

Dataforeningen, Bergen, 2. Desember 2002

# Managing Exploratory Testing

Parts of this presentation have been excerpted from:  
Rapid Software Testing, by James Bach, [james@satisfice.com](mailto:james@satisfice.com)  
Black Box Software Testing, by Cem Kaner, [kaner@kaner.com](mailto:kaner@kaner.com)  
High Speed WEB Testing, by Jarle Våga, [jarle.vaga@xit.no](mailto:jarle.vaga@xit.no)

Ståle Amland, Amland Consulting,  
Hulda Garborgsv. 2, 4020 STAVANGER, NORWAY  
Phone: +47 905 28 930, Fax: +4751585524  
e-mail: [stale@amland.no](mailto:stale@amland.no) WEB: [www.amland.no](http://www.amland.no)

## *Software Testing:*

*“There is artistry. There is craft. There is science. It is all three – as with all technologies.*

*If you understand the science and have mastered the craft, then it is meaningful to investigate, espouse, and perfect the art.”*

*Boris Beizer*

## *Software Testing:*

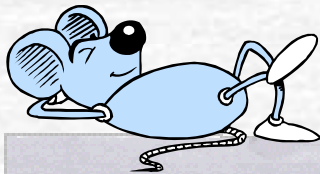
*“Testing is the process of executing a program with the intent of finding errors.”*

*“Testing is an extremely creative and intellectually challenging task.”*

*Glenford Myers in  
“The Art of Software Testing”*



# Presentation Outline



What is Exploratory Testing?

How to do Exploratory Testing?

How to Manage Exploratory Testing?

High Speed WEB Testing – Case Study



# Different testing approaches

- Skeptical approaches
- Analytical Approaches
- Information-driven approaches
- Time-honored but less effective approaches
- Experiential and intuitive approaches:
  - “Let’s think blue-sky, speculate and follow our intuition.”
  - “We have good hunches about where the bugs are lurking.”
  - “Let’s jump in and explore the system’s behavior hands-on, so we can decide how to test it.”
  - “Let’s find the important bugs fast, and worry about the test paperwork later.”
- Exploratory Approach ...

*Ross Collard (2002)*



# What is Exploratory Testing?

***"Exploratory testing involves simultaneously learning, planning, running tests, and reporting / troubleshooting results."***

*Dr. Cem Kaner (2001)*

***"Exploratory testing is an interactive process of concurrent product exploration, test design and test execution."***

***"To the extent that the next test we do is influenced by the result of the last test we did, we are doing exploratory testing."***

*James Bach, Satisfice (2001)*

## What is Exploratory Testing - continued

- - Let's structure and document our creative testing so we know where we have been
- "Thinking-While-Testing"!



## *Said about eXtreme Programming*

*Agile software development is not conventional software development done more quickly or done on tippie-toe. Agile software development is software done differently.*

*Ron Jeffries, (e-mail on agile-testing list, April 24, 2002)*

- *proven (no single technique is new)*
- *application oriented*
- *planned and disciplined*
- *controllable and reliable*
- *risk minimizing*
- *Two sides of extreme programming:*
  - *for the developer: freedom, flexibility, fun*
  - *for the manager: controllability, reliability, high quality*

