TODAY'S AGENDA

- Disadvantages of Manual Testing
- Principles of Software Testing
- Test Plan and it's benefits
- Test Plan Template
- Test Scenario + Template
- Test Case + Template
- Test Scenario vs Test Case
- Testing Techniques

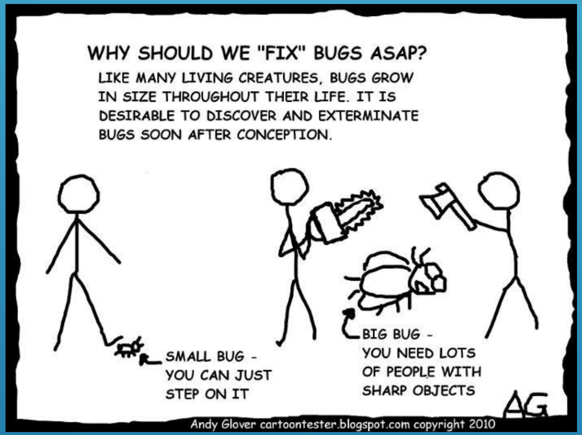
Challenges with Manual Testing

Boring
Time Consuming
Tedious
Mistakes, Errors.....
TEST PASS

7 PRINCIPLES OF SOFTWARE TESTING

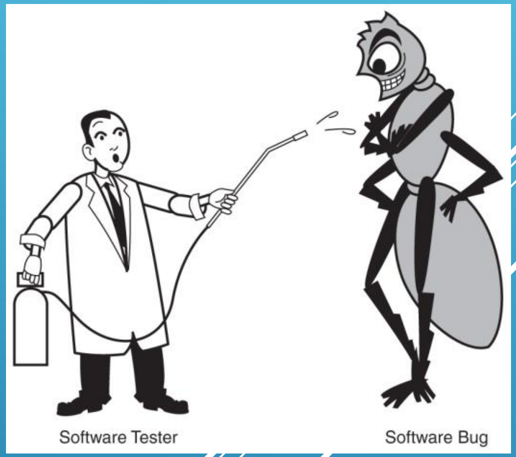
Exhaustive testing is not possible

Defect clustering


WHY SHOULD WE "FIX" BUGS ASAP?
LIKE MANY LIVING CREATURES, BUGS GROW IN SIZE THROUGHOUT THEIR LIFE. IT IS DESIRABLE TO DISCOVER AND EXTERMINATE BUGS SOON AFTER CONCEPTION.
SMALL BUG - YOU CAN JUST STEP ON IT
BIG BUG - YOU NEED LOTS OF PEOPLE WITH SHARP OBJECTS
AG
Andy Glover cartoontester.blogspot.com copyright 2010

