Debre Markos University
Institute of Technology
School of Computing, Software Eng A/program

Software Quality Assurance and Testing (SEng4112)

Chapter Two

Software Testing

Contents

- Introduction
- What is software testing
- Basics of Software Testing
- Software Testing Process

Introduction

- On average, software developers spend 50% of their programming time on finding and fixing bugs.
- Professional programmers have 1-3 bugs per 100 lines of code after it is "done".
- The global cost of software debugging rises to $312 billion annually.
- Software bugs cost the U.S economy an estimation of $59.5 billion per year.
- About $22.2 billion could be eliminated by improved testing that enables earlier and more effective identification and removal of defects.

Cont...

- Anyone who has worked with computers or software at tests to the wonders that technology can accomplish.
- But for software companies, creating the actual software is only half the battle; devising the method of testing it is equally challenging.
- And in order to create excellent software, an effective and efficient software testing is needed.

What is software testing?

- Software testing is
  - an activity of checking whether the actual results match the expected results and to ensure that the software is defect free.
  - the process of verifying and validating a software application to check whether it is working as expected or not.
- The intent is to find defects and improve the product quality.
- Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements.
- It can be done either manually or using automated tools.

Why software testing is required?

- No matter how well the software design looks on paper, once the development starts and you start testing the product, you will definitely find lots of defects in the design.
- It is not possible to create software with zero defects
- The purpose of testing is to find problems – find as many problems as possible
- The purpose of finding problems is to fix them – then fix the most important problems, as there isn't enough time to fix all of them.

Why...

- **Generally**, software testing is required
  - ✓ To ensure that the product is inline with the requirement of the client
  - ✓ To ensure that the system is free from any bug that can cause any kind of failure
  - ✓ To ensure the quality of the product
  - ✓ To check the reliability of the software
  - ✓ To make sure that the final product is user friendly
  - ✓ To stay in the business

Who does the software testing

- There is often a debate on who should actually test the software. People often question that why developers are not allowed to test.
- A developer generally checks his code several times before he submits it for testing and still in most cases it is never error-free because a developer is generally blind to his own mistakes.
- A tester on the other hand looks at software from clients' point of view. He is unbiased and his focus is only on the specifications and the requirements. So, a tester is able to look into areas that a developer may ignore. So, the testing should always be carried out by independent testers.

Who...

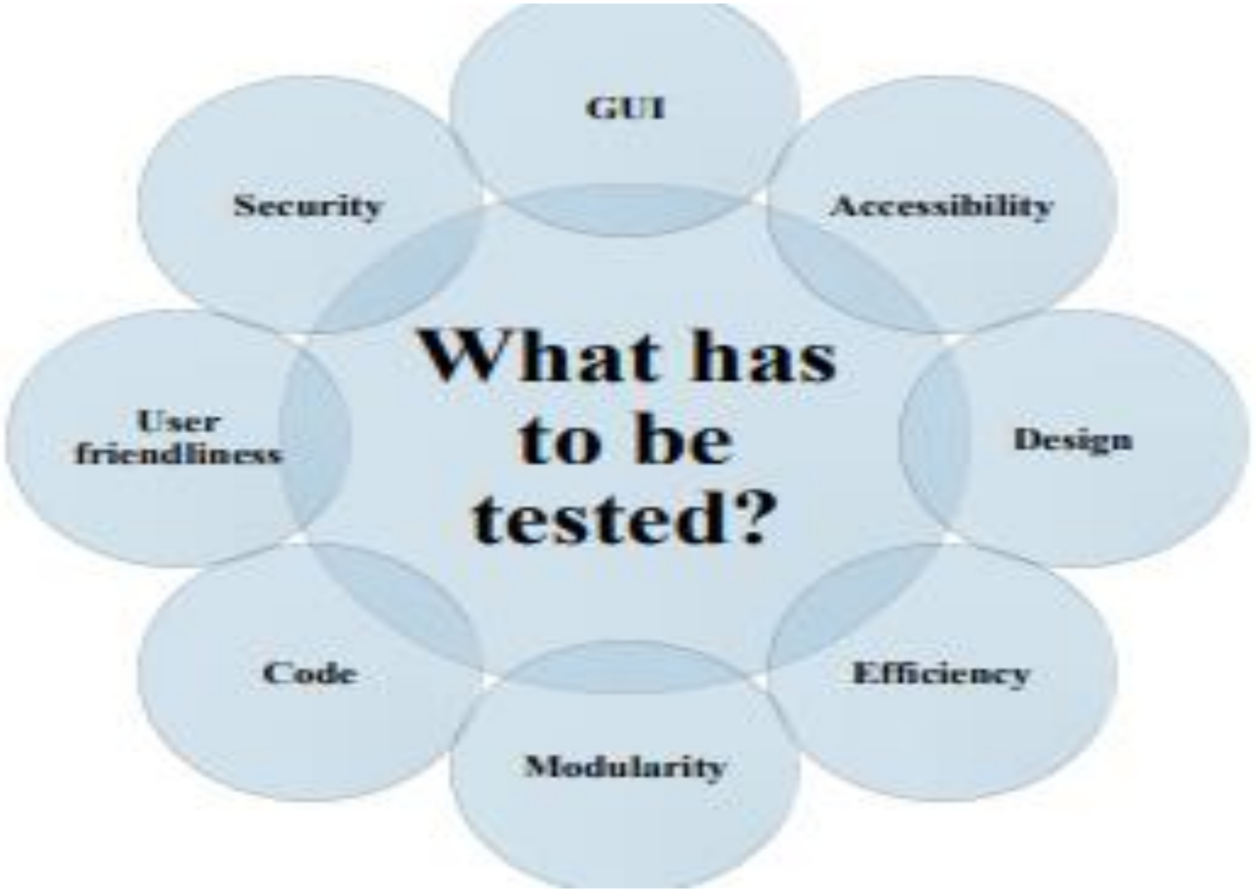
- Many times, developers share their work amongst each other and test each other's work. This is known as buddy/pair testing.
- When the development and testing teams are different, there is often a communication gap and sometimes, developers become careless towards coding and do not revise their code because they think that it is all a tester's job thereby increasing the burden on the tester.

