1) Identify a computer networked IT system to be tested in your organisation, and answer the following questions pertaining to it:

a) What is the purpose of the test that you will be conducting?

b) What data will you obtain from the test?

c) What factors are you aware of that could affect the test procedures that you will conduct?

d) What resources do you have available in order to conduct the test?

e) What compliance factors should you be aware of?

2) What are the objectives of system testing in your company?

#### Answer text

1). Identify a computer networked IT system to be tested in your organisation, and

answer the following questions pertaining to it:

a)What is the purpose of the test that you will be conducting?

The test will show if the system is working according and the results shows if

there is something wrong with it, so that we can provide solutions to those problems.

b)What data will you obtain from the test?

Information based on the test including all the errors found during the test and some technical issues that will need to be fixed.

c). What factors are you aware of that could affect the test procedures that you will conduct?

Poor Network

Slow processor on the device used to test the system.

A test is reliable if it accurately measures a quality and is repeatable.

d). What resources do you have available in order to conduct the test?

good and reliable device,

good internet connection,

fast processor that will also monitor the test in making sure that nothing goes wrong.

e). What compliance factors should you be aware of?

Safety: Safety is one of the key concerns that drive the need for compliance testing, Negligence, cutting corners, or unawareness of safety standards may result in safety issues that may be overlooked without external testing.

Quality: Compliance testing ensures the quality, efficiency, and efficacy of processes, products, and services. Periodic audits ensure that performance is as desired.

Legality: In the case of mandatory testing, it would be illegal to release a product or service into the market without it having proven to satisfy some stipulated.

● Finding defects which may get created by the programmer while developing the software.

● Gaining confidence in and providing information about the level of quality.

● To prevent defects.

● To make sure that the end result meets the business and user requirements.

● To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.

● To gain the confidence of the customers by providing them a quality product.

Objectives of Testing. The objectives of the testing are the… | by Senuri Samindi | Medium

#### Answer files

#### Comments

Comment:

### Question **2**

Complete

Mark 1.00 out of 1.00

Flag question

#### Question text

1) Based on the test identified in Specific Outcome 1, select a testing team in your company and prepare a test plan. Ensure that the following are included:

a) Description of the system to be tested

b) Test strategy to be followed

c) Test objectives

d) Test criteria

e) Resources including test team

f) Test environment

g) Schedule

h) Test scenarios and test data to be used, including expected outcomes

#### Answer text

1) Based on the test identified in Specific Outcome 1, select a testing team in your company and prepare a test plan. Ensure that the following are included:

a)Description of the system to be tested

Testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the

inner design of the code is not a pre-requisite and is done by the testing team. System testing is performed in the context of a System

Requirement Specification (SRS) or a Functional Requirement Specifications (FRS). It is the final test to verify that the product to be delivered meets the specifications mentioned in the requirement document. It should investigate both functional and non-functional requirements.

What is System Testing? Types &amp; Definition with Example

b)Test strategy to be followed

Optimisation of the approach to testing in software engineering is the best way to make it effective. A software testing strategy defines what, when, and how to do whatever is necessary to make an end-product of high quality.

Static Testing: The early-stage testing strategy is static testing: it is performed without actually running the developing product. Basically, such desk-checking is required to detect bugs and issues that are present in the code itself. Such a check-up is important at the pre-deployment stage as it helps avoid problems caused by errors in the code and software structure deficits.

Static Testing | Know Tools &amp; Essential Static Testing Technique Involved

c)Test objectives

● Finding defects which may get created by the programmer while developing the

software.

● Gaining confidence in and providing information about the level of quality.

● To prevent defects.

● To make sure that the end result meets the business and user requirements.

● To ensure that it satisfies the BRS that is Business Requirement Specification

and SRS that is System Requirement Specifications.

● To gain the confidence of the customers by providing them a quality product.

