

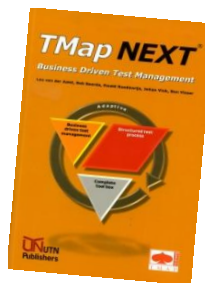
A large white silhouette of a person in a suit, standing with their back to the viewer and looking towards the right. The silhouette is filled with a pattern of small, white, stylized floral or leaf-like shapes. The person is holding a laptop bag or folder under their left arm.

The Many Flavors of Exploratory Testing

Gitte Ottosen
Managing Consultant
gitte.ottosen@sogeti.dk

Sogeti Profile

- Sogeti is a leading provider of structured testing solutions
- Part of the Sogeti Group, which brings together more than 20,000 professionals in 15 countries and is present in over 100 locations in Europe, USA and India
- Creators of the globally recognized methodologies TMap NEXT[®] and TPI NEXT[®]
- Well established cooperation with HP, IBM and Microsoft



Agenda

- Background
- Introducing different “flavors” of exploratory test
- Challenges – and what to do about it
- What we ended up with – our “flavors”
- What’s next



Background

Company Profile

- CMMI5 level software company
- Following several standards (AQAP, ISO9001 etc.)
- Developing mission critical software:
 - Defense command and control system
 - Defense communication system
 - Electronic patient journal
 - National security and intelligence
 - System integration, public sector

SYSTEMATIC

Why Moving Towards Exploratory Test?

- EUROSTAR 2004
 - Introduction to exploratory test - James Bach
 - Introduction to attacks – James Whittaker
- Soon after the story about “bug hunt” – Klaus Olsen
- Status on testing “back home”
 - The beginning of a structured approach to testing
 - Classical scripted test
 - Primarily focus on acceptance testing
 - To many defects ended in production

Definition of Exploratory Test

”An interactive process of simultaneous learning, test design, and test execution.”

Exploratory testing is not against the idea of scripting. In some contexts, you will achieve your testing mission better through a more scripted approach; in other contexts, your mission will benefit more from the ability to create and improve tests as you execute them. I find that most situations benefit from a mix of scripted and exploratory approaches.

James Bach
Exploratory Testing Explained



Exploratory Test - A Couple of Terms

- Charter:
 - States the mission and perhaps some of the tactics to be used.
 - Sometimes charters are written down.
- Timebox
 - Defines the period of time in which the tester design, learn and execute test
- Notes
 - Not mandatory for freestyle, but is recommended. Should be used when executing session based exploratory.
 - The only written output of a exploratory test except for bug reports.

Exploratory Flavors

- We tried a number of different flavors:
 - Freestyle exploratory
 - Session based exploratory
 - Testing tours
 - Bug hunts
 - General functionality and stability test procedure (Microsoft)



Sources:

