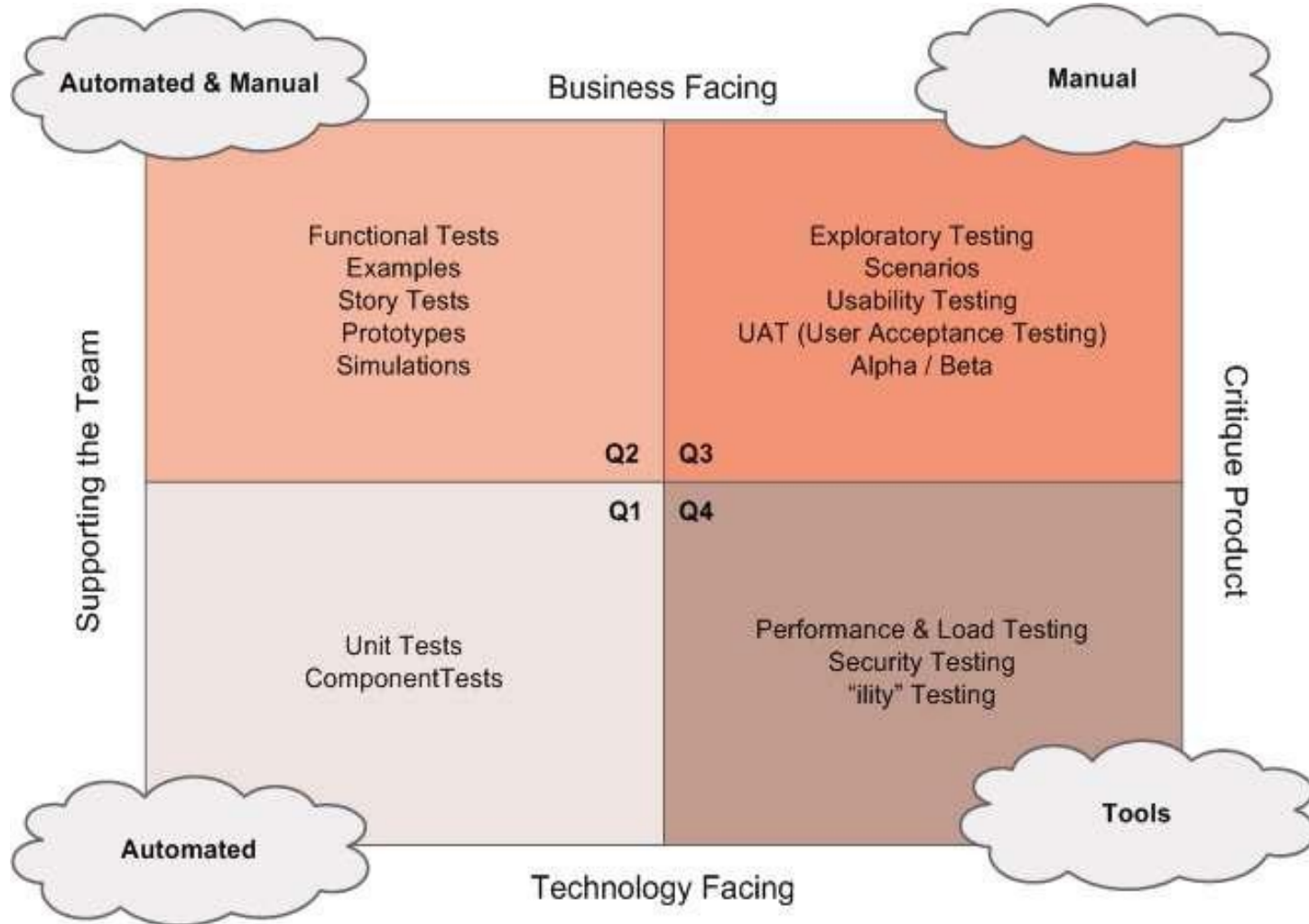


Agile acceptance testing tools round-up

Acceptance testing tools: Support the team and face the business



- Agile testing quadrants idea by Brian Marick, picture by Janet Gregory

Unit test = target for development:

```
@Test
public void testCanDoubleDown() {
    HandScore handScore = new HandScore(2, 13, 3);
    Assert.assertTrue(ANY.canDoubleDown(handScore));
    Assert.assertFalse(NONE.canDoubleDown(handScore));
    Assert.assertTrue(SOME_SOFT_AND_HARD.canDoubleDown(handScore));
    Assert.assertFalse(SOME_HARD.canDoubleDown(handScore));

    handScore = new HandScore(2, 14, 4);
    Assert.assertTrue(ANY.canDoubleDown(handScore));
    Assert.assertFalse(NONE.canDoubleDown(handScore));
    Assert.assertFalse(SOME_SOFT_AND_HARD.canDoubleDown(handScore));
    Assert.assertFalse(SOME_HARD.canDoubleDown(handScore));

    handScore = new HandScore(2, 12, 20);
    Assert.assertTrue(ANY.canDoubleDown(handScore));
    Assert.assertFalse(NONE.canDoubleDown(handScore));
    Assert.assertTrue(SOME_SOFT_AND_HARD.canDoubleDown(handScore));
    Assert.assertFalse(SOME_HARD.canDoubleDown(handScore));
}
```

But they aren't really readable

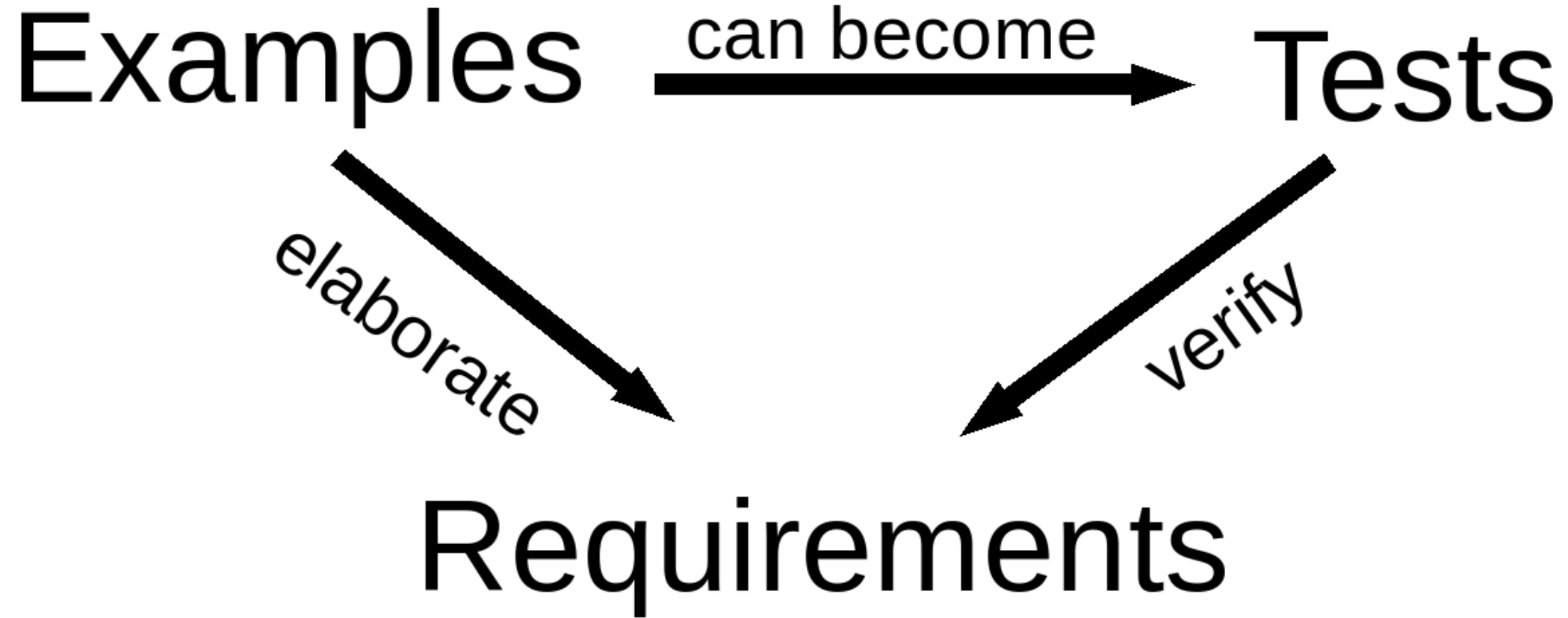
```
@Test