Objective test edu4

d)Test criteria

Test criteria help the tester to organise the test process. They should be chosen in accordance with the available test effort. Test coverage measures are defined as a ratio between the test cases required for satisfying the criteria and those of these which have been executed. The measures are used to obtain information about the completeness of integration tests. The approach is described for data flow and control flow oriented criteria and measures. The intention is to enable the tester to specify integration tests in advance in terms of effort, and to evaluate the results in terms of test completeness.

6 Basic Criteria For Testing Requirements - QATestLab Blog

e) Resources including test team

● Team meeting/training/webinars – GoToMeeting/WebEX/Adobe connect/Skype

● Screen recording software.

● Record your ideas or explain complex problems to your team – ScreenChomp

● Capture images and screen videos – SnagIt

● Free screen capture tool – qSnap

● Shared files or ideas on cloud space – Dropbox or JustCloud or GoogleDrive

The best screen recording software in 2021 | Zapier

f)Test environment

Testing is used to ensure the quality and functionality of the application, identify open bugs and review bug fixes. A test environment allows software developers to check how a code/program will behave in a live environment. The testing environment should closely resemble the production environment since it is one of the last safe places to find and fix environment-related bugs before moving into production. A test environment is built by setting aside storage, computing and additional resources required for testing. This may include new devices, physical or virtual, provisioned for testing use cases as defined by developers.

Performance Testing Environment: this environment allows developers to measure how fast an application responds to an interaction. Various aspects of an application, such as page load speeds, input processing, stability, reliability and so on, are tested in this environment.

System Integration Testing (SIT): the main purpose of system integration testing is to check if all modules, such as code modules, individual applications, and client and server applications, can communicate with each other as per the dependencies. It also ensures shared databases work as expected.

UAT: acceptance or user acceptance testing is conducted to check if the application meets the business requirements of the end users. This is the final stage of testing. If the features and functionalities of the application satisfy the end user, the application is moved to the production environment.

Types of Testing Environments – Test Environment Management (DOT) Com

g)Schedule

Making schedule is a common term in project management. By creating a solid

schedule in the Test Planning, the Test Manager can use it as tool for monitoring the

project progress, control the cost overruns.

● Help people outside the test team such as developers, business managers,

customers understand the details of testing.

● It guides our thinking. It is like a rule book, which needs to be followed.

● Important aspects like test estimation, test scope, so it can be reviewed by

Management Team and re-used for other projects.

To create the project schedule, the Test Manager needs several types of input as below:

● Employee and project deadline: The working days, the project deadline, resource

availability are the factors which affected to the schedule

● Project estimation: Base on the estimation, the Test Manager knows how long it

takes to complete the project. So he can make the appropriate project schedule

● Project Risk : Understanding the risk helps Test Manager add enough extra time

to the project schedule to deal with the risks

Rainbow Weekly Schedule Template

h)Test scenarios and test data to be used, including expected outcomes

Test Document in Good Shape: The best and simple way to organize your test document is by splitting it into many single useful sections. Divide the entire testing into multiple test scenarios. Then divide each scenario into multiple tests. Finally, divide each case into multiple test steps. If you are using excel, then document each test case on a separate sheet of the workbook wherein each test case describes one complete test flow.

Do not Forget to Cover the Negative Cases: As a software tester, you need to be innovative and draw up all the possibilities that your application comes across. We, as testers, have to verify that if any unauthentic attempt to enter the software or any invalid data to flow across the application should be stopped and reported. a negative case is as important as a positive case. Make sure that for each scenario, you have two test cases- one positive and one negative. The positive one should cover the intended or normal flow and the negative one should cover the unintended or exceptional flow. The more simple and clear-headed a test step is, the easier it would be to proceed with testing.

Prioritize the tests: We often have stringent timelines to finish up testing for an application. Here, we may miss testing some of the important functionalities and aspects of the software. In order to avoid this, tag a priority with each test while documenting it.

