What are Test Reports?

A software testing report is like a summary of all the tests we did, what we wanted to achieve with them, and what happened during the testing project. It helps us see how well the testing went and figure out why some tests didn't work as expected.

After the software testing team accomplishes the testing work, the team generates the test reports containing information like the details of the testing procedures, the overall quality of the application, the test results, and the identified defects. It is a consolidated document of the summary of all the testing activities.

The other term for a test reporting document is the test closure report. The various project stakeholders read this document. You can generate a test report by ensuring that it has the following features:

- **Detailed information:** You should include all the details of the testing activities. Although you should concisely mention these details, you should not write an abstract. A detailed yet concise test report enables the stakeholders to form a clear picture of the testing activities in their minds.
- **Clarity:** When you include information in this document, you should confirm clarity in all aspects of the testing process.
- **Format:** You should adhere to a standard template for the test report. When it is in a standard format, the stakeholders can quickly review and understand.
- **Specifications:** You should include all test result specifications with brevity. These specifications should be to the point.

Why Test Reports?

Test reports enable the stakeholders to estimate the efficiency of the testing and detect the causes that led to a failed or negative test report. The stakeholders can evaluate the testing process and the quality of the specific feature or the entire software application. They can clearly understand how the team dealt with and resolved the defects.

Using all the preceding information, they can make product release decisions. In these ways, the data in the test report proves vital for the business.

Various stakeholders; such as product managers, analysts, developers, and testers; read the test report to determine the origin of each issue and the stage at which it has surfaced. Using this report, they can locate the causes of negative test results and thoroughly analyze the cause for the issues, such as weak implementation, unstable infrastructure, mismanaged back end, and defective automation scripts.

After going through the test reports, one can have adequate information to find answers to the following questions:

- Has the team avoided unnecessary testing?

- How is the stability of the tests?
- Do the testers in the team have adequate skills to identify the issues in an advanced stage?
- What value has the testing team accomplished after all the testing activities?

Test reports are essential because they enable monitoring testing activities, contribute to quality improvement of the software application, and make early product releases feasible.

Benefits of Test Reports

Test reports are helpful in the following ways:

- The relevant stakeholders can get all the updates about the project's status and the software application's quality.
- Based on these updates, the stakeholders can direct corrective measures and amendments.
- Test reports help stakeholders to decide whether to ship the software application to the customer.
- The report acts as a justification for all the efforts of the testing team. After going through the report, the testing team can learn and apply learnings in the future to make testing even more effective.
- The test reports ensure that only quality software applications are provided a release.

Templates of a Test Report

Here are some possible templates for a Test Report.

Template 1:

Project Overview					
Project basic information					
Project name					
Product name and version					
Product description					
Project description	<the mission of the project> <the output product of the project>				
	Project type				
Project duration	Start date		End date		
Test report					
Test cycle					
EXECUTED	PASSED			1111 (1111 indicates that this cell is for numbers)	
	FAILED			1111	
	Total tests executed (PASSED + FAILED)				1111
PENDING					1111
IN PROGRESS					1111
BLOCKED					1111
SUB-TOTAL (TEST PLANNED)					1111
(PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED					1111

Template 2:

Functions	Description	Percentage of test cases executed	Percentage of test cases passed	Test cases pending	Priority	Remarks

Sections of a Test Report

There are various templates for test reports. Out of these, you can choose the template that matches your requirements and customize the sections per your project's nature.

The broad categorization of sections of a report is the following.

Document Control

This section consists of two points: the Revision History and the Distribution List.

- The Revision History consists of information such as the date on which the team created the Test report, the date of the previous update, and the updated sections.
- The Distribution List consists of information such as the resources in the testing process, such as the test engineers, test leads, and test managers. The other information concerns those who have reviewed the test report, such as product managers and other stakeholders.

Project Overview

You need to include the description of the project with details, such as the name of the project, the type of the project, the duration of the project, the version of the product, and the description of the product.

Test Objective

You need to include the purpose of the software testing. This consists of the objectives of each testing stage, such as performance testing, security testing, regression testing, UI testing, functional testing, etc.

Further, you can mention the software testing team's several activities as part of the software testing. This test report section shows that the QA team has clear concepts about the test object and the requirements.

Test Summary

One should include information such as the count of the executed test cases, the pertinent data of passed and failed tests, the pass/fail percentage, and comments. A team can present this information more finely using visual representations such as tables, charts, graphs, and color indications.

Included Areas

This section consists of descriptions of the testing areas of the software application along with its functionalities. It is not essential to include every test scenario in the minutest details. You have to include all the areas at a high level.

