A SYSTEMATIC VIEW OF USER ACCEPTANCE TESTING

Geoff Quentin
QBIT Ltd
Geoff Quentin

Geoff Quentin has an M.Sc. in Computer Science and is a Chartered Engineer. His career in IT spans nearly three decades. He has been running testing courses since 1981 and the current portfolio of courses can be found at www.qbit.co.uk.

Geoff has run courses in testing and related topics in the UK, the USA, Australia, the Middle and Far East and Europe, both publicly and in-house, and well over twenty thousand delegates world-wide have attended his courses. Geoff's earlier work experience research and development of printers, disk drives and cpus as well as training and project management of programmers and development staff.

Geoff has been a member of several working parties connected with testing and has written or contributed to a number of publications. He has spoken at STAR conferences before and other conferences in the UK, USA, Europe, the Middle East and Asia. He has also spoken to many UK groups such as the Quality Assurance Forum, the Software Quality Management Specialist Group and other branches of the British Computer Society, The Durham Centre for Software Maintenance, the NCC and many others.

Geoff was Founder Chairman of the British Computer Society Specialist Interest Group in Software Testing and served for several years as a member of ISEB (Information Systems Examinations Board), the organisation responsible for the syllabus and running of the certification scheme. His work with the BCS, together with his wide lecturing experience, have given him extensive knowledge, both practical and theoretical, of the whole area of testing software.
A Systematic View of User Acceptance Testing

A tutorial for STARWEST 2004 drawn from User Acceptance Testing
A three day course from QBIT Limited
www.qbit.co.uk
73 West Road Shoeburyness, Essex, UK
SS3 9DT, +44 (0) 1702 290558
About this presentation

- Presented by Geoff Quentin C Eng, M Sc. MBCS, founder chairman of the BCS Specialist Interest Group in Software Testing
- This talk is intended for all testers from junior user to acceptance test team leaders
Material drawn from the portfolio of courses

- Established over 20 years ago
- Given to many people >20,000
- Many countries
- Many projects
  - safety critical, financial, real time . . .
- The basis of formal qualifications
- www.qbit.co.uk for the full portfolio
Contents

• The basics of acceptance testing
  – the vital and specific activity
• Setting test objectives
  – making the tests bite within ISO 12207
• Acceptance testing throughout the project
  – not the same as system testing
• The acceptance test plan
• The test script
What I will cover

what acceptance testing is specifically
when acceptance testing should be started
who should be involved
what are the cost issues
what are the methods
what documentation should be produced
how this works within selected parts of ISO 12207
Three definitions

1. Test - any activity to identify or measure
2. Acceptance test - ensuring acceptability
3. Acceptance criteria - the preferred and expected results

Please note that the testing activity goes beyond just finding bugs
The acceptance testing model

Test

Accept

Reject

Test object

Tested object

With listed anomalies

Perhaps with anomalies

Copyright © QBIT Ltd August 2004  A systematic view of user acceptance testing 7
Acceptance?

• A deliverable needs to be shown to be acceptable:
  to users?
    users of systems?
    programmers using program specifications?
  to customers?
• If un-acceptable it must be possible to reject
The essential independence of the tester and the model of testing

The tester’s path

Start

Test point

The developer’s path

acceptable

Anomalies, issues, bugs, modifications
Analysis of an acceptance test

• A typical test objectives can be used
  – Use a phrase like: To show . . .
  – Or a phrase like: To demonstrate . . .
  – Must not use: To prove . . .
The test objective needs more detail

• To show that under specified circumstances the test object does not . . .
  – looking for fail safe attributes
• Specific anomalies accepted others not
• The test environment is part of the test
• All need to be defined before the test is run
• *May replace the detailed specification*
A selection of test objectives

• To show that when I . . .
• To demonstrate that when such and such is done the following happens . . .
And the acceptance test

• We must be able to describe each test
• Each test must fit within the strategy
• And each needs to be justifiable
• A good test represents the real world
• A good acceptance test represents typical usage of the system
• Testing may be analytical or empirical
Analytical and empirical testing

• Analytical
  – by analysis - use the review, prototypes and simulation
  – low cost - especially the review
  – the final deliverable may exist in many forms (paper being one)
  – anomalies found tend to improve the final deliverable

• Empirical
  – tends to be expensive
  – press a button and record what happens
  – done on the actual deliverable
  – tends to be late in the development life cycle
  – anomalies found tend to damage the final deliverable
  – also delay delivery and add costs
What is the acceptance criterion?

