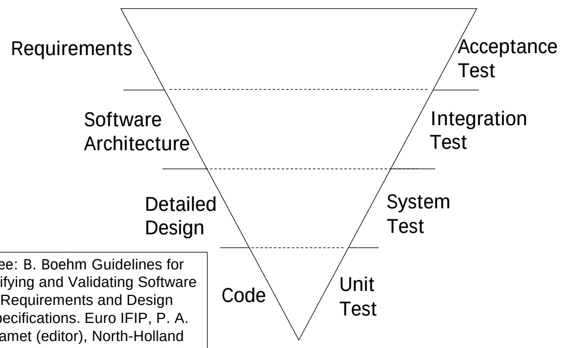
Acceptance Testing Tools - A different perspective on managing requirements

Wolfgang Emmerich
Professor of Distributed Computing
University College London
http://sse.cs.ucl.ac.uk

Learning Objectives

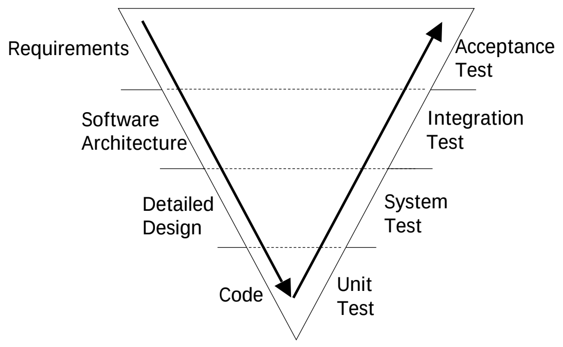
- Introduce the V-Model of quality assurance
- Stress the importance of testing in terms of software engineering economics
- Understand that acceptance tests are requirements specifications
- Introduce acceptance and integration testing tools for Test Driven Development
- Appreciate that automated acceptance tests are executable requirements specifications

V-Model in Distributed System Development



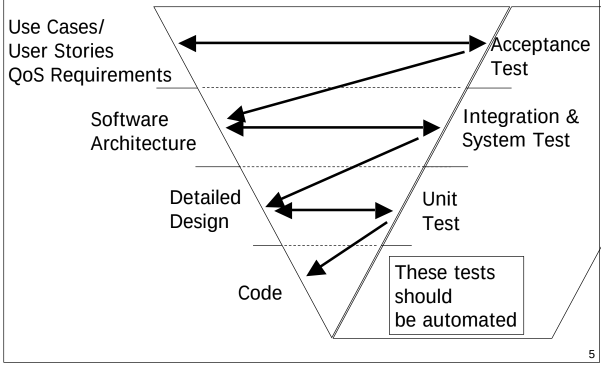
See: B. Boehm Guidelines for Verifying and Validating Software Requirements and Design Specifications. Euro IFIP, P. A. Samet (editor), North-Holland Publishing Company, IFIP, 1979.

Traditional Software Development



4

Test Driven Development of Distributed Systems


Use Cases/ User Stories QoS Requirements
Software Architecture
Detailed Design
Code
Acceptance Test
Integration & System Test
Unit Test
These tests should be automated

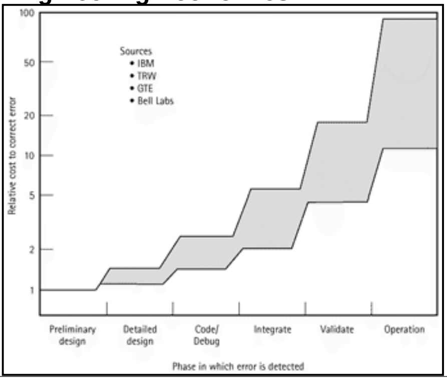
5

Advantages of Test Driven Development

- Early definition of acceptance tests reveals incomplete requirements
- Early formalization of requirements into automated acceptance tests unearths ambiguities
- Flaws in distributed software architectures (there often are many!) are discovered early
- Unit tests become precise specifications
- Early resolution improves productivity (see next slide)

6

Software Engineering Economics



See: B. Boehm: Software Engineering Economics. Prentice Hall. 1981

7

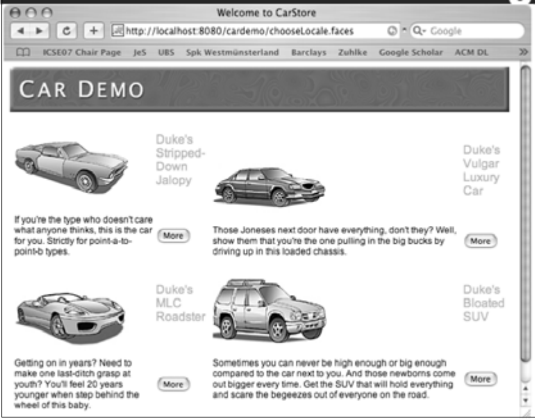
An Example

Consider an on-line car dealership

User Story:

- I first select a locale to determine the language shown at the user interface. I then select the SUV I want to buy. The system would allow me to customize it but I am happy with the base version. The dealership shows me the configuration and I confirm. I then enter my address and credit card details and the system confirms that the car will be shipped soon.