Pesticide paradox

Testing shows presence of defects/bugs



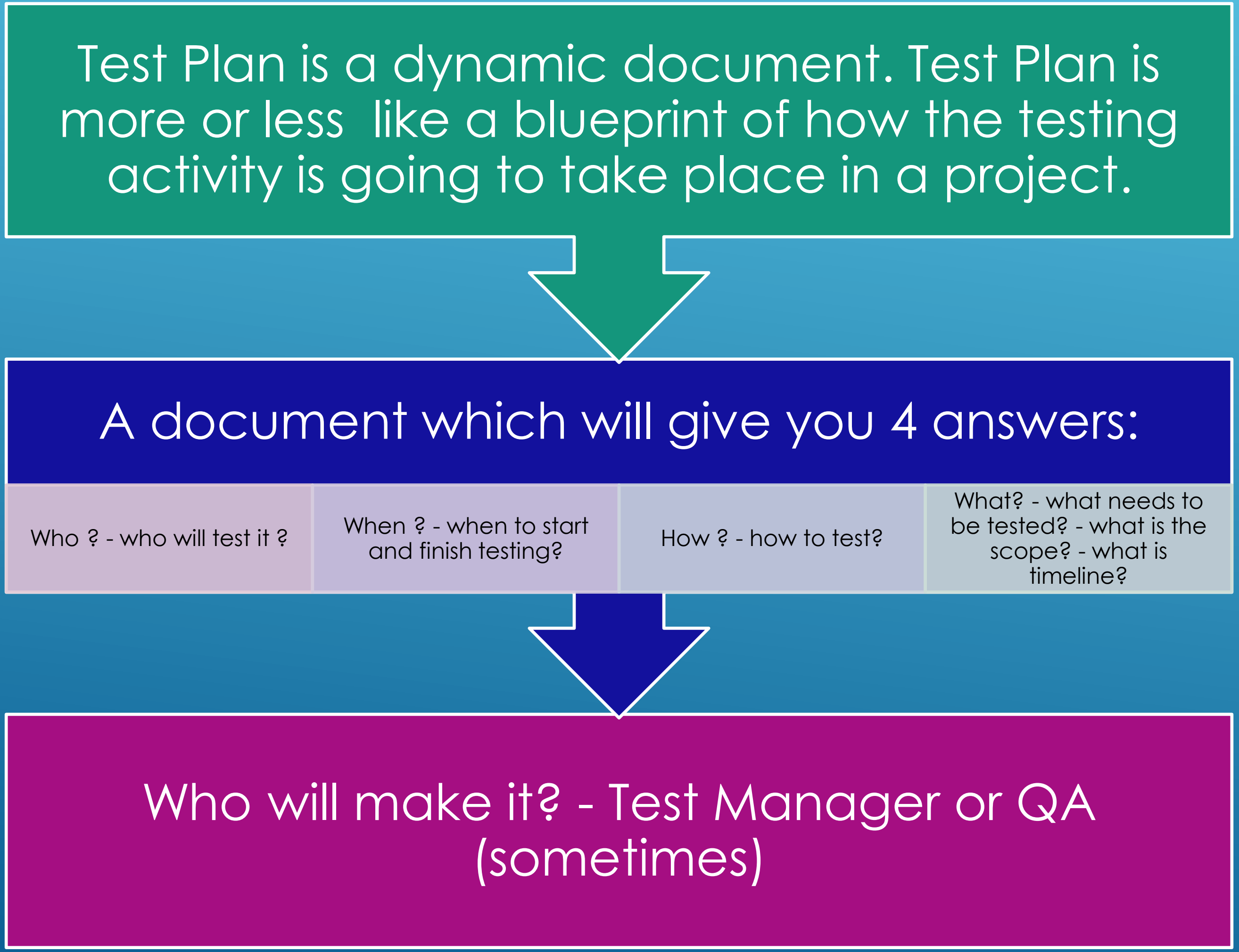
Absence of errors fallacy

Early testing

Testing is context dependent

Test Plan

Test plan is the high level document, which explains the roadmap of the project, which is describing the scope, approach, resources and schedule of intended test activities.



Test Plan is a dynamic document. Test Plan is more or less like a blueprint of how the testing activity is going to take place in a project.

A document which will give you 4 answers:

Who ? - who will test it ?	When ? - when to start and finish testing?	How ? - how to test?	What? - what needs to be tested? - what is the scope? - what is timeline?

Who will make it? - Test Manager or QA (sometimes)

WHY IS IT IMPORTANT TO HAVE TEST PLAN IN SOFTWARE TESTING ?

- **Planning makes a task execution in a systematic manner:** No matter which situation we are dealing with, we can make it simpler by proper planning (as it gives an ideal path for the execution process).

- **Test plan for finding a suitable way to execute the testing process:** Test plan give an inside look of the project and hence helping in developing a better testing approach.

- **We can avoid loopholes:** With test plan we can presume and assume most of the upcoming uncertain risks/issues of the application.

- **Efficient testing can be achieved with test plan:** by eliminating unnecessary tasks, processes from the application.

- **Effort will be minimised, and productivity will be maximised:** by avoiding unnecessary iterative processes which in turn reduce the effort and time

Test Plan Template

Introduction:

- Provide an overview of the test plan.
- Specify the goals/objectives.
- Example: Project_Name_1

References:

- List the related documents, with links to them if available, including the following:
- Example : Project Plan (SRS, BRS, PROJECT RELATED DOCUMENTS)

Test Items:

- List the test items (software/products) and their versions.
- Example: Testing should be done on both front end and back end of the application on windows/Linux Environments

Features to be Tested:

- List the features of the software/product to be tested.
- Provide references to the Requirements and/or Design specifications of the features to be tested.
- Example: Login page, dashboard, Reports

Features Not to Be Tested:

- List the features of the software/product which will not be tested.
- Specify the reasons these features won't be tested.
- Example : Payment using PayPal features is above to remove from the application. There is no need to test this feature.