- Specific and detailed - great
- Objective - good
- Subjective - poor
- Ambiguous - very poor
- Critical - worrying
- Unimportant - why do it?
- Not economically possible - review the project!
Acceptable to whom?

- Managers - cost and time
- Users - support the business process
- Developers - ?
- Auditors - compliance
- Trainers - achievable skills
- Customers - back for more
The relationship model

The acquirer

The supplier, developer and builder

So who does what testing and when?
The testing activities

Acceptance test

The acquirer

Acceptance test planning

System test planning

The supplier, developer and builder

System test
And when do we run this A-test?

• After delivery - can the goods be returned?
• Before delivery - can it be refused?
• During construction - impact of change?
• During design - modification of the design
• During specification - clarifying the requirement
• Before the project starts - is this possible?
And who runs the test?

- Nobody - oops!
- Specifier - bias?
- Developer - bound to pass (?)
- Builder - requirement not understood
- User - qualified? skilled enough?
- Test specialist - skilled but not the true user
- Does not matter or anybody - oops again!
And the economics of the A-test

- May be cheap and easy to run
- Could be expensive and difficult to run
- May be impossible to run
- Should be documented
- May be repeatable or ‘one-off’
- Decided by the benefits of success and the cost of failure
Testing within the development

The tester’s path

So what is the start point?

The developer’s path

This could be a unit test or an integration test or a system test or even an acceptance test.
The acceptance test among others

- Acceptance test - define this first
  - especially the commencement and completion
- System qualification test
  - developed from the acceptance test
- System test
- Software qualification test
- Integration test
- Unit test
The structure of ISO 12207

5. PRIMARY LIFE CYCLE PROCESSES

5.1 Acquisition

5.2 Supply

5.3 Development

5.4 Operation

5.5 Maintenance

6. SUPPORTING LIFE CYCLE PROCESSES

6.1 Documentation

6.2 Configuration Management

6.3 Quality Assurance

6.4 Verification

6.5 Validation

6.6 Joint Review

6.7 Audit

6.8 Problem Resolution

7. ORGANISATIONAL LIFE CYCLE PROCESSES

7.1 Management

7.2 Infrastructure

7.3 Improvement

7.4 Training

Acceptance tester

Developer system tester
The life cycle models - ISO 12207

Acquirer - 5 activities
Supplier - 7 activities
Developer - 13 activities
The acquirer - five activities

1. Initiation of acquisition - business objectives identification
2. Tender preparation - acceptance test strategy planning
3. Contract preparation - acceptance test planning
4. Supplier monitoring - acceptance test design
5. Acceptance and completion - acceptance test run

Many of the contractual requirements are best specified as acceptance test criteria
The supplier - seven activities

1. Initiation of supply
2. Preparation of response
3. Contract - with focus on the acceptance test
4. Planning
5. Execution & control
6. Review & evaluation - ensuring the supply will pass
7. Delivery & completion - supporting the acceptance test

Many of the contractual requirements are best understood by a study of the acceptance test criteria.
The developer - thirteen activities

1. Initiation of development - system test strategy to meet acquirer a-test
2. System requirements analysis - system test plan to meet acquirer a-test
3. System architectural design - system test design
4. Software requirements analysis
5. Software architectural design
6. Software detailed design
7. Software coding and testing
8. Software integration
9. Software qualification testing
10. System integration
11. System qualification testing - system testing
12. Software (system) installation
13. Software (system) acceptance support - acceptance testing

