

Building a Better Manual Test Case Spreadsheet

Why?

The first course I took post-University was a QA Testing course. It just wasn't covered enough at University. In the waterfall big telecom days, I wrote 600-page test documents and there would be corresponding forms where you'd fill in your manual test results. As the years went by, I found myself using things like HP Mercury Quality Center, recording test cases, cross-referencing requirements, recording manual results and uploading automated results.

This round of life has found me in a startup with no budget for something like TestRail. I tried out a few Jira packages but they seemed to be feature-poor and flaky.* So, back to basics: a spreadsheet. I have used many different versions of someone's idea of a test case spreadsheet over the years. I wanted something good. This is "good" in my mind but not perfect.

*Since creating this Sheet, I've been introduced to other Jira packages which seem more hopeful. Next to review is <https://marketplace.atlassian.com/apps/1211769/xray-test-management-for-jira?tab=overview&hosting=cloud> Nonetheless, this Google Sheet has some cool features that people may find interesting.

The "Good/Great!"

I wanted to maintain the test case titles/steps in only one place. (If the test case changed, the sheet used to record the results had to show those changes and allow you to easily jump from the results page to the full test case statement page.)

I wanted to be able to jump easily from the test case statement page to the results page. (So that you could fix the test case and then jump back to record the result)

I wanted to easily archive results from a test cycle. (You can just duplicate the Results tab and rename it to the archive tab name). You can also copy the whole Google Sheet.

The sheet had to be a "live"/"shared" thing so that multiple testers could make updates and you could label who was working on what.

There had to be a way to define “Common” test cases and sets of these common test cases so that they could all be reused. You want to be able to write a test case set for say phone number validation and then reference it in say 20 different high-level test cases that are part of say 5 different forms on app pages. Then when someone changes the code to validate the phone number differently, then you just update 1 place in the common test cases. (I have seen too much copy/paste .. maintenance nightmare, and wasted time in my life... common test cases needed to exist)

Common Test ID	Test Title	
User Name Format		
HT-C-MissingAtSign	@' Allowed	[Confirm a User Name without an '@' sign is invalid]
HT-C-UpperCase	Uppercase Allowed	[Confirm a User Name with all Capital letters is valid]
HT-C-LowerCase	Lowercase Allowed	[Confirm a User Name with a Lowercase letters is valid]
HT-C-Plus	Plus sign Allowed	[Confirm a User Name with a '+' sign is valid]
HT-C-Hyphen	Hyphen Allowed	[Confirm a User Name with a '-' is valid]
HT-C-Underscore	Underscore Allowed	[Confirm a User Name with a '_' is valid]
HT-C-Tilde	Tilde Allowed	[Confirm a User Name with a '~' is valid]
HT-C-8Char	8 characters or more	
HT-C-Weak	Cannot be particularly weak (like password123)	
HT-C-Mixed	Must contain some mix of letters, numbers, and/or symbols	
HT-C-NoBlankStart	Cannot start or end with a blank space	

CS-UserName-0001	[Valid - Invalid Formats]	HT-C-Missing@ Sign	@: Allowed [Confirm a User Name without invalid]	HT-C-UpperCa se	Uppercase Allowed [Confirm a User Name with all Capital letters is valid]	HT-C-LowerCa se	LowerCase Allowed [Confirm a User Name with a Lowercase letters is valid]	HT-C-Plus	Plus sign Allowed [Confirm a User Name with a "+" sign is valid]	HT-C-Hyphen	Hyphen Allowed [Confirm a User Name with a "-" is valid]	HT-C-Underscore	Underscore Allowed [Confirm a User Name with a "_" is valid]	HT-C-Tilde	Tilde Allowed [Confirm a User Name with a "~" is valid]
------------------	---------------------------	--------------------	---	-----------------	--	-----------------	--	-----------	---	-------------	---	-----------------	---	------------	--

[illegible]