public void testPlayersGetNewStatus() throws GameException {
    expectGameStart();
    Player p1 = new Player(BigDecimal.valueOf(17),BigDecimal.valueOf(17),"");
    Player p2 = new Player(BigDecimal.valueOf(18),BigDecimal.valueOf(18),"");
    EasyMock.expect(gameRules.getPlayers(gameStatus1)).andReturn(new Player[]{p1, p2}
    //ExecutionContext context = new ExecutionContext(GAME_ID, gameStatus1, v, "A", Tabl
    EasyMock.expect(gameRules.getStatusForPlayer(EasyMock.eq(p1), (ExecutionContext) Easy
    EasyMock.expect(gameRules.getStatusForPlayer(EasyMock.eq(p2), (ExecutionContext) Easy
    EasyMock.expect(gameRules.execute(new ExecutionContext(GAME_ID, gameStatus, v, "a", t
        (GameWalletImpl)EasyMock.anyObject()), EasyMock.eq(command))).andReturn(new E
    dd.notifyPlayer(EasyMock.eq(t.getId()),
        EasyMock.eq(BigDecimal.valueOf(17)),
        EasyMock.eq(new GameStatusDocument(1L, gameStatus1, "12345"))));
    EasyMock.expectLastCall();
    dd.notifyPlayer(EasyMock.eq(t.getId()),
        EasyMock.eq(BigDecimal.valueOf(18)),
        EasyMock.eq(new GameStatusDocument(1L, gameStatus, "12345"))));
    EasyMock.expectLastCall();
    EasyMock.replay(a, dd, gameStatus, gameStatus1, gameRules);
    executeCommand();
    EasyMock.verify(dd);
}
```