*Martin Lippert (University of Hamburg), ICSTEST 2002*



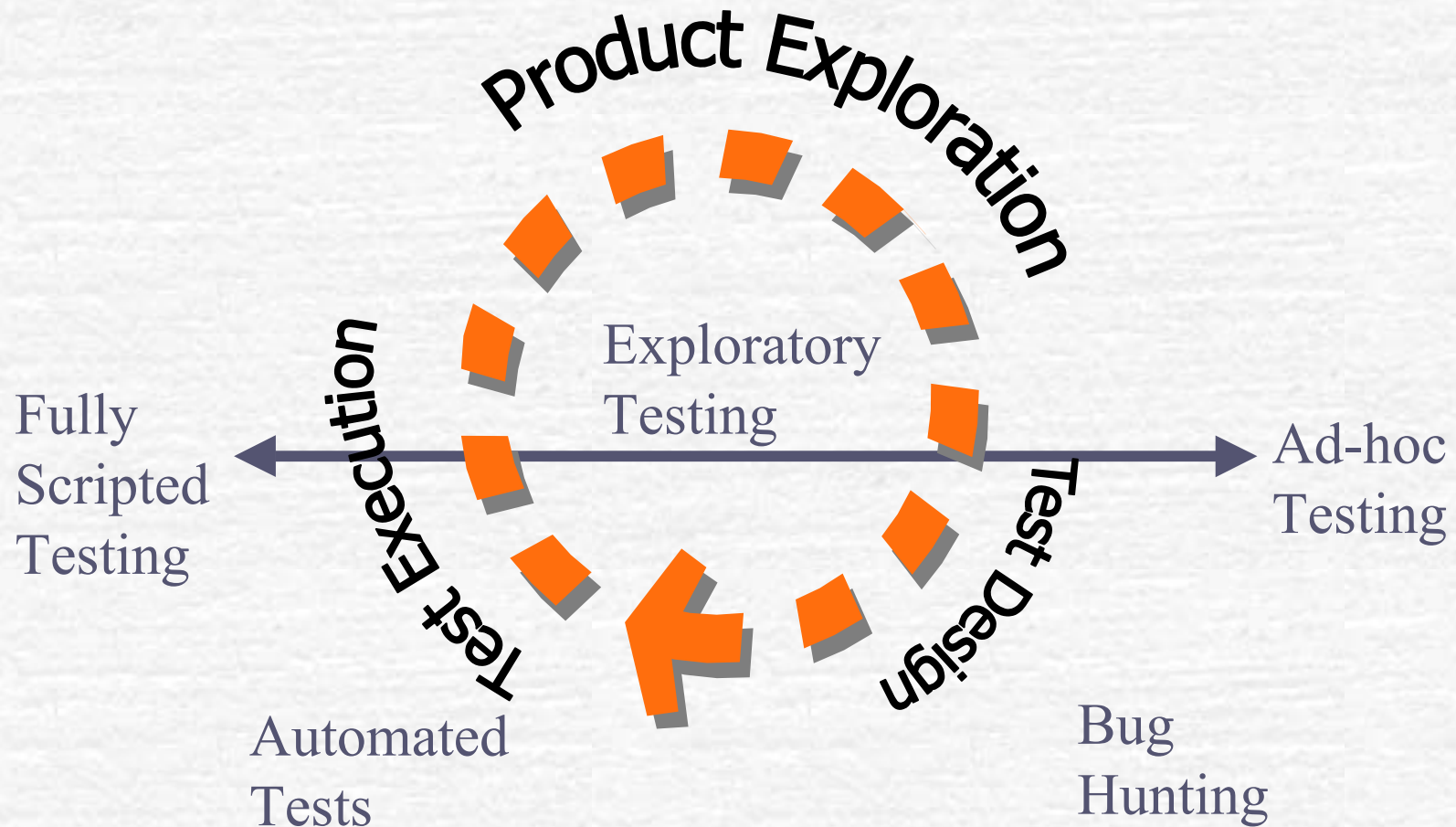
# *The eXtreme Programming and Exploratory Testing Analogy:*

*Agile software testing is not conventional (scripted) software testing done more quickly or done on tippie-toe.*

## *Exploratory Testing:*

- *proven (no single technique is new)*
- *application oriented*
- *planned and disciplined*
- *controllable and reliable*
- *risk minimizing*
- *Two sides of Exploratory Testing:*
  - *for the tester: freedom, flexibility, fun*
  - *for the manager: controllability, reliability, high quality*

# ET vs. Scripted Testing

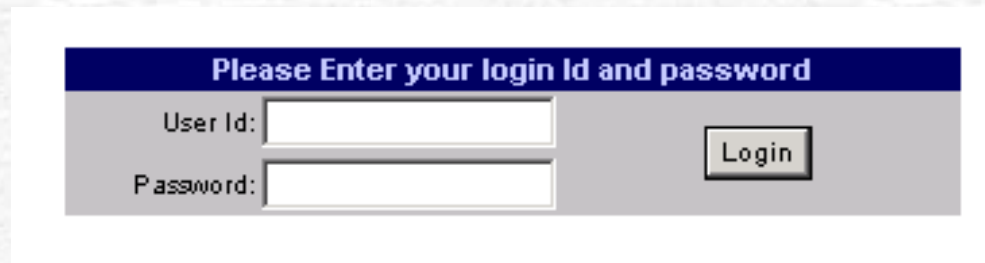


*Jarle Våga (2002)*



# What is Scripted Testing?

- Small (but realistic) example:



Please Enter your login Id and password

User Id:

Password:

Login

- How to script and test this login?  
(Functional tests only – not security!)