Who…

- Every development team should have dedicated testers and every project generally has at least one dedicated testing team.
- Some companies believe in having separate teams for different types of testing, this means different teams for usability, performance, security and other forms of testing.
- Some companies believe in outsourcing software testing work which means they hire a firm or independent testers or consultants to have a look at and test their project.

What has to be really tested

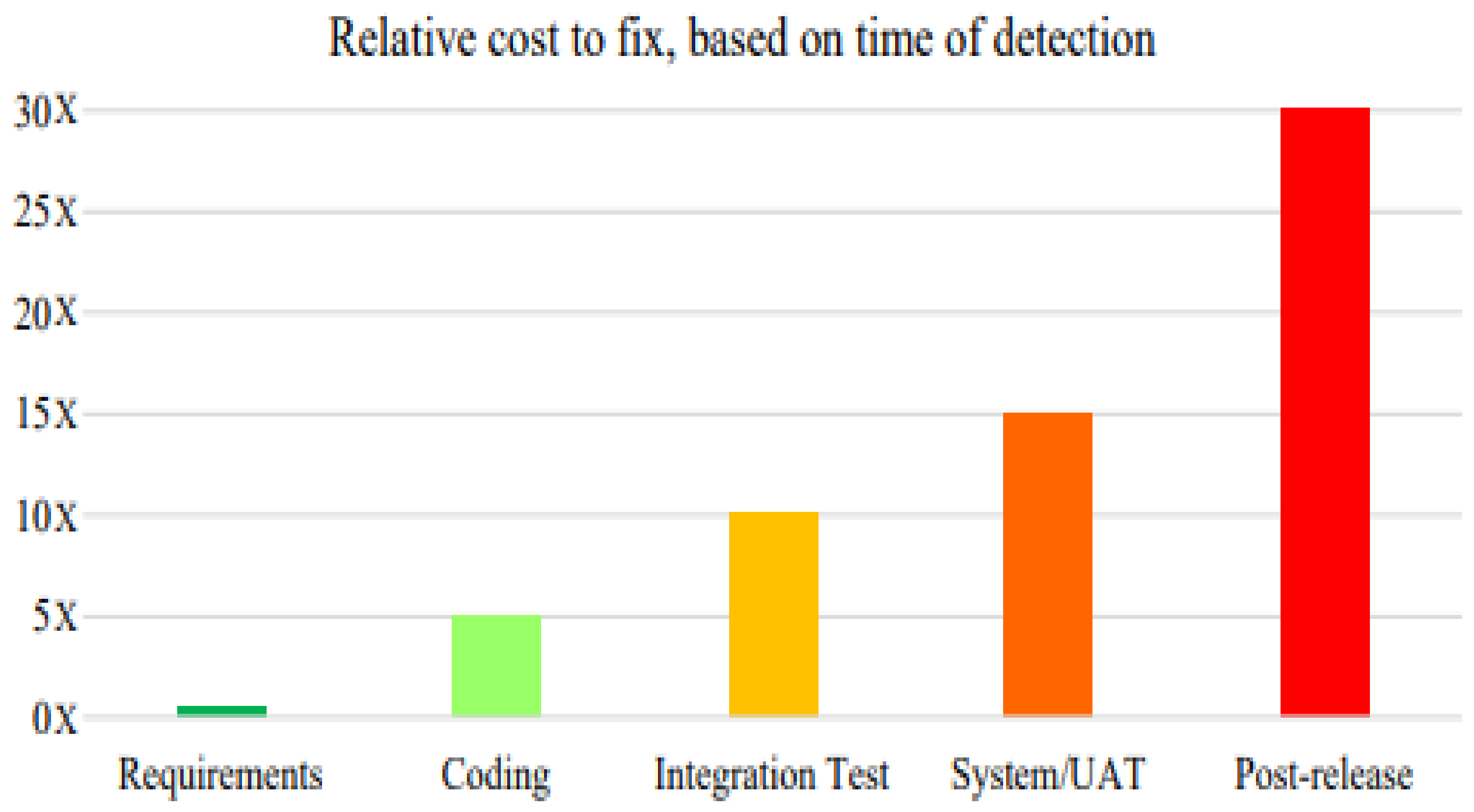
- The tester should have a good understanding about the project requirements.



When is the software testing done?

- The earlier the testing team starts testing the software, the easier it would be for the developers to complete the project on time and this would also save a lot of time, money and effort.
- Starting testing in the later stages of development can turn out to be an expensive matter as it is very difficult to rectify defects once the software has reached the final stages of development.
- Dividing software development into stages and then testing work done in every stage before moving on to the next stage helps in finishing the software development in time with good results.