Here's what I thought...

- When the order is in a “pending” state, we first check the account, and if approved move it to “confirmed” state
- When the order is manually confirmed, it moves from the “pending” to “confirmed” state even if the account does not have enough funds
- When the order is in a “pending” state for two days, we send an alert

Here is what they saw:

- ghorgh [the] [order] 'oH Daq [a] ["pending"]
[state] maH wa'DIch [check] [the] [account] 'ej
chugh [approved] vIH 'oH Daq ["confirmed"]
[state] ghorgh [the] [order] 'oH [manually]
[confirmed] 'oH vIHtaH vo' [the] ["pending"] Daq
["confirmed"] [state] 'ach chugh [the] [account]
ta'taH ghobe' ghaj yap [funds] ghorgh [the]
[order] 'oH Daq [a] ["pending"] [state] vaD cha'
jajmey maH ngeH [an] [alert]




FIT/FitNesse

TicketReviewTests.WinningsRecordedCorrectly - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:8888/TicketReviewTests.WinningsRecordedCorrectly



Test

Edit

Properties

Refactor

Where Used

Search

Files

Versions

Recent Changes

Draw on 02/01/2008 is open

Player john buys a ticket with numbers 1,3,4,5,8,10 for draw on 01/01/2008

Player john buys a ticket with numbers 1,3,4,5,8,10 for draw on 02/01/2008

Player john buys 5 tickets with numbers 3,6,9,12,15,18 for draw on 01/01/2008

Numbers 1,3,4,5,31,32 are drawn on 01/01/2008

Player	john	lists tickets for draw on	01/01/2008
value	numbers	is open	winnings
10	1,3,4,5,8,10	false	3
50	3,6,9,12,15,18	false	0

Done

5 key facts about FitNesse

- Web-based wiki system
- Aimed at collaborative test management
- Supports Java, .NET, C++, Python, Smalltalk...
- Based on FIT (and now Slim) test engines
- Uses tables to describe tests

FitNesse allows us to specify automated but human readable business tests!

The prize pool is divided among the winners using the following distribution for winning combinations (number of correct hits out of six chosen numbers).

The example below is for \$2M payout pool.

Prize Distribution for Payout Pool	2,000,000	
Winning Combination	Pool Percentage?	Prize Pool?
6	68	1,360,000
5	10	200,000
4	10	200,000
3	12	240,000

A PLAYER CAN BUY TICKETS FOR DIFFERENT DRAWS AND THEY WILL ALL SHOW UP SEPARATELY ON HIS STATEMENT.

Player buys a ticket with numbers	1,5,17,44,22,19	for the draw on	01/01/2009
-----------------------------------	-----------------	-----------------	------------

Player buys a ticket with numbers	1,5,17,44,22,19	for the draw on	02/01/2009
-----------------------------------	-----------------	-----------------	------------

Player buys a ticket with numbers	10,25,13,42,19,18	for the draw on	01/01/2009
-----------------------------------	-------------------	-----------------	------------

Player receives a statement	
-----------------------------	--

draw	numbers
01/01/2009	1,5,17,44,22,19
01/01/2009	10,25,13,42,19,18
02/01/2009	1,5,17,44,22,19

How FIT works

Test Table

The prize pool is divided among the winners using the following distribution for winning combinations (number of correct hits out of six chosen numbers). The example below is for \$2M payout pool.