Approach:

- Mention the overall approach to testing.
- Testing types, and the testing methods [Manual/Automated; White Box/Black Box/Gray Box]
- Example : We follow Agile Methodology in this project

Item Pass/Fail Criteria:

- Specify the criteria that will be used to determine whether each test item (software/product) has passed or failed testing.
- Example : All the major functionality of the application should work as intended and the pass percentage of test cases should be more than 95% and there should not be any critical bugs.

Suspension Criteria and Resumption Requirements:

- Specify criteria to be used to suspend the testing activity.
- Specify testing activities which must be redone when testing is resumed.
- Example : If any of the major functionalities are not functional or system experiences login issues than testing should suspended

Test Deliverables:

- List test deliverables, and links to them if available, including the following:
- Example : Test Cases & Bug Report

Test Environment:

- Specify the properties of test environment: hardware, software, network etc.
- List any testing or related tools.
- Example : Lambda, SauceLab, Browserstack

Estimate:

- Provide a summary of test estimates (cost or effort) and/or provide a link to the detailed estimation.
- Example : Time period

Schedule:

- Provide a summary of the schedule, specifying key test milestones, and/or provide a link to the detailed schedule.
- Example : Perform test execution 120 man-hours, Test Reporting – 30 man-hours

Staffing and Training Needs:

- Specify staffing needs by role and required skills.
- Identify training that is necessary to provide those skills, if not already acquired.
- Example : Wherever needed

Responsibilities:

- List the responsibilities of team/ role/ everyone.
- Example : Test plan should be prepared by test lead. Preparation and Execution of tests should be carried out by testers.

Risks:

- List the risks that have been identified.
- Specify the mitigation plan and the contingency plan for each risk.
- Example : In case of a wrong budget estimation, the cost may overrun. Contingency – Establish the scope before beginning the testing tasks and pay attention to the project planning and track the budget estimates constantly.

Assumptions and Dependencies:

- List the assumptions that have been made during the preparation of this plan.
- List the dependencies.
- Example : Design artefacts from customer (Assumption – approval is likely within 2 week, Dependencies – all depend on approval)

Approvals:

- Specify the names and roles of all persons who must approve the plan.
- Provide space for signatures and dates. (If the document is to be printed.)
- Example : Project manager should agree on completion of the project and determine the steps to proceed further.

Test Scenario

A Test Scenario is defined as any functionality that can be tested. It is a collective set of test cases which helps the testing team to determine the positive and negative characteristics of the project.

Why do we write Test Scenario?

- The main reason to write a test scenario is to verify the complete functionality of the software application.
- It also helps you to ensure that the business processes and flows are as per the functional requirements.
- Test Scenarios can be approved by 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.
- They serve as a quick tool to determine the testing work effort and accordingly create a proposal for the client or organize the workforce.
- They help determine the most critical end-to-end transactions or the real use of the software applications.
- Once these Test Scenarios are finalized, test cases can be easily derived from the Test Scenarios.

Test Scenario Template

- **Module** – The module or the component of the application.
- **RequirementId** – This field is optional and can be linked to the SRS.
- **TestScenarioId** – This field is the identifier of the test scenarios.
- **Description** – The description field describes the purpose of the test scenario.

Test Scenario Template

Module	RequirementId	TestScenarioId	Test Scenario Description
Login	[Optional]	TS_01	Verify that user is able to login with valid credentials
	[Optional]	TS_02	Verify that user is not able to login with invalid credentials
	[Optional]	TS_03	Verify mandatory field check validation on the login page