When…



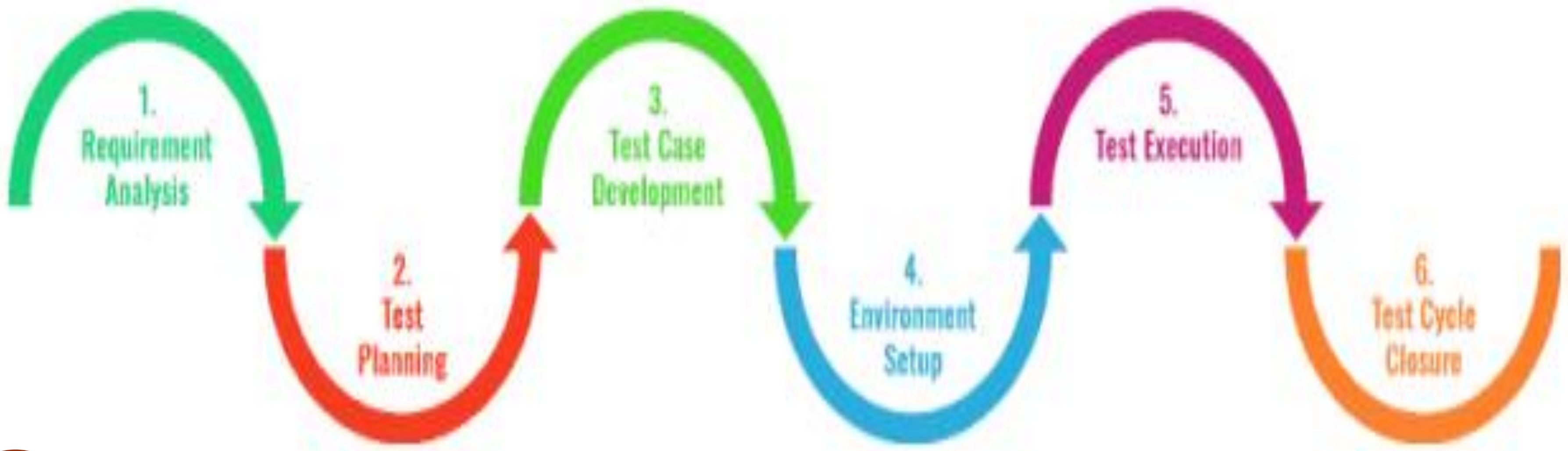
Source: National Institute of Standards and Technology

Software testing process

- A process is a series of activities performed to fulfill a purpose and produce a tangible output based on a given input.
- Processes can be described and hence monitored and improved. A process description must always include:
  - A definition of the input
  - A list of activities—the procedure
  - A description of the output

Software testing life cycle (STLC)

- Software testing life cycle defines the stages in testing of software.
- It is executed in a systematic and planned manner.
- In STLC process, various activities are carried out to improve the quality of the product.
- STLC, in general, comprises of the following phases:



1. Requirement Analysis

- Before we go for designing any software, we need to analyze the requirements. Like
  - ✓ What is the purpose of the application?
  - ✓ How many users are going to use this?
  - ✓ What is the traditional process of the client and how our software is going to help our client?
- Here testing team gathers as much information as possible about the software which they are going to test.
- It's obvious that if the testing team has good knowledge about the software, then it can test it very well.

2. Software test planning

- Software test planning is the practice of documenting software testing requirements in an organized manner.
- The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

Cont...

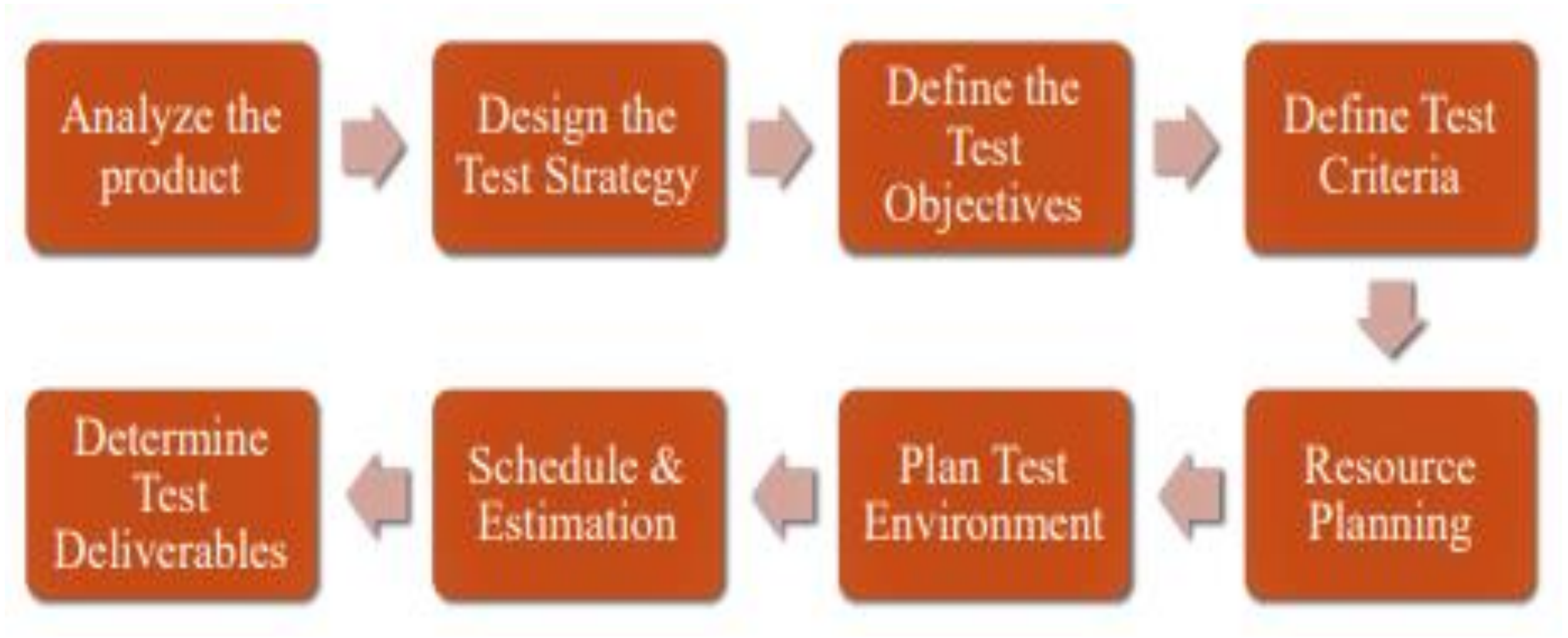
- A test plan is a detailed document that outlines the
  - ✓ test tasks, (what)
  - ✓ test strategy, (how)
  - ✓ test approach, (how)
  - ✓ testing objectives, (why)
  - ✓ resources (manpower, software, hardware) required for testing,
  - ✓ test schedule, (when)
  - ✓ test estimation and (how much)
  - ✓ test deliverables. (outputs)

