TESTSTORMING™ – A COLLABORATIVE TECHNIQUE FOR RAPID TEST DESIGN

RANDALL W. RICE, CTFL, CTFL-AT, CTAL

WWW.RICECONSULTING.COM

© 2015, Rice Consulting Services, Inc.
THE NEED

• You are pressed for time to devise new tests for new or modified software.
• You have run short of ideas.
• You have limited perspective and knowledge.
• You have limited or inadequate specifications, documentation or other information.
THE PROBLEM WITH TEST DESIGN

• It is often:
  • Performed by individuals working alone
  • Based on flawed specifications
  • Very nuanced
    • Not a simple translation from specs to test case
  • Lacking in rigor
  • Outdated by the next release
BENEFITS OF COLLABORATION IN TEST DESIGN

• Shared problem solving ability
• Increased knowledge of the application to be tested
• Increased knowledge and experience in test design
• Higher levels of ownership in the test design activity
• Greater creativity in ways to create tests
• More objectivity in prioritizing tests
• Better communication about the project, the tests and the problems to be solved
The problem isn’t identifying enough test cases – they are everywhere you look.

The challenge is to identify the right test cases – the ones that add the most value to a test.
TESTSTORMING COMBINES THREE BASIC TECHNIQUES

- Brainstorming – to generate a lot of ideas quickly, then refine them.
- Test design – to identify ideas and conditions that truly exercise the software to be tested.
- MindMapping – to provide a way to document ideas and associate them logically.
COMMON MISCONCEPTION

• Don’t make the common mistake of thinking of brainstorming only as a group of people tossing about ideas.
  • While that is how many people perceive brainstorming, there is actually a creative process behind the technique.
OTHER USES

- TestStorming can also be applied to:
  - Risk identification and assessment
  - Root Cause Analysis
  - Test strategy development
USEFUL CONTEXTS

• While TestStorming can be used in just about any project context, agile teams seem to benefit most due to the need for collaboration and rapid test design.

• TestStorming is especially helpful in designing:
  • Mobile application tests
  • Acceptance tests (non-specification based)
  • Risk-based tests
  • Technical tests, such as security or performance
WHAT IS REQUIRED?

• A good facilitator
  • Ideally, trained for this role
• Knowledgeable and creative people who are not afraid to fail or look foolish
• The ability to collaborate with others
• A way to capture ideas
• A way to format ideas (such as a mind map)
MINDMAPPING

• You can use a whiteboard.
• But...there are some great open source tools.
  • Xmind - http://www.xmind.net/download/mac/
• The concept is to branch off a major topic, then develop sub-topics.
• The advantage of tools is that they produce output that can be preserved, updated and distributed.
THE FACILITATOR

• This is the key role.
• Like in reviews, the facilitator does not inject themselves into the session.
• Their role is to keep the session on track and to keep the flow of ideas coming.
THE RECORDER

• Someone needs to capture the ideas.

• The facilitator typically doesn’t have the ability to both facilitate the session and record ideas accurately.

• The ideas often flow quickly, so the recorder needs to be able to keep up with the flow.
GROUND RULES

• There are no “bad” ideas.
  • No ideas are rejected at first.
• Everyone contributes.
• No personal criticisms.
• The session is limited in time.
  • No more than 2 hours.
GETTING STARTED

• This is “a” process, not the only process.

• It has served me well, but you may want to adapt it for your own purposes and culture.
THE PROCESS

Step 1 – Decide on How to Capture Input
Step 2 – Define the Scope of the Session
Step 3 – Define the Focus of the Session
Step 4 – Assign a Leader
Step 5 – Invite the Participants
Step 6 – Conduct the Session
Step 7 – Filter, Refine and Combine Tests
Step 8 – Conclude the Session
Step 9 – Refine the Tests
AFTER THE SESSION

Step 10 – Perform the Tests
Step 11 – Update the Tests as Needed
STEP 1 – DECIDE ON HOW TO CAPTURE INPUT