Test Case

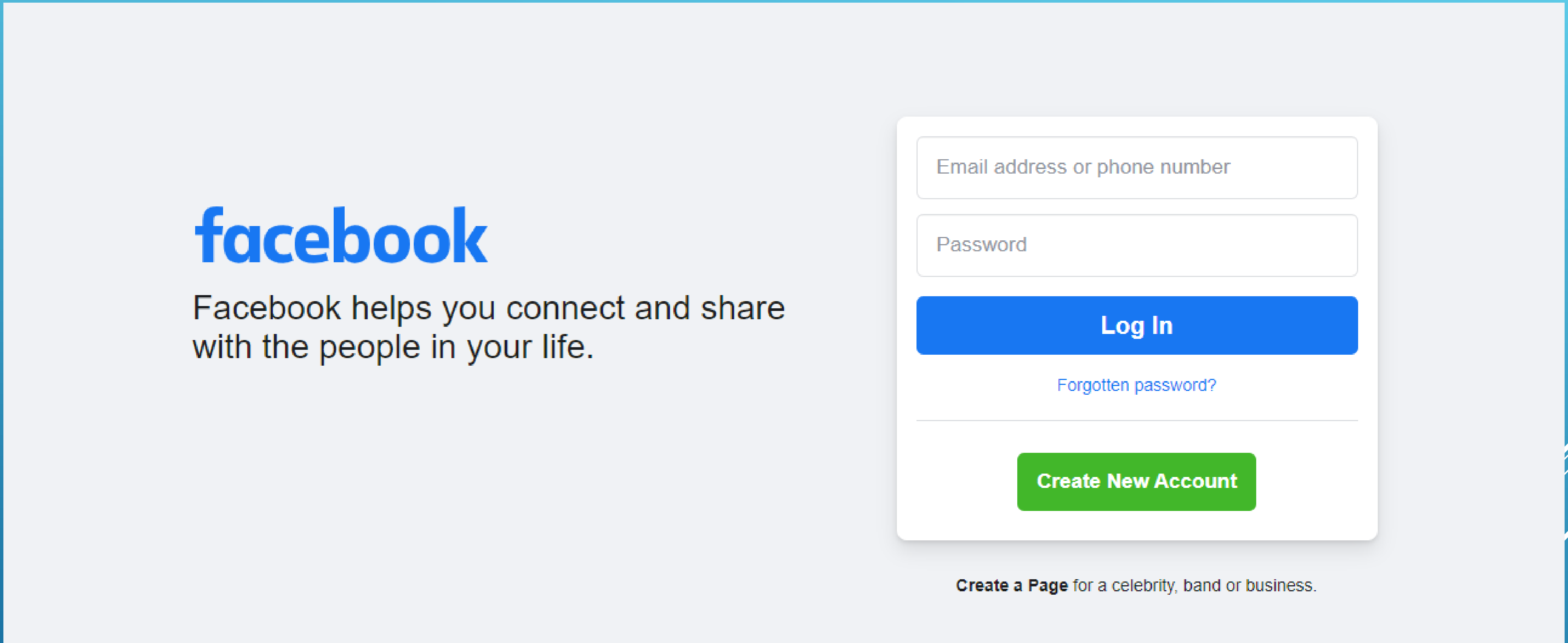
A **TEST CASE** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

Why do we write Test Cases?

- Test cases help to verify conformance to applicable standards, guidelines and customer requirements.
- Helps you to validate expectations and customer requirements.
- Increased control, logic, and data flow coverage.
- You can simulate 'real' end user scenarios.
- Exposes errors or defects.
- When test cases are written for test execution, the test engineer's work will be organized better and simplified.

Test Case Template

Test Case Id	Test Case Description	Pre –Condition	Test Steps	Test Data	Expected Results	Actual Results	Status	Defect ID	Comments
FB_Login_01	To Verify User Should Open Web Page		1. Open Chrome 2. Enter URL on address bar	URL: www.facebook.com	Facebook webpage should open successfully.	Facebook webpage opened successfully.	Pass		
FB_Login_02	To Verify User should be logged in with valid credential	User should have a registered Facebook account.	1. Enter Valid Username 2. Enter Valid Password 3. Click on Log in Button	Username: demo12345@gmail.com Password: test123	User should be logged in successfully.	User logged in successfully.	Pass		
FB_Login_03	To verify user Should not be Login with Valid Username and Invalid Password	User should have a registered Facebook account.	1. Enter Valid Username 2. Enter Valid Password 3. Click on Log in Button	Username: demo12345@gmail.com Password: test123	User should not be able to login.	User logged in.	Failed	FB_Login_03	



facebook

Facebook helps you connect and share with the people in your life.

