that implementation of any new feature or fixes does not affect any other part or functionality of the application.	Retesting is the form of testing the application after fixing of defects for those test cases which were failed in last execution.

Regression testing	Retesting
As a part of regression testing, new changes in the application should not affect the existing functionalities.	As a part of retesting, defect verification is done.
Based on the project requirement, regression testing can be parallel performed with retesting.	Retesting is performed before regression testing because of its high priority.
Also known as generic testing and is done for passed test cases.	Also known as planned testing and is only done for failed test cases.
As manual testing can be time consuming and expensive, automation can be done for regression testing.	Automation cannot be done for retesting.

**Q #25) Explain User Acceptance testing.**

**Answer:** User acceptance testing is usually performed after the product is thoroughly tested. In this form of testing, software users or say, client, itself use the application to make sure if everything is working as per the requirement and perfectly in the real world scenario. UAT is also known as End-user testing.

**1. What is the difference between Quality Assurance (QA) and Quality Control (QC)?**

**Quality Assurance:** Quality Assurance involves in process-oriented activities. It ensures the prevention of defects in the process used to make Software Application. So the defects don't arise when the Software Application is being developed.

**Quality Control:** Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

**2. What is the difference between Preventative and Reactive approaches in testing?**

**Preventive approach:** It is also known as Verification Process. This approach is to prevent defects. In this approach, tests are designed at early stages of SDLC i.e., before the software has been produced. Here in this approach testers try to prevent defects in the early stages. It comes under Quality Analysis (QA).

**Reactive approach:** It is also known as Validation Process. This approach is to identify defects. In this approach, tests are designed to execute after the software has been produced. Here we try to find defects. It comes under Quality Control (QC).

**3. Why are you in QA?**

I am in QA because I like this job.

Read more on [why did you choose Quality Assurance as a career](#)

**4. List out the roles of Quality Assurance engineer?**

A software quality assurance engineer usually involves in the following tasks.

- QA Team is responsible to monitor the entire development process.
- They are responsible to track the outcomes of each phase of SDLC and adjust them to meet the expectation.
- They are responsible to read and understand the requirement documents.
- Analyze test requirements, and design and execute tests.
- Develop test cases and prioritize testing activities.
- Record problems and issues in accordance with the project's problem and issue management plans.
- Work with the application team and/or client to resolve any issues that arise in the testing process.
- Carry out regression testing every time when changes are made to the code to fix defects.
- Have to interact with the clients to better understand the product requirements.
- Participate in walkthroughs of testing procedures.

### **5. Explain the process of QA testing?**

In simple words, QA testing process is a step by step process which involves analyzing requirement documents, preparing test strategy, test plan and test cases, executing test cases when the build is ready. In the execution process QA's perform different types of testing to make sure the software reaches or exceeds the expectation.

[Read more..](#)

### **6. What is the role of documentation in QA?**

Documentation plays a vital role in Quality Assurance. All the documents involved in SDLC such as Business Requirement Specifications, Designs, Inspection reports, Configurations, Code changes, Test Strategy, Test plans, Test cases, Bug reports, User manuals should be documented.

- Documentation helps us to achieve high quality software product.
- Documentation is necessary to make things more real
- We could use documentation as a reference material and reuse it when necessary
- We could save lot of organization's time, effort and money by maintaining proper documentation.
- Proper documentation makes easy for the client to review the software process.

### **7. What is quality audit?**

Quality audit is the process of systematic and independent examination of a software product or process to assess compliance with specifications, standards, agreements and other relevant criteria.

### **8. Mention what are the test artifacts involved in QA?**

The test artifacts involved in QA are Test Strategy, Test Plan, Test Scenarios, Test Cases, Test Summary Report, Bug Report etc.,

Read more and download [complete set of test artifacts](#) from here..

## **9. Have you written Test Strategy?**

Usually, test strategy document will be prepared by Test Managers or Project Managers. If you are applying for a Project Manager position and you have experience in preparing Test Strategy document then you can say Yes else say I know what is a test strategy and its purpose but I never got a chance to write Test Strategy document.

## **10. What is a Test Strategy and what does it include?**

Test Strategy is a high level document (static document) and usually developed by project manager. It is a document which captures the approach on how we go about testing the product and achieve the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as base.

