AIRA Annual Meeting 2013

October 9, 2013

OPTIMIZING ACCEPTANCE TESTING

Authors:

Katie Reed

Gary Wheeler

Jennifer Neuls

Tina Skinner

Eric Schuh

What is User Acceptance Testing

- UAT
- Verification the application/functionality meets the requirements defined in the Design Document or Change Request.
- Requirements are translated into Test Cases to conduct UAT.

Who is responsible for UAT

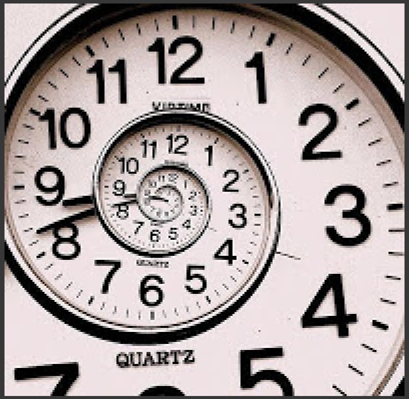
- Business Owner
- Typically UAT will be the basis for formal sign off on acceptance of changes to system

Avoid assumptions

- You are accepting the enhancement(s).
- This applies if “IT” is internal or external, either way you are the customer.
- Confirm prior to conducting UAT.
  - Ensure you know what **steps** of the SDLC have occurred prior to your starting UAT.
  - Clearly define **expectations** before you start in terms of what types of testing you expect to occur prior to start

When do you conduct UAT

- Design Document or Change Request
- Architecture Design Document
- Detailed Test plan
  - Specific to Design Document
  - Build in for Regression Testing
- Peer Review Design Walkthrough
- Unit Testing
- Peer Code Review
- Detailed Test Plan conducted
- System/Integration/Load Testing
- Regression Testing
- **User Acceptance Testing**

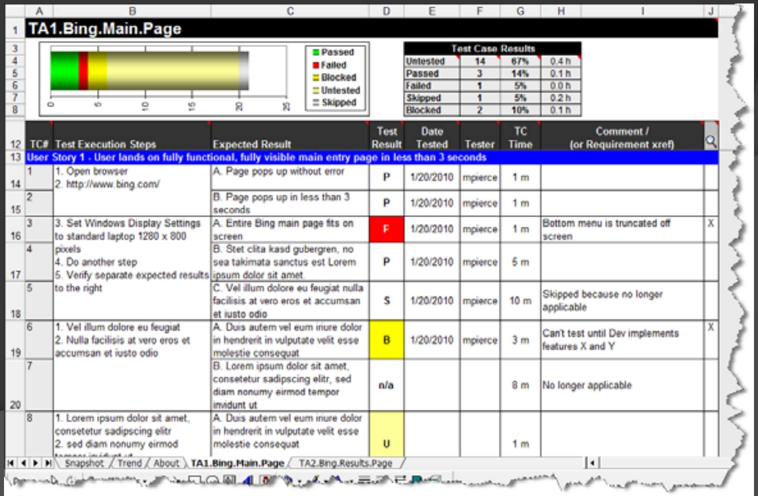


How To Prepare

- Think through timing and resource impact on your staff.
- Ideally start writing test cases and plans after requirements are approved. Work from approved requirements.
- UAT should not be conducted until the entire test plan is written.
- Create test data files for all scenarios prior to testing.
- Factor in UAT into the overall project plan

Types of Test Cases

- Requirements based test cases
- Data Driven
- Business Process, Workflow, User scenario – Ensures the system supports the workflow or user needs of the system