A number of articles from satisfice.com: by James and Jon Bach

Tutorial on Session based exploratory: by Jon Bach

Book "Exploratory Software Testing": by James Whittaker

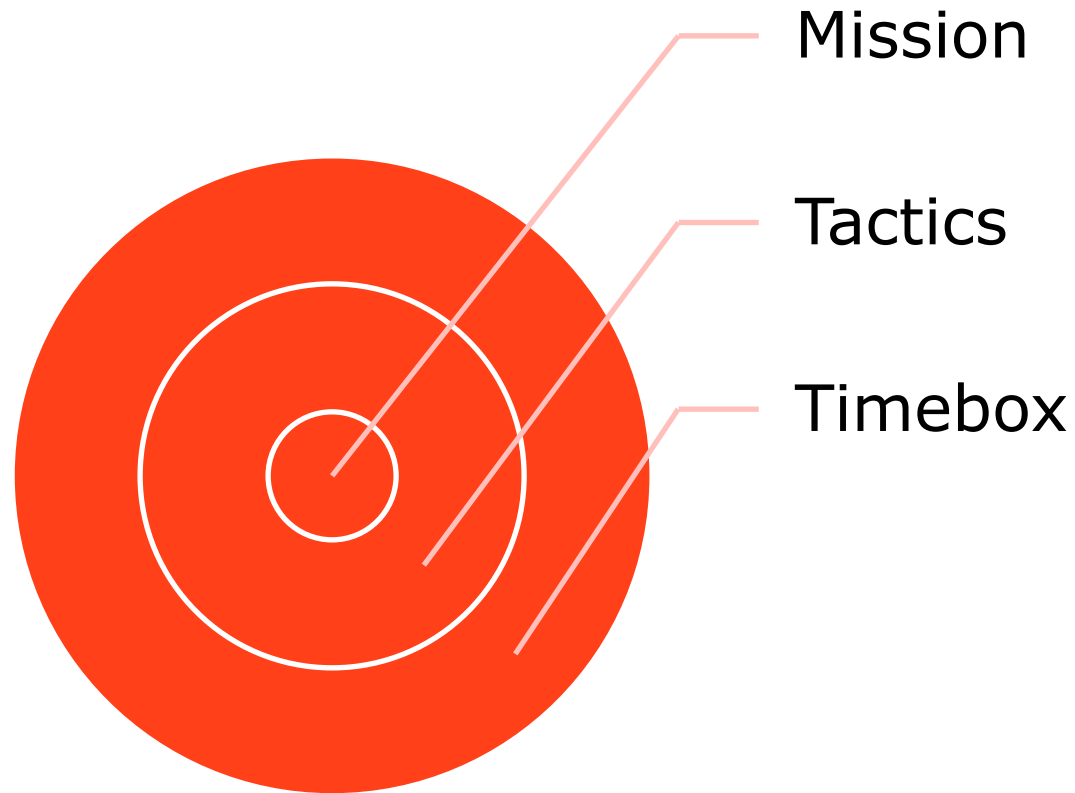
Tutorial on testing tours: by James Whittaker

Bug hunt, presentation: by Klaus Olesen



Freestyle Exploratory

Freestyle Exploratory



Freestyle Exploratory - Example

- Test object creation in SitaWare Headquarter
- Test medicine prescription in the medicine module, ensure all different prescription types are covered
- Identify primary operational workflows in SitaWare Frontline and run through each one.

Freestyle Exploratory - Comments

- What have been tested?, Missing tangible evidence
 - Only output is bug reports
 - No notes
 - No coverage
- A great tool when you haven't got anything else
- Be careful with too large time boxes
- Be careful with too complex/big charters
- Hard to convince management
- Often misunderstood
- Cannot stand on its own in a structured process

A large, solid red silhouette of a person in a suit, seen from the back and slightly to the side. The person is holding a mobile phone to their ear with their right hand. The silhouette is positioned on the left side of the slide.

Session Based Exploratory

Sessionbased Exploratory test

- A "session"
 - A mission
 - Time box
- The Report
 - "Coverage"
- Debriefing
- Tool supported



Session Based Exploratory - Report and Debriefing

- Session charter (includes a mission statement, and areas to be tested)
- Tester name(s)
- Date and time started
- Task breakdown (the TBS metrics)
- Data files
- Test notes
- Issues
- Bugs

Debriefing

- **P**ast
- **R**esults
- **O**bstacles
- **O**utlook
- **F**eelings

Session Based Exploratory - Comments

- Be careful with too large time boxes
- Be careful with too complex/big charters
- Challenge to get testers to make good notes in a defined structure
- No success in implementing note/coverage tool
- Who to make the debriefing? Someone who knows about the SUT in detail or the test manager?
- Makes it easier to keep track on what is tested



