Test Cases, Test Suites and Test Case management systems

Why do we need such a thing?

Astea Solutions QA Team

astea SOLUTIONS

Questions

- Define the terms failure, fault, and error
- What is defect masking?
- Explain the difference between testing and debugging
- List several misconceptions about testing
- Why do we test

What is test case?

What is test case?

- Document
- Contains defined test conditions
  - preconditions for execution
  - input data
  - expected outputs or the expected behavior of the test object
- Should have a high probability of revealing previously unknown faults

Why do we need to create test cases?

- Requirements traceability
  - Structured way to verify customer needs are satisfied
- Requirements traceability matrix
  - Shows the relationship between requirements and test cases
  - Table that shows many to many relationships
  - No full names are displayed - it is just a summary
  - When a requirement is changed

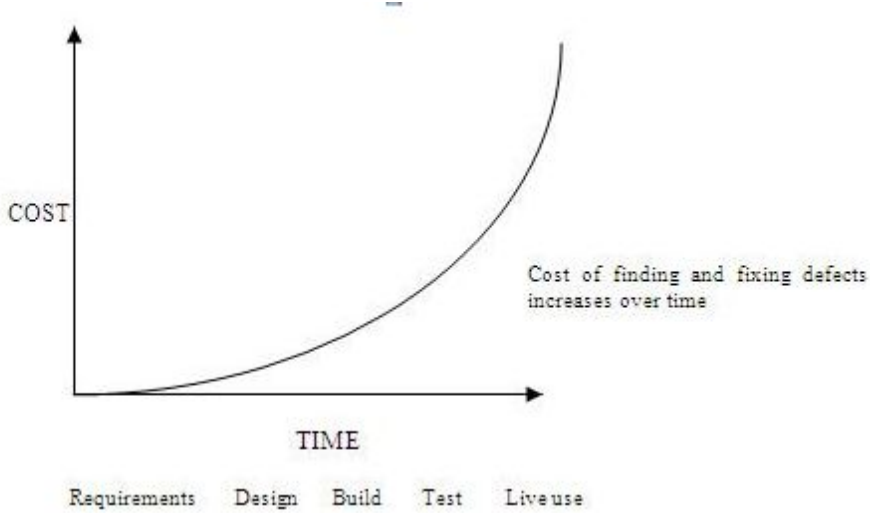
Requirements traceability matrix (RTM)

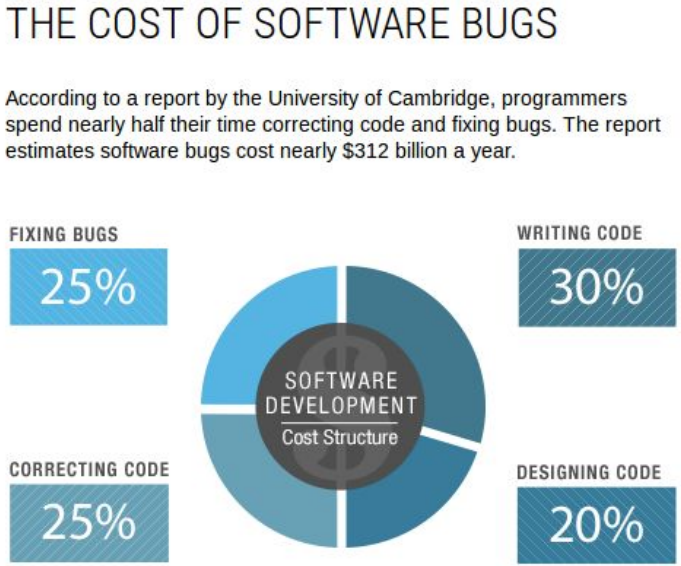
	A	B	C	D	E	F	G	H	I	J	K	L
1	Requirements		Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10
2	Test Cases	Totals	1	2	4	2	2	5	5	5	2	4
3	TC1	3	X		X							X
4	TC2	1		X								
5	TC3	1		X								
6	TC4	2				X	X					
7	TC5	2				X	X					
8	TC6	3						X	X	X		
9	TC7	3						X	X	X		
10	TC8	3						X	X	X		
11	TC9	3						X	X	X		
12	TC10	3						X	X	X		
13	TC11	1									X	
14	TC12	1									X	
15	TC13	2			X							X
16	TC14	2			X							X
17	TC15	2			X							X

astea SOLUTIONS

Why do we need to create test cases(2)

- Early bug detection





Testing SHOULD start as early as possible!

astea SOLUTIONS

Why do we need to create test cases(3)

- Audit trail of testing done
  - Formal documentation what will be/was tested
  - Prove that some functionality is tested

Why do we need to create test cases(4)

- Re-usability/Regression testing
  - Can be executed over and over on each iteration
  - Used for test automation

- Regression testing - the process of testing changes to computer programs to make sure that the older programming still works with the new changes.