Read more on [detailed explanation of Test Strategy..](#)

## **11. Have you written Test Plan?**

Usually, test plan document will be prepared by Test Leads or Test Managers. If you are applying for a Test lead position and you have experience in preparing Test Plan document then you can say Yes else say I know what is a test plan and its purpose but I never got a chance to write Test Strategy document.

## **12. What is a Test Plan and what does it include?**

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product. Test Plan document is derived from the Product Description, SRS, or Use Case documents for all future activities of the project. It is usually prepared by the Test Lead or Test Manager and the focus of the document is to describe what to test, what not to test, how to test when to test and who will do what test. Also, it includes the environment and tools needed, resource allocation, test technique to be followed, risks and contingencies plan. A test plan is a dynamic document and we should always keep it up-to-date. Test plan document guides us how the testing activity should go on. Success of the testing project completely depends on Test Plan.

Read more on [detailed explanation of Test Plan..](#)

## **13. What is a Test case template?**

A test case template is a document comes under one of the test artifacts, which allows testers to develop the test cases for a particular test scenario in order to verify whether the features of an application are working as intended or not. Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results. Most of the companies are using test case management tools such as Quality Center (HP QC), JIRA etc., and some of the companies still using excel sheets to write test cases.

## **14. What are the key components of a test case template**

The key components of a test case template are Project name, Module name, Created by, Date of creation, reviewed by, date of review, executed by, Date of execution, test scenario, tase case id, test case description, Precondition, Test steps, Test data, expected result, post condition, actual result, status of the bug.

Check the below video on how to write effective test cases.

### **15. How do you decide when you have tested enough?**

This is one of the most important questions in terms of ISTQB. Option will be tricky and you have to choose the right one.

As a project manager or project lead, sometimes you might face a situation to call off the testing to release the product early. In those cases, you have to decide whether the testers have tested the product enough or not. There are many factors involved in the real time projects to decide when to stop testing.

- if we reach Testing deadlines or release deadlines
- By reaching the decided pass percentage of test cases
- if the risk in the project is under the acceptable limit
- if All the high priority bugs and blockers are fixed
- if we met the acceptance criteria

As per ISTQB, It depends on the risks for the system being tested.

### **16. What are the key components of a bug report?**

Bug report is aka defect report, it conveys the detailed information (such as environment details, steps to reproduce etc.) about the bug to the developers. It allows developers to replicate the bug easily. The key components of a bug report are Defect Id, title of the defect, Reporter Name, Defect Report Date, Reporter designation, Project name, Release Version, Environment details, Priority of the bug, Severity of the bug, Status of the bug, Defect Description, Steps to reproduce the bug, Expected result, Actual result, Attachments if any and Defect closed date.

Read more on [how to write a good report..](#)

### **17. Tell me some key points to consider while writing a bug report.**

- i. Reproduce the bug 2-3 times.
- ii. Use some keywords related to your bug and search in the Defect Tracking Tool.
- iii. Check in similar modules.
- iv. Report the problem immediately.
- v. Write detailed steps to reproduce the bug.
- vi. Write a good defect summary. Watch your language in the process of writing the bug report, your words should not offend people. Never use capital letter whilst explaining the issue.
- vii. Advisable to Illustrate the issue by using proper screenshots.
- viii. Proofread your bug report twice or thrice before posting it.

### **18. What are the advantage and disadvantages of Automated Testing?**

**Advantages:**

1. Automation testing is faster in execution
2. It is cheaper compared to manual testing in a long run
3. Automated testing is more reliable
4. Automated testing is more powerful and versatile
5. It is mostly used for regression testing
6. It does not require human intervention. Test scripts can be run unattended
7. It helps to increase the test coverage

**Disadvantages:**

1. It is recommended only for stable products
2. Automation testing is expensive initially
3. Most of the automation tools are expensive
4. It has some limitations such as handling captcha, fonts, color
5. Huge maintenance in case of repeated changes in the requirements

Not all the tools support all kinds of testing. Such as windows, web, mobility, performance/load testing

**19. What is the difference between build and release?**

**Build:** A build is a version of a software. Every build has a number for identification purpose. Build is a pre-release version of a Release. Build is given to testing team by developers to test the application locally. Build numbers are incremental.

