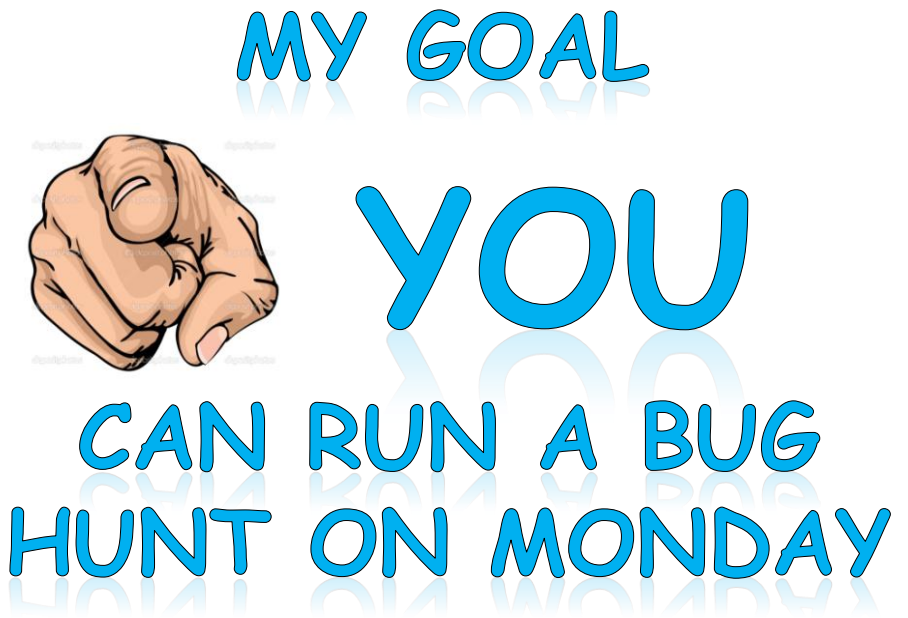
Advance your testing: Go on Bug Hunt!

Presented by Klaus Olsen
softwaretest.dk

On ANZTB TEST 2013 ADVANCING EXPERTISE IN SOFTWARE TESTING

Conference in Canberra
6 June 2013



softwaretest.dk

Klaus Olsen

- Founder and owner of the company Softwaretest.dk in 2000
- Has used the past 20 years to focus on software testing, test process improvements and teaching

- Author of "Softwaretest – how to get started" in Danish
- Board member of the non-profit organization TMM*i* ® Foundation
- TMMi Management Executive Chair
- Member of ISTQB, representing Denmark
- Co-author of ISTQB Foundation and Advanced Syllabus
- Certified ISTQB Foundation and Test Manager Advanced
- Certified Scrum Master

softwaretest.dk

Public presentation includes:

- *EuroSTAR´98 in Münich, Germany.*
- *Second World Congress on Software Quality 2000 in Yokohama, Japan.*
- *EuroSTAR´2001 in Stockholm, Sweden.*
- *Quality Week 2001 in San Francisco, USA.*
- *EuroSTAR´2003 in Amsterdam, Holland.*
- *ASTA 2007 in Seoul, Korea.*
- *Test 2008 in New Delhi, India.*
- *EuroSTAR´2008 in Haag, Holland.*
- *ANZTB Test2009 conference Sydney, Australia.*
- *JSTQB カンファレンス 2010, in Tokyo, Japan.*
- *ASTA 2010 in Seoul, Korea.*
- *Czechtest 2011 in Prague, Czech Republic.*
- *TAPOST 2011 in Riga, Latvia.*
- *TMMi Seoul International Conference 2011 in Seoul, Korea.*
- *Czechtest 2012 in Prague, Czech Republic.*
- *Nordic Testing Days 2012 in Tallinn, Estonia.*

Contact Klaus by mail klaus@softwaretest.dk

FIRST SOME FUN

HTTP://WWW.YOUTUBE.COM/WATCH?V=ZNM0ENUCO5I

AND THEN SOME BUG HUNTING

TESTING CAN BE FUN

softwaretest.dk

Agenda

- Introduction
- Test in Pairs
- Exploratory Testing
- Bug Hunting
- Real World Experience

softwaretest.dk

What can be achieve?

- Case 1; 32 faults identified during 45 minutes Bug Hunt in software ready for acceptance test - according to supplier!
- Case 2; 72 faults identified during 2 hours Bug Hunt this was also in software ready for acceptance test according to supplier!
- Case 3, Bug Hunting during 3 days with participation of developers, designers, architects and testers, in total 20 persons.
  - 60 man days of test executed in 3 calendar days.
  - A conservative estimate of the efficiency is that this equals 75 man days of “normal” work.
- Case 3 identified 180 faults during 3 calendar days!

softwaretest.dk

When can Bug Hunting be used?

