Exploratory Testing 101

February 3, 2012
• Expectations
Outline

• Exploratory Testing
• Session-Based Test Management (SBTM)
• Thread-Based Test Management (TBTM)
• xBTM
Exploratory Testing
What is Exploratory Testing?
Exploratory Testing

• What is Exploratory Testing?
  ‣ A software testing approach
  ‣ Simultaneous learning, test design and test execution
  ‣ Introduced by Cem Kaner in 1983

• Why the name Exploratory Testing?
  ‣ To distinguish it from ad hoc testing
  ‣ To emphasise the exploration
Exploratory Testing

• What is Exploratory Testing *not*?
  - Ad hoc testing
  - Sloppy testing
  - Careless testing
  - Unstructured testing
  - Undocumented testing
  - Unskilled testing
Exploratory Testing

- Scripted
- SBTM
- TBTM
- Freestyle Exploratory Testing

Exploratory Testing
• Why Exploratory Testing?
Exploratory Testing

Why Exploratory Testing?
- Less preparations
- Do not need complete specifications or requirements
- Bugs found quickly
- Adaptable and flexible
- Creative, fun and stimulating
• Are there any disadvantages?
Exploratory Testing

- Are there any disadvantages?
  - Cannot review in advance
  - Reproducibility
Exploratory Testing

Bug

Test case
• Run test case 1st time
• 3 bugs found
• Run test case 2\textsuperscript{nd} time

![Diagram showing exploratory testing process with icons for bugs, found bugs, corrected/known bugs, and test cases.]

- Bug
- Found bug
- Corrected/known bug
- Test case
Exploratory Testing

• 0 bugs found

[Diagram showing the process of exploratory testing with symbols for bugs, found bugs, corrected/known bugs, and test cases.]
**Exploratory Testing**

- **Exploratory testing, 1st time**

![Grid with arrows and stars indicating test cases and bugs](grid_diagram.png)

- **Bug**
- **Test case**
Exploratory Testing

- 3 bugs found
### Exploratory Testing

- **Exploratory testing, 2\textsuperscript{nd} time**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Bug" /></td>
<td><img src="image" alt="Found bug" /></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Corrected/known bug" /></td>
<td></td>
<td><img src="image" alt="Test case" /></td>
</tr>
</tbody>
</table>

- Bug
- Found bug
- Corrected/known bug
- Test case
• 4 bugs found
• **What makes a good exploratory tester?**
  ‣ What kind of skills are needed?
• **Where do you start when there is no script?**
  ‣ San Francisco Depot: Mnemonic to test systematically
  ‣ Heuristic Test Strategy Model
San Francisco Depot (SFDPO)
- Mnemonic to test systematically
  - Structure: What the product is
  - Function: What the product does
  - Data: What the product processes
  - Platform: What the product depends on
  - Operations: How the product will be used
Heuristic Test Strategy Model (HTSM)
- Set of patterns to design test strategy
- What to think about when creating tests
Exploratory Testing

Quality Criteria Categories
- Capability
- Reliability
- Usability
- Security
- Scalability
- Performance
- Installability
- Compatibility
- Supportability
- Testability
- Maintainability
- Portability
- Localizability

Operational Criteria

Development Criteria

Product Elements
- Structure
- Functions
- Data
- Platform
- Operations
- Time

HTSM

Project Environment
- Customers
- Information
- Developer Relations
- Test Team
- Equipment & Tools
- Schedule
- Test Items
- Deliverables

Test Techniques
- Function Testing
- Domain Testing
- Stress Testing
- Flow Testing
- Scenario Testing
- Claims Testing
- User Testing
- Risk Testing
- Automatic Testing
Exploratory Testing

Do you know who your customers are?
- Whose opinions matter?
- Who benefits or suffers from the work you do?
- Do you have contact and communication with your customers?
- Maybe they can help you test?
- Maybe your customers have strong ideas about what tests you should create and run?
- Maybe they have conflicting expectations.
- You may have to help and resolve those.

Project Environment

Customers

Information
- Developer Relations
- Test Team
- Equipment & Tools
- Schedule
- Test Items
- Deliverables
Customers. Anyone who is a client of the test project.
- Do you know who your customers are? Whose opinions matter? Who benefits or suffers from the work you do?
- Do you have contact and communication with your customers? Maybe they can help you test.
- Maybe your customers have strong ideas about what tests you should create and run.
- Maybe they have conflicting expectations. You may have to help identify and resolve those.