## When to use Exploratory Testing? (1)

- There is little or no specifications and / or requirements
- You have little or no domain knowledge
- You don't have time to specify, script and test

**Uncertainty and Time Pressure!**



## When to use Exploratory Testing? (2)

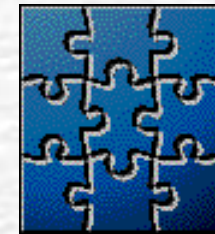
- A common goal of exploration is to *probe* for weak *areas* of the program.
- Test team's resource consumption per week:
  - 25% of the group's time developing new tests
  - 50% executing old tests (including bug regression)
  - 25% on exploratory testing

*Cem Kaner (2001a)*



# Testing is done in Context

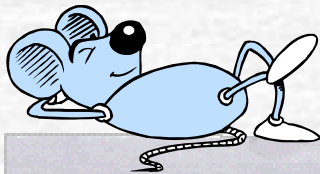
- The value of any practice depends on its context.
- There are good practices in context, but there are no best practices.
- People, working together, are the most important part of any project's context.
- Projects unfold over time in ways that are often not predictable.
- The product is a solution. If the problem isn't solved, the product doesn't work.
- Good software testing is a challenging intellectual process.
- Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.



<http://www.context-driven-testing.com/>



# Presentation Outline



What is Exploratory Testing?

How to do Exploratory Testing?

How to Manage Exploratory Testing?

High Speed WEB Testing – Case Study



# ET Planning

- *"We plan as much as we can (based on the knowledge available), when we can (based on the time available), but not before."*
- *"When we apply Exploratory Planning to Testing, we create Exploratory Testing."*

*Lee Copeland, Exploratory Planning, 2001*



# Exploratory Testing Tasks

	<b>Explore</b>	<b>Design Tests</b>	<b>Execute Tests</b>
<b>Product</b> <i>(coverage)</i>	Discover the elements of the product.	Decide which elements to test.	Observe product behavior.
<b>Quality</b> <i>(evaluation)</i>	Discover how the product should work.	Speculate about possible quality problems.	Evaluate behavior against expectations.
<b>Techniques</b>	Discover test design techniques that can be used.	Select & apply test design techniques.	Configure & operate the product.

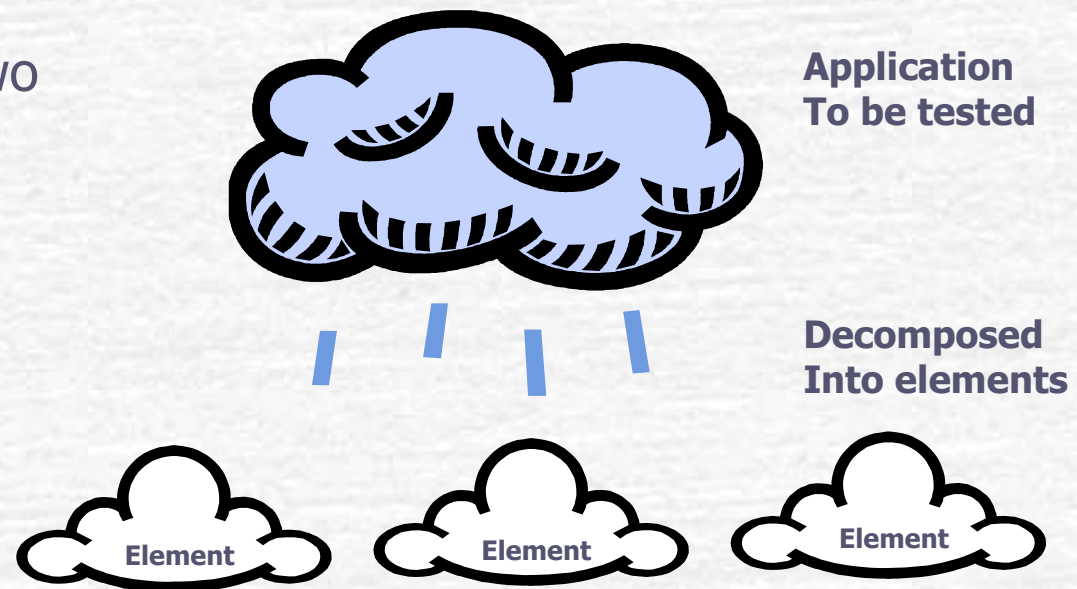
<b>Testing notes</b>	<b>Tests</b>	<b>Problems Found</b>
----------------------	--------------	-----------------------

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



# Preparation

- Decompose the Product into elements
  - Areas / Groups of functions or features you expect to be able to test within a day or two
- Identify the Quality Criteria
  - Ref. Satisfice Testing Model
- Check your toolbox
  - Test techniques

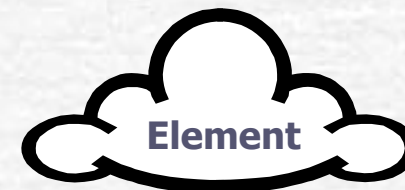


**Each element is expected to be tested within a day or two.**



## ET Planning – Architecting Charters

- Pick an element to test
- Decompose further – into smaller elements
  - Each new element should be tested within an hour or two
  - Prepare a charter for each element



Element  
To be tested

Decomposed  
Into smaller  
elements /Charters



Each Charter is expected to be tested  
within an hour or two.