**Release:** A release is the distribution of the final version of an application to the customer by software development team.

**20. What is bug leakage and bug release?**

**Bug Leakage:** A bug which is actually missed by the testing team while testing and the build was released to the Production. If now that bug (which was missed by the testing team) was found by the end user or customer then we call it as Bug Leakage.

**Bug release:** Releasing the software to the Production with some known bugs then we call it as Bug Release. These known bugs should be included in the release note. In other case, releasing the software to the testing team with some known bugs whose severity and priority is low. These bugs can be removed before releasing to production.

**21. What is Bug triage?**

Bug triage is a formal process to find which bugs are important by prioritizing them based on their severity, frequency, risk and other important parameters. Testers assign priority (high, medium, low) to each and every bug in a bug triage meeting and based on the priority those bugs will be fixed in an order. By doing this we could save a lot of organization's time.

## 22. Explain bug life cycle.

Bug life cycle is also known as Defect life cycle. In Software Development process, the bug has a life cycle. The bug should go through the life cycle to be closed. Bug life cycle varies depends upon the tools (QC, JIRA etc.,) used and the process followed in the organization. [Read more..](#)

## 23. What is MR and ER?

**MR:** MR stands for Modification Request. It is used to change the existing functionality in a software, it is usually requested by clients.

**ER:** ER stands for Enhancement report. It is used to add a new feature in a software. It is usually requested by clients.

## 24. Mention some of the types of software testing?

There are more than 100 types of software testing.

**Must Read:** [100+ Types of Testing](#)

## 25. What is CRUD testing?

CRUD (Create, Read, Update and Delete) is another term used for Black box testing. CRUD testing is another term for database testing.

Read more on [Black box testing](#) here..

- C – Create – Creating a new Transaction
- R – Read/Retrieve – Searching or viewing a transaction
- U – Update – Editing or modifying an existing transaction.
- D – Delete – Deleting a transaction from the database

**Must Learn:** [SQL Tutorial for Software Testers](#)

## 26. What is a Cookie testing?

A Cookie is also known as HTTP cookie, web cookie, internet cookie, browser cookie.

Read more on [Cookie testing..](#)

## 27. What is Cross browser testing?

Cross Browser Testing is a type of non-functional test which helps us ensure that our website or web application works as expected in various web browsers. We could do Cross Browser Testing on different browsers both manual and automated way. To do Cross Browser Testing manually, we (Software Testers) create tests for each

browser and execute it manually on each browser. To do it in an automated way, we could create Selenium tests with multiple conditional statements that execute test cases based on specified browser type. Every browser displays a website in their own style. We usually cannot have all the browsers on one machine. Each browser is designed by a different vendor. So each browser has their own features to showcase their unique presence. While testing a website, we need to ensure that our website is appearing same across all the browsers. To do this we need to have all the browsers. Fortunately, there are some tools to perform cross-browser testing without testing individually in a manual way.

Read more on [Cross browser testing](#).

## 28. What is the difference between Compatibility testing and Cross browser testing?

**Compatibility testing:** Testing an application on different hardware or software platform is Compatibility testing.

**Example:** Different devices such as iPhone, Samsung etc., Different operating system such as Windows, Linux etc.,

**Cross browser testing:** Testing a web application on different browsers is Cross browser testing. Cross browser testing is a subset of Compatibility testing.

**Example:** Google Chrome, IE 10, IE 11, Firefox 43 etc.,

## 29. What is Configuration management?

Configuration management is a process followed during the project life cycle to control and document each and every change.

## 30. What are the various tools you have used in testing process?

The tools which I have used during testing process are as follows.

**Test Management Tools:** JIRA, [TestLodge](#), Quality Center

**Test Case Management Tools:** [TestCaseLab](#)

**Defect Tracking Tools:** Bugzilla, MantisBT

**Automation Tools:** QTP/UFT, [Selenium](#), LoadRunner

**GUI Tools:** [Froglogic Squish](#)

**Cross Browser Testing Tools:** [CrossBrowserTesting](#), [BrowserStack](#)

## Conclusion:

I would like to conclude this 'Software QA Interview Questions And Answers' post here. If you have any questions, please comment below and we will try to include those in this list of Software QA Interview Questions.