- When a new release of software is ready for test, a Bug Hunt will very clearly read the temperature ~ quality of the software.
- As an entry-criteria for new phases. View it as a ”smoke test” executed by people, instead of automated test, if you don´t have any of these.
- As team-motivation, when test execution becomes day to day work, and the auto-pilot is taking over, a Bug Hunt can be what you need to add extra adrenalin to your test.

softwaretest.dk

Ingredients in a Bug Hunt

- Test in pair, 2 people with one computer.
- Exploratory Testing.
- Different attacks, to make the software break down.
- Soap opera scenarios, short, wild exaggerated, but with humour and often very good at identifying faults.

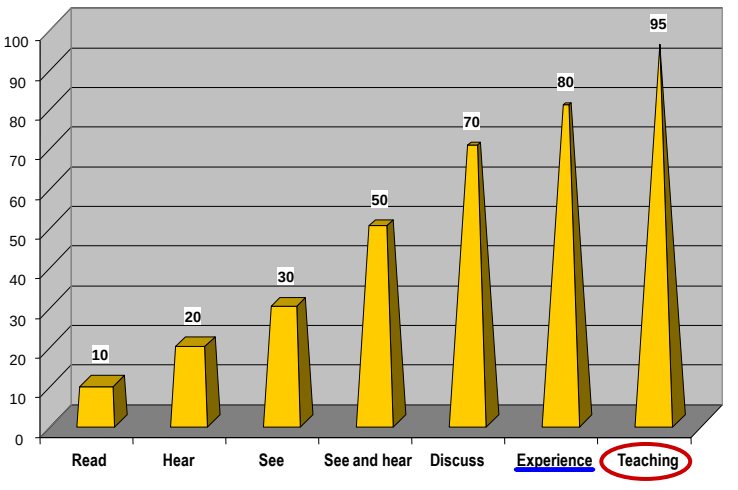
softwaretest.dk

Agenda

- Introduction
- Test in Pairs
- Exploratory Testing
- Bug Hunting
- Real World Experience

softwaretest.dk

Learning ...



Stephen R. Covey, *The 7 Habits of Highly Effective People* → **William Glasser**

softwaretest.dk

Exploratory Testing

- ”**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 we did, we are doing exploratory testing.”

  James Bach, Satisfice

- **Exploratory Testing** is test we didn´t plan and document prior to test execution. What most of us see as one of the native laws of testing, we must be able to define an expected result before we start testing, isn´t true in exploratory testing.

- But as an important note, when doing Exploratory Testing we document all faults we identify carefully enough in order for others to reproduce the faults.

softwaretest.dk

When can Exploratory Testing be used?

- On projects where you don´t have:
  - Enough time to work with test planning.
  - Enough time to document test cases with input and expected output data.
- On projects where there isn´t enough people assigned to testing.
- On projects without any documented requirement or very weak requirement specification.
- When you work with Risk Based Testing, low risk areas could be done using Exploratory Testing.
- During Bug Hunting.

If Exploratory Testing is used it is my recommendation that a limit of 50% of all test should be Exploratory Testing!

softwaretest.dk

Exploratory Testing, step by step

- Exploratory Testing can be described as a goal oriented wandering.
- There is a mission described in a charter, but there is no planned route you need to take.

Step 1:
- Create a charter describing what and how and which way you want to test.

Step 2:
- Describe duration of your test.

Step 3:
- The two people in the Pair decides whether or not to break down the charter in more details, depending on there own needs.

softwaretest.dk

Charter template used as documentation

PROJECT NAME		SOFTWARETEST.DK	
Area		Tester	
Environment		Delivery	
To be filled in after completion of testing			
Number of faults		Date	
Number of issues		Duration	

Charter: *(You should see 4 lines of hidden text in orange below this line as a writing guideline, if you don't please select Tools, Options, View and select Hidden text and OK.)*
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.
Specific charters provide better focus, but take more effort to design.
Charter …

Time Box:
Each test is planned to be executed in one of the 3 fixed durations below:
1. *Short time box: 60 minutes (+/-15)*
2. *Normal time box: 90 minutes (+/-15)*
3. *Long time box: 120 minutes (+/-15)*

Suggest time box:

Task Breakdown:
Testers own breakdown of above charter.

Test Notes:
Notes from test execution.

Faults:
Headlines of faults encountered during test execution, to be raised with more detail in Fault tracking system after the time box session.
The following ..