Importance of Software Test Plan

- Test plan guides the testing phases to be followed.
- Test plan helps us determine the effort needed to validate the quality of the application under test
- Helps people outside the test team such as developers, business managers, customers understand the details of testing.
- It can be reviewed by the management team and re-used for other projects.

How to write a Software Test Plan

- Follow the eight steps below to create a test plan as per IEEE 829 Standard

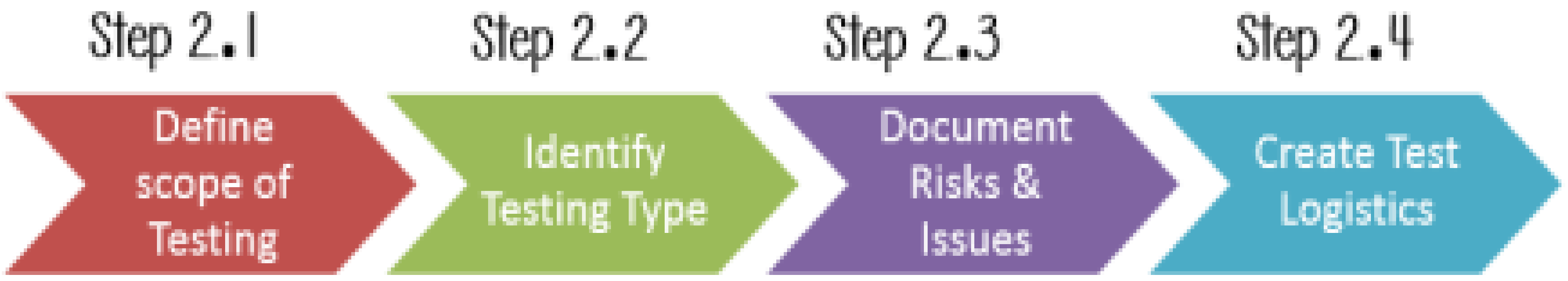


I. Analyze the product

- You must learn a product thoroughly before testing it. How can you test a product without any information about it? The answer is **impossible**.
- You should research clients or the end users to know their needs and expectations from the application.
  - ➢ Who will use the software?
  - ➢ What is it used for?
  - ➢ How will it work?
  - ➢ What software/ hardware the product uses?

II. Develop Test Strategy

- Test strategy is a critical step in making a test plan.
- A test strategy is a high-level document which is usually developed by test manager.
- This document defines:
  - The project's testing objectives and the means to achieve them
  - Determines testing effort and costs
- You should follow the steps below



2.1. Define Scope of Testing

- Before the start of any testing activity, scope of the testing should be known. You must think hard about it.
- The components of the system to be tested (hardware, software, middleware, etc.) are defined as "in scope"
- The components of the system that will not be tested also need to be clearly defined as being "out of scope."
- Defining the scope of your testing project is very important for all stakeholders. A precise scope helps you
  - Give everyone confidence & accurate information of the testing you are doing
  - All project members will have clear understanding about what is tested and what is not

How you determine scope of

- To determine scope, you must have
  - Precise customer requirement
  - Project budget
  - Product specification
  - Skills and talent of your test team
- For instance, if a customer wants you to conduct stress and load testing, but the project budget does not permit to do so. In such a case what will you do?

Cont...