## **Regression Testing**

The definition of regression testing, as stated in [ISTQB glossary](#):

“Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.”

## **When to perform Regression Testing?**

Regression testing is performed when any of the following situation happens during software development lifecycle:

1. A bug is fixed.
2. A change in requirements is implemented.
3. A new business rule, feature or functionality is implemented.
4. A new module, component or subsystem is developed and integrated to the existing modules or system.

## **Challenges of Regression Testing**

Regression testing is easy to define and understand, but baffling when it comes to perform regression testing of a software product. One reason is the dynamic nature of software product. Further complexity is added as the product’s functionalities and capabilities expand. Lastly, the quality assurance team might also face time limitations and pressure from the management when they are testing the application for regression.

You need to understand the challenges well, before you can craft a counter strategy for those challenges. Let’s have a look at the common challenges of regression testing, listed below:

**TRY REQUEST FREE - #1 BUG TRACKING TOOL**

## **Knowledge of Existing Application**

It happens that new testers join the team, as the workload increases. The new team members gain knowledge of the new modules, subsystems and components assigned to them. This knowledge might be sufficient for major functional testing, but insufficient for the regression testing. You can not perform complete regression testing with partial knowledge.

Hence, it becomes a challenge for the testers to deliver the knowledge of existing features and functionalities to the new members. Similarly, new testers are not very comfortable in gaining knowledge of what happened in the past.

Sometimes this reluctance is result of the fact that existing application already contains a lot of modules or subsystems, with myriad business rules implemented. At other times, the testers that tested the early versions of the application might leave the team and consequently no one in the team might have detailed, correct knowledge of early features and business rules.

In such scenarios, it is mandatory that a consistent test management and requirement management approach is followed so that new testers can also perform regression testing effectively.

## **Expensive for Business**

Another challenge associated with the regression testing is the portrayal of its value to business people. It might happen that business units of the organization see regression testing as an

expense to the organization with little value to the business. They might consider it a drain of resources and manpower to test again and again, something which has already been developed, tested and deployed at early stages. Hence, the management can be reluctant in allocating the budget, time and resources to conduct the activity of software regression testing. Hence, the management can be annoyingly critical and demand justifications for every cycle of regression testing.

### **TRY REQTEST FREE FOR TEST MANAGEMENT**

As quality assurance manager, it is your duty to convince the management about the need and importance of regression testing. You will need to intelligently devise a regression test strategy which is efficient and effective. Remember that management is interested in giving you minimal resources, time and budget so make request accordingly and be prepared to justify your demand.

#### **Less Time for Regression Testing**

When testers are told to perform regression testing, they are tempted to perform exhaustive testing of the software product. However, the quality assurance team is not given much time to perform regression testing. Therefore, it is wiser of the quality assurance manager to make sure that the testing team does not exhaust all energies on a few modules – missing out the testing of other modules due to shortage of team. This means that the quality assurance manager needs to devise an intelligent methodology for regression testing to assure that every required test case has been executed within the limited span of time.

#### **Easy to lose track**

As new features and changes are implemented in the software product, more test cases are added to the regression test suite. As the product functionalities expands, testers are overwhelmed by the regression test cases and they fall victim to lose the track of test cases, overlooking the important test cases. This can be prevented by regularly monitoring the regression test suite and deleting the obsolete test cases. It is also important to avoid any duplication of test cases, else they will add to the unnecessary effort and frustration for testers.

#### **Selection of Regression Test Cases**

Selection of test cases for the regression test suite is yet another challenge of regression testing. You want to make sure that all points of integration and changes are being verified by the regression. At the same time, you can not overload your regression test suite with test cases as you get less time for regression testing. As a result, you need to choose the test cases wisely, regularly monitor the test suite and perform periodic cleanup for removing obsolete or unnecessary test cases from the regression test suite.

Another problem arises when there is a conflict of requirements. You need to be really mindful of the updated requirements and corresponding test cases. Any carelessness might result in retaining the test cases that have now become invalid and removing the valid test cases. Furthermore, make sure that all required test cases, for the new scenarios, have been created and moved to the regression test suite.