These must match
The 1 2 3 review model of testing

1. Verification
2. Validation
3. Certification
This leads to a revised Gantt chart

Parallel activities of requirements analysis and test planning

Iterative communications between acquirer and supplier/developer

A preview of the revised chart follows and we will return to this later
<table>
<thead>
<tr>
<th>ID</th>
<th>ISO 12207 Process</th>
<th>Dependency</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1</td>
<td>Initiation of acquisition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1.2</td>
<td>Tender preparation</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.1.3</td>
<td>Contract preparation</td>
<td>5.1.2</td>
<td></td>
</tr>
<tr>
<td>5.1.4</td>
<td>Supplier monitoring</td>
<td>5.1.3</td>
<td></td>
</tr>
<tr>
<td>5.1.5</td>
<td>Acceptance &amp; completion</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.2.1</td>
<td>Initiation of supply</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.2.2</td>
<td>Preparation of response</td>
<td>5.2.1</td>
<td></td>
</tr>
<tr>
<td>5.2.3</td>
<td>Contract</td>
<td>5.2.2</td>
<td></td>
</tr>
<tr>
<td>5.2.4</td>
<td>Planning</td>
<td>5.2.3</td>
<td></td>
</tr>
<tr>
<td>5.2.5</td>
<td>Execution &amp; control</td>
<td>5.2.4</td>
<td></td>
</tr>
<tr>
<td>5.2.6</td>
<td>Review &amp; evaluation</td>
<td>5.2.5 &amp; 5.3.1</td>
<td></td>
</tr>
<tr>
<td>5.2.7</td>
<td>Delivery &amp; completion</td>
<td>5.2.4 &amp; 5.2.5</td>
<td></td>
</tr>
<tr>
<td>5.3.1</td>
<td>Initiation of development</td>
<td>5.2.4</td>
<td></td>
</tr>
<tr>
<td>5.3.2</td>
<td>System requirements analysis</td>
<td>5.3.1</td>
<td></td>
</tr>
<tr>
<td>5.3.3</td>
<td>System architectural design</td>
<td>5.3.2</td>
<td></td>
</tr>
<tr>
<td>5.3.4</td>
<td>Software requirements analysis</td>
<td>5.3.3</td>
<td></td>
</tr>
<tr>
<td>5.3.5</td>
<td>Software architectural design</td>
<td>5.3.4</td>
<td></td>
</tr>
<tr>
<td>5.3.6</td>
<td>Software detailed design</td>
<td>5.3.5</td>
<td></td>
</tr>
<tr>
<td>5.3.7.1</td>
<td>Software coding</td>
<td>5.3.6</td>
<td></td>
</tr>
<tr>
<td>5.3.7.2</td>
<td>Software testing</td>
<td>5.3.5; 5.3.7.1</td>
<td></td>
</tr>
<tr>
<td>5.3.8</td>
<td>Software integration</td>
<td>5.3.7.2</td>
<td></td>
</tr>
<tr>
<td>5.3.9</td>
<td>Software qualification testing</td>
<td>5.3.8; 5.3.8</td>
<td></td>
</tr>
<tr>
<td>5.3.10</td>
<td>System integration</td>
<td>5.3.9</td>
<td></td>
</tr>
<tr>
<td>5.3.11</td>
<td>System qualification testing</td>
<td>5.3.1; 5.3.10</td>
<td></td>
</tr>
<tr>
<td>5.3.12</td>
<td>Software installation</td>
<td>5.3.11</td>
<td></td>
</tr>
<tr>
<td>5.3.13</td>
<td>Software acceptance support</td>
<td>5.3.12</td>
<td></td>
</tr>
</tbody>
</table>
Wow - that was complicated!

- Need to establish more guidelines
- Need to analyse what frequently occurs
- Need to establish what **should** occur
- Need to establish the basis of the argument
We need to define each individual acceptance test

At the start a test is documented as:

– identification of the test object
– the test objective(s)
– the pre-conditions
– a description of the test environment
– inputs
– expected outputs
At the conclusion of test execution

The test documentation records:

– status of the test object
– the test objective(s)
– the pre-conditions
– a description of the test environment
– inputs
– expected outputs
– the actual outputs
– an analysis of the results
So what is a good acceptance test?

A good acceptance test:
– is repeatable
– represents reality
– is understood by all involved
– influences design
– demonstrates compliance
– is prioritised
– is driven by the business requirement
The generic test process (GTP)

We need a generic test process to run the tests
A good test process includes
  analysis, design, execution and reporting of tests
Widely applicable and scalable
Manageable
Efficient
Effective
The seven steps of the GTP

1. TEST STATUS REPORTING
2. TEST STRATEGY
3. TEST PLAN
4. TEST ANALYSIS
5. TEST DESIGN
6. TEST SCHEDULING
7. TEST EXECUTION
The QBIT GTP

document flow

1. TEST STATUS REPORTING
2. TEST STRATEGY
3. TEST PLAN
4. TEST ANALYSIS
5. TEST DESIGN
6. TEST SCHEDULING
7. TEST EXECUTION