- Well, in such case you need to convince the customer that stress and load testing is extra work and will consume significant resources. Give him data supporting your facts. Tell him if we conduct these tests, the budget will increase by XYZ amount. If the customer disagrees, you have to modify your “in scope” and “out of scope” testing
- As the software requirement specs, only focus on testing the **functions** and external interface (**in scope** testing)
- Non-functional testing such as stress and load testing currently will not be tested. (**out of scope**)

2.2. Identify Testing Type

- A testing type is a standard test procedure that gives an expected test outcome.
- Each testing type is formulated to identify a specific type of product bugs. But, all testing types are aimed at achieving one common goal "early detection of all the defects before releasing the product to the customer"
- Your team cannot have enough efforts to handle all kind of testing.
- As test manager, you must set **priority** of the testing types
  - ➢ Which testing types should be **focused**?
  - ➢ Which testing types should be **ignored** for saving cost?

2.3. Document Risk & Issues

- Risk is future's uncertain event with probability of occurrence and a potential for loss.

Risk	Mitigation
Team members lack the required skill for website testing.	Plan training courses to skill up your members
The project schedule is too tight; it's hard to complete this project on time	Set test priority for each of the test activity.
Test manager has poor management skill	Plan leadership training for the manager
Lack of cooperation negatively affects your employees' productivity	Encourage each team member in his task, and inspire them to greater efforts.
Wrong budget estimate and cost overruns	Establish the scope before beginning work, pay a lot of attention to project planning and constantly track and measure the progress

2.4. Create Test Logistics

- In test logistics, the test manager should answer the following questions:
  - Who will test?
  - When will the test occur?
- You may not know exact names of the tester who will test, but the **type of tester** can be defined.
- To select the right member for specified task, you have to consider if his skill is qualified for the task or not, also estimate the project budget. Selecting wrong member for the task may cause the project to **fail** or **delay**.
- A person having the following skills is most ideal for performing software testing:
  - Ability to understand customers point of view
  - Strong desire for quality
  - Attention to detail
  - Good cooperation
- In your project, the member who will take in charge for the test execution is the **tester**. Based on the project budget, you can choose in-source or outsource member as the tester.

III. Define Test Objective

- Test objective is the overall goal and achievement of the test execution. The objective of the testing is
  - finding as many software defects as possible;
  - ensure that the software under test is **bug free** before release.
- To define the test objectives, you should do the following two steps:

  1. List all the software features (functionality and non-functional) which will be tested.
  2. Define the **target** or the **goal** of the test based on the above features

IV. Define Test Criteria

- Test criteria is a standard or rule on which a test procedure or test judgment can be based. There're two types of test criteria:

(a) Suspension criteria

- Specify the critical suspension criteria for a test. If the suspension criteria are met during testing, the active test cycle will be **suspended** until the criteria are **resolved**.

Example: if your team members report that **40%** of test cases failed, you should **suspend** testing until the development team fixes all the failed cases.

(b) Exit Criteria

- It specifies the criteria that denote a **successful** completion of a test phase.
- The exit criteria are the targeted results of the test and are necessary before proceeding the next phase of development.

Example: 95% of all critical test cases must pass.

Cont…

- Some methods of defining exit criteria are by specifying a targeted run rate and pass rate.
  - Run rate - is the ratio between **number of test cases executed/total test cases** of test specification.

For example, the test specification has a total of 120 TCs, but the tester only executed 100 TCs. So, the run rate is 100/120 = 0.83 (83%)

  - Pass rate - is the ratio between **number of test cases passed / test cases executed**.

For example, in the above example, 100 TCs executed and 80 TCs passed. So, the pass rate is 80/100 = 0.8 (80%)

- This data can be retrieved in test metric documents.
- **Run rate** is mandatory to be 100% unless a clear reason is given.
- **Pass rate** is dependent on project scope, but achieving high pass rate shall be a goal.

V. Resource Planning

- Resource plan is a detailed summary of all types of resources required to complete project task. Resource could be human, physical (equipment and materials) and financial needed to complete the test.
- The test manager can make the correct schedule & estimation for the project based on the resources he has.
- This section represents the recommended resources for your project.

VI. Plan Test Environment

- A testing environment is a setup of software and hardware on which the testing team is going to execute test cases.
- The test environment consists of real business and user environment, as well as physical environments, such as server, front end running environment.
- To finish this task, you need a strong cooperation between test team and development team
- You should ask the developer some questions to understand the web application under test clearly.
- Here're some recommended questions. Of course, you can ask the other questions if you need.

    1. What is the maximum user connection which this website can handle at the same time?

    2.What are hardware/software requirements to install this website?

    3. Does the user's computer need any particular setting to browse the website?

VII. Schedule & Estimation

- In the test estimation phase, suppose you break out the whole project into **small tasks** and add the estimation for each task as shown below.

Task	**Members**	**Estimate effort**
Create the test specification	Test Designer	170 man-hour
Perform Test Execution	Tester, Test Administrator	80 man-hour
Test Report	Tester	10 man-hour
Test Delivery		20 man-hour
Total		280 man-hour

VIII. Determine Test Deliverables

- Test deliverables are list of all the documents, tools and other components that have to be developed and maintained in support of the testing effort.
- There are different test deliverables at every phase of the software testing process.
- They can be grouped into **three**.

1) Test deliverables provided before the testing phase.

2) Test deliverables are provided during the testing

3) Test deliverables are provided after the testing cycles is over.

Cont...

1) **Test deliverables provided before the testing phase.**
   - Test plans document.
   - Test cases documents
   - Test design specifications.
2) **Test deliverables are provided during the testing**
   - Test scripts
   - Simulators.
   - Test data
   - Test traceability matrix
   - Error logs and execution logs.
3) **Test deliverables are provided after the testing cycles is over.**
   - Test results/reports
   - Defect report
   - Release notes

Outline of a test plan

- Test plan format and content may vary depending upon different standards. The following format details the points usually covered in a test plan.