Excluded Areas

This section consists of the testing areas of the software application that the testing team did not include. You have to mention every such testing area with the specific reason for exclusion. An example of such a reason is restricted access to device availability. It can happen that when the customers know about the excluded areas, they might raise the alarm. This is the main reason to jot down what the QA team needs to test, along with the expectations relevant to that specific area.

Testing Approach

You have to include information about what the testing team has tested and how this team has tested the specific area. Further, you have to mention the testing approaches of the team and the details of the various steps.

Defect Reports

Generally, the bug report has information about all the defects. However, if you include the defects' information in the test report, it is advantageous for the utility of the test reports. This report comprises information such as the total count of bugs the team has handled during the testing and the current status of these bugs (such as open, closed, and resolved).

The other details are the bugs marked as 'Deferred,' the bugs marked as 'Not a Bug,' reopened bugs, open bugs of the previous release, new bugs found, and total open bugs.

This is a crucial ingredient of the test report because the metrics in this report are adequate for correct decision-making and product improvement.

Platform Details

Currently, the testing team validates the software application across multiple platforms. Due to this, the team tests across various browsers, devices, and OS. You need to include all the platforms and environments related to testing the software application.

Knowledge Management

This consists of details such as the lessons the team has learned during the current testing cycle and a list of issues that need particular attention. This is followed by information about the work of the testing team in ensuring the application's quality. Also, it details the list of enhancements the company should implement in future testing cycles. Lastly, there are some suggestions or remarks for all the varied stakeholders.

Gross Summary

In this section, you need to include the feedback of the testing team, which is an overall opinion of the application under test. Through this summary, you communicate the details about the critical issues and those issues that are still open to the customer. After reading the summary, the customer can set expectations about the software application's shipping date.

In the case of larger organizations, the above sections of a test report are inadequate. The team should include additional data such as video recordings, screenshots, network traffic, logs, and other pertinent data, which enable the team to implement data-driven decision-making.

The testing team should generate the test report ideally at the end of the testing cycle. The motive behind this time selection is that the team should be able to include information about the regression tests.

However, the team should arrange for an adequate time interval between the date of submission of the test report and the date of shipping the software application to the customer. The reason for having this time interval is to enable the customer and the stakeholders to understand the overall health of the software application and the testing cycle so that the pertinent teams can make the necessary amendments.

After the test report generation, the testing team should share it with stakeholders, the customer, and all the team members. Due to this step, all members can get an overview of the testing cycle, which enables them to conclude about the ways to improve further. This report serves as an explanatory text for the novices on the team.

Types of Test Reports

Test reports are of the following three types.

1. **Test Incident Report**

During the testing cycle, the team enters the defect repository whenever they encounter a defect. This is termed the Test Incident Report. In this report, every defect is associated with a unique ID to facilitate the identification of the incident. Further, the team highlights the high-impact test incidents in the test reports.

2. **Test Cycle Report**

In every test cycle, the team plans and runs specific test cases. For each test cycle, the team uses a different software application build. The motive for doing so is that the team expects the software application to stabilize as it passes through the various test cycles.

The team creates the Test Cycle Report, which consists of the following information:

- A summary of all the steps the team has implemented in the test cycle.
- The defects the team detected during the specific test cycle based on impact and severity.
- The progress of the team in terms of the fixed defects from the previous cycle to the current cycle.
- The defects that the team has not fixed in the cycle.
- The observed variations in the schedule or efforts.

3. **Test Summary Report**

In the last step of the test cycle, the team recommends whether the product is suitable for release. The team creates the Test Summary Report, which consists of the summary of the outcome of a test cycle. This report includes the following sections:

- The Test Summary Report ID.
- Identification of the test items included in the report with the Test ID.
- The deviations from the test plans and the test procedures, if any.
- The result summary includes all the results with their resolved incidents and solutions.
- A well-rounded assessment and recommendation for the release, including the fit-for-release assessment and release recommendations.

The Test Summary report is of two types:

- The report generated at the end of every phase is known as the Phase-wise Test Summary.

- The final Test Summary Report.

How to create a Test Report?

Let us consider that AB is an online travel agency for which an organization is developing an ABC software application. The software testing team does the following while generating a test report.

The team notes all the activities it has done during the testing of the ABC application. Then, it documents the overview of the application.

The ABC application provides services to book bus tickets, railway tickets, hotel reservations, domestic and international holiday packages, and airline tickets. To do all these services, the application has modules such as Registration, Booking, Payment, etc. The team includes all such information in the testing report document.

Now let's see the steps to create a test report for an online travel agency.