- **Tools**
  - Freemind
  - Xmind
  - Word
  - Spreadsheets
    - METS
- **Whiteboard**
- **Flipcharts**
MINIMAL ESSENTIAL TEST STRATEGY (METS)

- Developed by Greg Paskal
  - www.gregpaskal.com
- A way to quickly define features and functions to be tested by levels of criticality.
- Test time can be estimated along with potential defect severity.
- Also available as an iPhone app.
<table>
<thead>
<tr>
<th>Category</th>
<th>Critical</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Display completes loading</td>
<td>Loading completes in a reasonable amount of time</td>
<td>Display consistent between device brands and versions</td>
<td>Display loading time reasonable under load</td>
</tr>
<tr>
<td>Graphics</td>
<td>Graphics, text and other elements seem to be in correct locations</td>
<td>Display contains all navigational elements</td>
<td>Resolution is correct</td>
<td></td>
</tr>
<tr>
<td>Graphics</td>
<td>Graphics loading completely</td>
<td>Graphics completed loading within consistent time frame</td>
<td>Consistently loading every time</td>
<td>Graphics load in low bandwidth conditions</td>
</tr>
<tr>
<td>Scaling, cropping or image quality correct</td>
<td>Link associated with graphic is correct</td>
<td>Grapics cached on device correctly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rollover graphics displaying correctly</td>
<td>Graphic rollover state providing correct transition illusion</td>
<td>Preloaders working correctly for quick image redraw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical text within graphic is legible</td>
<td>Correctly spelled text within graphic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forms</td>
<td>Menus are functional</td>
<td>Menu contains all desired options</td>
<td>Submitted form contains menu selection(s)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Menu items are spelled correctly</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buttons are functional</td>
<td>Submitted form contains button selection</td>
<td></td>
<td>Button text is spelled correctly</td>
<td></td>
</tr>
<tr>
<td>Checkboxes are functional</td>
<td>Selection of multiple checkboxes is possible</td>
<td>Submitted form contains checkbox selection(s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checkbox text is spelled correctly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text fields and boxes are functional</td>
<td>Text field and boxes have correctly spelled default text</td>
<td>Submitted form contains text field and text box information</td>
<td>Text fields allow enough room for a typical data entry</td>
<td></td>
</tr>
<tr>
<td>Search box is functional</td>
<td>Hitting Return/Enter submits form</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forms submitting correctly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form data being received</td>
<td>Data from submitted form validated and correct</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form validation working correctly</td>
<td>Invalid data input not allowed</td>
<td>Error messages provide helpful guidance</td>
<td>Error messages contain no typos</td>
<td></td>
</tr>
</tbody>
</table>
# Functional Test Grid

