SO YOU THINK YOU CAN WRITE A TEST CASE

Srilu Pinjala (Sridevi)
PNSQC
ASSUMING

- Audience has experience estimating tests and scenarios.
- Audience attempted to write test cases at some point.
Can’t understand test cases written by others.

New testers are dependent on legacy testers.

Test Cases are not aiding to test accurately or completely.
WHY ARE WE HERE?

- Test Cases written during a project do not suffice in supporting the product through its life.
INTRODUCTION

Difference between TESTING and TEST CASE

Testing is **ACTION**

Test Case is a **DOCUMENT**
Test Case is mere text. It is instructions and verification in its simplest form.

Test Case is a journal with information about what was tested and how.

Test Case belongs to the product.
PURPOSE OF A TEST CASE

“A Test Case has two purposes: to expose an error, or to demonstrate correct execution.” (Jorgensen, 2014)

- It explains when and how to test.
- It serves as a guide to learn about the product.
- It exposes errors in case of regression.
TYPES OF TEST CASES

(Not to be confused with the types of testing.)
EXPLORING

✓ To actually test elements, functions, flows, and more.
✓ To test the heck out of the product.

PILOTING

✓ To navigate to the test elements, that need to be tested.
✓ To avoid testing the same thing twice.
TYPES OF TEST CASES

EXPLORING

PILOTING
## TYPES OF TEST CASES

<table>
<thead>
<tr>
<th>EXPLORING</th>
<th>PILOTTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Unique</td>
<td>✓ Usual</td>
</tr>
<tr>
<td>✓ One action</td>
<td>✓ One set of actions</td>
</tr>
<tr>
<td>✓ Verify specifics</td>
<td>✓ Verify generic</td>
</tr>
<tr>
<td>✓ Do not cram</td>
<td>✓ Do not cram</td>
</tr>
</tbody>
</table>
Identify the test case user base

- Author
- Other testers
- UA Testers
- End Users

- Business Analysts
- Developers
- Project, Product, Program managers

All of the Above and more
WHEN TO START WRITING TEST CASES?
Requirements are written to create products that are powerful, with the hopes that the whole process and the product can be controlled.

Every Freaking time!!
Products could end up different, even when the acceptance criteria is met.
TAKES ON A NEW LIFE

• Due to many moving parts, the product ends up slightly or entirely different with some expected features.

• It is what was asked for, but somehow different.
They are abstract and not clear.
Interpreting them leaves to assumptions.

Testing a requirement is not always possible.
BEST TIME TO WRITE

- As soon as the requirement is signed off
- Before the application is developed
- Before starting any testing

After the first round of Testing
AFTER THE FIRST ROUND OF TESTING

✓ GUI is available for exploring
✓ Environment, database, services, etc. are evident
✓ Questions can be answered clearly

Testing is not a one time thing. Test Cases serve many QA cycles.
WHAT DOES THE TEST CASE BELONG TO?
SPEC, PROJECT, PRODUCT?

- Verifying requirements is not enough testing
- They are project specific for a limited time
- Test Cases are for supporting the product
- They have to be written, edited, and deleted as the product changes through its life

Test Case belongs to the Product
KEY COMPONENTS OF A TEST CASE
### Key Components

<table>
<thead>
<tr>
<th>Test Step</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ User actions</td>
<td>✓ Verification of GUI and Data changes</td>
</tr>
<tr>
<td>✓ Data inputs</td>
<td>✓ Specifics of changed state</td>
</tr>
<tr>
<td>✓ Instructions to the tester</td>
<td>✓ Specifics of changed state</td>
</tr>
</tbody>
</table>

_Everything else are bells and whistles._
## HOW NOT TO WRITE

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test step</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell phone charges</td>
<td>Charge the cell phone</td>
<td>Cell phone charged fully</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test step</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Website</td>
<td>Go to Home Page</td>
<td>Page Loads fully</td>
</tr>
</tbody>
</table>
HOW TO WRITE
LIKE TALKING TO A ROBOT

- Robots do not understand logic
- They execute specific instructions
- They are only as good as your instructions
CASE IS ABOUT HOW
TEST STEP - EXAMPLES

- Go to URL – www.pnsqc.org
- Type in field Username (myname)
- Check ON checkbox ALL
- SQL - [SELECT userID, userKey FROM Users
  WHERE userKey = 7]
- Go back one page; Refresh page; Go forward one page.
TEST STEP - DO

- Keep the steps clear, simple, specific (CSS)
- One step at a time / One function at a time
- Instruct to use specific data. Explain where to get it.
- Present Tense – test case should be in present tense
TEST STEP – DON’TS

✓ Jargons, abbreviations, project specific terms to describe. They change!!