Testing Tours

Testing Tours

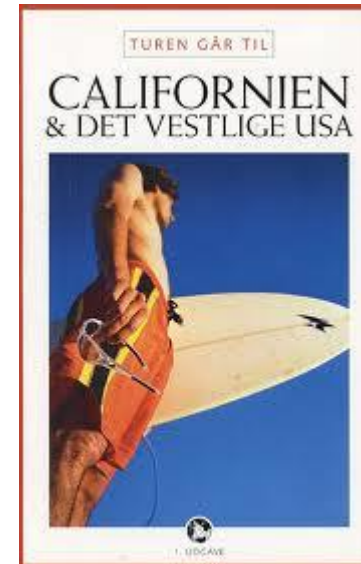
Exploratory testing without good guidance is like wandering around a city looking for cool tourist attractions. It helps to have a guide and to understand something about your destination. *James Whittaker*

- Theme/behavior based rather than feature focused
- Decompose the test based on intent rather than application structure.
- The tourist and the tours



Testing Tours - Example

- The landmark tour
- The FedEx tour
- The museum tour
- The supermodel tour



Testing Tours - Comments

- Hard to introduce to testers
- Hard to keep track on coverage
- Hard to communicate to management
- When to apply?
- A good way to get another perspective than workflows or features
- A good input for bug hunts



The Bug Hunt

Bug Hunts

- A competition:
 - Pair test
 - Free style Exploratory
 - Based on a charter
 - Time box 1 hr

Severity	Points
1	25
2	16
3	9
4	4
5	2

Team No.				
Run #				
Page #				
Build / Baseline				



PACCS No.	Description	Category	Points	Judge

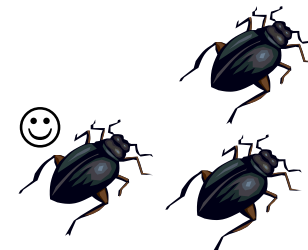
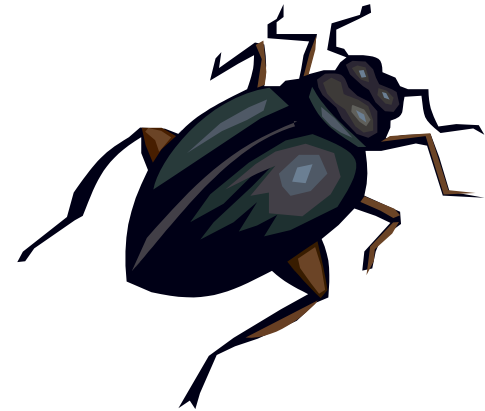
Prices and Experiences

- To the team that finds the biggest (most serious) bug (determined by the “judges”)
- To the team that finds the most bugs (sum of points)

- We spent 112 hours on the first event
 - 6 teams of 2 people with 4 runs of 1 hour (48 hours)
 - Rest of the time spent on; Preparations and defect reporting
 - Facilitators
 - Judges (architects) – they were *very* busy
- Found 76 defects
 - Worth a total of 573 points
- Equivalent to 0.68 defects/hour
 - Better than our normal average finding rate
- Best bug was a security loop-hole

Bughunt - Comments

- A Bug-Hunt Is:
 - NOT a replacement for structured/scripted tests
 - Cannot cope with hard-to-test features and long scenarios
 - Coverage
 - Number of testers needed
 - But great as a supplement
 - Finds new errors
 - Get different profiles to pair test
 - ... and a fun team building exercise 😊



A large, solid red silhouette of a person in a suit, seen from the back and slightly to the side. The person is holding a mobile phone to their ear with their right hand. The silhouette is positioned on the left side of the slide, partially overlapping the text.

General Functionality and Stability Test Procedure - Microsoft

General Functionality and Stability Test Procedure - Microsoft



Purpose of
the product

Identify
functions

Areas of
potential
instability

Test each
function

Consistency
verification
test.