Information. Information about the product or project that is needed for testing.
- Are there any engineering documents available? User manuals? Web-based materials?
- Does this product have a history? Old problems that were fixed or deferred? Pattern of customer complaints?
- Do you need to familiarize yourself with the product more, before you will know how to test it?
- Is your information current? How are you apprised of new or changing information?
- Is there any complex or challenging part of the product about which there seems strangely little information?

Developer Relations. How you get along with the programmers.
- Hubris: Does the development team seem overconfident about any aspect of the product?
- Defensiveness: Is there any part of the product the developers seem strangely opposed to having tested?
- Rapport: Have you developed a friendly working relationship with the programmers?
- Feedback loop: Can you communicate quickly, on demand, with the programmers?
- Feedback: What do the developers think of your test strategy?
Oracles and Heuristics
Exploratory Testing

- **How do you know if it is a bug?**
  - How do you know if the milk is still ok?
  - Does my traffic light have any “bugs”?
  - How do we recognise problems?
• What is an oracle?
  ‣ “In Classical Antiquity, an oracle was a person or agency considered to be a source of wise counsel or prophetic predictions or precognition of the future, inspired by the gods. As such it is a form of divination.” Wikipedia
• **What is an oracle?**
  ‣ Oracles are one kind of heuristic – rule of thumb
  ‣ An oracle is any principle or mechanism by which you recognize problems
• You are testing a web-based application
  ‣ You know absolutely nothing about it
  ‣ What oracle can you use?

James Bach
• Can we trust oracles?

  ‣ “The Pythia, when about to deliver, would chew leaves from Apollo's sacred **laurel** tree and would then sit on her holy tripod, seated in the innermost sanctum, over a crack on the rock from where **noxious volcanic fumes** emanated. **Dazed and disoriented**, she would then be ‘possessed by the voice of Apollo’.” *Wikipedia*
Can we trust oracles?
- Oracles are fallible
- Oracles are not the truth
- You can reach the wrong conclusion using the wrong oracle
- You can use the wrong conclusion using the right oracle too…
• HICCUPPS Heuristic
  ‣ Test oracles mnemonic

  ‣ History
  ‣ Image
  ‣ Comparable Product
  ‣ Claims
  ‣ User Expectation
  ‣ Product
  ‣ Purpose
  ‣ Statutes
Exploratory Testing

- **HICCUPPS Heuristic**
  - Is this defect inconsistent with the product **history**?
  - Is this defect inconsistent with the **image** our company (or project team) is attempting to portray?
  - Is this defect inconsistent with a **comparable** product?
  - Is this defect inconsistent with **claims** made about the product?
  - Is this defect inconsistent with **user** expectations about the product?
  - Does this defect show an inconsistency within the **product**?
  - Is this defect inconsistent with the **purpose** of the product?
  - Does the product comply with **statutes**?
Session-Based Test Management
• What is Session-Based Test Management (SBTM)?
  ‣ Tool-supported testing approach
  ‣ Introduced by Jonathan Bach and James Bach in 2000
  ‣ Structured and documented exploratory testing
Why Session-Based Test Management (SBTM)?

- Management control
- Metrics reporting
- Accountability
- Documentation
- Rapid defect discovery
- Flexibility
How does SBTM work?

- Work in sessions
- Time-box
- Uninterrupted
- Reviewable
- Feedback (debriefing)

- Test charter: Mission for the session
- Session report
Session-Based Test Management

• Sessions
  ‣ Short: 60 min
  ‣ Normal: 90 min
  ‣ Long: 120 min

  ‣ N.B.: Only plan about 60% of your time
Session-Based Test Management

- Point of attack
  - Reconnaissance
  - Analysis
  - Deep coverage
Session-Based Test Management

- Test Charter
  - Mission for the session
  - How to test
  - What kind of problems to look for
  - Often created in advance
  - Extent and level of detail flexible
Session-Based Test Management

- Test Charter – Example GMail

<table>
<thead>
<tr>
<th>TEST CHARTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyse Chat function and report potential problems</td>
</tr>
<tr>
<td>BROWSER: Firefox</td>
</tr>
<tr>
<td>OS: Windows 7</td>
</tr>
<tr>
<td>STRATEGY: Function Testing</td>
</tr>
</tbody>
</table>
Session-Based Test Management

- Session Report
  - Date, Time & Tester
  - Test charter
  - Area
  - Time break-down
    - Test design
    - Test execution
    - Test reporting
    - Other (e.g. interruptions and setup)
  - Bugs found
  - Issues found
  - Opportunity
  - Notes

![Time break-down chart]
• Session Report – Example GMail

TESTER: Christin  
DURATION: Normal  

TEST DESIGN AND EXECUTION: 60  
BUG INVESTIGATION AND REPORTING: 25  
SESSION SETUP: 5  

CHARTER VS. OPPORTUNITY: 100/0  

TEST NOTES: Focused on actual chat and adding contacts, did not look at Chat History.  

