

# Test Strategies in Agile Projects



## sast Q 15



# Contents

- Agile testing
- Test methods, tools and planning
- Definition of Done
- User Stories
- Test ideas and guidelines
- Test charters and exploratory testing
- Test reporting
- When to stop testing

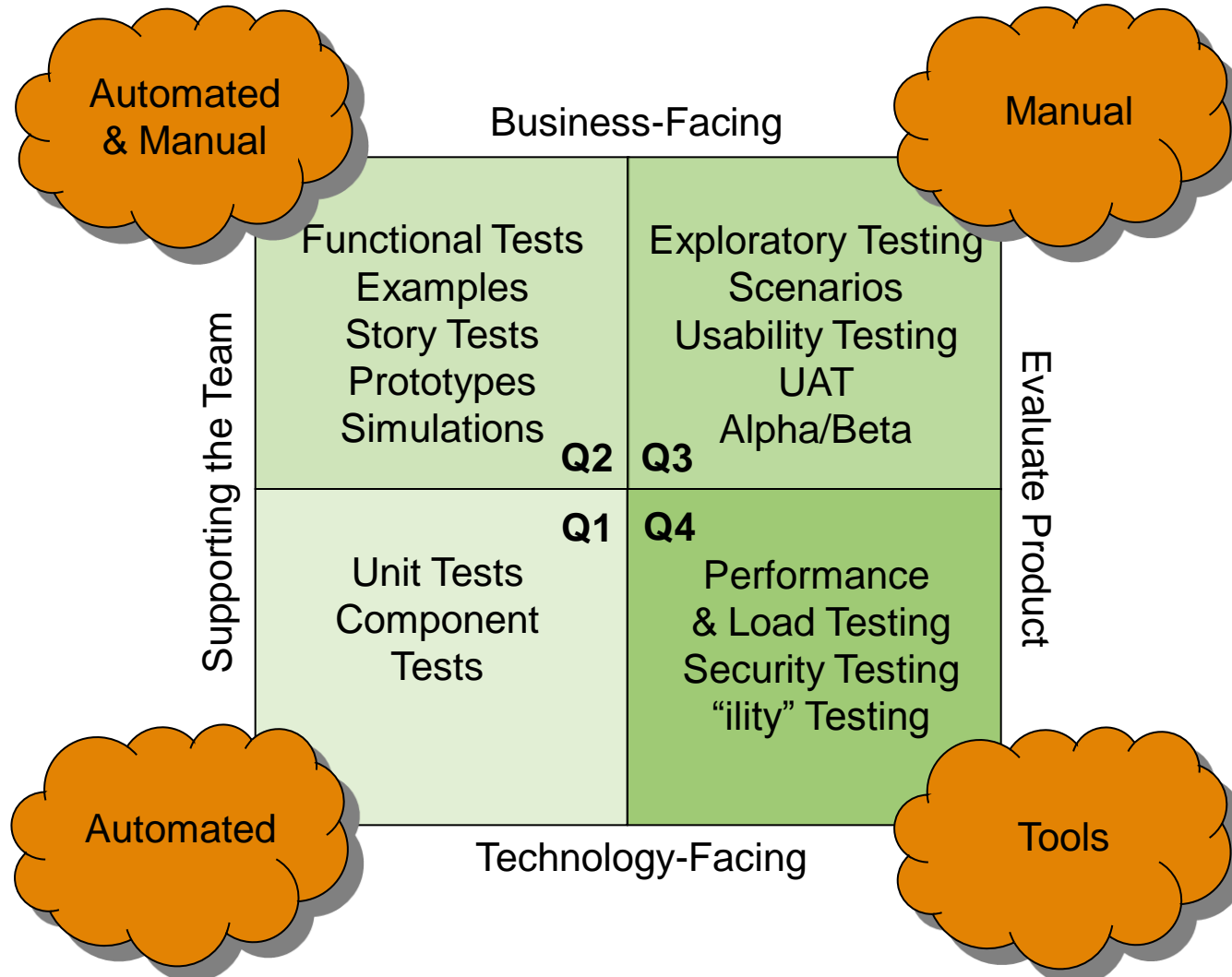


# Agile Testing

- An iterative process from a customer's perspective
- Testing is performed early and often
- Testers are part of the development team
- “User Stories” are tested
- Close cooperation with developers and customers
- Continuous integration and regression tests
- All test results are logged
- Defects are reported