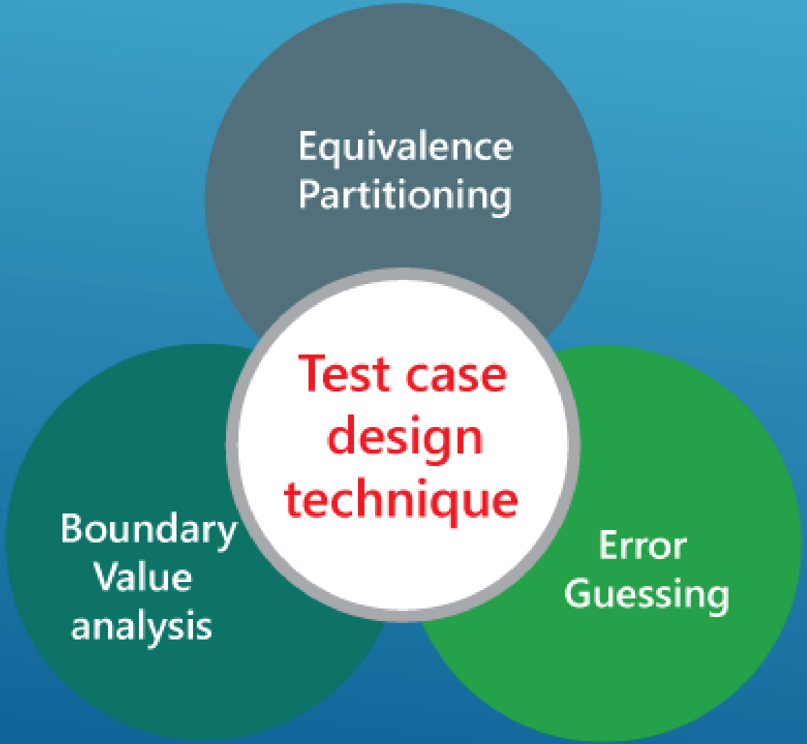
Create a Page for a celebrity, band or business.

Test Scenario Vs Test Case

Test Scenario	**Test Case**
A test scenario contains high-level documentation which describes an end to end functionality to be tested.	Test cases contain definite test steps, data, expected results for testing all the features of an application.
It focuses on more "what to test" **than** "how to test".	A complete emphasis on "what to test" **and** "how to test.".
Test scenarios are a one-liner. So, there is always the possibility of ambiguity during the testing.	Test cases have defined a step, pre-requisites, expected result, etc. Therefore, there is no ambiguity in this process.
Test scenarios are derived from test artifacts like BRS, SRS, etc.	Test case is mostly derived from test scenarios. Multiple Test case can be derived from a single Test Scenario
It helps in an agile way of testing the end to end functionality	It helps in exhaustive testing of an application
Test scenarios are high-level actions.	Test cases are low-level actions.
Comparatively less time and resources are required for creating & testing using scenarios.	More resources are needed for documentation and execution of test cases.

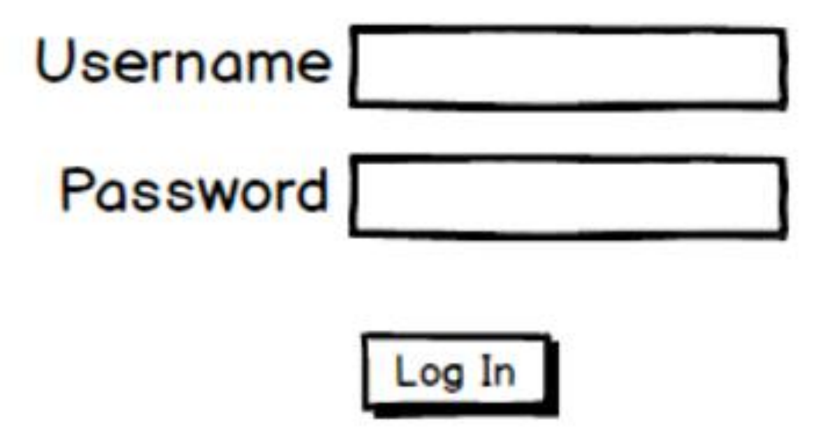
SOFTWARE TESTING TECHNIQUES

- Helps you to design better test cases.
- It reduces time required for performing testing of a software due to less number of test cases.
- Helps to increase test coverage.