Step 1: Create a Testing Scope

The team mentions those modules or areas that are in scope, out of scope, and untested owing to dependencies or constraints.

- **In-scope:** We completed the functional testing of the following modules:
    - User registration
    - Registration confirmation
    - Ticket booking
    - Hotel package booking
    - Payment
- **Out of scope:**
    - Multi-tenant user testing
    - Concurrency
- **Untested modules:**
    - The User Registration page that has the field values in mixed cases

Step 2: Test Metrics

Test metrics include the following:

- The count of planned test cases
- The count of executed test cases

- The count of passed test cases
- The count of failed test cases

The usage of test metrics is to analyze test execution results, the status of the cases, and the status of the defects, among others. The testing team can also generate charts or graphs to represent the distribution of defects: function-wise, severity-wise, or module-wise.

Step 3: Implemented Testing Type

The team includes all the types of testing it has implemented on the ABC application. The motive for doing so is to convey to the readers that the team has tested the application properly.

- **Smoke testing**

  When the QA team receives the build, the team implements smoke testing to confirm whether the crucial functionalities are working as expected. The team accepts the build and commences testing. After the software application passes the smoke testing, the testing team gets the confirmation to continue with the next type of testing.

- **Regression testing**

  The team conducts testing not on a particular feature or defect fix but on the entire software application. It consists of defect fixes and new enhancements. This testing confirms that after these defect fixes and new enhancements exist in the software application, the application has rich functionality. The team adds and executes new test cases to the new features.

- **System Integration testing**

  The team performs system integration testing to ensure that the software application is functioning as per the requirements.

Step 4: Test Environment and Tools

The team notes all the details of the test environment used for the testing activities (such as Application URL, Database version, and the tools used).

The team can create tables in the following format.

Application URL	
Database	
Tools used	

Step 5: Learnings during the Testing Process

The team includes information such as the critical issues they faced while testing the application and the solutions devised to overcome these issues. The intention of documenting this information is for the team to leverage it in future testing activities.

The team can represent this information in the following format.

Sr. No.	The Issue-Faced	Solutions

Step 6: Suggestions or Recommendations

The team notes suggestions or recommendations while keeping the pertinent stakeholders in mind. These suggestions and recommendations serve as guidance during the next testing cycle.

Step 7: Exit Criteria

When the team defines the exit criteria, it indicates test completion on the fulfillment of specific conditions, such as the following:

- The team has successfully executed all its planned test cases.
- The team has closed all the critical issues.
- The team has planned the actions for all open issues, which it will address in the next release cycle.

Step 8: Sign-off

If the team has fulfilled the exit criteria, the team can provide the go-ahead for the application to 'go live.' If the team has not fulfilled the exit criteria, the team should highlight the specific areas of concern. Further, the team should leave the decision about the application going live with the senior management and other top-level stakeholders.

4. Test Execution Report

The software Test Execution report is one of the important deliverables offered to the client after completing the software testing and development process. It provides a comprehensive testing process summary with accurate details and information. The report should be well-organized, concisely providing all essential information.

It usually follows a set of templates that contains an overview, summary, and analysis of the test plan, test cases, issues encountered during validation tests, defects list, and defect tracking information. The report also includes detailed information on the various tools used for testing and their effectiveness in finding critical defects in the developed software product.

A typical test report has the following template:

- Test Summary Report Identifier
- Summary
- Variances
- Comprehensive Assessment
- Summary of Results
- Evaluation

- Summary of Activities
- Approval

What is a Test Log?

The test log is one of the essential test artifacts created during the test execution. It contains a historical record of events related to each test. Also, it provides information related to different test operations that include a source of issues and reasons for the failed operation.

You can use a test log to enable post-execution debugging of failures and defects related to the product or application. When we talk about test artifacts, it helps to establish transparency among the team members and is appropriately recorded with accurate information. With the help of test artifacts, it becomes easier for the concerned team members to track the changes done in the software and become aware of the latest progress related to testing activities.

Here are some key test artifacts that are used during the testing process.

- Test strategy.
- Test plan.
- Test case.
- Test data.
- Requirement traceability matrix.
- Test coverage matrix.
- Test scripts.
- Test log.
- Defect reports.
- Test closure report.

The following operations you can do when you are working with logs:

- **View**: You can view the required test results in the supported format. It contains the event details that are part of a test run or scheduled run.
- **Export**: You can export the contents of the log to a text file.
- **Import**: You can import the contents of a log if the need arises.

Status of the Test Log

The log contains the following statuses for each test run:

- **Pass**: It indicates that all the verification points matched or received as per the expected response..

