Exploratory Testing an API

by Maaret Pyhäjärvi

@maaretp

http://maaretp.com
Testers don’t break your code, they break your illusions about the code.

-- adapted from James Bach
Exploratory testing is a systematic approach for discovering risks using rigorous analysis techniques coupled with testing heuristics.
What Testing gives Us

Testing as performance

Exploratory Testing

Guidance
Understanding
Models
Serendipity

Granularity
Regression
Feedback
Spec

Unit Testing

Testing as artifact creation

@maaretp
http://maaretp.com
## Testing is Everywhere!

<table>
<thead>
<tr>
<th></th>
<th>Before implementing</th>
<th>While implementing</th>
<th>Before production</th>
<th>While in production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing as</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>artifact creation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing as</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(exploration)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Ways to think of testing peak at different moments with regards to adding a capability*
An application programming interface (API) is a set of routines, protocols, and tools for building software and applications.
Product is my external imagination

I am my developer’s external imagination
Teaching Exploratory Testing in a Mob
Let's Test!
A Unit Testing Library
The Test Target
Mob Testing – Rules for the Exercise

No thinking as the driver

Yes, and…

Kindness, consideration and respect
One Idea Per Sticky Note ≠ Few Words ≠ Easy To Read

Capital Letters ≠ Lowercase Letters

Avoid Too Much Writing On One Single Note As It Takes A While For People To Read The Whole Thing

It Doesn’t Help To Write In Cursive Or In Pen

From Michael Sahota
Takeaways

Exploratory Testing an API
[UseReporter(typeof(DiffReporter))]

0 references | 0 changes | 0 authors, 0 changes

public void MyFirstTest()
{
    Approvals.Verify("This");
}

@maaretp

http://maaretp.com
[UseReporter(typeof(()))]

public void MyFirst()
{
    Approvals.Verify
}

- DiffReporter
- DirectoryAssert
- DivideByZeroException
- DllNotFoundException
- Does
- DotNet4Utilities
- Double
- DuplicateWaitObjectException
- Encoder
- EncoderExceptionFallback
- EncoderExceptionFallbackBuffer

twitter: @maaretp
http://maaretp.com
How to Explore

• Do something with it
• Find out what is the common thing to do with it
• Find out what you could do with it

• Reflect after anything and everything – make notes
  – What do we know from other connections?
  – What do we know from empirical evidence?
  – How do we turn it all into empirical evidence?
Structure
Function
Data
Platform
Operations
Time

*SFDPOT heuristics from James Bach / Michael Bolton, Rapid Software Testing*
Approvers do

- Formatting
- Sorting
- File Extensions
- Scrubbing (removing common inconsistencies)
- Serialization (saving to a file)
- Mocking
- Proxying
- Rendering
- Execution (e.g. retrieve the URL)
- Aggregating test cases
- File naming
- PRINCIPLE: "Every time you handle this type of object, you do these things to it."
Reporters do

- Waiting
- Scrubbing (removing common inconsistencies)
- Execution
- Launching
- Serialization
- Decompilation
- Chain of responsibility
- Creating Approved file
- Environmental awareness

I LEARNED ABOUT FUNCTIONS
I LEARNED ABOUT THE ENVIRONMENT

IDE
Solution structure
Diff Tools (LOTS)
Documentation

Delivery (nuget,...)
Approval Tests

Test Runner (nUnit, MSTest)

Language (C#,...)

Operating System

Version control
A Few Ideas of Exploratory Testing

• It’s not just about GUI and finalized features
  – You can explore an API
  – You can adapt to known limitations

• It’s not just for functional
  – You should do exploratory performance/security … testing

• It’s not without automation
  – Sometimes you need to do things humans can’t do! – exploratory test automation

• Repeating is seldom an issue
  – Vary the data, the environment, the story around your testing – exploratory regression testing
Patterns of Exploration for APIs

Lessons Distilled
Working with limited understanding: Focus!
Calls and Operations. Inputs and Outputs. Exceptions.
Target of your testing vs. the environment it depends on
Specific user with a specific purpose
Why would anyone use this?
Think lifecycle
Versioning and Deprecation
Google for Concepts
like “Conservative Overloading Strategy”
Fast-track your understanding: collaborate
Documentation Matters a Lot for APIs
Explore to create documentation
Some patterns become visible with repetition
Disposable test automation
Mob Programming Guidebook

BY Maaret Pyhäjärvi AND Llewellyn Falco

This is the pocket guide to help you get started Mob Programming successfully, and avoid the common pitfalls and misunderstandings.

Add to Wish List

#MobProgrammingGuidebook
Maaret Pyhäjärvi
Email: maaret@iki.fi
Twitter: @maaretp
Web: maaretp.com
Blog: visible-quality.blogspot.fi

(please connect with me through Twitter or LinkedIn)