Prize Distribution for Payout Pool	2,000,000	
Winning Combination	Pool Percentage?	Prize Pool?
6	68	1,360,000
5	10	200,000
4	10	200,000
3	12	240,000

Fixture

```
public class PrizeDistributionForPayoutPool:fit.ColumnFixture {
    private WinningsCalculator wc = new WinningsCalculator();
    public int winningCombination;
    public int PoolPercentage()
    {
        return wc.GetPoolPercentage(winningCombination);
    }
    public decimal? payoutPool;
    public decimal PrizePool()
    {
        if (payoutPool == null) payoutPool = Decimal.Parse(Args[0]);
        return wc.GetPrizePool(winningCombination, payoutPool.Value);
    }
}
```

Domain code

```
public class WinningsCalculator
{
    public int GetPoolPercentage(int combination)
    {
        switch(combination) {
            case 6: return 68;
            case 5: return 10;
            case 4: return 10;
            case 3: return 12;
            default: return 0;
        }
    }
    public decimal GetPrizePool(int combination, decimal payoutPool)
    {
        return payoutPool * GetPoolPercentage(combination) / 100;
    }
}
```

Where it is really good

- Specifying business logic for complex domains at domain/API level
- Live documentation
- Collaboration between testers, business and developers

TextTest

The screenshot displays the TextTest GUI with two windows. The main window, titled 'TextTest static GUI : management of tests for TextTest', shows a tree view of test categories. The 'TestSelf' category is expanded, showing 'TestSelection' and 'Console'. The 'Console' sub-category is selected, displaying a list of tests with their status and details. The status bar indicates '15 of 21 tests completed'.

The 'TextTest dynamic GUI : testing TextTest under /...' window shows a summary of test results. The status bar indicates '15 of 21 tests completed'. The 'Status' tab is active, showing a table of test results:

Status	Number	Visible
Not started	0	<input checked="" type="checkbox"/>
Pending	1	<input checked="" type="checkbox"/>
Running	5	<input checked="" type="checkbox"/>
Succeeded	12	<input checked="" type="checkbox"/>
Failed	3	<input checked="" type="checkbox"/>
Differences	3	<input checked="" type="checkbox"/>
catalogue different	1	<input checked="" type="checkbox"/>
output different	1	<input checked="" type="checkbox"/>
errors different	1	<input checked="" type="checkbox"/>

The 'TestSelf' window shows the following test results:

Test Name	Status	Details
NoFilters	Success	
TestNameFilter	Success	
MultipleTests	Success	
MultipleTestsWithSpace	Success	
TestSuiteFilter	Success	
TestSuiteFullPath	Failed	output different
FileNameFilter	Success	
FileFilterInConfig	Failed	catalogue different
FileMetaFilter	Success	
FileFilterMultiApp	Failed	errors different
FileFilterMultiPaths	Success	
FileFilterCombine	Running	
CombineSeveralConfig	Success	
GrepFilter	Running	
GrepFilterOtherFile	Success	
TestNameSubstrings	Running	
RegExpTestName	Success	
MultipleCopies	Success	
CompositeCopies	Running	
AutoSortTestSuites	Running	
AutoReversedSortTestSuites	Pending	

Shortcuts: **New**
 TextTest started at 15Sep13:09:13.

Shortcuts: **New**
 Started 21 tests at 15Sep13:09:12.

5 Key facts about TextTest

- Tool for managing system behaviour changes
- Works by analysing text files, not API assertions
- Written in Python, but integrates with almost anything
- Integrates with xUseCase recorders
- Two GUI tools for test management

Smart text file comparison

output.demo vs. output.demo - TkDiff 4.0

File Edit View Mark Merge Help

1 : 8c8 Merge: Diff: Mark:

C:\TEXTTEST\TESTS\demo\AddDup\output.demo JUNE~1\ALLTIP~1\LOCALS~1\Temp\demotmp03May145954\AddDup\outp

1		1	
2	'set new movie name to' event created with arguments 'Star Wars'	2	'set new movie name to' event created with arguments 'Star Wars'
3		3	
4	'add movie' event created with arguments "	4	'add movie' event created with arguments "
5	Adding movie 'Star Wars'. There are now 1 movies.	5	Adding movie 'Star Wars'. There are now 1 movies.
6		6	
7	'add movie' event created with arguments "	7	'add movie' event created with arguments "
8	! ERROR : 'Star Wars' has already been added!	8	! Adding movie 'Star Wars! There are now 2 movies.
9		9	
10	'close' event created with arguments "	10	'close' event created with arguments "
11	Exiting the video store!	11	Exiting the video store!

xUseCase recording

- Record GUI interaction and assign meaningful names to actions
- Compose actions into tests
- Manage tests from a higher level
 - Easier to read
 - More resilient to change
 - Focused on the correct level of abstraction

Where it is really good:

- Workflows (specify and inspect steps, not APIs)
- Integrations
- Non-intrusive verifications (no code changes)
- Regression testing of working systems
- Low-level system testing

Robot Framework

The screenshot displays the Robot IDE interface. The window title is "Robot IDE". The menu bar includes "File", "Edit", "Tools", and "Help".