Why do we need to create test cases(5)

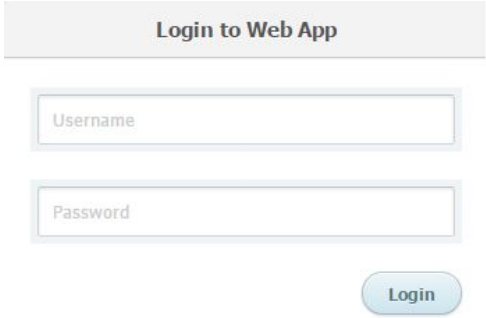
- Knowledge transfer

Test case structure

- ID
- Name
- Priority
- Date/Author
- Description
- Preconditions
- Test data
- Steps
- Expected result

Test Case Demo

Test Case DeMo-1: Demo		
Author:	admin	
Summary: Log into the system		
Preconditions: Google Chrome		
#:	Step actions:	Expected Results:
1	Enter “user1” in username field	“user1” is entered
2	Enter “password1” in password field	“password1” is entered
3	Click on “Login” button	User is successfully logged in
Execution type:	Manual	
Keywords:	login	



astea SOLUTIONS

Test case types

By Details:

- High level
  - Input data, steps and expected result are not specified
  - Each QA execute it differently and can find new bugs
  - Does not guarantee complete tests even of major functionality
  - Easy to maintain

High level test case demo

Test Case DeMo-2: High level demo		
Author:	admin	
Summary: Enter positive integer in age field and submit it		
Preconditions: Google Chrome		
#:	Step actions:	Expected Results:
1		
Execution type:	Manual	
Keywords:		

Test case types (2)

By Details:

- Low level
  - Detailed steps and expected result with given input data
  - Every QA will execute it exactly the same and will not miss regression
  - Narrows testers’ horizon and creativity
  - Hard to maintain

Low level test case demo

Test Case DeMo-3: Low level demo		
Author:	admin	
Summary: Here we will make a low level test		
Preconditions: Google Chrome		
#:	Step actions:	Expected Results:
1	Enter 42 in the field	42 is entered
2	Click on Submit button	The data is submitted successfully
Execution type:	Manual	
Keywords:	age	

Test case types (3)

By Expected result:

- Positive
  - Verify main functions are working correctly (best/happy path)
  - Verify cases of expected use
  - Ensure customers can achieve what product is designed for
  - Should always be performed

Positive test case demo

Test Case DeMo-4: Positive test demo		
Author:	admin	
Summary: Here we will make a positive test		
Preconditions: Google Chrome		
#:	Step actions:	Expected Results:
1	Enter 42 in the field	42 is entered
2	Click on Submit button	**The data is submitted successfully**
Execution type:	Manual	
Keywords:	age	

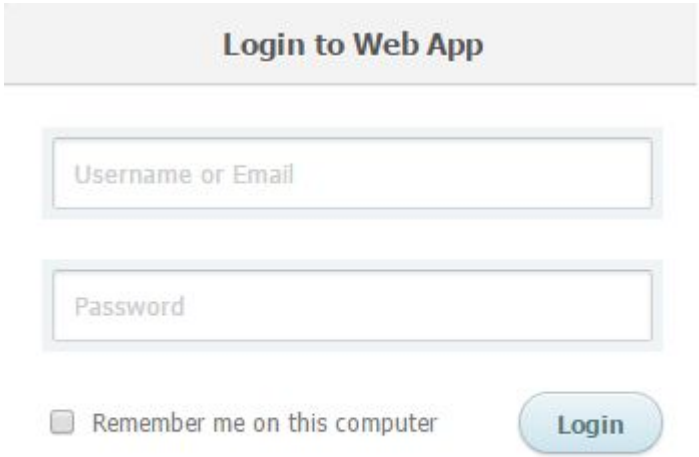
Test case types (4)

By Expected result:

- Negative
  - Verify system does not crash in case of invalid data
  - Verify customer will not get bad experience in case of incorrect or malicious usage
  - Should be performed in mission critical applications or products used by real customers