BUGS: #1726, Cannot set status to “Busy”  

ISSUES: Not sure what values status can have
Session-Based Test Management

- Debriefing - PROOF
  - Past
  - Results
  - Obstacles
  - Outlook
  - Feelings

- Frequency depends on team
Session-Based Test Management

• Metrics
  ‣ Bugs found
  ‣ Issues found
  ‣ On-charter vs opportunity
  ‣ Session vs non-session work
  ‣ Number of sessions over time
Session-Based Test Management

- Estimating Time & Reporting Status
  - Estimate number of charters, time for each charter is given

<table>
<thead>
<tr>
<th>Charter</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charter #1</td>
<td>John</td>
</tr>
<tr>
<td>Charter #2</td>
<td>Lisa</td>
</tr>
<tr>
<td>Charter #3</td>
<td>Anne</td>
</tr>
</tbody>
</table>

- Charters planned: 78
- Charters run: 12
- New charters added: 5

(78 + 5) - 12 = 71 charters left to run
71 * 90 min = 106 hours

1. **Blocked**
2. **Run**
3. **Waiting**

Charter assigned to: [Diagram]
Regression Testing – Example Gmail

- 90 min sessions

Key Area #1: Chat
- Charter #1: Add contact
- Charter #2: Change status
- Charter #3: Chat

Key Area #2: Create mail
- Charter #4: Write mail
- Charter #5: Attachments
- Charter #6: Cc and Bcc
Session-Based Test Management

- Regression Testing – Example GMail
  - Key Area as charter
  - 2 session, 3 hours

Charter #1: Chat
- Add Contact
- Change Status
- Chat

Charter #2: Create mail
- Write mail
- Attachments
- Cc and Bcc
Session-Based Test Management

- Regression Testing – Example Gmail

Key Area #1: Chat
- Charter #1: Add contact
- Charter #2: Change status
- Charter #3: Chat

Key Area #2: Create mail
- Charter #3: Write mail
- Charter #4: Attachments
- Charter #3: Cc and Bcc

15%
5%
80%
65%
25%
10%
Regression Testing – Example GMail
- Key Area as charter
- 2 session, 1 ½ hours

Charter #1: Chat
- Add Contact 15 min
- Change Status 70 min
- Chat 5 min

Charter #2: Create mail
- Write mail 60 min
- Attachments 20 min
- Cc and Bcc 10 min
**Tools**

- **Bach Scan Tool**
  - Scans session reports

- **Rapid Reporter**
  - Note testing application
  - [http://testing.gershon.info/reporter/](http://testing.gershon.info/reporter/)

- **Session reporter**
  - Management and recording tool
  - [http://sessiontester.openqa.org/](http://sessiontester.openqa.org/)

- **SBTExecute**
  - Calculate metrics
Visual Test Design
Visual Test Design

- Visual Test Design – A picture is worth a test case
  - Easy to understand
  - Easy to communicate
  - Fast to create and update
  - Reusable

Especially useful in exploratory testing
Visual Test Design

• Example
  ‣ Arrows are actions
  ‣ Boxes are verification points
  ‣ Make visual test design part of charter
## Visual Test Design

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Expected result</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter user name “user” in field <strong>User Name</strong></td>
<td>Text “user” displayed in field</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Enter password “password” in field <strong>Password</strong></td>
<td>Password characters are hidden</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Click button <strong>Log In</strong></td>
<td>User is logged in</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Wait 2 minutes</td>
<td>After 2 minutes, the user is automatically logged out</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Enter user name “user” in field <strong>User Name</strong></td>
<td>Text “user” displayed in field</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Enter password “p@ssword” in field <strong>Password</strong></td>
<td>Password characters are hidden</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Click button <strong>Log In</strong></td>
<td>Error message “Incorrect password” is displayed. User is not logged in.</td>
<td></td>
</tr>
</tbody>
</table>
Visual Test Design

Log in:
Valid user name
Valid password

Logged in

Wait for automatic log out

Logged out

Log in:
Valid user name
Invalid password

Error message
Not logged in
Visual Test Design

Log in:
Valid user name
Valid password

Logged in

Wait for automatic log out

Logged out

Log in:
Valid user name
Invalid password

Error message
Not logged in

How is the password displayed?