# Doing Exploratory Testing

- Keep your mission clearly in mind.
- Keep notes that help you report what you did, why you did it, and support your assessment of product quality.
- Keep track of questions and issues raised in your exploration (logging and debriefing)
- To supercharge your testing, pair up with another tester and test the same thing on the same computer at the same time.



*From Rapid Software Testing, copyright © 1996-2002 James Bach*



## Charter: A clear mission for the session

- A charter may suggest what should be tested, how it should be tested, and what problems to look for.
- A charter is not meant to be a detailed plan.
- General charters may be necessary at first:
  - “Analyze the Insert Picture function”
- Specific charters provide better focus, but take more effort to design:
  - “Test clip art insertion. Focus on stress and flow techniques, and make sure to insert into a variety of documents. We’re concerned about resource leaks or anything else that might degrade performance over time.”

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



## Charter – summary...

- “Architecting the Charters” i.e. Test Planning
- Brief information / guidelines on:
  - Mission: **Why** do we test this?
  - **What** should be tested?
  - **How** to test (approach)?
  - **What problems** to look for?
- Might include guidelines on:
  - **Tools** to use
  - Specific Test **Techniques** or **tactics** to use
  - What **risks** are involved
  - **Documents** to examine
  - Desired **output** from the testing



## Charter Example

# Charter: Search Engine (Company internal)

- What:
  - Search Engine to look up other sources of information in the company (list of sample information sources: A, B, C etc.). Standard and Advanced search must be tested.
- Why:
  - To test the search feature with single information sources and multiple sources, to see that the retrieved information is presented consistently and according to standard, and that the retrieved information is correct.
- How:
  - Search from the WEB portal as well as continue searching in the result list (advanced search – refining the search)
- Expected problems:
  - Some information not found.
  - Not possible to navigate to information found (jumping between information sources)
  - Information found not presented consistently independent of sources
- References:
  - Requirement specification section x.11





## Charter Example

# Charter: Publishing Information on the WEB

- What:
  - Testing the feature "Publishing" in the publishing tool "A"
- Why:
  - To test the the link / interface between the WEB portal and the documentation handler / publishing tool, so new articles published with "A" can be viewed from the WEB. The published information should be sorted in different groups on the WEB: "Top Stories", "Latest News" and "News Archive"
- How:
  - The publishing tool "A" will only be available to a limited number of users. These users will get some product training and will then be part of the test team.
- Expected problems:
  - Non in particular.
  - The publishing tool "A" is not perceived as user friendly. This might cause some unexpected problems.
- References:
  - Requirement specification section x.13.2
  - Publishing tool "A" user manual





# Taking Notes

- Test Coverage Outline/Matrix
- Evaluation Notes
- Risk/Strategy List
- Test Execution Log
- Issues, Questions & Anomalies
  - It would be easier to test if you changed/added...
  - How does ... work?
  - Is this important to test? How should I test it?
  - I saw something strange...

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



# ET Execution - Sessions

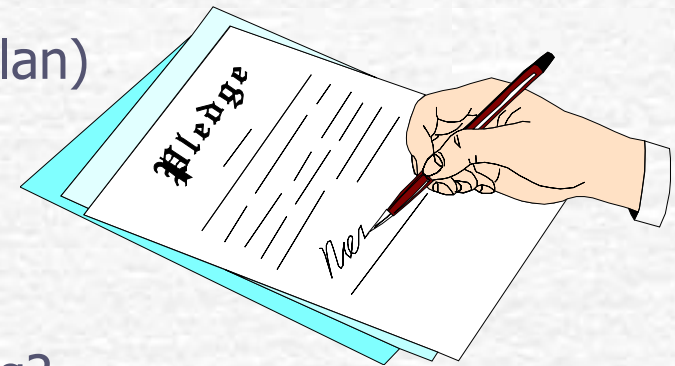
- Session
  - Basic testing work unit
  - Uninterrupted
    - e-mail, meetings, telephone calls etc.
  - Reviewable
    - A report should be produced
  - Chartered
    - Mission associated with this session; What are we testing? What are we looking for?