<table>
<thead>
<tr>
<th>Category</th>
<th>Critical</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts</td>
<td>Can user change account details such as shipping address.</td>
<td>Can user delete an account?</td>
<td>Does account time out require re-login after specified inactivity?</td>
<td>Is the user notified when why they must re-login?</td>
</tr>
<tr>
<td>Logging in</td>
<td>Can user request forgotten login information?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shopping Cart</td>
<td>Can an item be placed into the cart?</td>
<td>Can multiple items be put into shopping cart?</td>
<td>Do items consistently carry through purchase session?</td>
<td>Is tax being calculated correctly?</td>
</tr>
<tr>
<td></td>
<td>Can an item be removed from the cart?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can more than one item be removed at a time?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Catalog</td>
<td>Can user get online catalog?</td>
<td>Does online catalog reflect entire product line?</td>
<td>Can user sort catalog items buy price, size or color?</td>
<td>Does the catalog item reflect the product being sold?</td>
</tr>
<tr>
<td></td>
<td>Can user sort catalog items buy price, size or color?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactional</td>
<td>Can user back out of an order?</td>
<td>Can user save an order for later?</td>
<td>When user comes back to saved order is it accurate in items in cart and pricing?</td>
<td>Is the user notified when a transaction is complete?</td>
</tr>
<tr>
<td>Search</td>
<td>Can user search from home display?</td>
<td>Do the search results link correctly to the found content?</td>
<td>Is the found content logical to the original search?</td>
<td>Were search results sorted according to the business rules?</td>
</tr>
<tr>
<td>Store Locator</td>
<td>Is the locator functional from the home display?</td>
<td>Are the locator results correct according to business rules.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Downloading</td>
<td>Is the file accessible for downloading?</td>
<td>Does the download work correctly once on the customers computer?</td>
<td></td>
<td>Are the file size(s) and operating system requirements correct?</td>
</tr>
<tr>
<td>File Uploading</td>
<td>Can a customer upload a file?</td>
<td>Is the upload functionality providing an intact file on the business side once upload is complete?</td>
<td></td>
<td>Are the upload instructions clear and understandable?</td>
</tr>
<tr>
<td>Test</td>
<td>Category</td>
<td>Importance</td>
<td>Time Required</td>
<td>Potential Severity</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>----------</td>
<td>------------</td>
<td>---------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Display completes loading</td>
<td>Display</td>
<td>Critical</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Graphics, text and other elements are in correct locations</td>
<td>Display</td>
<td>Critical</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Display loading completes in a reasonable time.</td>
<td>Display</td>
<td>High</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Display has intuitive and logical flow.</td>
<td>Display</td>
<td>High</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Display loading is consistent between device branks and versions</td>
<td>Display</td>
<td>Medium</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Display has correct color contrast</td>
<td>Display</td>
<td>Medium</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Display loading time reasonable under load.</td>
<td>Display</td>
<td>Low</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>Graphics loading completely</td>
<td>Graphics</td>
<td>Critical</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>No scaling, cropping or image quality problems</td>
<td>Graphics</td>
<td>Critical</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Rollover graphics displaying correctly</td>
<td>Graphics</td>
<td>Critical</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
STEP 2 – DEFINE THE SCOPE OF THE SESSION

• This includes both the time to be spent in the session and the number of features to be addressed.
STEP 3 – DEFINE THE FOCUS OF THE SESSION

• Having a particular focus prevents the team from digressing into areas that may have limited value or importance in the test.

• For example, the focus of a TestStorming session might be the security aspects of a mobile device.

• In fact, the focus could be to design tests for the secure ordering process using a mobile device.
STEP 4 – ASSIGN A FACILITATOR

• Without a facilitator, the session can drift into non-productive discussions.

• Also, the facilitator may need to “prime the pump” of ideas by tossing out some initial suggestions.

• Sometimes, a session may stall.
  • This requires the facilitator to inject some remarks or ideas to get more ideas.

• The facilitator should be someone with subject matter knowledge and test knowledge.

• The facilitator should also know how to lead dynamic group discussions, mediate disputes and keep a session on track.
STEP 5 – INVITE THE PARTICIPANTS

- The success of a TestStorming session depends on the quality of the people in the room. You want people who are:
  - Creative thinkers
  - Critical thinkers
  - Knowledgeable about the context of how the application will be used
  - Knowledgeable about organizational goals for the application
  - Understanding of different types of users
  - Knowledgeable about test design (you want people who can think of strong tests)
  - Courteous to others in the session
  - Able to contribute ideas (otherwise, they are not really needed in the session)
STEP 5 – INVITE THE PARTICIPANTS (2)

• Another way to identify participants is by role, such as:
  • Testers
  • Developers
  • Users
STEP 6 – CONDUCT THE SESSION

• Several things are discussed at the outset, including:
  • The basis for test design, such as requirements, use cases, user stories, user experience, and the application itself
  • Known risks
  • Known issues
  • Past defect trends
  • The nature of the application and specific features to be tested
  • The context of how the application is used
  • The objective of the session
GATHERING INPUT

• Participants contribute their ideas for test design. This can be done in a variety of ways:
  • Answering context-free questions, such as:
    • What are the most important things to be tested?
    • What are the least important things to be tested?
    • Where is the risk in the features to be tested?
    • Which unusual events could occur?
    • Which events would be most commonly performed by the average user?
  • Each person writing their “top 5” tests on index cards, from which the leader will use as input to the list of conditions to test.
  • Going around the room in a round-robin format, making suggestions about conditions to test.
    • This allows people to build on other people’s ideas.
Our goal is to test the account set-up function.
And the process continues until the ideas stop flowing.
STEP 7 – FILTER, REFINE
AND COMBINE TESTS