Negative test case demo

Test Case DeMo-5: Negative test demo		
Author:	admin	
Summary: Here we will make a negative test		
Preconditions: Google Chrome		
#:	Step actions:	Expected Results:
1	Enter “abcd” in the field	“abcd” is entered
2	Click on Submit button	**Data is not submitted.**
Execution type:	Manual	
Keywords:	age	



astea SOLUTIONS

Test case execution and expected result

- Test case execution is finished once we compare Actual vs. Expected results
  - PASS – Actual result corresponds to Expected result
  - FAIL – Actual result does not correspond to Expected result
  - Blocked – other bug prevents us from executing this test case
- Best theoretical case: one expected result
  - Focus on only one thing while testing
- **The reality**: many expected results
  - Each step can have one or more expected result
  - Keep number of expectations low (1-3)
  - Combine only related (to each other) expected results
  - Can distract the person executing the test case
  - Optimizes test effort

Test case maintainability

- **Maintainability** - The readiness of a software product to be modified to correct defects, to meet new requirements, to make future maintenance easier, or to adapted to a changed environment
- Software changes often and a lot
  - This requires changes (add, update, remove) in test cases
- Design test case to be easily modified
  - Separate common steps in dedicated document – Knowledge base
  - Test case maintainability depends of its level of details
  - Do not add obvious or unrelated steps

Test case maintainability

TEST CASE STEPS
1. Go to http://main.sharelane.com. 2. Click link "Test Portal". 3. Click link "Account Creator". 4. Press button "Create new user account". 5. Copy email to the clipboard.
6. Go to http://main.sharelane.com. 7. Paste user email into textbox "Email". 8. Enter **password** into textbox "Password". 9. Press button "Login".
10. Enter **search keyword** into textbox "Search". 11. Press button "Search".
12. Press button "Add to Cart".
13. Click link "Test Portal". 14. Click link "Credit Card Generator". 15. Select needed card from drop-down menu. 16. Press button "Generate Credit Card". 17. Copy card number to the clipboard.
18. Go to http://main.sharelane.com. 19. Click link "Shopping cart". 20. Press button "Proceed to Checkout". 21. Select appropriate value from drop down menu "Card Type". 22. Paste card number into text box "Card Number". 23. Press button "Make Payment".
24. Write down order id: ___.
25. Run **SQL1**

LOGICAL MODULE
1. Create new user.
2. Login.
3. Find a product.
4. Add product to Shopping Cart.
5. Generate Credit Card
6. Do Checkout.
7. Get Order ID.
8. Query DB.

Bad practices

- Dependency between test cases
  - Independent test case does not refer to another test case and does not rely on other test case to bring the system in certain state
- Poor step description
  - Things clear and obvious now for you can become unclear in a month

Bad practices (2)

- Poor expected result description
  - It should be clear what and why this is the expected result
  - Do not make references to an external documents
  - Everything works as expected
- Long or complex test cases
  - Other people are going to execute your test cases !!!

Good practices

- Stage of the project
  - High level test cases are preferred in early stages of the project
  - The more mature projects become and the more users are using it, the level of details and number of test cases increases
- Always keep your test cases up to date
- Test the simple stuff (if time permits) cases for specific part of the project or specification

Good practices (2)

- Put yourself in the shoes of the end user

Good practices (2)

- Put yourself in the shoes of the end user
- Update test cases if new scenarios come to your mind
- Importance of the project
  - Critical projects should be thoroughly tested
  - This implies detailed test cases covering different situations
- Complexity of the project
  - Important projects are complex
  - Require more test data

Test suite

- Combination of test cases that check specific part of the project or specification
- Test cases and test suites are used to formally document test process
- Structure:
  - Author
  - Spec ID
  - Developer
  - Priority (usually 1-4)
  - OVERVIEW – represents the IDEA of the test suite
  - Test cases – Execution result

Test Case management systems

- Microsoft Excel
- Testlink
- Zephyr
- QA Complete, HP Quality center etc.

TestLink

- TestLink is a web based test management and test execution system. It enables quality assurance teams to create and manage their test cases as well as to organize them into test plans. These test plans allow team members to execute test cases and track test results dynamically.
- Open Source Software

TestLink demo

https://oblacheto.com/testlink

Summary

- What is a test case?
- Test case structure?
- Test case types?
- Good and bad practices

Resources

https://github.com/TestLinkOpenSourceTRMS/testlink-code/raw/testlink_1_9/docs/testlink_user_manual.pdf

Homework

- Go to https://oblacheto.com/testlink, register, create a test suite with your Faculty number and make 8 test cases (for an application/website of your choice):
  - 2 high level positive test cases
  - 2 high level negative test cases
  - 2 low level positive test cases
  - 2 low level negative test cases
- Export the test suite and submit it in moodle

Questions