#### **How to make your Regression Testing Effective?**

Phew! We just discussed the various challenges at every step of regression testing. Now, you must have become uncomfortable and looking for the ways to overcome those challenges. Do not worry! We have got a few tips for you that will help you formulate an effective regression testing strategy. You can improve your regression testing by incorporating the following elements in your regression testing strategy:

## **Closely Monitor the Changes**

Closely monitoring of changes hold a vital importance in regression testing. We discussed that regression testing is performed when a change is made into the existing application. This follows that you need to closely monitor the changes in the requirements and their corresponding impact on the functionality of the application. Upon doing so, you will need to modify the test cases in your regression test suite i.e. add new test cases, delete the obsolete test cases, modify any expected result or test steps.

## **New Test Cases for Impact of Changes**

As new features and changes are implemented in the software product, they are integrated with the existing functionalities. These integrations can be really tricky and are most susceptible to the bugs and issues. This means that you can make your regression test strategy effective by analyzing the impact of changes and integrations of different modules, systems or sub-systems.

**TRY REQUEST FREE FOR REQUIREMENT MANAGEMENT**

The tip is to identify the points of integration and analyze how the changes, new features or bug fixes might have impacted the older features and functionality. Create new test cases for testing and verifying the correctness of integrated application. These test cases can be added to your regression suite now.

## **Identify Problematic Areas**

Another important step towards effective regression testing is to focus the problematic areas. Due to several factors, some modules of software have less bugs than others such as difference in complexity level of each module, expertise of the developer, clarity of requirements. This is also evident from the bug reports that some modules have larger number of issues, as compared to others. You can analyze the bug reports and identify major problematic areas. Similarly, you can analyze the user reported issues that occurred due to overlook in regression testing. In this manner, you will get a clear picture of problematic areas and tricky integration points. This follows that you need to focus and prioritize testing of these risky features. Take special care that you don't remove the corresponding test cases, during periodic cleanup of regression test suite.

## **Maintain Regression Test Suite**

Maintain a repository of regression test suites. Whenever a new change or integration occurs, create respective test cases and add to the regression test suite. You can also move some of the existing test cases into the list of regression test cases. The goal is to maintain all test cases at one place which should be executed at the time of every regression test cycle.

Not all test cases qualify for regression test suite – you selectively choose test cases into your regression suite. Regular screening of test cases is one of the ways to ensure accuracy and efficiency of test suites. Optimize your regression test suite by the categorization of test cases into reusable, re-testable and obsolete test cases. A well maintained regression test suite can be really handy for performing effective regression testing.

## **Periodic Cleanup of Obsolete Test Cases**

As we have discussed in challenges of regression testing that quality assurance team gets limited time, resources and considerable pressure from the management while performing regression testing. Therefore, it becomes essential for the quality assurance manager to conduct a periodic

cleanup of the regression test suite. This will help the QA to keep their priorities correct and spend their time on executing the required test cases only.

For example, you can delete test cases of outdated requirements. Otherwise, there can be a conflict if original and updated requirements state two different things. Similarly, there is a possibility that you created test case for verifying the integration of two modules, say Module A and Module B, and have verified its functionality several times. Now, there is no change or requirement that could have impacted the functionality of integrated Module AB. In such scenario, you can remove the test cases that were prepared for verification of integrated Module AB.

### **Random Testing of User Scenarios**

No matter how careful you have been while creating test cases and how many scenarios you have covered, nothing beats random testing at the end. Dedicate some amount of time to random testing of the application. These random tests might cover complete process cycles. You might also assume the roles of different users and try to depict real world scenarios as the users interacting with the system.

### **Rotate Your Workforce**

This point is more linked to human resource management. The quality assurance manager manages the team of people, so he should also take care of the motivation level and interest of his team members. If you ask the same tester to perform regression testing every time, he may get bored, disinterested or develop the tunnel vision. Eventually, he will lose the motivation and the quality of testing might drop which means defects could slip through into live release. So, a better approach is to keep rotating your workforce and give different testers the task of regression.

---

**What challenges of regression testing do you face? How do you overcome those challenges? How do you make your regression testing strategy effective? Do share your thoughts and feedback in the comment section!**