8



Duke's Stripped-Down Jalopy

If you're the type who doesn't care what anyone thinks, this is the car for you. Strictly for point-a-to-point-b types.

Duke's Vulgar Luxury Car

Those Joneses next door have everything, don't they? Well, show them that you're the one pulling in the big bucks by driving up in this loaded chassis.

Duke's MLC Roadster

Getting on in years? Need to make one last-ditch grasp at youth? You'll feel 20 years younger when step behind the wheel of this baby.

Duke's Bloated SUV

Sometimes you can never be high enough or big enough compared to the car next to you. And those newborns come out bigger every time. Get the SUV that will hold everything and scare the begeezes out of everyone on the road.

9

Acceptance Testing the Car Dealer Web Site

- Translate each user story into a number of acceptance tests
- Cover both main flow of events and alternative flows
- Automate all acceptance tests, otherwise
  - you can't be agile!
  - it's too expensive
  - you'll get bored with manual testing
  - you might not detect defects that are re-introduced after they have been fixed
  - you might not be aware that a fix has broken other parts of the system
- Run automated acceptance tests whenever new candidate release is to be deployed.
- If you find a defect in a deployed system write a new test case that catches the defect before fixing it.

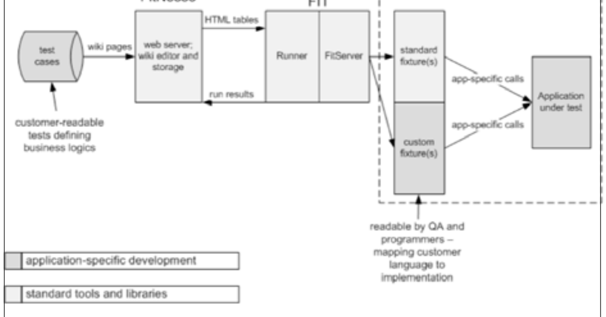
10

Test Automation Tools

- Automating tests is hard
- Fortunately it can be simplified by test automation tools
- There are numerous commercial tools and a few open source tools available developed by the agile development community
- Of these we discuss Fit/FitNesse http://www.fitnesse.org in more detail.

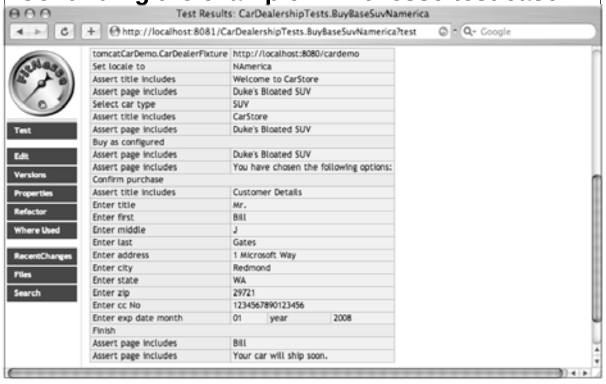
11

Automated Acceptance Tests with FIT/FitNesse


FitNesse
FIT
test cases
wiki pages
web server; wiki editor and storage
HTML tables
run results
Runner
FitServer
standard fixture(s)
custom fixture(s)
app-specific calls
app-specific calls
Application under test
customer-readable tests defining business logics
readable by QA and programmers – mapping customer language to implementation
application-specific development
standard tools and libraries

12

Continuing the example: A FitNesse test case


Test Results: CarDealershipTests.BuyBaseSuvNamerica
http://localhost:8081/CarDealershipTests.BuyBaseSuvNamerica?test
Google
Test
Edit
Versions
Properties
Refactor
Where Used
RecentChanges
Files
Search

tomcatCarDemo.CarDealerFixture	http://localhost:8080/cardemo		
Set locale to	NAmerica		
Assert title includes	Welcome to CarStore		
Assert page includes	Duke's Bloated SUV		
Select car type	SUV		
Assert title includes	CarStore		
Assert page includes	Duke's Bloated SUV		
Buy as configured			
Assert page includes	Duke's Bloated SUV		
Assert page includes	You have chosen the following options:		
Confirm purchase			
Assert title includes	Customer Details		
Enter title	Mr.		
Enter first	Bill		
Enter middle	J		
Enter last	Gates		
Enter address	1 Microsoft Way		
Enter city	Redmond		
Enter state	WA		
Enter zip	29721		
Enter cc No	1234567890123456		
Enter exp date month	01	year	2008
Finish			
Assert page includes	Bill		
Assert page includes	Your car will ship soon.		