Test plan identifier
Introduction
Test items
Features to be tested
Features not to be tested
Approach
Item pass/fail criteria
Suspension criteria
Test deliverables
Testing tasks
Environmental needs
Roles and Responsibilities
Staffing and training needs
Schedule
Risks and contingencies
Approvals

Brief description

- **Test Plan Identifier**: Provides a unique identifier for the document. Every deliverable has a unique identification number which could be numeric or alphanumeric based on the company configuration management.
- **Introduction:** Brief introduction about the project and objective of the current release. Project could be platform configuration tool and objective could new mobile App interface or new feature / enhancement in existing product or defect fixes.
- **Test item:** Introduction and overview of Software Under Test.
- **Features to test:** In scope features. This could be newly added or updated features. Indirect features that have technical or functional dependency on newly added or updated features.

Cont...

- **Features not to test:** Out of scope features. Excluded product features from current Test Plan. [Note: Provide reasoning for exclusion, like, non-impacted / less impacted / less priority features, as applicable.]
- **Approach**: Strategy to test the software. Includes types of tests and how to test. Functional, performance, security testing using combined [manual + automation], manual only, automation only approach.
- **Test deliverables:** All the deliverables from the testing e.g. approaches, test cases, reports etc.
- **Item pass/fail criteria:** Entry and Exit criteria for all items. **E.g**.
  - ➢ Test Case: All Steps passed
  - ➢ Feature: All test cases executed and no defects are detected.
- **Testing tasks:** All tasks / steps to execute for test planning and execution
  - **Environmental needs:** Infrastructure required for application and testing.

Cont...

- **Responsibilities:** Roles and responsibilities for various testing / supported activities.
- **Staffing and training needs:** Training / hiring needs to bridge the gap of available and expected skill.
- **Schedule:** Test estimation (Efforts) and high-level schedule. Schedule should be for key deliverables or important milestones. Ideally, all test deliverables included in the test plan should be scheduled. Detailed test schedule (at feature or defects or resource level) is prepared at appropriate time during test execution. Risks and Mitigation: Risk identification for applicable items, assumptions, and mitigation plan.
- **Approvals**: Approvals and sign of dates.

Test plan is a guideline based on which test execution should be tracked. For successful testing and good product test delivery, it is important to update and make required changes in the plan as per changes in the any of the parameter which was basis of the test plan.

3. Test Design

- In software engineering, test design is the process of creating and writing test artifacts for testing a software.
- Tasks include:

✓Identifying test basis

✓Developing test scenarios

✓Identifying and describing test cases

✓Developing test suite

✓Identifying and structuring test scripts

Test Basis

- Test basis is the source to create test scenarios and test cases.
- It can be
  - the application itself or
  - the requirement documents such as
    - SRS (Software Requirement Specification),
    - BRS (Business Requirement Specification), etc.

Test Scenario

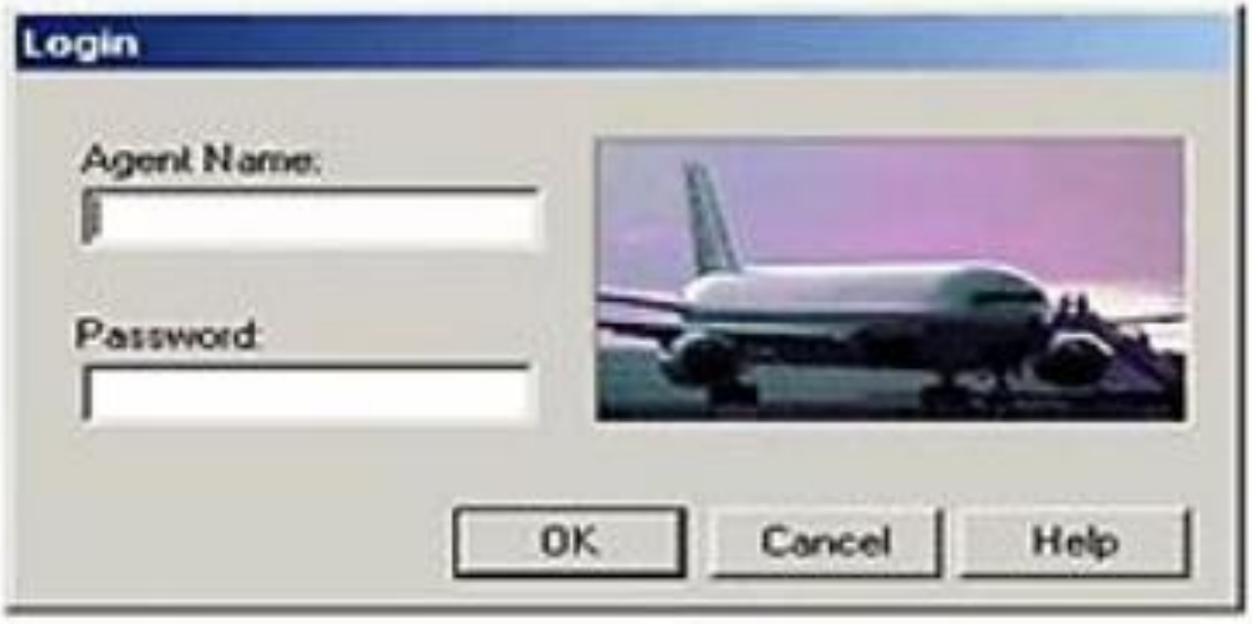
- A **test scenario** is any functionality that can be tested.
- It is also called test condition or test possibility.
- As a tester, you may put yourself in the end user's shoes and figure out the real-world scenarios and use cases of the Application Under Test (AUT).
- **Test scenario** is what to be tested and a **test case** is how to be tested.