# Agile Testing Quadrants



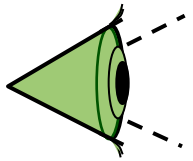
Brian Marick's agile testing matrix

# Test Methods and Techniques

- Requirements based testing
- Design based testing
- Risk based testing
- Exploratory testing
- Error guessing
- Taxonomy based testing
- Attack based testing
- Model based testing
- Scenario based testing
- Combinatorial testing
- Value based testing
- Prototyping

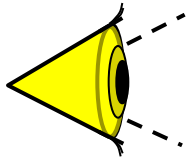


# Three Different Views



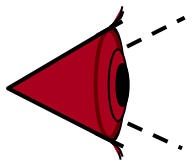
**User**

What do the user want to do with the system?



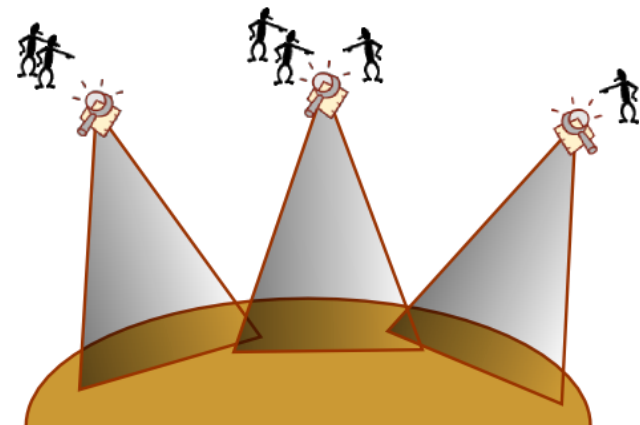
**System**

What should the system be capable of doing?



**Risks**

What problems may occur?





# How to Explore and Learn



# Test Planning

- What
- Why
- Who
- Where
- When
- How
- Dependencies
- Risks
- Prio
- Time





# Useful Tools for Agile Testing

<http://www.opensourcetesting.org/> <http://www.satisfice.com/tools.shtml>

- |                        |   |
|------------------------|---|
| <b>Allpairs</b>        | Test Case Generation Tool for combinatorial testing   |
| <b>PICT</b>            | Generation of combinatorial tests using orthogonal arrays <a href="http://www.pairwise.org/tools.asp">http://www.pairwise.org/tools.asp</a>                         |
| <b>Perlclip</b>        | Testing of text fields or documents with different kinds of stressful inputs  |
| <b>SpectorPro</b>      | Logging/recording of all activities on a PC<br><a href="http://www.spectorsoft.com/">http://www.spectorsoft.com/</a>  |
| <b>TestExplorer</b>    | Session based ET <a href="http://www.testexplorer.com">http://www.testexplorer.com</a>  |
| <b>Session Tester</b>  | An exploratory testing tool for managing and recording session-based testing <a href="http://sessiontester.openqa.org/">http://sessiontester.openqa.org/</a>        |
| <b>Resource Viewer</b> | Is intended for viewing of resources in executable files<br><a href="http://www.glocksoft.com/resource_viewer.htm">http://www.glocksoft.com/resource_viewer.htm</a> |
| <b>Rasta</b>           | Keyword Driven Test Automation<br><a href="http://rasta.rubyforge.org/index.html">http://rasta.rubyforge.org/index.html</a>   |
| List of Testing Tools: | <a href="http://www.aptest.com/resources.html">http://www.aptest.com/resources.html</a>   |