Fixtures

- Automating these tests requires translation of the domain specific language used in the acceptance test into invocations of the system under test.
- Performed in FIT through Fixtures.
- Fixtures are a thin layer of “glue” code.
- May require specific drivers to interface with the system (e.g. httpunit for web pages, JDBC for databases or JMS for message queues).

14

Continuing the example - `CarDealerFixture`

```
…
public boolean setLocaleTo(String to){
  WebForm localeForm;
  WebRequest setLocaleRequest;
  if (to.equals("NAmerica") || to.equals("France") ||
      to.equals("SAmerica") || to.equals("Germany")) {
    carDealer=new WebConversation();
    carDealerURL=new GetMethodWebRequest(args[0]);
    try {
      HttpUnitOptions.setExceptionsThrownOnScriptError( false );
      carDealerResponse=carDealer.getResponse(carDealerURL);
      localeForm=carDealerResponse.getFormWithID("localeButtons");
      SubmitButton lang = localeForm.getSubmitButton("localeButtons:"+to);
      setLocaleRequest=localeForm.getRequest(lang);
      carDealerResponse=carDealer.getResponse(setLocaleRequest);
    } catch (Exception e){e.printStackTrace(); return false;}
    return true;
  }  else
    return false;
}
```

15

Building testable distributed systems

- Automated tests are distributed systems, too.
- Need “interfaces” for your fixtures in the system under test
- Example:

```
<!-- chooseLocale.jsp -->
<h:form id="localeButtons”> …
 <h:panelGrid id="buttons" columns="4"
       summary="#{bundle.chooseLocale}"
       title="#{bundle.chooseLocale}">
  <h:commandButton id="NAmerica”
      action="storeFront”
      value="#{bundle.english}” …>
  </h:commandButton> …
</h:form>
```

```
…
public boolean setLocaleTo(String to){
  WebForm localeForm;
  WebRequest setLocaleRequest;
  carDealer=new WebConversation();
  carDealerURL=new GetMethodWebRequest(
  try {
      HttpUnitOptions.setExceptionsThrow
      carDealerResponse=carDealer.getResponse(carDealerURL);
      localeForm=carDealerResponse.getFormWithID("localeButtons");
      SubmitButton lang = localeForm.getSubmitButton("localeButtons:"+to);
      setLocaleRequest=localeForm.getRequest(lang);
      carDealerResponse=carDealer.getResponse(setLocaleRequest);
  } catch (Exception e){e.printStackTrace(); return false;}
    return true;
}
```

16

User Story revisited

- Acceptance tests lead to better understanding of user story.
- Keep user story updated and stored alongside tests
- Example:

Before:

I first select a locale to determine the language shown at the user interface. I then select the SUV I want to buy. The system would allow me to customize it but I am happy with the base version. The dealership shows me the configuration and I confirm. I then enter my address and credit card details and the system confirms that the car will be shipped soon.

After:

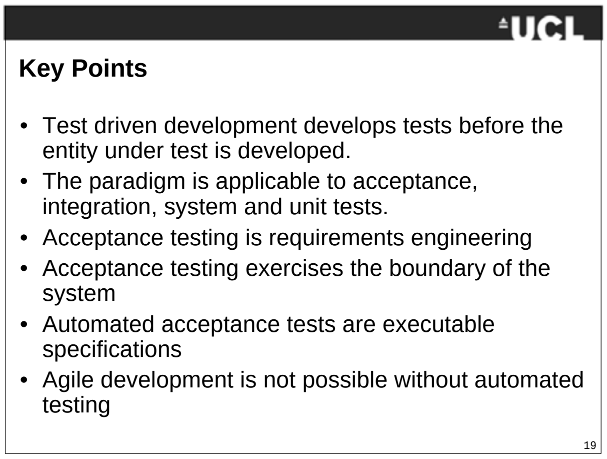
From the choice of supported locales (NAmerica, SAmerica, France and Germany) I choose NAmerica. I then select that I want to buy the SUV. The system would allow me to customize it but I am happy with the base version. The dealership shows me the configuration and I confirm. I then enter my address and credit card details and the system confirms that the car will be shipped soon.

17

But …

- How can you write the tests without having the distributed system yet?
- Solutions:
  - In Agile development most often you have parts of the distributed system already
  - You also often already have the middleware
  - Green-field developments are rare and you already have existing components
  - Use Mock components and objects for the really new stuff.
  - Build Mock user interfaces (relatively fast using JSPs) - this also helps in eliciting new requirements

18



Key Points

- Test driven development develops tests before the entity under test is developed.
- The paradigm is applicable to acceptance, integration, system and unit tests.
- Acceptance testing is requirements engineering
- Acceptance testing exercises the boundary of the system
- Automated acceptance tests are executable specifications
- Agile development is not possible without automated testing

19