...Test Scenario

Example 1:Test Scenario for Flight Reservation

- For the Flight Reservation Application, a few test scenarios would be

➢ **Test Scenario 1**: Check the Login Functionality



...Test Scenario

- **Test Scenario 2:** Check that a New Order can be created
- **Test Scenario 3:** Check that an existing Order can be opened
- **Test Scenario 4:** Check that a user, can FAX an order
- **Test Scenario 5**: Check that the information displayed in the HELP section is correct
- **Test Scenario 6:** Check that the information displayed in About section, like version, programmer name, copy right information is correct

...Test Scenario

- Apart from these six scenarios, here is the list of all other scenarios
  - Update Order
  - Delete Order
  - Check Reports
  - Check Graphs and so on.
- We have already learned exhaustive testing is not possible.
- Suppose you have time only to execute 4 out of those 6 scenarios which two low priority scenarios of these six will you eliminate.
- I am sure most of you would have guessed scenarios 5 & 6 since they are not the core functionality of the application. This is nothing but test prioritization.

Why to create test scenarios?

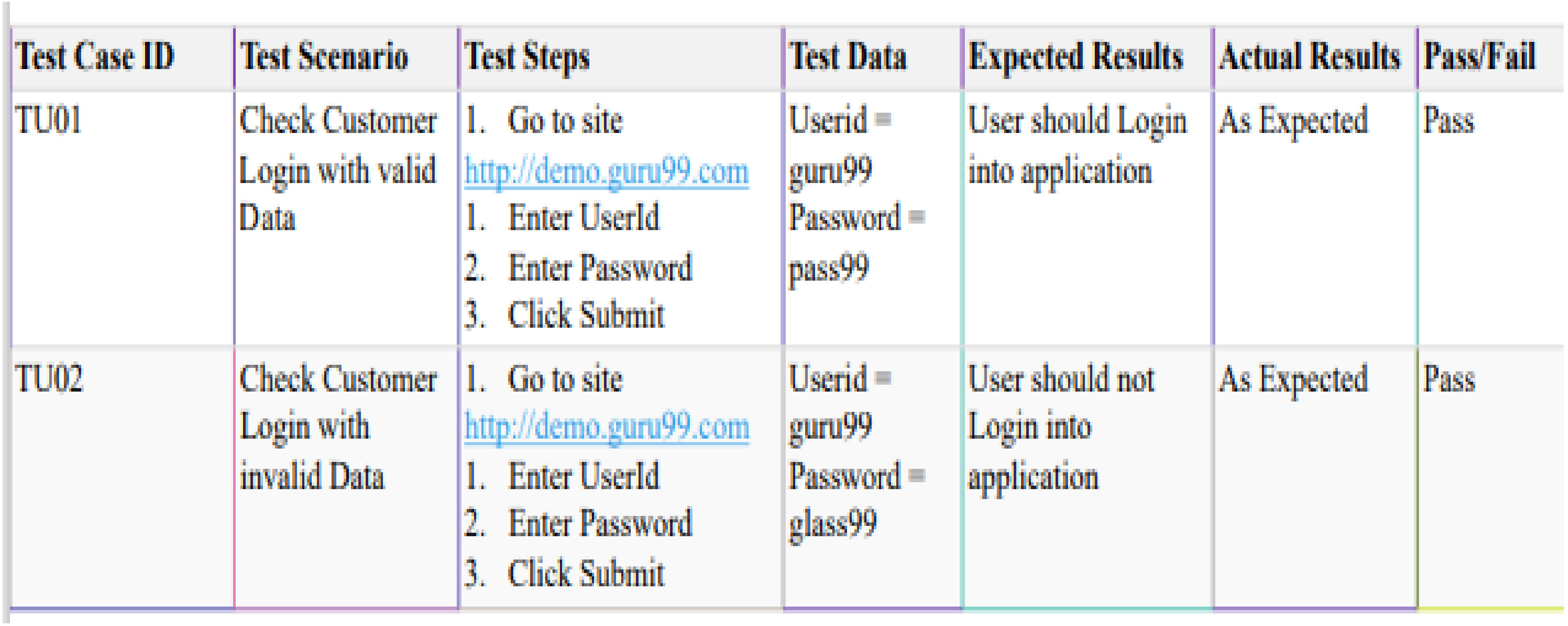
- Test scenarios are created for the following reasons:
  - To ensure complete test coverage
  - To get approval from various stakeholders like business analyst, developers, customers to ensure the application under test is thoroughly tested. It ensures that the software is working for the most common use cases.
  - To quickly determine the testing work effort and accordingly create a proposal for the client or organize the workforce.
  - To determine the most important end-to-end transactions or the real use of the software applications.