# ET Documentation

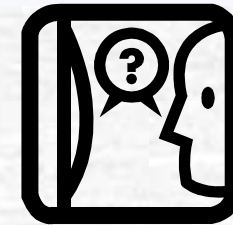
- Planning:
  - Charter – overview of what to test (plan)
    - Might be a flip chart on the wall
  - Mission – What are we looking for?
- Execution
  - Notes – what happened during testing?
    - What did I do? Why did I do it?  
Used to assess product quality after test.
  - Data files – input data used for testing
  - Bug reports – enough details to recreate the test / bug
  - Track questions and Issues





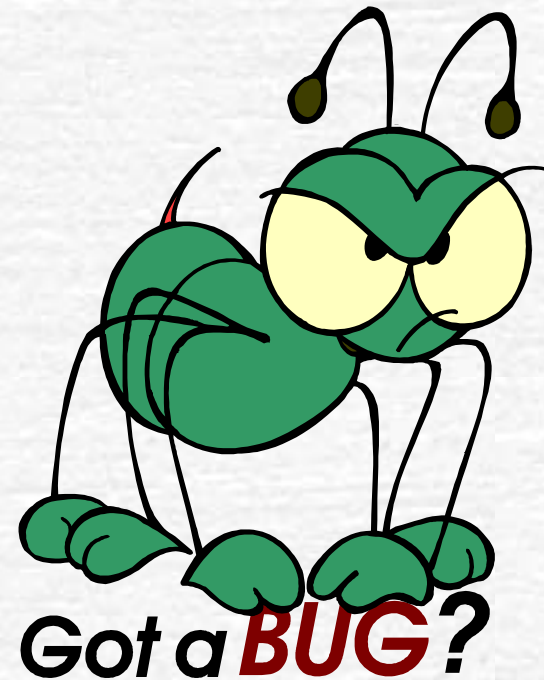
# Approaches / Styles of ET

- At the heart of all ET styles:
  - *Questions and Questioning Skills*
- Characterize the styles with respect to each other:
  - Do they focus on:
    - Method of questioning?
    - Method of describing or analysing the product?
    - The details of the product?
    - The patterns of use of the product?
    - The environment in which the product is run?
  - To what extent would this style benefit from group interaction?



# Styles of Exploration

- Hunches
- Models
- Examples
- Invariances
- Interference
- Error Handling
- Troubleshooting
- Group Insight
- Specifications





# Exploratory Testing Skills

- General systems modelling and dimensional analysis
- Inferencing out loud
- Experiment design
- Technical story telling
- Use of mnemonics and heuristics
- De-biasing (individual or team)
- Exploratory investigation
- Risk analysis
- Bug advocacy



*“I thought skill is the ability to something,  
more or less.*

*Skill varies from person to person.  
It’s distinct from talent and knowledge.*

*A technique, by contrast, is a way of doing  
something; a sort of recipe.*

*Skill belongs to a person, technique is  
universal.”*

*James Bach, Satisfice Inc.*

**James@satisfice.com**



# Heuristics (and rules) and Skills

*“... we relate to heuristics as a tool to apply; something that might help us do the right thing in a given situation, whereas we relate to a rule as something to comply with; something that defines right behavior.*

*Using heuristics properly requires that you exercise discretion and judgment, on some level; whereas judgment may get in the way of rules. It's helpful to have contradictory heuristics, because that's like having a variety of advice available before making a decision; whereas contradictory rules make compliance impossible.*



# Consistency Heuristics - "HICCUPP"

- Consistent with History: Present function behavior is consistent with past behavior.
- Consistent with our Image: Function behavior is consistent with an image that the organization wants to project.
- Consistent with Comparable Products: Function behavior is consistent with that of similar functions in comparable products.
- Consistent with Claims: Function behavior is consistent with what people say it's supposed to be.
- Consistent with User's Expectations: Function behavior is consistent with what we think users want.
- Consistent within Product: Function behavior is consistent with behavior of comparable functions or functional patterns within the product.
- Consistent with Purpose: Function behavior is consistent with apparent purpose.

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



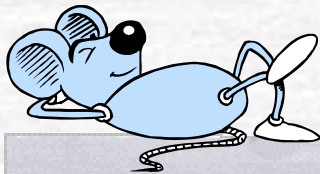
# ET – The Vanguard

*The object of your mission is to explore the Missouri river, & such principal streams of it, as, by its course and communication with the waters of the Pacific ocean...may offer the most direct & practicable water communication across this continent for the purposes of commerce.*

- Thomas Jefferson's letter to Meriwether Lewis, June 1803



# Presentation Outline



What is Exploratory Testing?

How to do Exploratory Testing?

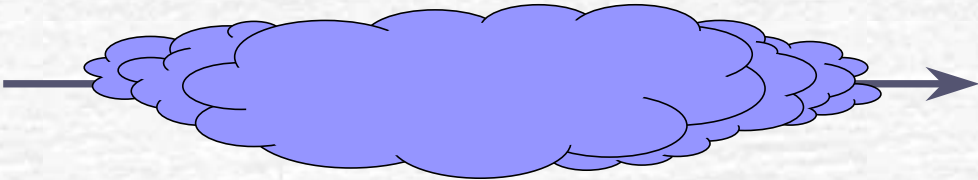
How to Manage Exploratory Testing?