Down load the white paper from www.qbit.co.uk

A systematic view of user acceptance testing 38
The seven steps of the GTP leading to test execution

1. TEST STATUS REPORTING
2. TEST STRATEGY
3. TEST PLAN
4. TEST ANALYSIS
5. TEST DESIGN
6. TEST SCHEDULING
7. TEST EXECUTION
The GTP and the status indicators

1. TEST STATUS REPORTING
   - test reference assigned

2. TEST STRATEGY
   - business objectives identified
   - critical success indicators established
   - test strategy defined

3. TEST PLAN
   - test objects identified
   - test objectives set
   - objects and objectives prioritised

4. TEST ANALYSIS
   - attributes defined
   - criteria / conditions specified
   - criteria / conditions prioritised

5. TEST DESIGN
   - method identified
   - method created
   - data created
   - test procedure created
   - environment set up

6. TEST SCHEDULING
   - data assigned
   - resources allocated
   - pre-conditions met
   - test ready to run

7. TEST EXECUTION
   - re-test needed
   - regression test needed
   - accept with qualifications
   - accepted
Remember that acceptance testing is unfortunately often an iterative process.

Detection → Location → Correction → Relocation → Quality check the correction

Iteration is costly
More definitions

Re-test
repeat the test to demonstrate correction
doubles to time and cost of the individual test

Regression test
repeat the test to demonstrate absence of knock-on effects
this may double the total test budget (or more)
More about an acceptance test

- The attribute to be tested must be identified
- The attribute must be testable!
- A test method must be available
- Completion criteria must be agreed
- Must be worth doing
- If not done the risk or failure may rise
A selection of attributes

• The system must be user friendly
• The system must record ‘time of transaction’
• The system must be reliable

All need further qualification
Essential support

• Testing provokes change so:
  – Change must be managed
  – Versions must be maintained
  – Configuration management must be provided
• Relate all changes back to the business
• Traceability is essential
The project as viewed by the majority of staff

New system in use

Implementation

So what are the success indicators?
The balancing act

Cost

Function

Performance

Implementation date

Each iteration increases this

Each iteration delays this

Copyright © QBIT Ltd August 2004  A systematic view of user acceptance testing 47
The supplier/developer view using a simple life cycle

Fixed implementation date

Specify
Design
Build
(Test) Implement
Success is elusive as propagation of errors can occur down the life cycle.

Real world requirement

Specify
Design
Build
Implement

Correct
Correct
Correct

Error
Error
Error

Design and build on errors
Hidden and long term

Real world realisation

Copyright © QBIT Ltd August 2004 A systematic view of user acceptance testing 49
A simple project plan

Specification ___
Design (user) ___
Design (tech) _____
Design (prog) _____
Code ___
Unit test ___
System test _____
Acceptance test _____
Installation _
Use the system ___
A simple project plan showing the standard Quality Control points

- Specification
- Design (user)
- Design (tech)
- Design (prog)
- Code
- Unit test
- System test
- Acceptance test
- Installation
- Use the system
Remember that testing is an iterative process

Detection  Quality check the correction

Location  Relocation

Correction

And each iteration costs both time and money
Now to return to ISO 12207

• Scalable
• Robust
• Comprehensive
• Customer driven (acquirer driven)
The acquirer - supplier/developer divide

The ISO 12207 model
Business success drives the acceptance test process - the five acquirer activities

1. Initiation of acquisition - business objectives identification
   * identify the success indicators and measure current performance
   * propose desired performance
2. Tender preparation - acceptance test strategy planning
   * decide how the acceptance testing is to be done
3. Contract preparation - acceptance test planning
   * identify functions, performance attributes and acceptance criteria
4. Supplier monitoring - acceptance test design
   * design tests based on new and established business processes
5. Acceptance and completion - acceptance test run
We start with the business objectives

What is the business?
What constitutes success?
How is success recognised?
Who is going to identify success?
What can be measured to gauge the success?
What is it worth to be successful?
What will it cost if only partially successful?
What are the failure costs and consequences?
Recall the 1 2 3 model of testing

Verification
Validation
Certification

Contract for system delivery

Type 1 testing
Type 2 testing
Type 3 testing
The acceptance tester’s work

1. From the Initiation of Acquisition
   – identify the success indicators - drivers
   – and the risks - prioritisers

2. From the Tender Preparation
   – create the acceptance test strategy

3. From the Contract
   – high level test plan

4. From the Supplier Monitoring
   – the design of the tests
First the critical success indicators - the CSI - from the initiation activity

1. Risks and benefits
   Establish traceability using a test management tool

2. And the overall test strategy must be created as part of the tender process
Next during contract preparation the acceptance test plan is created.