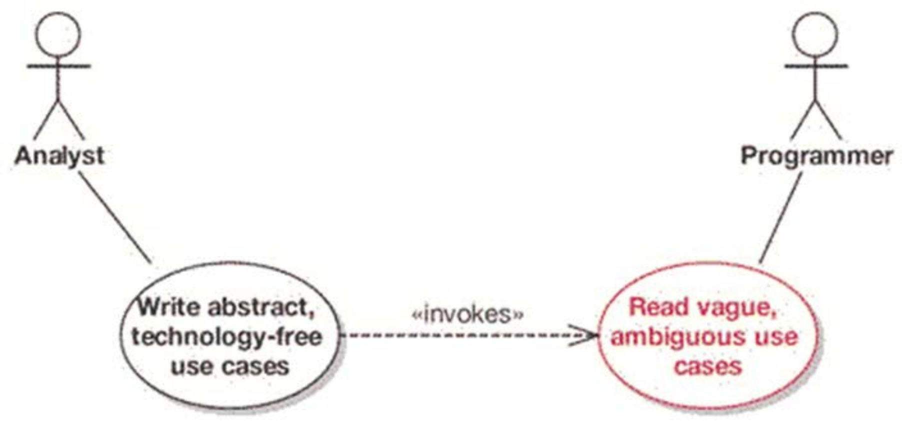
TA1.Bing.Main.Page

Test Case Results			
Untested	14	67%	0.4 h
Passed	3	14%	0.1 h
Failed	1	5%	0.0 h
Skipped	1	5%	0.2 h
Blocked	2	10%	0.1 h

TC#	Test Execution Steps	Expected Result	Test Result	Date Tested	Tester	TC Time	Comment / (or Requirement xref)	
User Story 1 - User lands on fully functional, fully visible main entry page in less than 3 seconds								
1	1. Open browser 2. http://www.bing.com/	A. Page pops up without error	P	1/20/2010	mpierce	1 m		
2		B. Page pops up in less than 3 seconds	P	1/20/2010	mpierce	1 m		
3	3. Set Windows Display Settings to standard laptop 1280 x 800 pixels 4. Do another step 5. Verify separate expected results to the right	A. Entire Bing main page fits on screen	F	1/20/2010	mpierce	1 m	Bottom menu is truncated off screen	X
4		B. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.	P	1/20/2010	mpierce	5 m		
5		C. Vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio	S	1/20/2010	mpierce	10 m	Skipped because no longer applicable	
6	1. Vel illum dolore eu feugiat 2. Nulla facilisis at vero eros et accumsan et iusto odio	A. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat	B	1/20/2010	mpierce	3 m	Can't test until Dev implements features X and Y	X
7		B. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut	n/a			8 m	No longer applicable	
8	1. Lorem ipsum dolor sit amet, consetetur sadipscing elitr 2. sed diam nonumy eirmod tempor invidunt ut	A. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat	U			1 m		

Test Plan Development

- Think through happy path and document
- Think through worst case scenario
- Modifications of the two above


Analyst
Programmer
Write abstract, technology-free use cases
«invokes»
Read vague, ambiguous use cases