High Speed WEB Testing – Case Study

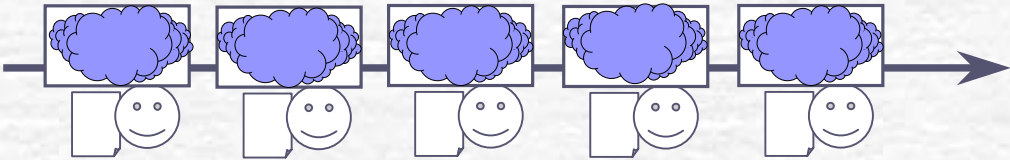


# The Test Session

- Charter
- Time Box
- Reviewable Result
- Debriefing



**VS.**



*From Rapid Software Testing, copyright © 1996-2002 James Bach*



## Charter – summary...

- “Architecting the Charters” i.e. Test Planning
- Brief information / guidelines on:
  - Mission: **Why** do we test this?
  - **What** should be tested?
  - **How** to test (approach)?
  - **What problems** to look for?
- Might include guidelines on:
  - **Tools** to use
  - Specific Test **Techniques** or **tactics** to use
  - What **risks** are involved
  - **Documents** to examine
  - Desired **output** from the testing



## Time Box: Focused test effort of fixed duration

Short: 60 minutes (+-15)

**Normal: 90 minutes (+-15)**

Long: 120 minutes (+-15)

- Brief enough for accurate reporting.
- Brief enough to allow flexible scheduling.
- Brief enough to allow course correction.
- Long enough to get solid testing done.
- Long enough for efficient debriefings.
- Beware of overly precise timing.

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



## Reviewable Result: The session sheet

- Charter
  - #AREAS
- Start Time
- Tester Name(s)
- Breakdown
  - #DURATION
  - #TEST DESIGN AND EXECUTION
  - #BUG INVESTIGATION AND REPORTING
  - #SESSION SETUP
  - #CHARTER/OPPORTUNITY
- Data Files
- Test Notes
- Bugs
  - #BUG
- Issues
  - #ISSUE

```
CHARTER
-----
Analyze MapMaker's View menu functionality and
report on areas of potential risk.

#AREAS
OS | Windows 2000
Menu | View
Strategy | Function Testing
Strategy | Functional Analysis

START
-----
5/30/00 03:20 pm

TESTER
-----
Jonathan Bach

TASK BREAKDOWN
-----

#DURATION
short

#TEST DESIGN AND EXECUTION
65

#BUG INVESTIGATION AND REPORTING
25

#SESSION SETUP
20
```

*From Rapid Software Testing, copyright © 1996-2002 James Bach*



## Debriefing: Measurement begins with observation

- The manager reviews session sheet to assure that he understands it and that it follows the protocol.
- The tester answers any questions.
- Session metrics are checked.
- Charter may be adjusted.
- Session may be extended.
- New sessions may be chartered.
- Coaching / Mentoring happens.

*From Rapid Software Testing, copyright © 1996-2002 James Bach*

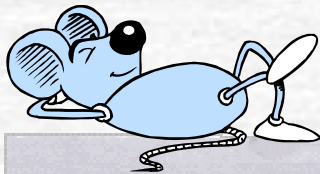


# Summary

- Exploratory Test Management
  - Requires skills and experience – practice
  - Be prepared! Improve your skills!
  - Charter – Define your mission
  - Take notes
  - Test in Sessions
  - Track Session Metrics



# Presentation Outline



What is Exploratory Testing?

How to do Exploratory Testing?

How to Manage Exploratory Testing?

High Speed WEB Testing – Case Study



# High Speed WEB Testing\*

## A Case Study

With permission from Jarle Våga, XIT AS,  
jarle.vaga@xit.no,  
Oslo, Norway

\* contribution to the book "Software Quality and Testing in Internet Times", see references



# Presentation Outline

- About the WEB application
- Preparation
  - Environment and people
- Test Planning
- Test execution
- “The web testing experience...”

*“It’s on the Web. Go find the bugs before the client does!”*



# Application: Basic functionality

- WEB publishing tool:
  - “Content Creator” creates pages
    - Multi dimension / organisational views
    - Multi language
  - “Topic Manager” have to approve each topic before publication

*“One company - One message”*



# Preparation...

- Documentation
  - System documentation not available – only high level standard product description
- Changes on the test environment
  - Constant changes during planning period





# What to do...

- Features to be tested:
  - Functional test
  - Search facilities
  - Browser tests
    - Internet Explorer 4.0 and up
    - Netscape 4.0 and up
  - Operating Systems:
    - Windows NT
    - Windows 98
    - Windows 2000
- ...would this be enough? And how to do it??





