Acceptance Tests

Determining That a Story Is Complete

Acceptance Tests

- Also called Customer Written Tests
  - Should be developed by or with the customer
- Purpose is to determine if a story has been completed to the customer's satisfaction
- Client equivalent of a unit test
  - On the level of a story
  - Black box test
- Not the same as a developer's unit test
  - On the level of methods/classes/algorithm
  - White box test

Benefits to Acceptance Tests

- Can serve as a contract for the client/developers
  - Requires stories are testable
  - User stories are understandings; acceptance tests are requirements the developers must meet
- Client can track progress by observing the total number of acceptance tests growing and % of passing tests increasing
- Developers get more confidence that work is being done and can cross stories off their list when acceptance tests pass

Writing Acceptance Tests

- Sooner or later?
  - If sooner, can help drive the development. However, as you work on a story, the understanding may change
  - If later, can avoid changes that may result but also reflect the story that was actually implemented
  - Your call as to when to solicit acceptance tests
    - Could be around story gathering, after stories are complete, after an iteration and can be displayed to the customer, when stories mostly complete, etc.
- If a story can't be tested then it needs to be clarified with the customer (or perhaps removed)

Acceptance Tests in XP

- Simple version
  - Customer writes the acceptance tests with help from the developer and the user stories
  - Developers write code to make the acceptance tests pass, reports results to the customer
- Using an acceptance test framework
  - Customers write acceptance tests in some format (e.g. fill in tables in a spreadsheet)
  - Framework maps tests to code stubs that will perform the tests
  - Developer fills in the code for the framework that will perform the actual tests
  - Upon running tests the framework automatically maps the results to a format for the customer to understand (e.g. HTML)
  - Framework makes it easier to run regression tests, allow the customer to track progress
    - Not required for this class; could run tests on top of JUnit or other framework
    - Fit framework described in Chapter 14 along with examples

Sample Acceptance Test

- Writing cash register software
- Acceptance Test: Shopping cart for generating a receipt
  - Create a shopping cart with:
    - 1 lb. coffee, 3 bags of cough drops, 1 gallon milk
    - Prices: Coffee $6/lb, cough drops $2.49/bag, milk $4.95/gallon
    - Verify total is $18.42
- Test might span multiple stories (fill shopping cart, checkout, view receipt…)
- Other tests might verify sales tax is calculated correctly, coupons properly discounted, etc.
- Not comprehensive tests, but specific cases to test user stories and functionality

Writing Acceptance Tests

- You can write most of them just like a unit test
- Invoke the methods that the GUI would call

```
inventory.setPrice("milk", 4.95);
inventory.setPrice("cough drops", 2.49);
inventory.setPrice("coffee", 6.00);

order.addItem("milk", 1);
order.addItem("cough drops", 3);
order.addItem("coffee", 1);

order.calculateSubtotal();

assertEquals(order.receipt.getsubtotal(), 18.42);
```

- Easy to automate

Running Acceptance Tests

- You can also run them manually, such as through a GUI interface
  - Select milk from the drop down menu
  - Enter 1 and Click on “add” button
  - Select coffee from the drop down menu
  - Enter 1 and Click on “add” button
  - Select cough drops from the drop down menu
  - Enter 3 and Click on “add” button
  - Verify shopping cart subtotal displays $18.42
- Useful to run, avoid relying completely on this technique as it is slow, time consuming, and hence not feasible for regression testing

Fit format example

Press miscButton		
Check display	Enter Unit Price	
Enter unitPrice	800	
Check display	800	
Press	Enter	
Check display	Enter number of items	
Enter 5		
Check display	5	
Press doneButton		
Check display	4000	
Check totalCost	4000	
Press endButton		

Automating GUIs

- Possible to automate GUI testing as well
- Program simulates (or records) clicking, dragging, etc. on the app and re-creates them
  - Ex. Test Automation FX
    - http://www.testautomationfx.com/tafx/tafx.html
  - (google others, keyword GUI testing)

Acceptance Tests Are Important

- Gives customer some satisfaction that features are correctly implemented
- Not the same as Unit Test
  - Unit tests could pass but acceptance tests fail, especially if acceptance test requires the integration of components that were unit-tested