Test Case Results Page

HT-Login-004	LINK	Tap "Sign In" button, confirm "Invalid password" Exception Pop-up Message	Passed
HT-Login-005	LINK	[Valid - Invalid Formats] HT-C-MissingAtSign '@' Allowed [Confirm a User Name without an '@' sign is invalid] HT-C-UpperCase Uppercase Allowed [Confirm a User Name with all Capital letters is valid] HT-C-LowerCase Lowercase Allowed [Confirm a User Name with a Lowercase letters is valid] HT-C-Plus Plus sign Allowed [Confirm a User Name with a '+' sign is valid] HT-C-Hyphen Hyphen Allowed [Confirm a User Name with a '-' is valid] HT-C-Underscore Underscore Allowed [Confirm a User Name with a '_' is valid] HT-C-Tilde Tilde Allowed [Confirm a User Name with a '~' is valid]	Passed
HT-Login-005b	LINK	Confirm User Name limit Cannot start or end with a blank space	Passed
HT-Login-006	LINK	Confirm password limit 8 characters or more 8 characters or more	Passed
HT-Login-007	LINK	Confirm password limit Cannot be particularly weak (like password123) Cannot be particularly weak (like password123)	Passed
HT-Login-008	LINK	Confirm password limit Must contain some mix of letters, numbers, and/or symbols Must contain some mix of letters, numbers, and/or symbols	Passed
HT-Login-009	LINK	Confirm password limit Cannot start or end with a blank space Cannot start or end with a blank space	Passed
HT-Login-010	LINK	Go to and confirm Login Page Step 1 Step 2 Step 3 Step 4	Passed

The other copy/paste/maintenance waste was restating the “context” where a test case or step was to be done. The Web/App world is generally a hierarchy where you go to some page or pane within a page, go to some sub-page, fill in some fields and click some button. Then fill in more fields and click some other button on that same sub-page. The problem is test result forms tend to be linear; hoping/dreaming that a setup statement in the test case, tells everything you need to know. And that setup gets copy-pasted and becomes a maintenance problem. To address this, sort of, in a minimalistic, “Startup” kind of way ... the test cases can be placed in any cell on a ROW after column H. So, you can show a hierarchy by the placement of the test case description cells. You can even use multiple cells in a row to break things into Steps.

24	HT-Login-001	RESULT				Go to and confirm Login Page	
25	HT-Login-002	RESULT					Enter a valid user name - email address
26	HT-Login-003	RESULT					Enter a password that isn't correct for the account
27	HT-Login-004	RESULT					Tap "Sign In" button, confirm "Invalid password" Exception Pop-up Message
28	HT-Login-005	RESULT					[Valid - Invalid Formats] HT-C-MissingAtSign '@' Allowed [Confirm a User Name without an '@' sign is invalid] HT-C-UpperCase Uppercase Allowed [Confirm a User Name with all Capital letters is valid] HT-C-LowerCase Lowercase Allowed [Confirm a User Name with a Lowercase letters is valid] HT-C-Plus Plus sign Allowed [Confirm a User Name with a '+' sign is valid] HT-C-Hyphen Hyphen Allowed [Confirm a User Name with a '-' is valid] HT-C-Underscore Underscore Allowed [Confirm a User Name with a '_' is valid] HT-C-Tilde Tilde Allowed [Confirm a User Name with a '~' is valid]
			CS-UserName-0001				
33	HT-Login-009	art				Confirm password limit	Cannot start or end with a blank space Cannot start or end with a blank space
34	HT-Login-010					Go to and confirm Login Page	Step 1 Step 2 Step 3 Step 4
35							

But what do you see on the Results page when you have this hierarchy?

Well, every ROW is assumed to be a test case. So, you don't have "Setup Rows", you have Test Cases that are run before test cases in a sub-hierarchy position. Each Results page test case is just the TEXT for that ROW. It is literally all concatenated together. You have to know that the test case is in a sub-hierarchy position. The only way to know is to hope the linear flow of tests makes this obvious OR you click the "Link" back from the test case result page test case item to the full test case statement page, where you would see the hierarchy layout of the tests and observe the parent test cases giving the context.

There isn't a formal place in the hierarchy for prerequisites, setup, test data, or expected results. You have to work with just the series of free-form test sheet cells.

Note: There is an Image Link column on the "TestTree" tab (full test case statement page) where a link to the expected result images could be stored for each test case.

The "Bad"

The truly "Bad" is having to ALWAYS update each tab \$B2 with the URL of that exact tab page. Once you set it up, you are good to go. (See Sheet Readme tab for how getting the URL resulted in problems, it has to be hardcoded in a cell)

There are natural sheet limitations

Clicking on a "Link" or any formula can result in editing the formula and messing things up. Then you will have to copy the formula from a non-edited cell.

Yes, columns B and C, in Results, have formulas in them. Yes, if you insert a Row, you will have to copy those formulas from another Row. This is a spreadsheet. This isn't a custom app.

I'd NOT recommend having multiple people update the same sheet when they are first getting to understand how this works. Maybe even give them only read-only access, print them a results tab sheet and have one person enter the results. Then when people are more familiar (trusting), open it up to more people making changes.

THE LINK TO THE SHEET!!!!

https://docs.google.com/spreadsheets/d/1mx90J--iOOGhOAO2RwE_MBREsYadhUJZk887vEtjTVA/edit?usp=sharing

Licensing

MIT License

Copyright (c) 2023 Howard Thompson

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.