Did you press **Enter** or click the **Log In** button?

How long did it take?
• Model-Based Test Design
  ‣ Initial step towards automation
  ‣ yEd to make graphs
  ‣ GraphWalker to generate test sequences from yEd files
    http://graphwalker.org/ (open source)
  ‣ Must be able to loop
Visual Test Design

Log in:
Valid user name
Valid password

Logged in

Wait for automatic log out

Logged out

Log in:
Valid user name
Invalid password

Error message
Not logged in
Thread-Based Test Management
Thread-Based Test Management

What is Thread-Based Test Management (TBTM)?

- Activity-based approach
- Introduced by Jonathan Bach and James Bach in 2010
- Embraces the fact that activities change over time
- Cf. conversation threads
Tester1
Anyone read that article on exploratory testing?
Tester1
Anyone read that article on exploratory testing?

Tester2
Do you have the link?
Tester1
Anyone read that article on exploratory testing?

Tester2
Do you have the link?

Tester1
I'll look for it, hold on
Anyone read that article on exploratory testing?

Do you have the link?

I'll look for it, hold on

Are you going to the conference this year?
Tester1
Anyone read that article on exploratory testing?

Tester2
Do you have the link?

Tester1
I'll look for it, hold on

Tester3
Are you going to the conference this year?

Tester2
Yes, are you?
Thread-Based Test Management

Tester3
Yep, maybe we can rent a car and drive together?
Tester2
Sure, will you book one?

Tester3
Yep, maybe we can rent a car and drive together?
Tester1
Here it is http://link.ca

Tester2
Sure, will you book one?

Tester3
Yep, maybe we can rent a car and drive together?

Tester3
Ok, Friday – Sunday, right?
Tester1
Here it is http://link.ca

Tester2
Sure, will you book one?

Tester2
Exactly

Tester2
Thanks

Tester3
Ok, Friday – Sunday, right?

Tester3
Yep, maybe we can rent a car and drive together?
Tester1
Here it is http://link.ca

Tester2
Sure, will you book one?

Tester3
Yep, maybe we can rent a car and drive together?

Tester2
Exactly

Tester2
Thanks

Tester1
No problem, let me know what you thought

Tester3
Ok, Friday – Sunday, right?
Tester1
Anyone read that article on exploratory testing?

Tester2
Do you have the link?

Tester1
I'll look for it, hold on

New thread

Tester3
Are you going to the conference this year?

Tester2
Yes, are you?

Thread dropped
Thread-Based Test Management

Tester1: Here it is http://link.ca

Tester3: Ok, Friday – Sunday, right?

Tester2: Sure, will you book one?

Tester3: Yep, maybe we can rent a car and drive together?

Tester2: Exactly

Tester1: No problem, let me know what you thought
Thread-Based Test Management

• What is a thread?
  ‣ A test idea or test activity
  ‣ A thread can be interrupted and resumed
  ‣ Parallel threads

• What differs a thread from a session?
  ‣ A session charter is a *commitment* to complete a task
  ‣ A session is time-boxed
  ‣ TBTM is a generalisation of SBTM
Thread-Based Test Management

- Example: Test online book store
  - 5 different books in stock
  - Shop and pay now
  - Shop and save order for later

- Threads
  - Add book(s) to cart and check out
  - Add book(s) to cart and save cart
Thread-Based Test Management

Add book(s) to cart and check out

Add book(s) to cart and save cart
Thread-Based Test Management

Need credit card to test check out function

Drop thread 1, pick up thread 2
Thread-Based Test Management

New functionality: Ship books as gift
Must be tested now
All other threads dropped
Thread-Based Test Management

Testing threads 1 and 2 in parallel

All 5 books tested

Milestone

Failure: Testing blocked
Thread-Based Test Management

Decision to remove functionality

Testing cancelled
Thread-Based Test Management
Thread-Based Test Management

• Using SBTM
  ‣ Test charter: Add book and check out
    • Aborted – no credit card
  ‣ Test charter: Add book and save cart
    • Aborted – new functionality has priority
  ‣ Test charter: Ship book as gift
    • Aborted – decision to remove functionality
• Threads
  ‣ Following a thread corresponds to what SBTM calls opportunity
Thread-Based Test Management

- Please remember:
  - This is a contrived example to make a point
  - It is oversimplified
Thread-Based Test Management