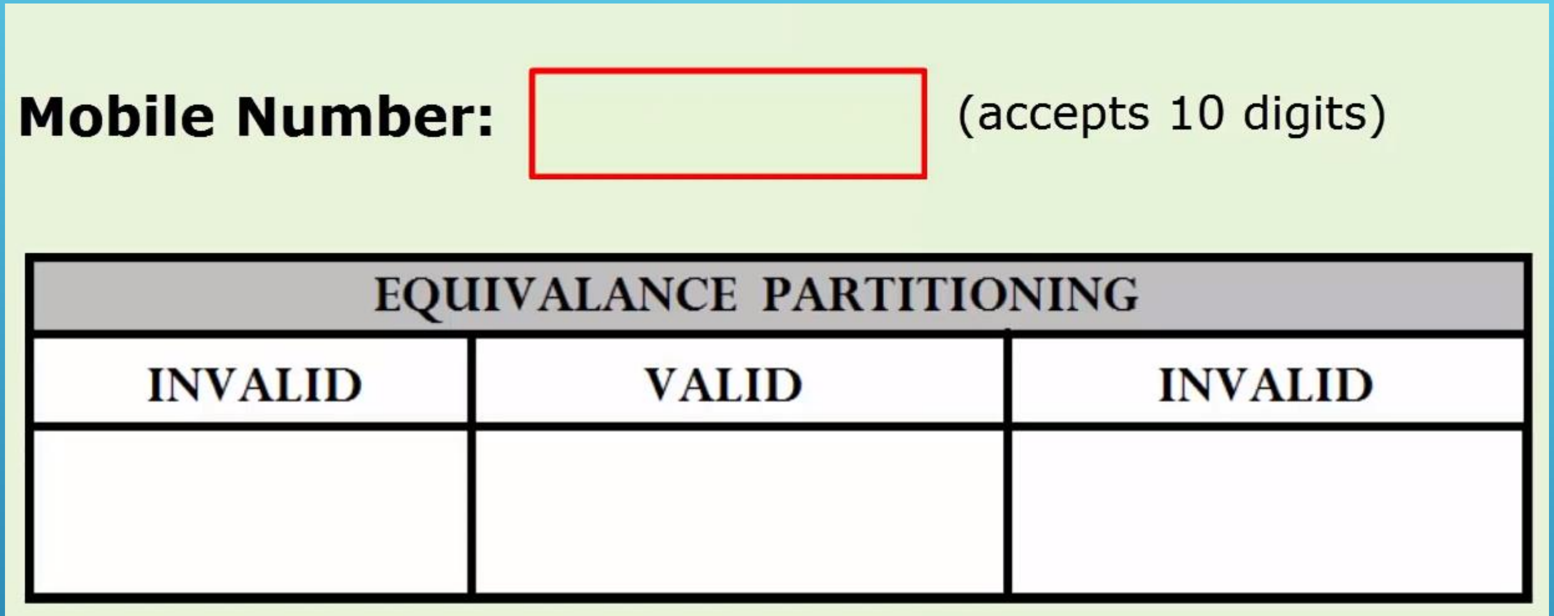


ECP (EQUIVALENCE CLASS PARTITIONING)

- It is used for Black box testing.
- Divides set of test conditions into partitions that can be considered the same.
- If one condition valid in the group, others are valid too.
- If one condition is invalid in the group, others are invalid too.
- The key goal is to complete the test coverage and to lessen duplication.



ECP EXERCISE

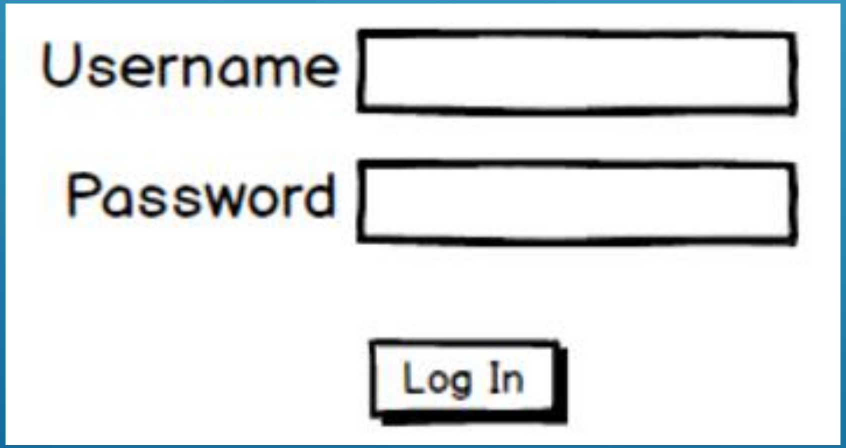


Mobile Number: (accepts 10 digits)