Left Panel (Project Structure):

- Golden
 - User Keywords
 - Variables
- *Tests
 - Test Cases
 - User Keywords
 - My Test Setup**
 - My Suite Setup
 - My Test Teardown
 - My Suite Teardown
 - User Keyword
 - Variables

Right Panel (My Test Setup):

My Test Setup

Arguments: [] Edit Clear

Documentation: [] Edit Clear

Timeout: [] Edit Clear

Return Value: [] Edit Clear

Add Row Add Column

1	Log	Hello from test setup				
2	Shou					
3	Should Be Empty	Builtin				
4	Should Be Equal	Builtin				
5	Should Be Equal As Integers	Builtin				
6	Should Be Equal As Numbers	Builtin				
	Should Be Equal As Strings	Builtin				
	Should Be True	Builtin				
	Should Contain	Builtin				

Keyword Details (for "Should Be Equal"):

Arguments: [first | second | msg=None | values=True]

Fails if the given objects are unequal.

- If `msg` is not given, the error message 'first != second'.
- If `msg` is given and `values` is either

5 key facts about Robot framework

- Keyword-based
- Works on html, tsv and rst (wiki-like) files
- UI tool for management
- Written in python, supports Java as well
- Extended by various libraries and scripts

Keyword DSL testing

Running Keyword Conditionally - Mozilla Firefox

file:///tmp/robotframework-2.1/doc/examples/conditional_execution/running_keyword_conditional | jbehave

Running Keyword Conditionally

Setting	Value	Value	Value	Value
Documentation	Adding conditions to test cases should be avoided if possible. However, sometimes that kind of construct is needed. For this purpose BuiltIn library contains keywords "Run Keyword If" and "Run Keyword Unless" which use is demonstrated in this suite.			

Variable	Value	Value	Value	Value
`\${COUNTRY}`	Robotland			

Test Case	Action	Argument	Argument	Argument
Running Test Based On The Condition	[Documentation]	If you change `\${COUNTRY}` value or give --variable COUNTRY:your_country from command line the test is executed differently than with default value.		
	Run Keyword If	'Robotland' == `\${COUNTRY}`	Hello World	
	Run Keyword Unless	'Robotland' == `\${COUNTRY}`	Log	Hello `\${COUNTRY}`!

Done

S3Fox

Where it is really good:

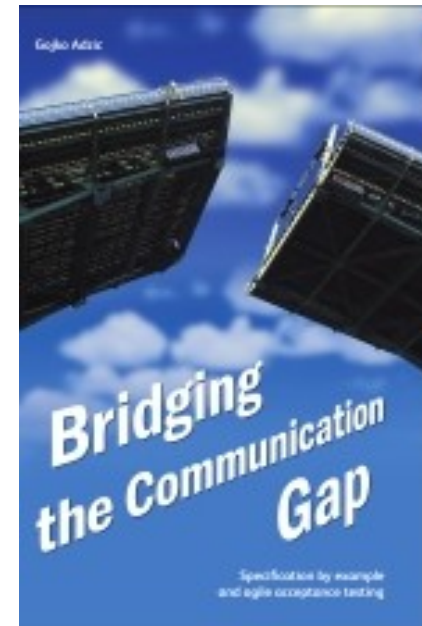
- Scripting/workflows
- UI/Web testing (with Selenium)
- DSL testing
- Technical tests
- Testers building complex scripts themselves

Links

- <http://fitnessse.org>
- <http://www.concordion.org>
- <http://texttest.org>
- <http://www.jbehave.org/>
- <http://robotframework.org>
- <http://gojko.net>

Bridging the Communication Gap

- learn how to improve communication between business people and software implementation teams
- find out how to build a shared and consistent understanding of the domain in your team
- learn how to apply agile acceptance testing to produce software genuinely fit for purpose
- discover how agile acceptance testing affects your work whether you are a programmer, business analyst or a tester
- learn how to build in quality into software projects from the start, rather than control it later



<http://www.acceptancetesting.info>

upcoming events....

- 23/6: Testable software is good software
- 22/7: Agile Testing: Tools and Approaches
- 26/8: Fast Track Test-Driven Development:
Testify your project