- The essence of TBTM:
  - Activities change over time
  - “For me test is a verb. Testing is something that I do, not so much something that I create.”*
  - Focus on “doing” and not “getting done”

Thread-Based Test Management

- **Actions**
  - *Create* new threads
  - *Drop* threads - pause testing
  - *Pick up* dropped threads – resume testing
  - *Knot* threads – reach a milestone
  - *Untangle* threads – new knowledge
  - *Tie off* threads – stop testing
  - *Cut* threads – cancel testing
  - *Comb* threads – organise test activities
Thread-Based Test Management

• Where do threads come from?
  ‣ Heuristics (e.g. HTSM)
  ‣ Experience
  ‣ Previous projects
  ‣ Self-generating
Thread-Based Test Management

- **How do you do it?**
  - List ideas for test activities – each activity is a thread
  - Arrange threads in a *mind map*
  - Which thread is most important right now?
  - On which thread can we make the most progress right now?
  - Threads dropped and picked up
  - Parallel threads
• How do you keep track of threads?

- Large attachments
  - Many mails
  - Many contacts

- Many mails
  - Long strings
  - Special characters

- Stress testing

- Search mail

- GMail

- Chat
  - Change status
    - Add contact
    - History

- Compose mail
  - Cc
  - Bcc
  - Attachments
Thread-Based Test Management

- How do you keep track of threads?
  - Prioritise threads

Diagram:
- GMail
  - Large attachments
    - Many mails
    - Many contacts
  - Long strings
    - Search mail
  - Special characters
  - Stress testing
  - Search mail
  - Compose mail
  - Chat
    - Add contact
    - History
  - Change status
    - Cc
    - Bcc
    - Attachments
Thread-Based Test Management

- How do you keep track of threads?
  - Assign threads

![Diagram showing GMail with options for stress testing, search mail, compose mail, chat, and change status with contact history, Cc, and attachments.]
Thread-Based Test Management

- How do you keep track of threads?
  - Mark progress
Thread-Based Test Management

- How do you keep track of threads?
  - Make notes

Diagram:
- GMail
  - Stress testing
  - Search mail
  - Compose mail
  - Chat
    - Add contact
    - Change status
  - Large attachments
  - Many mails
    - Steve
    - Many contacts
  - Long strings
    - Special characters
  - Cc
  - Bcc
  - PDF
  - BIN
  - Text formatting

Note:
- Edit Notes: Cannot set status to "Busy"
- Reported bug, ID: 84768.
Thread-Based Test Management

• Why Thread-Based Test Management (TBTM)?
  ‣ Works even in chaotic and difficult environments
  ‣ A way to handle interruptions
  ‣ Can have very long or very short threads (not limited by time-boxing)
  ‣ Easy and quick to get started
  ‣ Traceability and documentation as needed
xBTM
• **What is xBTM?**
  ‣ Combines *Session-Based Test Management* (SBTM) and *Thread-Based Test Management* (TBTM)

• **Why the name xBTM?**
  ‣ Cf. xUnit (collection of code-driven testing frameworks)
  ‣ $x = S\ (Session)$ or $T\ (Thread)$
  ‣ Best of both worlds!
  ‣ Choose SBTM and/or TBTM depending on context
• **How is it done?**
  ‣ Start by making a mind map – test plan

  ‣ Use SBTM when possible
    • Group threads to create charters
    • Write session reports

  ‣ Use TBTM when SBTM is not an option
    • Test threads

  ‣ Update mind map continuously
• **Planning**
  ‣ Identify function areas
  ‣ Test techniques
  ‣ Use HTSM
  ‣ Test threads
  ‣ Arrange threads in mind map
• **Design**
  
  ‣ Create test charters were appropriate

---

TEST CHARTER: Copied recipients

Analyse Copy recipients function

BROWSER: Firefox

OS: Windows 7

STRATEGY: Function Testing
• **Execution & Reporting**
  ‣ Test – threads and charters
  ‣ Add threads and charters as needed
  ‣ Session reports
  ‣ Metrics
**Closure – Artifacts**

- “Final” version of mind map
- Session reports
- Metrics

**TESTER:** Christin  
**DATE:** Feb. 7, 2012  
**DURATION:** Normal

**TEST DESIGN AND EXECUTION:** 60  
**BUG INVESTIGATION AND REPORTING:** 25  
**SESSION SETUP:** 5

**CHARTER VS. OPPORTUNITY:** 100/0

**TEST NOTES:** …
Reading tips

• Recommended books
  ‣ Testing Computer Software
  ‣ Lessons Learned in Software Testing