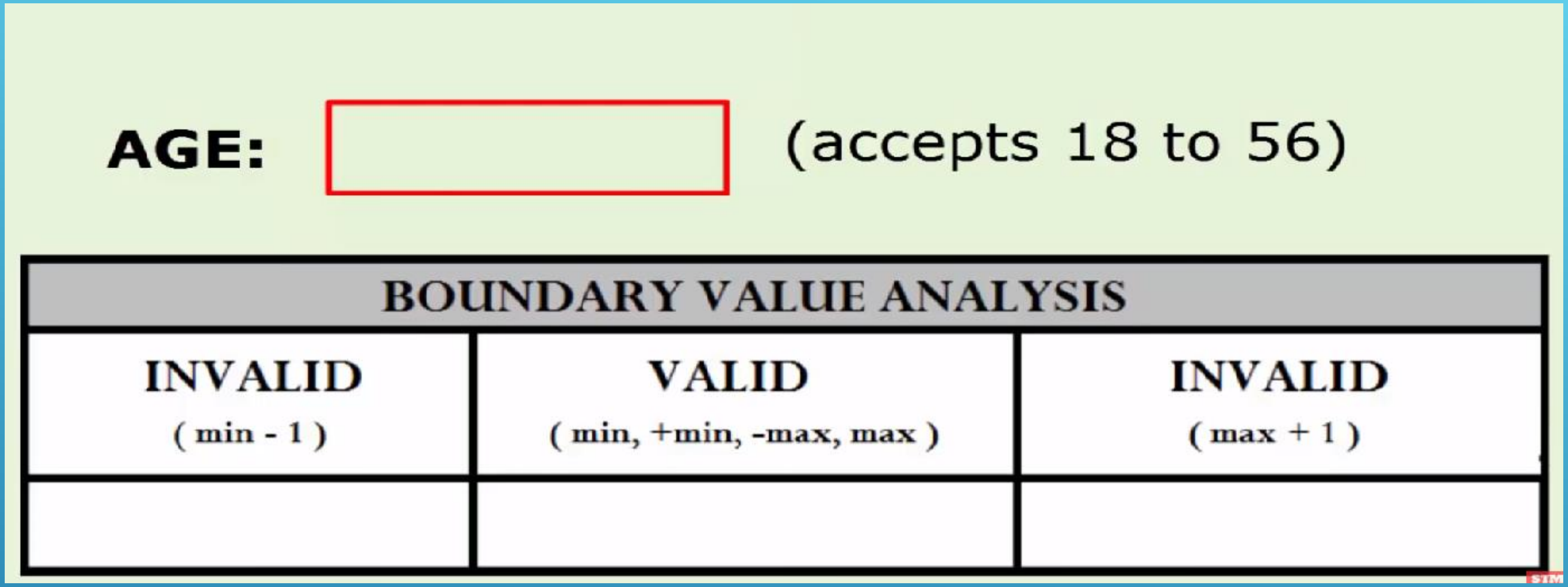
EQUIVALANCE PARTITIONING		
INVALID	VALID	INVALID

BVA
(BOUNDARY VALUE ANALYSIS)

- It is used for Black box testing – also called range checking
- Helps to test boundary values because the input value near the boundary have higher chances of error.
- Based on testing at the boundaries between partitions.
- Includes maximum, minimum. Inside and outside boundaries.



BVA -EXERCISE



AGE: □ (accepts 18 to 56)

BOUNDARY VALUE ANALYSIS		
INVALID (min - 1)	VALID (min, +min, -max, max)	INVALID (max + 1)

ERROR GUESSING

- Mainly unit testing technique – testers also use it.
- Based upon our previous experience – not requirement based.
- The technique counts a list of possible errors or error-prone situations. Then tester writes a test case to expose those errors.