# Definition of Done – Functional Level

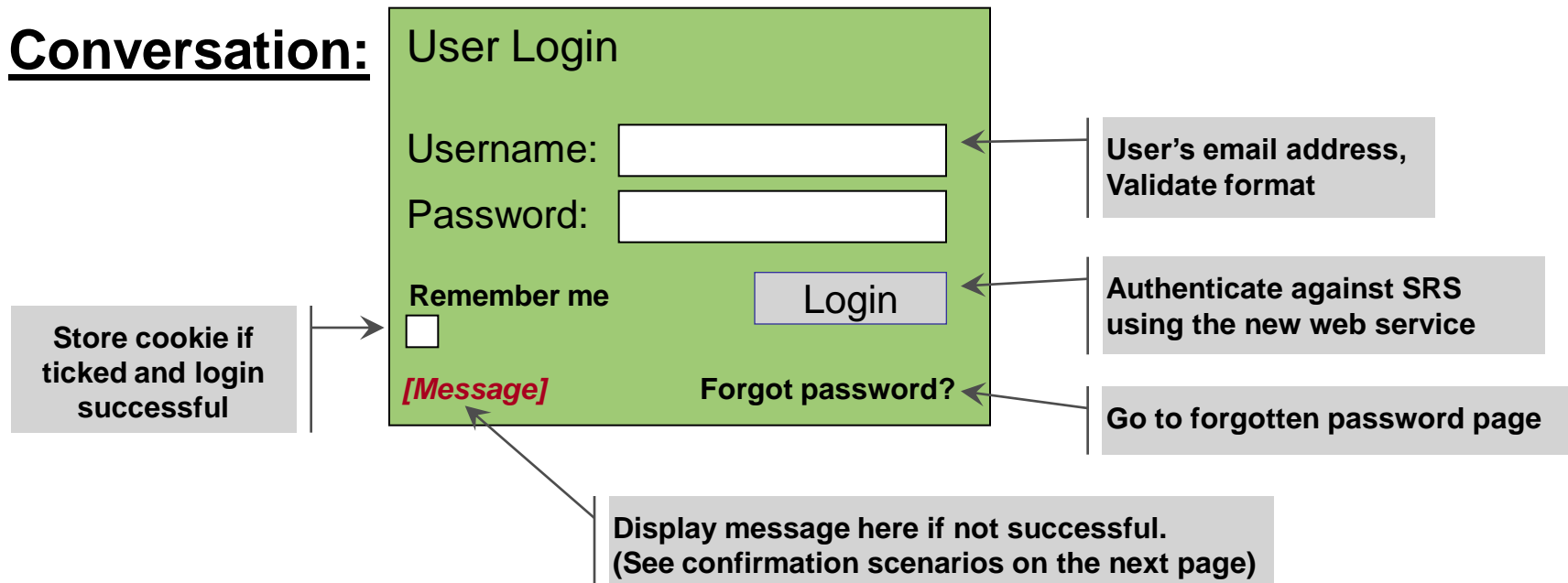
Test Aspect	Criteria for Done
Functional testing	<ul style="list-style-type: none"><li>➤ 100% requirements coverage.</li><li>➤ 100% coverage of the main flows in the operational scenarios.</li><li>➤ 100% of the highest risks covered.</li><li>➤ 100% of externally observable system states covered.</li><li>➤ 100% of externally observable failure modes covered.</li><li>➤ Operational manuals tested.</li><li>➤ All failures found are reported.</li><li>➤ Boundary Values, Equivalence Classes and Input partitioning testing made for all input data.</li><li>➤ All combinations of input and output parameters and values covered (pair-wise coverage).</li></ul>

# User Stories 1(2)

## Card:

As a **registered user**, I want to **log in**, so I can **access subscriber content**

## Conversation:



# User Stories 2(2)

## Confirmation:

**Success** Valid user logged in and referred to the home page

- a) Valid user name and password
- b) “Remember me” ticked – Store cookie/automatic login next time
- c) “Remember me” not ticked – Manual login next time
- d) Password forgotten and a correct one is sent via email

**Failure** Display message:

- a) “Email address in wrong format”
- b) “Unrecognized user name, please try again”
- c) “Incorrect password, please try again”
- d) “Service unavailable, please try again”
- e) Account has expired – refer to account renewal sales page

# Test Questions

- Which user/usage goals should be met?
- What user problems should be solved?
- Which user benefits should be achieved?
- Why does the orderer/customer want the system?
- Who are the customer(s) and the target user group?
- Which functions and characteristics are included?
- What are the most common and critical parts of the functionality from the users point of view?
- Are there any performance requirements included?
- What is an acceptable response time for the users?
- How tolerant should the system be to faulty input or user actions?



# Test Ideas

What do we need to find out about the system?  
-----

- 1 What happens if .....
- 2 What should happen when .....
- 3 Will the system be able to fulfil all its requirements?
- 4 What are the expectations and needs from the customer?
- 5 In what way may the system fail?
- 6 What problems were found in the previous release?
- 7 Are the requirements and the input specifications possible to understand and test (sufficient testability)?
- 8 Will the system be reliable and resist failure in all situations?
- 9 Will the system be safe in all configurations and situations it will be used?
- 10 How easy is it for real users to use the system?
- 11 How fast and responsive is the system?
- 12 Is it easy to install (and configure) onto its target platform?
- 13 How well does it work with external components and configurations?
- 14 How effectively can the system be tested (e.g. can log files be read)?