# Who?

- 12 testers
  - Different business areas
  - Different countries
  - Marketing and publishing people, a few IT literates
- 2 test coordinators
- Test Environment manager



# Test Environment

- 20 PCs
- 5 browsers (versions)
- 3 Operating systems
- 12 people
- 2 user groups simultaneously:
  - Publishers / Content Creator
  - Content Approvers / Topic Manager



## How to...

- Test scripts - abandoned
  - Unstable system
  - Unavailable documentation
- Pair testing
  - To motivate the testers and overcome any problems related to inexperienced people
- Exploratory testing
  - Will depend on the testers ability to explore the system to find bugs





# The Test Plan (Charters)

- Features to be tested
  - Important to be precise, this helps focusing the pair testing
- Features not to be tested
  - Just as important, used to limit the test
- Approach
  - Preparations and requirements
  - No specific test script (but rather “guidelines”)
  - Use of role-play and test matrixes
- Item pass/ fail criteria
  - ...in order to judge the outcome of the test.



# Test Execution

- Major difficulties with the set-up of the test environment
  - Browsers not installed
  - Win2000 pc's not available
- Functionality test according to guidelines
  - Creation of pages in the portal using role-play with pairs (Content Creator/Topic Manager)
  - Using a matrix, where roles, OS and browsers were integrated.
- Browser and op.sys. test
  - Matrix with 5 browsers and 3 op.sys.
- Yellow pages (search functionality)
  - re-use of pages published
- Exploratory test
  - what happens if...

# Execution – The Matrix

The upper matrix indicate how the groups worked on different combinations of browsers and OS'. The lower how the groups were divided into different user groups at the start of the test, for each change of operating system they were alternated.

I.e. Group 3 started as Content Creator working on NT with Netscape 6, and moved to W98 with Explorer 5 after lunch, working as Topic Manager. Different “test guidelines” was developed for each combination, which gave us  
 $OS * Browser = 3 * 5 = 15$  different guidelines.

Group #		Operating System		
Browser		W-NT	W98	W2000
	Netscape 4	6	1	
	Netscape 6	3	4/6	5
	Iexplore 4	1		
	Iexplore 5	4/5	2/3	
Time	Iexplore 5.5	2	5	6
	Day 1	10:00-12:00	13:00-15:00	15:00-16:30
	Day 2	09:00-17:00	09:00-17:00	13:00-16:00

User group	Groups					
	1	2	3	4	5	6
<i>Content Creator</i>	•		•		•	
<i>Topic Manager</i>		•		•		•



## Execution – Exploratory pair testing

- Plan – increase coverage by using the testers creativity
- ...which is what happened
- It can be difficult to keep testers “on track”
- ...but easy to motivate, and many bugs found.



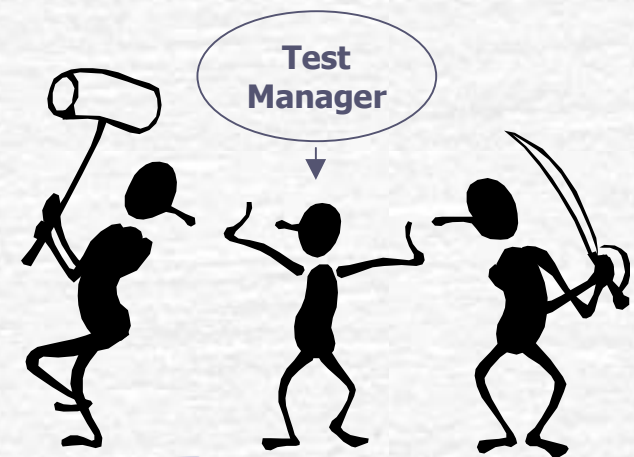
## Execution – Debriefing and Notes

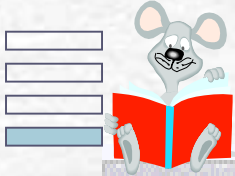
- Continuous guiding / coordination required
- Taking Notes did not seem to be a problem (due to testers working in pairs)
- High number of duplicate error messages due to no clear mission set for each pair
- Debriefing takes time! Need to be planned



# Acceptance Test Summary

- Approx. 150+ bugs reported (incl. duplicates)
  - Mainly due to lack of limitations in the system, you could do “anything”.
  - Developer accepted only approx. 60
- Corrupted pages: Web portal made inaccessible
  - Menu disappeared when using signs like (“#\*\’/@ etc”) in page title
    - Customer: Severe safety risk
    - Developer: Tester have misused the system