Sample: Use Case Development

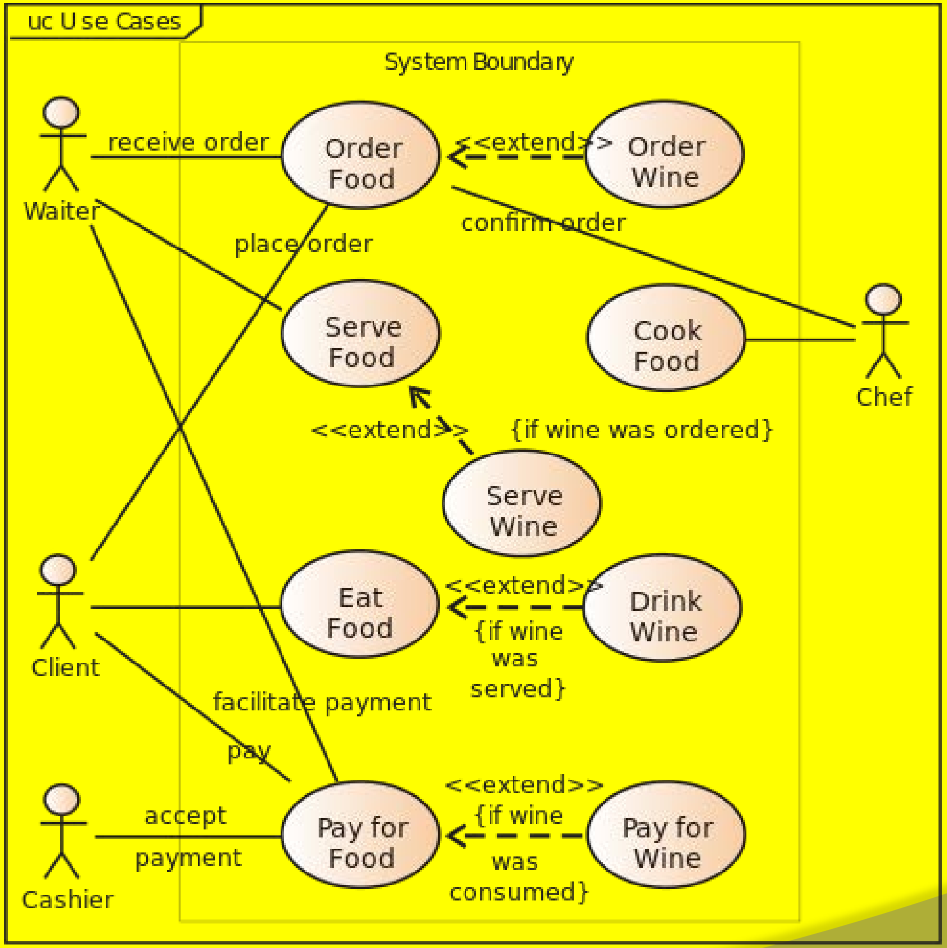
What is a Use Case: From WIKIPEDIA

- In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

Use Case or….

- Use Case is just one example.
- Need to create what is appropriate and sustainable.
- Could be a diagram, could be prose….
- Could be created together with vendor/IT staff. Process may help understand both sides better.

Sample Use Case



Source: WIKIPEDIA

Sample Test Plan

- Identify Section being tested (order food)
- First describe Test Condition (there may be a few or many) order food – happy path
- Second describe steps or conditions to complete action (triggering event) – walk in restaurant, sit, receive menu, choose items, provide order to waiter
- Third describe expected result – order place successfully
- Fourth describe actual result -
- Determine Pass/Fail

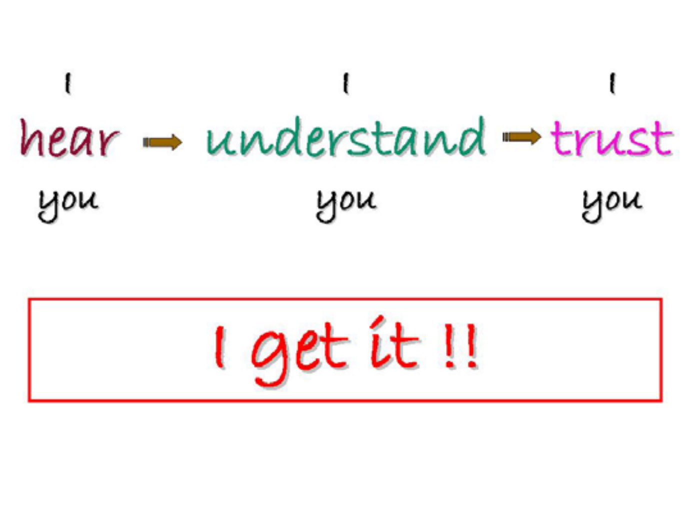
Sample Test Plan

- Identify Section being tested (order food)
- First describe Test Condition (there may be a few or many) order food – unhappy path - allergy
- Second describe steps or conditions to complete action (triggering event) – walk in restaurant, sit, receive menu, choose items, provide off menu order to waiter, waiter denies order due to can’t accommodate allergies
- Third describe expected result – order not placed successfully
- Fourth describe actual result -
- Determine Pass/Fail

What can be leveraged?

- Test Plans
- Test files – data exchange
- Test Populations
- Scheduler Test Decks

During UAT



I hear you → I understand you → I trust you

I get it !!

- Communication
  - How to communicate
  - Who is point of contact
  - How to manage bugs/issues, resolution, additional releases into UAT

- How long is enough?
  - Know your projected timeframes.
  - Ask for IT/Vendor estimates for testing.
  - Will you conduct 100% of their testing? Will you have your own testing?
  - Preferably you hit testing hard up front so resources are most likely available for quick resolution. Longer you wait, harder it is……

Challenges

- Time Pressure
  - Move pieces out faster (not Agile)
  - Introduce work from dynamic development environment
  - Limited regression testing (can’t, cause not done)
- Lack of Resources
  - Don’t forget help desk, trainers, health departments may be able to help
- Requirements vs what you thought you wanted.

Things to keep in mind:

- UAT duration, should be known up front as your are planning the schedule. Sets better timeframe expectations.
- UAT is testing approved requirements. Helps to use design document as frame of reference when working through UAT.
- If you can spend time up front planning the better UAT will go.
- Engage the same personnel as much as possible.

QUESTIONS?

catherine.reed@hp.com