Prepare the function catalogue and identify the associated non-functional attributes.

3. And the overall test strategy must be followed.
Then during supplier monitoring

Add the business processes and the tests

Copyright © QBIT Ltd August 2004  A systematic view of user acceptance testing 61
The result is the early creation of:

- Test strategy
- High level test plans
- Low level test plans
- Test scripts

- What the acceptance tester now needs is notification that the system is ready for testing - the test item transmittal report
The acceptance tester’s view of the project

This early test preparation work has the effect of removing many problems from both the specification and the design and hence removing iteration from the empirical acceptance testing.
The simple project plan now shows this early planning work:

- Specification
- Design system
- Design software
- Design (prog)
- Code
- Plan unit test
- Unit test
- Plan system test
- System test
- Plan acceptance
- Acceptance test
- Installation

QC
The simple project plan showing concurrent activities and improved quality control

Specification _____
Design system _____
Plan acceptance ______QC
Design software _______
Plan system test ______________QC
Design (program) _______
Plan unit test _____QC
Code _______
Unit test _______QC
System test _______QC
Acceptance test ____QC
Installation ____QC

Copyright © QBIT Ltd August 2004  A systematic view of user acceptance testing 65
The system tester’s view and ISO 12207

- Specify business requirements
- System requirements analysis
- System architectural design
- System qualification testing
- Test item transmittal report
- System qualification test team
Now we can make sense of the 12207 model

<table>
<thead>
<tr>
<th>ID</th>
<th>ISO 12207 Process</th>
<th>Dependency</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1</td>
<td>Initiation of acquisition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1.2</td>
<td>Tender preparation</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.1.3</td>
<td>Contract preparation</td>
<td>5.1.2</td>
<td></td>
</tr>
<tr>
<td>5.1.4</td>
<td>Supplier monitoring</td>
<td>5.1.3</td>
<td></td>
</tr>
<tr>
<td>5.1.5</td>
<td>Acceptance &amp; completion</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.2.1</td>
<td>Initiation of supply</td>
<td>5.1.1</td>
<td></td>
</tr>
<tr>
<td>5.2.2</td>
<td>Preparation of response</td>
<td>5.2.1</td>
<td></td>
</tr>
<tr>
<td>5.2.3</td>
<td>Contract</td>
<td>5.2.2</td>
<td></td>
</tr>
<tr>
<td>5.2.4</td>
<td>Planning</td>
<td>5.2.3</td>
<td></td>
</tr>
<tr>
<td>5.2.5</td>
<td>Execution &amp; control</td>
<td>5.2.4</td>
<td></td>
</tr>
<tr>
<td>5.2.6</td>
<td>Review &amp; evaluation</td>
<td>5.2.5 &amp; 5.3.1</td>
<td></td>
</tr>
<tr>
<td>5.2.7</td>
<td>Delivery &amp; completion</td>
<td>5.2.4 &amp; 5.2.5</td>
<td></td>
</tr>
<tr>
<td>5.3.1</td>
<td>Initiation of development</td>
<td>5.2.4</td>
<td></td>
</tr>
<tr>
<td>5.3.2</td>
<td>System requirements analysis</td>
<td>5.3.1</td>
<td></td>
</tr>
<tr>
<td>5.3.3</td>
<td>System architectural design</td>
<td>5.3.2</td>
<td></td>
</tr>
<tr>
<td>5.3.4</td>
<td>Software requirements analysis</td>
<td>5.3.3</td>
<td></td>
</tr>
<tr>
<td>5.3.5</td>
<td>Software architectural design</td>
<td>5.3.4</td>
<td></td>
</tr>
<tr>
<td>5.3.6</td>
<td>Software detailed design</td>
<td>5.3.5</td>
<td></td>
</tr>
<tr>
<td>5.3.7.1</td>
<td>Software coding</td>
<td>5.3.6</td>
<td></td>
</tr>
<tr>
<td>5.3.7.2</td>
<td>Software testing</td>
<td>5.3.5; 5.3.7.1</td>
<td></td>
</tr>
<tr>
<td>5.3.8</td>
<td>Software integration</td>
<td>5.3.7.2</td>
<td></td>
</tr>
<tr>
<td>5.3.9</td>
<td>Software qualification testing</td>
<td>5.3.8; 5.3.8</td>
<td></td>
</tr>
<tr>
<td>5.3.10</td>
<td>System integration</td>
<td>5.3.9</td>
<td></td>
</tr>
<tr>
<td>5.3.11</td>
<td>System qualification testing</td>
<td>5.3.1; 5.3.10</td>
<td></td>
</tr>
<tr>
<td>5.3.12</td>
<td>Software installation</td>
<td>5.3.11</td>
<td></td>
</tr>
<tr>
<td>5.3.13</td>
<td>Software acceptance support</td>
<td>5.3.12</td>
<td></td>
</tr>
</tbody>
</table>
Planning the Acceptance Test