#### Answer files

#### Comments

Comment:

### Question **3**

Complete

Mark 1.00 out of 1.00

Flag question

#### Question text

1) In your team, conduct the test as planned in Specific Outcome 2. Document the following:

a) Actions taken to prepare for the test

b) The test procedures followed

c) Any problems identified and recommended action

#### Answer text

1) In your team, conduct the test as planned in Specific Outcome 2. Document the following:

a). Actions taken to prepare for the test:

Basic functionality testing: Begin by making sure that every button on every screen works. You also need to ensure that you can enter simple text into each field without crashing the software. You do not have to try out all the different combinations of clicks and characters, or edge conditions, because that is what your testers do, and they are really good at that.

If the feature is designed to be accessed by way of an API, you need to run tests to make sure that the basic API functionality works before submitting it for more intensive testing. If your basic functionality testing detects something that does not work, that is fine. Just tell them that it does not work, that you are aware of it, and that they should not bother trying it. You can fix it later, just do not leave any surprises in there.

Code review: Another pair of eyes looking at the source code can uncover a lot of problems. If your coding methodology requires peer review, perform this step before you hand the code over for testing. Remember to do your basic functionality testing before the code review, though.

Static code analysis: There are tools that can perform analysis on source code or byte code without executing it. These static code analysis tools can look for many weaknesses in the source code, such as security vulnerabilities and potential concurrency issues. Use static code analysis tools to enforce coding standards, and configure those tools to run automatically as part of the build.

17 Ways to Prepare for a Test - wikiHow

b) The test procedures followed:

● Step-1: Assess Development Plan and Status.

● Step-2: Develop the Test Plan.

● Step-3: Test Software Requirements.

● Step-4: Test Software Design.

● Step-5: Build Phase Testing.

● Step-6: Execute and Record Result.

● Step-7: Acceptance Test.

● Step-8: Report Test Results.

TEST PLAN: What is, How to Create (with Example)

c) Any problems identified and recommended action:

Building test tools instead of testing: Smart testers are constantly looking for better ways to create and distribute manual test steps. Automation testers try to improve things by building new frameworks or utilities. That’s a good practice when you have the time, but if it was not part of your plan then your resources are being diverted from completing the project. The problem is not that they are working to improve your tools and processes, the problem is when the activity gets substituted for actual testing. If that’s happening, you’ve got to get the team back on track as quickly as possible. Put the tool work on hold or schedule time for tools in a separate project, give them their own summary, goals and metrics.

The tested system isn’t testable enough:

At first this statement doesn’t make sense, because anything can be tested. But if your developers work with you to make it more testable, the testing can be more robust.

Some tips on how to avoid this:

Get in line with development as soon as possible. It saves time in the long run. This works double for developers when test time comes around – because if the requirements are unclear or haven’t been thought of from the tester’s perspective, they may not be testable because of the lack of clarity. For instance, having a requirement that simply reads, “The application should be responsive” is not quantifiable. Does that mean the application loads in 10 seconds? Does that mean every click needs less than a 5-second delay to the next screen? These specifics are essential for the tester’s job.

#### Answer files

#### Comments

Comment:

### Question **4**

Complete

Mark 1.00 out of 1.00

Flag question

#### Question text

1) Record the outcomes of the test conducted and draft a report. The report should include:

a) What was tested and how it was tested.

b) The results of the test conducted.

c) An analysis of the results as documented based on the documented test scenarios.

d) Any problems identified and action taken to address these.

2) How does recording ensure that the data is sufficient to meet the purpose of the test in the company?

3) What tools and information systems exist in your company to assist with planning and execution of tests, and capturing of results?

#### Answer text

1) Record the outcomes of the test conducted and draft a report. The report should include:

a) What was tested and how it was tested:

An early start to testing reduces the cost and time to rework and produce error-free software that is delivered to the client. However in Software Development Life Cycle(SDLC), testing can be started from the Requirements Gathering phase and continued till the deployment of the software.

It also depends on the development model that is being used. For example, in the Waterfall model, formal testing is conducted in the testing phase; but in the incremental model, testing is performed at the end of every increment/iteration and the whole application is tested at the end.

Testing is done in different forms at every phase of SDLC:

● During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.

● Reviewing the design in the design phase with the intent to improve the design is also considered as testing.