# Guidelines for Agile Testing

1. Test in pairs
2. Prepare test charters in advance
3. Use exploratory testing
4. Build tests incrementally
5. Use test design patterns
6. Perform keyword/data driven tests
7. End-to-end testing
8. Scenario based testing
9. Use automation for test data generation and execution
10. Frequent regression testing
11. Documentation testing
12. Log everything you do



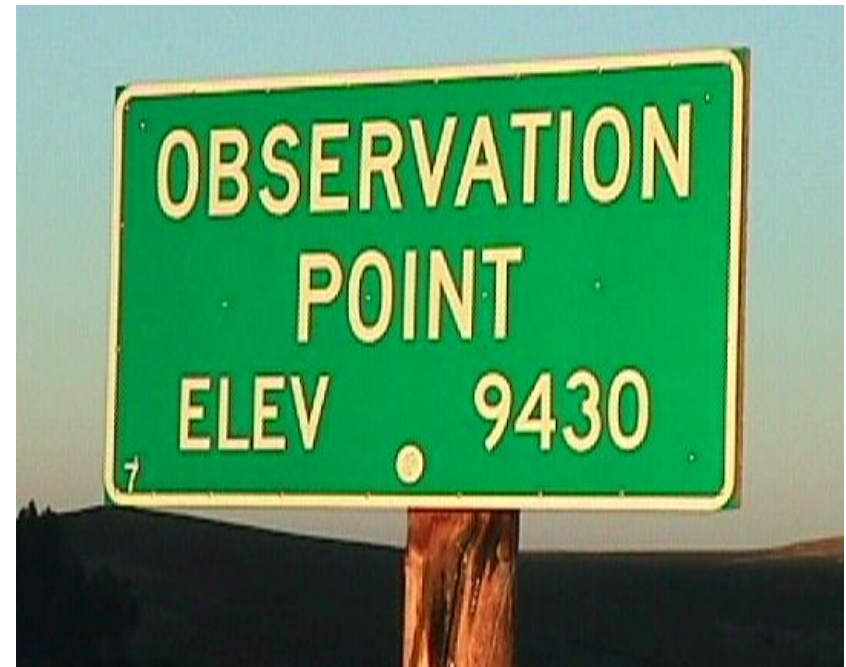
# Test Charter 1(3)

## Test Charter no # and <title>

Actor	< Type of user >
Purpose	< Describe the function, web page , test idea, ... to be tested >
Setup	< Preconditions i.e. concerning HW, content of data base(s),... >
Priority	< Importance of risk, function, web page, .... >
Reference(s)	< Requirement, risk, test ideas, ... >
Data	< Whatever is needed for the activities, files >
Activities	< A list of actions and test ideas >
Oracle notes	< How to evaluate the product for correct results >
Variations	< Alternative actions and evaluations >

# Exploratory Test Execution

1. Observe
2. State questions
3. Form hypothesis
4. Design the experiment
5. Test the hypothesis
6. Draw conclusions
7. State additional questions



# Test Reporting

## Test effort and perceived Quality Level – Including risks and test coverage

Test area	Initial Risk Level	Needed test effort	Current risk level	Current test effort	W 1	W 2	W 3	W 4	W 5	W 6	Q ass.	Comments
Area 1	Low	Low	Low	None				0	0	0		Feature(s) not yet delivered from design and integration. Definition of Done not fulfilled for functional testing. No testing possible.
Area 2	Medium	Medium	Low	High	1	1	1+	2	2	2+		On track, no faults.
Area 3	High	High	High	Blocked			1	1	1+	1+		Crashes, IR12345
Area 4	High	High	Medium	Pause	1	1	1+	1+	1+	2		IR1212 under investigation.
Area 5	Medium	Medium	Medium	High	1	2	2+	2+	3	3		Configuration problems.

# When to Stop Testing

## Coverage

All planned/required Test Charters/sessions and characteristics tests have been run and passed according to the current **risk areas/levels** and where faults have been found. The coverage objectives have been reached that were stated in the test goals (e.g. System Requirements, User Stories).



## Quality

### Testing should stop when:

- The probability of remaining faults has been reduced to a level that can be accepted by the customer.
- No open priority A Incident Reports.
- The systems' **risk level** is within acceptable limits (i.e. no critical **risks** remain unsolved).
- The Definition of Done for all testing activities have been fulfilled.
- The product values have been demonstrated and accepted (i.e. implicit and explicit quality attributes are satisfied).

## Time

When the agreed ship date has been reached.