• This is where the magic happens.
• After the team has contributed their ideas and it is apparent that the flow of input is slowing down, the leader asks the team to take a step back and look at the ideas put forth.
• From the test ideas suggested, the team organizes them into categories such as:
  • Functions
  • Risks
  • User personas
STEP 7 – FILTER, REFINE AND COMBINE TESTS (2)

• During the categorization process, some tests may be eliminated, others may be combined and enhanced.

• One of my favorite test design methods is what I call “toggling” conditions.
  • For example, if someone suggests testing a new customer placing an order, create a test condition for an existing customer placing an order.
STEP 8 – CONCLUDE THE SESSION

• At the end of the session, the facilitator should recap the tests suggested.
• For the sake of time, a summary may be best instead of reading each and every condition, test or idea.
• If a whiteboard or flipchart is used to capture ideas, take pictures of it to e-mail to the team and others.
• If a tool is used, then the chart can be saved and distributed in either an editable format or in image format.
STEP 9 – REFINE THE TESTS

• The purpose of the TestStorming session is to gather ideas for testing and to get those ideas to the point of use as strong, effective tests.

• The next steps occur after the session when test analysts take the great ideas and use them for robust test design.

• In some cases, the output from a TestStorming session might be a good set of test conditions.

• In other cases, the output may be good ideas of thing to test – much like you would have in a checklist or defect taxonomy.
STEP 9 – REFINE THE TESTS (2)

• Regardless of the detail obtained from the session, the ideas and/or conditions will then need to be formed into executable test cases with expected results.
  • Or, into the format used by your organization
  • In some situations, you may not know enough about the application to define expected results.
    • In that case, you will need to treat the TestStorming output as a place to start exploring the application.
STEP 10 – PERFORM THE TESTS

- Since the main objective is to collaborate on tests, the way we know whether they are effective tests is to actually perform them.
- The main determinant is not whether the tests pass or fail, but rather, do they exercise the application in a way that provides coverage, measures risk and goes beyond simple functional tests?
STEP 11 – UPDATE THE TESTS AS NEEDED

• After performing the tests, you will have a feel for whether or not to include the tests in future cycles.

• Perhaps you may decide to build upon certain tests.

• You may also see where some of the tests need to be modified to be more effective in finding failures.
A REALLY COOL VARIATION

• Real-time TestStorming using the application.

• Have one person “driving” the application and project it onto a screen at the front of the room.

• Starting with a designated person, each person suggests a test, building on the previous person’s test.

• The tests are performed in an exploratory way and any defects, issues or questions are noted.
SUMMARY

• There is power in collaboration and test design is an active that benefits greatly from group interaction.

• TestStorming is where test design meets brainstorming to define creative and effective tests.

• To be successful, you need a good facilitator, good participants and a way to capture the input from multiple people.

• Thankfully, there are some great free tools available.

• The rest is up to you to put your team’s creativity to work!
BIO - RANDALL W. RICE

• Over 34 years experience in building and testing information systems in a variety of industries and technical environments

• ISTQB Certified Tester – Foundation level (CTFL), Foundation Level Agile Tester (CTFL-AT) and Advanced Level (CTAL) Full

• Director, American Software Testing Qualification Board (ASTQB) since 2006.

• Chairperson, 1995 - 2000 QAI’ s annual software testing conference

• Co-author with William E. Perry, Surviving the Top Ten Challenges of Software Testing and Testing Dirty Systems

• Principal Consultant and Trainer, Rice Consulting Services, Inc.
CONTACT INFORMATION

Randall W. Rice, CTAL
Rice Consulting Services, Inc.
P.O. Box 892003
Oklahoma City, OK  73189
Ph: 405-691-8075
Fax: 405-691-1441
Web site: www.riceconsulting.com
e-mail: rrice@riceconsulting.com