Test Case

- A test case is a set of actions executed to verify a particular feature or functionality of your software application.
- A test case is a document which consists of a set of conditions or actions which are performed on the software application in order to verify the expected functionality of the feature.
- Here we describe the end to end logical flow of a specific requirement with test data, prerequisites and expected results
- Now, consider the Test Scenario: **Check Login functionality there are many possible cases like**
- Test Case 1: Check results by entering valid User Id & Password
- Test Case 2: Check results by entering Invalid User ID & Password
- Test Case 3: Check response when User ID is Empty & Login Button is pressed, and many more.

Cont…

- Below is format of a standard login Test case



Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	1. Go to site http://demo.guru99.com 1. Enter UserId 2. Enter Password 3. Click Submit	Userid = guru99 Password = pass99	User should Login into application	As Expected	Pass
TU02	Check Customer Login with invalid Data	1. Go to site http://demo.guru99.com 1. Enter UserId 2. Enter Password 3. Click Submit	Userid = guru99 Password = glass99	User should not Login into application	As Expected	Pass

Characteristics of Good Test Case

1. **Test Cases need to be simple and transparent**

- Create test cases that are as simple as possible. They must be clear and concise as the author of test case may not execute them.
- Use assertive language like go to home page, enter data, click on this and so on. This makes the understanding of the test steps easy and test execution faster.

2. **Create Test Case with End User in Mind**

- Ultimate goal of any software project is to create test cases that meets customer requirements and is easy to use and operate. A tester must create test cases keeping in mind the end users' perspective.

3. **Avoid test case repetition.**

- Do not repeat test cases. If a test case is needed for executing some other test case, call the test case by its test case id in the pre-condition column

Cont…

4. Do not Assume

- Do not assume functionality and features of your software application while preparing test case. Stick to the specification documents.

5. Ensure 100% Coverage

- Make sure you write test cases to check all software requirements mentioned in the specification document. Use Traceability Matrix to ensure no functions/conditions is left untested.

6. Test Cases must be identifiable.

- Name the test case id such that they are identified easily while tracking defects or identifying a software requirement at a later stage.

7. Implement Testing Techniques

- It's not possible to check every possible condition in your software application. Testing techniques help you select a few test cases with the maximum possibility of finding a defect. **Example**: BVA, EP, etc.

Cont...

8. Self cleaning

- The test case you create must return the Test Environment to the pre-test state and should not render the test environment unusable. This is especially true for configuration testing.

9. Repeatable and self-standing

- The test case should generate the same results every time no matter who tests it

10. Peer Review.

- After creating test cases, get them reviewed by your colleagues. Your peers can uncover defects in your test case design, which you may easily miss.

Test Suit

- In software engineering, a test suite is a collection of test cases that are intended to be used to test a software program, to show that it has some specified set of behaviors.
- Test suite is a container that has a set of tests which helps testers in executing and reporting the test execution status.
- A test suit is a collection of test cases that are grouped for test execution purposes.

Test Script

- A test script is a set of instructions (written using a scripting/programming language) that is performed on a system under test to verify that the system performs as expected.
- Test scripts are used in automated testing.

Test Data

- Test data is the documented data that is basically used to test the software program.
- Test data may be produced by the tester, or by a program or function that aids the tester. Test data may be recorded for re-use, or used once and then forgotten.
- Test data is divided into two categories.
  - ➢ **Positive test data** which is generally gives to system to generate the expected result and
  - ➢ **Negative test data** which is used to test the unhandled conditions, unexpected, exceptional or extreme input conditions.
- ***If the test data is inadequately designed, then such test inputs will not cover all possible test scenarios, which impact the quality of the software application under test***

4. Test Environment Setup

- Every software is developed by considering hardware and software requirement.
- Here we setup the testing environment (e. g. server/client/network, or install the setup if it is window's application etc.) with the goal of replicating the end-users' environment.
- This step decides on which Platform/OS the tester needs to perform testing of project.
  - ✓ Example: Use only Internet Explorer, Resolution must be like 136 x 768, or in many large applications they indicate the minimum hardware and software requirement like Photoshop, latest version of Matlab, Oracle, large games etc.
- In this phase, according to requirements, they configure required hardware and software.
- Tester may or may not involve in setting the configuration.

5. Test Execution

- Once the preparation of test case development and test environment setup is completed, then test execution phase can be kicked off.
- Test execution is the process of executing the code and comparing the expected and actual results.
- In this phase, testing team start executing test cases based on prepared test planning & prepared test cases in the prior step.
- Once the test case is passed then same can be marked as passed.
- If any test case is failed then corresponding defect can be reported to the developer team via bug tracking system & bug can be linked for corresponding test case for further analysis.
- Ideally, every failed test case should be associated with at least single bug.

Cont…

- Once the bug fixed by development team then same test case can be executed based on your test planning.
- If any of the test cases are blocked due to any defect then such test cases can be marked as Blocked, so we can get the report based on how many test cases passed, failed, blocked or not run etc.
- Once the defects are fixed, same Failed or Blocked test cases can be executed again to re-test the functionality.

6. Test Reporting and closure

- Call out the testing team member meeting & evaluate cycle completion criteria based on test coverage, quality, cost, time, and critical business objectives.
- Discuss what all went good, which area needs to be improved & taking the lessons from current STLC as input to upcoming test cycles, which will help to improve bottlenecks in the STLC process.
- Test cases & bug reports will analyze to find out the defect distribution by type and severity.
- Once complete the test cycle then test closure report & test metrics will be prepared.

THE END
THANK YOU!!!