✓ Repeat above steps. They change!!

✓ Test a feature. Tell how to test.

✓ Valid data, invalid data, etc. Give examples of valid and invalid
RESULTS - EXAMPLES

- Page appears with *title* Login
- Header consists of *link* Click here
- *Error message “so and so”* appears.
- Auto-complete appears with 5 matching suggestions.
- SQL verification – columns UserID & Pass
- SQL Result – Top 5 match the search results
RESULT – DO’S

✓ Results should be tangible and verifiable.
✓ Keep it clear, simples, specific (CSS)
✓ one step can have multiple verifications.
✓ List as many verifications as necessary (title, logo, text, images, links, error messages, etc.)
✓ Describe verbatim / WYSWIG
RESULTS – DON’TS

✓ Loads successfully – Explain successful
✓ It has no errors – Explain what kind of errors
✓ System accepts – Explain how the system accepts by verifying text, database, etc.
✓ It works – Explain how to verify it works
TEST CASE LIFE CYCLE
Life Cycle

- Test Case
- Integrate
- Design
- Explore
- Map
- Revise
START HERE

- **Study / Revise** – the application
- **Map** – the pages, elements, functions or features
- **Explore** – the interfaces, databases, environments, services, etc.
- **Design** – test cases with appropriate names and organize them in suitable categories.
- **Integrate** – the mini test cases and update related information needed for test execution.
STUDY / REVISE

Study the product

- No. of pages, each page
- CSS and unique parts
- Authentication roles
- Session timeouts
- Notifications and Errors
- Many more
PICK A STYLE

- Model the tests around the GUI features, core functions, flows, data, etc.
- There is no wrong or right way of testing. The decision is based on your best judgement.
- If it does not work, change it. Its okay.
**page Login**

<table>
<thead>
<tr>
<th>field</th>
<th>Email or phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>field</td>
<td>Password</td>
</tr>
<tr>
<td>checkbox</td>
<td>Keep me signed in</td>
</tr>
<tr>
<td>button</td>
<td>Sign In</td>
</tr>
<tr>
<td>link</td>
<td>Can’t access your account?</td>
</tr>
<tr>
<td>link</td>
<td>Sign in with a single-use code</td>
</tr>
<tr>
<td>logo</td>
<td>Outlook</td>
</tr>
</tbody>
</table>
- Split the application into modules or pages
- Identify the common CSS elements
- List the unique elements and their hierarchy
- Use a spreadsheet to list the elements by their hierarchy.
- Hierarchy – If functions in Level-1 fail, there is not point in testing the functions in Level-2 under that function.
EXPLORE

- URL – invoke in multiple browser and operating systems.
- Page – refresh, resize, go back and forth
- Authenticate – login flow, user roles, etc.
- Profile information – set up, edit, delete profiles.
- Search feature – autocomplete, results, etc.
- Text fields – field validation (alpha, numeric, chars)
- Links – what do they open. Buttons – what do they process
- Session time out, etc.
<table>
<thead>
<tr>
<th>Page</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Go to URL Hotmail.com | Page open with title – Sign In  
Left section –  
Field – enter your phone number and we’ll send you the download link;  
Button – Send My Link  
Right section –  
Logo – Outlook  
Fields – Email or Phone, Password  
Checkbox – Keep me signed in  
Button – Sign in  
Links – What’s this?, Can’t access your account?, Sign in with a single-use code  
Footer section –  
Links – Contact Us, Terms of Use, Privacy & Cookies, Link Disclaimer  
Disclaimer ©Copyright |
Design test cases for all the known scenarios with data, environment, services information.