The Consistency Verification Test

- Regression test
- The test must be specific enough that it can be repeated by different testers, and all testers will get the same results.
- Cover each of the most important primary functions with a simple test.
- Specify and archive any data needed for the test.
- Specify some complex data for use in the test.
- Use specific file names and path names.
- Make the test as short and simple as you reasonably can, while meeting these requirements

General functionality and stability test procedure - Comments

- The best of Exploratory Test but also gives you:
 - More structured documented result
 - Regression test
 - Visible coverage
 - Enough "proof" to satisfy certifications (e.g. AQAP)
- Take care not to make too big charters (time box)
- The test notes are still crucial
- Missing debriefing



General Challenges

And what we did to mitigate

Challenges

- Management understanding
- Tester understanding
- The notes
- Sourcing
- The regression test
- How to get visible coverage
- Mapping to requirements



How we Solved it?

- For classic/traditional testers:
 - An introduction/workshop
 - Pair testing (sourcing tester + “local” tester)
- For management:
 - An introduction
 - Prove in battle
- For a “structured house”: a way to keep track enforcing:
 - Notes
 - Debriefing
 - Requirement traceability
 - “Documentation” of regression test

Test Plan : TRP014 Test Execution Plan 5.3 Hotfix 2 with german features [Build : build01]

Settings

Test Plan: TRP014 Test Execution Plan 5.3 Hotfix 2 with german fea
Build to execute: build01
Update tree after every operation
Export Test Plan

Filters

Test Case ID: HQ5-
Test Case Title:
Test Suite:
Priority: [Any]
Execution type: [Any]
Assigned to: include unassigned Test Cases
Result on: [Any]
Build chosen for execution

Apply Reset Filters Advanced Filters

- SitaWare HQ5 / TRP014 Test Execution Plan 5.3 Hotfix 2 with germ
 - TSP064 ET Smoketest for Sitaware Headquarters (1)(1, 0, 0, 0)
 - TSP122 Auditing (1)(0, 1, 0, 0)
 - HQ5-506:Auditing
 - TSP123 Active directory support (1)(0, 1, 0, 0)
 - TSP124 Communication HQ one way gateway support (1)(0, 1, 0, 0)

Date	Build	Tested by	Status	Test Case Version	Delete	Run mode
27/07/2011 13:19:40	build01	admin	Passed	1		

Notes

Defect ID:
Requirements Verified:
Execution Time:
Attached files :

Upload new file



Summary

Preconditions

Execution type : Manual

Defect ID:

Requirements Verified: Yes No



Execution Time:

Notes / Description

Result

- Not Run
- Passed
- Failed
- Blocked


Save execution

Save and move to next

Important Notice: Once a Result is updated from 'Not Run' to another value, you cannot set it back to 'Not Run'. You can still set the Result to any other value.

The "Test Case"

Functionality			
Purpose of functionality			
Function outline			
Notes			
Observations			
Product Name		Product Version incl. Release candidate	
Test setup configuration	Products	Hardware	Software
Date		Time spent	
Tester initials			
Signature			
Conclusion	<input type="checkbox"/> Accepted <u>without comments</u>	<input type="checkbox"/> Not <u>accepted major problems</u>	
	<input type="checkbox"/> Accepted <u>with comments</u>	<input type="checkbox"/> Not <u>accepted not conducted</u>	
State of Quality			

A large, solid red silhouette of a man in a suit, seen from the back and slightly to the side. He is holding a mobile phone to his ear with his right hand. The silhouette is positioned on the left side of the slide.

What we Ended up With

Our "clones"

Our Implementation

- Microsoft approach with session based debriefing
- Charters based on test design for story test
- Freestyle charters to fill in the gaps
- Bug hunts when appropriate
- Heuristics - checklists



Is it Exploratory Testing?

I think so - just in our context

if we cannot call it that then lets call it:

Examining

Inquiry

Expedition

Investigation

Inquisition

Reconnaissance

Questioning

Scrutiny

Looking Ahead - Where to go?

- End to end test as exploratory charters
- More use of heuristics and checklists
 - E.g. EDI message types
- Use the fundamentals of exploratory testing whenever possible
- Learn and try out.... 😊

Questions?