Done in the following steps:
1. Identification of Business Objectives
2. Creation of the test strategy
3. Test planning can be split
   3.1 High level test plan with graded functions and associated non-functional attributes
   3.2 Low level test plan identifying all the scripts needed
4. Detailed test design (the creation of the test cases/scripts)
The initial acceptance test plan

• Aim to produce the best risk reduction
• Graded requirements
• Vital non-functional attributes
• Relate the plan to the business objectives
• Test the plan with the rest of the users
• A good test plan influences the development plan
Follow this with more detail

• The initial plan is at high level
• The test objectives are at high level
• We now need detailed lower level test plans
• The tests need to be recorded using test scripts
• These will be at a lower level
• We need to analyse what can and cannot be tested
• We need to catalogue the scripts
• We will be part of business process re-engineering
Test methods

• Use the 1 2 3 review model
• If a test cannot be created then the design is probably defective (at least ambiguous)
• Create tests from the specification and design documents
• Relate the tests to the functions and business processes to be tested
Work at the business process level

Identify the user level business processes
Each will be self contained
Each will have input(s) and output(s)
Each will have test script(s)

1. Normal
2. Exception
3. Error
4. Performance
Remember the relationships and maintain traceability

1. Add the business processes (not use-case)
2. Add the test(s)
The test creation methods

- Explore the life of the major business components
- Explore the flow of data through the business processes
- Model the relationships between the major business components
- Divide and classify live data
- Analyse specific business rules
- Identify and analyses specific break points in the rules
The business process

The actual user event
- immediate relevance
- real world
- script creation
- script testing
## The script

### The script will have at least:

<table>
<thead>
<tr>
<th>Ref</th>
<th>Test object</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective(s) and completion criteria</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test environment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Expected</th>
<th>Actual</th>
<th>Analysis</th>
</tr>
</thead>
</table>

A good script is a tested script may also have test metrics
Each test supported and authenticated by

- Prototyping
- Simulation
- Modelling
- Mechanisation

We must know in advance the attributes we are looking for - good time for exploratory testing
The tools to support the testing

- Traceability manager
- Test management
- Configuration management
- Change management
- Test capture
- Test playback
- Incident management
- Load simulation
Acceptance test coverage

- Functions
- Non-functional attributes
- Performance
- Usability
- Fit with the business processes
- Internal controls
- General build quality
The relationships develop further as the tests are run and results recorded.
Test documentation

Documents to assist test planning
Documents to assist test design
Documents to assist test running
Documents to report the status of the work

Use existing standards
ISO 12207
ISO 15504
IEEE 1012
IEEE 1028
IEEE 829
BS 7925
So who does what acceptance test work?

Need to produce a comprehensive table of:

– the project stages
– the deliverables
– those involved
– when they are involved
– the contributions from those involved
Typical job titles of those involved

- Testing specialist - test analyst
- Trainer
- Auditor
- Technical author
- True end user
- User line managers
- Manager - Project sponsor
The typical problems

• Not enough
  – time
  – budget
  – resources
  – tools
  – techniques

Overcoming the problems is the responsibility of management
Conclusions

From the ISO 12207 view we have covered:

– what acceptance testing is all about
– when it should be started
– who should be involved
– the CSI
– the functions and non-functional attributes
– the business processes
– the test scripts
– the documentation
Thank you

• Any more information required?

geoff@qbit.co.uk