- **Fail**: It indicates that at least one of the verification points is not working as per the expected response.
- **Error**: It indicates that there is no valid response received from the server or the primary request was not sent successfully.
- **Inconclusive**: This status message is shown when there is no conclusive evidence on the verdict

Components of Test Log

The logs generated after test execution contain entries that present comprehensive and relevant information about different aspects of the test runs. The following components are part of it.

- Date and time details of when the failure, event, or any other significant issue was encountered.
- The details related to the event logged.
- Different statuses recorded by the team.
- Contextual details and information.
- Error details and anomalies.

Test Log Template

The testing team can follow the test log template standardized as per IEEE Standard 829-1998, a universal standard. The template provides the following details:

- **Identifier**: It is a unique identifier for the log. It defines the software level it is associated with. Key details such as version date and number are also a part of the log identifier.
- **Description**: Once the unique identifier is assigned, the team offers a detailed description of the teams that need to be tested. Any other supporting information, reference, or dependencies are also highlighted in the description part.

  Other information included in the description are: date and time, test environment, case, and procedural specifications.

- **Activity and event details**: The team defines the list of activities and events that need to be performed by them.

Advantages of Test Log

Even though creating logs is usually considered a time-consuming activity, it is one of the most essential and beneficial tasks. In addition to tracking testing activities, it allows the team to address various software application issues.

Below are a few of the advantages of a logs.

- It provides information related to passed and failed tests.
- It allows the team to track and monitor test execution activities and functionalities closely.
- Provides the complete summary related to the test run.
- It allows the team to reinvent their testing strategies.
- The respective team can take appropriate measures related to recovery and reporting.

Test Log vs. Test Case

In this section, let's try and compare these two based on different parameters.

Test Log	Test Case
It is a crucial artifact where detailed information about each test helps you to decide the software quality and performance.	It is a comprehensive document containing elements ranging from test data, preconditions, expected results, and post-conditions. It helps the testing team to make informed decisions about the working of software applications.
It supports different statuses based on the success or failure of each test.	The test case can either pass or fail based on functionality.
This is prepared by the testing team	It is prepared by the testing team, but developers can also contribute to it.
There is a standard template designed as per IEEE standards.	No standard template is followed for this, as different teams can customize it according to their testing needs and objectives.

Who creates a Test Log?

Test Log is all about monitoring and tracking critical software testing activities. The primary responsibility lies with the testing team because they know the software life cycle from a testing perspective. Logs are prepared whenever tests are executed, or test scripts are implemented by the team. Its entries can include references to images, files, and other important information related to the application.

Testers often work with business analysts and product owners to develop high-level test scenarios and review end-to-end test cases. Using this approach, product owners are closer to the users and have a better understanding of how they approach the problem. To ensure that the technically complicated functionality workflows are thoroughly tested, it is better to consult the developers. That is why the testing team needs timely inputs from required stakeholders to update the logs with correct, up-to-date information.

Role of Software testers in Test Log Report

Organizations are always looking to hire exceptional software testers who can create an effective and detailed log report. It is always a preferred option to have software testers be technically sound to contribute to the product's success in the long run. Software testers, the chief contributors to the log, need to understand the domain and product requirements.

Some of the key activities performed by software testers:

- Ensure that accurate recording of the observed outcome of each test executed in the test cycle.
- Validate that the logs are uniquely identified and stored against the correct test cycle.
- Monitor the issues or errors in the logs and share them with required project stakeholders.
- Take appropriate recovery and reporting actions.
- Help to improve the log templates.
- Understand the testing lifecycle and bring improvements in the long run.

Strategies for dealing with Test Log Data

You need to figure out the overall strategy for dealing with testing log data along with using the logging tools which are best suited for your test environment and the organization’s specific testing needs. The best place to begin is to take full advantage of the testing system’s built-in logging features.

Suppose you need logging capabilities that the test system does not provide. In that case, you can consider integrating third-party test logging tools with your testing system to provide the required features and capabilities.

Some strategies you can adopt in the long run include

- **Full log integration support:** You can have a system that allows you to integrate with all the required log details. Many benefits are offered when you are providing full log integration support. It is beneficial to the teams that are working in close collaboration. You can keep track of the system values logged at the testing time against the values logged in real-time.
- **Dashboard support to view data trends:** If you use a log integration tool that provides dashboard support, you can get a quick summary of the log data from different sources. The dashboard is a much easier way to manage your logs and establish key relationships between data requirements.

How to share a Test Log with the team?

Once the logs are generated, they should be shared with stakeholders, customers, and the team. Logs provide team members with a comprehensive view of the testing cycle, which can be used to improve the testing process.