● Testing performed by a developer on completion of the code is also categorised as testing.

It is difficult to determine when to stop testing, as testing is a never-ending process and

no one can claim that a software is 100% tested. The following aspects are to be

considered for stopping the testing process:

● Testing Deadlines

● Completion of test case execution

● Completion of functional and code coverage to a certain point

● Bug rate falls below a certain level and no high-priority bugs are identified

● Management decision.

Best Practices for a Secure Software Development Life Cycle (SDLC)

b) The results of the test conducted:

Testing is Too Expensive: pay less for testing during software development or pay more for maintenance or correction later. Early testing saves both time and cost in many aspects, however reducing the cost without testing may result in improper design of a software application rendering the product useless.

Time-Consuming: during the SDLC phases, testing is never a time-consuming process.

However diagnosing and fixing the errors identified during proper testing is a time-consuming but productive activity.

A Tested Software is Bug-Free: No one can claim with absolute certainty that a software application is 100% bug-free even if a tester with superb testing skills has tested the application.

Missed Defects are due to Testers:

It is not a correct approach to blame testers for bugs that remain in the application even

after testing has been performed. This myth relates to Time, Cost, and Requirements

changing Constraints. However the test strategy may also result in bugs being missed

by the testing team.

The results of the t-test conducted to determine whether there were... | Download Scientific Diagram

c) An analysis of the results as documented based on the documented test scenarios:

Only Fully Developed Products are Tested: testing depends on the source code but reviewing requirements and developing test cases is independent from the developed code. However iterative or incremental approach as a development life cycle model may reduce the dependency of testing on the fully developed software.

Complete Testing is Possible: It is possible that all paths have been tested by the team but occurrence of complete testing is never possible. There might be some scenarios that are never executed by the test team or the client during the software development life cycle and may be executed once the project has been deployed.

How to Review SRS Document and Create Test Scenarios – Software Testing Training Course on Live Project – Day 2

d) Any problems identified and action taken to address these:

A Tested Software is Bug-Free: No one can claim with absolute certainty that a software application is 100% bug-free even if a tester with superb testing skills has tested the application.

Missed Defects are due to Testers: It is not a correct approach to blame testers for bugs that remain in the application even after testing has been performed. This myth relates to Time, Cost, and Requirements changing Constraints. However the test strategy may also result in bugs being missed by the testing team.

Testers are Responsible for Quality of Product: testers responsibilities include the identification of bugs to the stakeholders and then it is their decision whether they will fix the bug or release the software. Releasing the software at the time puts more pressure on the testers, as they will be blamed for any error.

Test Automation will used wherever possible to Reduce Time: it is true that Test Automation reduces the testing time, but it is not possible to start test automation at any time during software development. Test automaton should be started when the software has been manually tested and is stable to some extent. Moreover, test automation can never be used if requirements keep changing.

What is automated QA testing?

2) How does recording ensure that the data is sufficient to meet the purpose of the test in the company:

It is only through an effective data analysis that the management can keep a track of their performance. It is through this that the company develops an insight about where it stands in the market. Data analysis is the basic and most crucial component of business and company research and therefore, it becomes all the more essential to maintain the neutrality, credibility, quality and authenticity of the collected data. Data analysis will only provide a realistic insight and will suggest real-time and practical measures if the data is genuine and free from any machine or manual errors, discrepancies or loopholes. Therefore, it is essential for data analysts and company management to ensure the accuracy of data and information collected on a regular basis from various company operations and procedures. The efficacy and accuracy of the data collection process can be improved by incorporating the following measures in the data collection techniques. Data resources play the most crucial role in effective data collection and further analysis. You need to ensure the reliability and credibility of the systems and personnel responsible for data and information generation.

Chapter 7: Audit evidence

3) What tools and information systems exist in your company to assist with planning and execution of tests, and capturing of results:

Practice-Test.

Free A+ Software Practice Test - CertBlaster

Requirements and Test Management for Jira (RTM).

3 Ways to Bring Requirements and Test Management Into Jira | Stiltsoft