We can consider:

- Tester intuition
- Past failures (error prone areas)
- “What craziest thing I can do ?”
- Brain storming

HOMEWORK

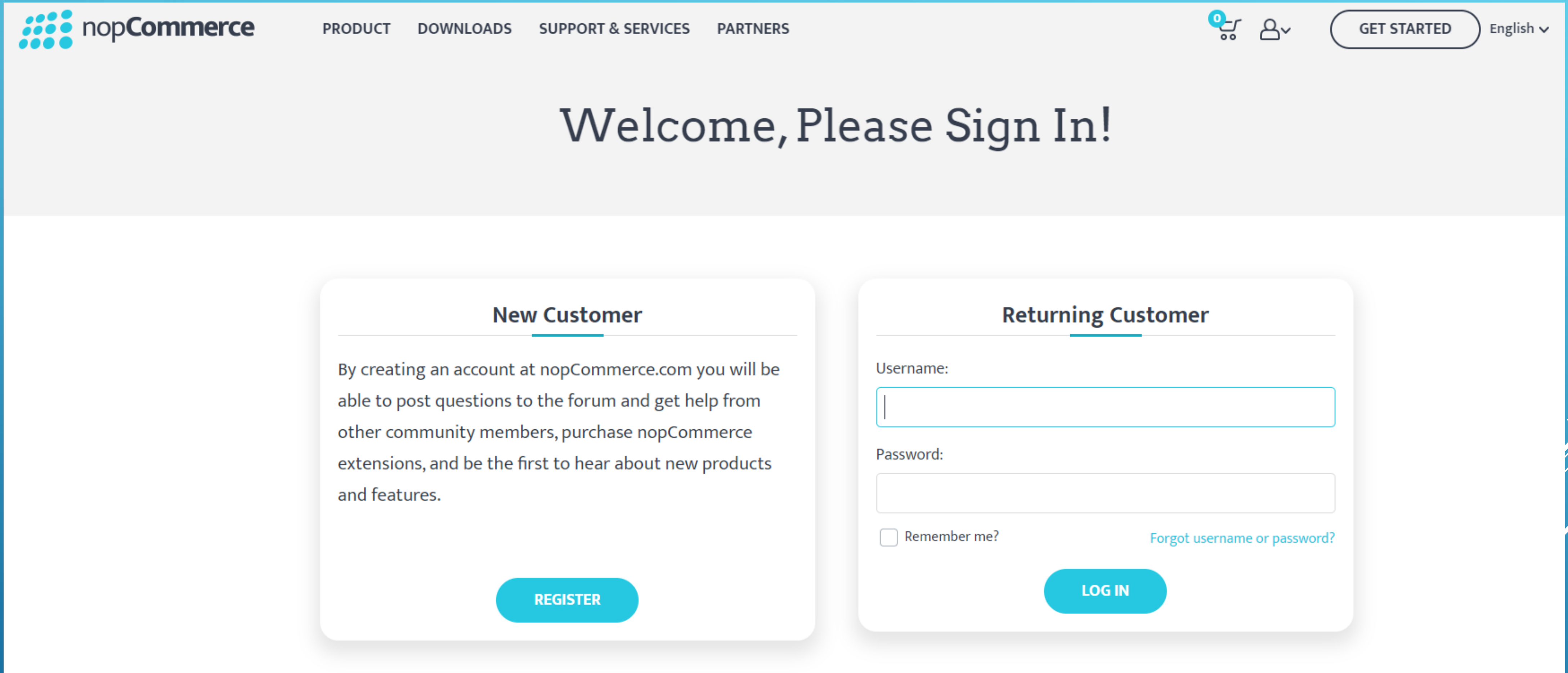
- 7 Principles to know by heart with definition example
- Login page test cases for nopCommerce to be sent by email: unifytesting57@gmail.com

Deadline: Wednesday Midnight Latest

Login Page Test Scenario

https://www.nopcommerce.com/

- User should be logged in with valid credential.
- User should NOT be logged in with INVALID credential.
- Username should contain letter and numbers only.
- Username should not be left blank.
- Username should not be more than 40 characters.
- Username should not start with or contain any special characters.
- Password should be at least 6 characters.
- Password should contain combination of uppercase, lowercase, numbers and special characters.
- Password should not contain spaces.
- Password should not be more than 40 characters.



Welcome, Please Sign In!

New Customer

By creating an account at nopCommerce.com you will be able to post questions to the forum and get help from other community members, purchase nopCommerce extensions, and be the first to hear about new products and features.

Returning Customer

Username:

Password:

□ Remember me?