- They maybe used individually or in sets.
- Multiple scenarios may be listed in one test case. (This is not same as cramming)
- Maintain a separate suite of these test cases to be called easily.
- Update here and related test sets update instantly.
### DESIGN TEST CASES

<table>
<thead>
<tr>
<th>Test Step</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select any amount of text. Click icon CUT (Home &gt; Clipboard &gt; Cut)</td>
<td>The selected text is cut (disappears) from the page / cell.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Step</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select any amount of text. Click icon Copy (Home &gt; Clipboard &gt; Copy)</td>
<td>The selected text persists on the page. It does not disappear from the page.</td>
</tr>
</tbody>
</table>
INTEGRATE

- Integrate the individual test cases to form scenario based test cases from end to end telling a story.
- Add information about data inputs and verification. Any other verifications too.
- These test cases may now be linked to any requirements as they fit.
- Organize the test cases in test execution order and dependency.
THE CYCLE GOES ON

- Test case is not a one time thing.
- Write it, but don’t forget it. Visit them from time to time.
- **CHANGED FEATURES** – update existing test cases.
- **NEW FEATURE** New test case will be needed.
- Edit, Merge and delete as necessary.
- Having an Owner for the test cases is recommended.
When a exploratory test case or condition or data is updated, the Full test case is automatically updated.

Updating one test case will update the test sets across the suite.
TEST CASE NAMING

CONVENTION

Naming is a part of Test Case DESIGN
NAME IS ABOUT WHAT

- First part – belongs to the product
- Second part – belongs to the module, page
- Third part – belongs to the feature
- Fourth part – belongs to the function

(It could have more parts)
Examples

- Ms.PPT.Invoke
- Ms.PPT.mHome_iCut
- Ms.PPT.mHome_iCopy
- Ms.PPT.mHome_iPaste
- Ms.PPT.mHome_iPaste_oPaste
- Col.Search
- Col.nav.Home

Clearly identifying test cases make it easy to sort through them to identify:

- What is covered.
- What is not covered.
- Eliminates redundancy
- Locating, locating, locating
GO THE DISTANCE
ENABLE THE TESTER
PROVIDE TOOLS
ACCESSORIZED THE
TEST CASES
TEST DATA

- List test data in the test step  OR
- List test data in a separate column
- Provide exact data, SQL queries and verification
- Maintain Wikis, Excel sheets, etc. with Test data, environment information and screen shots.
- Provide ideas and tools to execute tests faster.
Test data may be provided in data sheets.

Certain test cases require the same steps and flow, but require variety of data to result various expected results.

Each iteration does not need a new test case.

Pair the test case with a data sheet.
We are intelligent beings, but not as smart as we assume we are. If you think you cannot error, you don’t know yourself.
If you can read this, You just missed typo bugs!!

Instruct to copy paste text in a Word document, to not miss typos.

Don’t forget to compare it to previous versions.
So, be Specific about what to look for.

- Highlight what to look for and what to ignore.
- Avoid missing on actual bugs as well as invalid bugs.
- Going the extra mile, saves tons of time.
FIND THE DIFFERENCES

- Explain what is acceptable
- Explain localization and marketplace exceptions
- Explain if variations are expected and known between browsers and devices.
PROVIDE THE TOOLS

- Provide tools to get the job done fast and Right!!
- Testing is not the place to test the testers about testing a product they don’t know.
LESSONS LEARNT

- School teaches lessons and gives tests. Software gives you tests and teaches you lessons.
- Note down what you learnt in Clear, Simple, Specific instructions.
- Don’t assume the tester will understand and know how to test. They are just executing your instructions.
Test Case belongs to the product. Its name should be associated to the product, module, feature and functions.

Test Cases support the product through its life, so don’t write the them on assumptions based on requirements.

Write test cases after exploring first hand.

Exploratory test cases explore every inch of the product.

Pilot test cases merely navigate through the product.
Srilu is leaving the building. I hope you had fun!!

Want a laugh – http://qasrilu.blogspot.com
Want to connect – https://linkedIn.com/in/SriluPinjala
Want to join PNSQC – Srilu.PNSQC@gmail.com