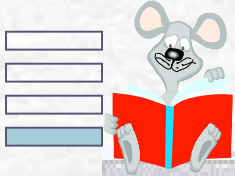




# Case Summary

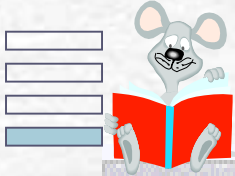
- The portal was NOT recommended for pilot production after the test.
- Using pair- and exploratory methods proved successful
- Has happened since then:
  - The two test coordinators scripted tests for the 60 bugs fixed and participated in a 2 days re-test of the application
  - Application went live after this total of 4 days testing – and has been in production without significant problems for more than a year





# Lessons learned...(1)

- Pair testing
  - Speeded up reporting errors
  - Excellent to promote creativity
  - Encouraged discussion
  - Demanded a “strong voice” to keep everyone focused!
  - Enables effective testing of short-lived, intense projects...
  - but has also proved very useful on many test projects, also larger (i.e. at Microsoft, ref. J.Bach)



## Lessons Learned...(2)

- Exploratory testing
  - Requires skilled testers
  - Requires some pre-set restraints, and thorough planning
  - Requires a system for briefing and reporting/ documenting (ref. J.Bach - "Session based test management")
  - Requires guidance from test manager
  - ..but when the requirements are fulfilled, it proved to be very successful.
  - Easy to motivate
  - MANY deviations found
  - ...and it gave testers a greater feeling of participation



## Learn more...

- Exploratory Testing:
  - Rapid Software Testing, by James Bach, [james@satisfice.com](mailto:james@satisfice.com)
  - Black Box Software Testing, by Cem Kaner, [kaner@kaner.com](mailto:kaner@kaner.com)
  - Amland Consulting; London, Copenhagen and Stockholm:
    - [http://www.amland.no/training\\_schedule.htm](http://www.amland.no/training_schedule.htm)
- People Issues, test techniques, inspections:
  - Grove Consultants, [www.grove.co.uk](http://www.grove.co.uk)
  - + many, many more



# References/Bibliography (1)

- Amland, S., 1999. Risk Based Testing and Metrics, EuroSTAR '99 Preceding, Barcelona, Spain, <http://www.amland.no/articles>
- Argus, C. & Johnson, B., Ad Hoc Software Testing; Exploring the Controversy of Unstructured Testing, [http://www.testingcraft.com/ad\\_hoc\\_testing.pdf](http://www.testingcraft.com/ad_hoc_testing.pdf)
- Bach, James, 1999a. Risk-Based Testing. How to conduct heuristic risk analysis, Software Testing & Quality Engineering Magazine, November/December 1999, vol. 1, issue 6, <http://www.stqemagazine.com>. See also [www.satisfice.com](http://www.satisfice.com)
- Bach, James , 1999b. What is Exploratory Testing? Satisfice.com
- Bach, Jonathan, 2000, Session Based Test Management, STQE Magazine, 11/00, [www.satisfice.com](http://www.satisfice.com)
- Beizer, Boris, Software Testing Techniques, Van Nostrand Reinhold. NY. 1990.
- Collard, Ross, 2002, to be published, [rcollar@attglobal.net](mailto:rcollar@attglobal.net)



## References/Bibliography (2)

- Kaner, C., Falk, Nguyen, 1999, Testing Computer Software, John Wiley & Sons
- Kaner, C., 2000. Rethinking Software Metrics, Evaluating measurement schemes, Software Testing & Quality Engineering Magazine, March/April 2000, vol. 2, issue 2.
- Kaner, C., 2001a, Black-Box Software Testing, training at Satisfice July 2001.
- Kaner, C., Bach, J., Pettichord, B., 2001b, Lessons Learned in Software Testing, John Wiley & Sons; <http://www.testinglessons.com/>, ISBN: 0471081124
- Kaner, C., Bach, J. 2001c, Exploratory Testing in Pairs, presentation at STAREast, Orlando, FL, [www.kaner.com](http://www.kaner.com)
- Keith, Geordie, 2002, "All Hands on Deck: How we used our whole company to test", STQE Magazine, vol. 4, issue 4, July / August 2002.
- Lyndsay, James and Eeden, Niel van, 2002, "Adventures in Session-Based Testing", paper at StickyMinds.com, will also be presented at EuroSTAR 2002
- Våga, J., Amland, S., 2002. High Speed Web Testing, contribution to the book "Software Quality and Testing in Internet Times", Editor: Dirk Meyerhoff et al, Springer-Verlag, 2002, ISBN: 3-540-42632-9

# Contact Details



Ståle Amland  
Amland Consulting  
Hulda Garborgsv. 2  
N-4020 STAVANGER  
Norway

Phone: +47 905 28 930

Fax: +47 51 58 55 24

e-mail: [stale@amland.no](mailto:stale@amland.no)

WEB: [www.amland.no](http://www.amland.no)