Issues:
Problems encountered during test execution which isn't faults, but things we want to capture for later investigation, or it could be ideas for further testing, but out of scope for this time box session.
During ..

softwaretest.dk

A famous example

"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

Robinson, H., Microsoft, Exploratory Modelling

softwaretest.dk

Teach the person next to you

Use 2 minutes teaching Exploratory Testing

softwaretest.dk

Soap opera scenarios

- They are created with inspiration from real world.
- They are just much more compact:
  - One week can be presented in a 30 minutes episode on TV!
- They are much more extreme:
  - lead character gets married,
  - man and wife gets 3 kids,
  - man dies,
  - woman gets married again,
  - 2 more kids are born,
  - one child gets cancer and dies,
  - a love affair from youth arrives and new problems are created, wife leaves man….

softwaretest.dk

Why use soap opera scenarios

- It is an effective way to team up business domain knowledge with test experienced people. Use pair test design and pair test execution.
- Test are covering the system under test more broad, this is a black box approach.
- Less depended on requirement specification.
- Bigger opportunities to detect ”Design Holes”.
- They are fun to create, they challenge your creativity, test becomes interesting instead of boring.

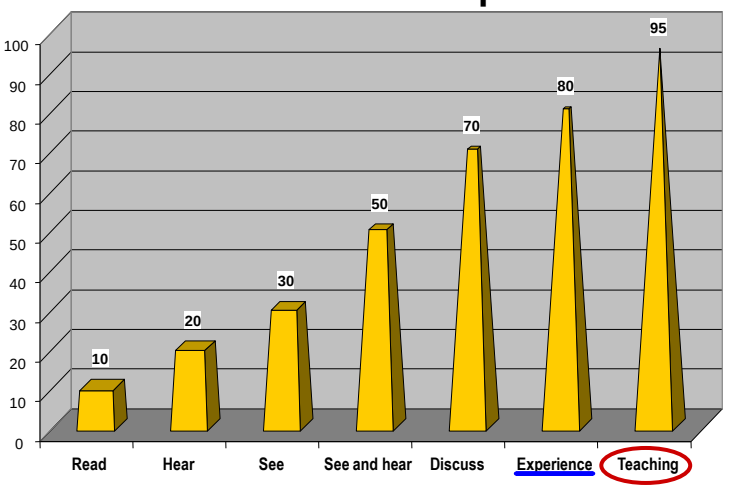
softwaretest.dk

Agenda

- Introduction
- Test in Pairs
- Exploratory Testing
- Bug Hunting
- Real World Experience

softwaretest.dk

Learning - again
but now the other person



Stephen R. Covey, *The 7 Habits of Highly Effective People* → **William Glasser**

softwaretest.dk

Attacks for Bug Hunting

- Use input values that forces all error messages to be activated.
- Use input data that forces the software under test to re-create default values.
- Try all possible characters.
- Force the input area into overflow,
  - Use long strings of input, larger than what the application is designed to handle.
- Force illegal output values.
- Test with illegal operators and data.

Find more attacks in "*How to Break Software*" by James A. Whittaker

softwaretest.dk

3 Artifacts in Bug Hunting

Bells

Clock

Prizes

softwaretest.dk

Bells

Bells are used to sound when a fault is found

softwaretest.dk

Clock, fix time

A Bug Hunt is always within a fix time period

softwaretest.dk

Prizes

- ✓ The pair who reports the most serious fault, from a business perspective, are awarded a prize
- ✓ The pair who reported most faults are awarded a prize

softwaretest.dk

2 Roles in Bug Hunting

Coach

Referee

softwaretest.dk

Coaching

- It is recommend to use a coach during the hunt
- The coach advise on new attacks, blind roads

softwaretest.dk

Referee

- A referee is used to judge if a fault is identified
- All faults must be reproducible when the judge is watching

softwaretest.dk

Bug Hunting - How to get started

1. Make sure you have a test environment with data and access.
2. Create charters for the Bug Hunt as used in Exploratory Testing.
3. Prioritise these charters based on risk impact and likelihood.
4. Pair people, based on skills and domain knowledge.
5. Define a period for how long the Bug Hunt should be running e.g. 30, 60, 90 or 120 minutes.
6. Handout paper template documents to be used for each identified faults.
7. Handout bells for each pair to be used when new defects are identified, and the referee is needed to approve it.
8. The referee makes together with the coach the final decision on which faults was the best, and the pair who reported this fault is awarded 1. prize in the Bug Hunt.
9. Hand out the prize to the pair with most reported faults.

softwaretest.dk

Teach the person next to you

Use 2 minutes teaching Bug Hunting

softwaretest.dk

Agenda

- Introduction
- Test in Pairs
- Exploratory Testing
- Bug Hunting
- Real World Experience

softwaretest.dk

Case 1

Bug Hunting in 1 hour

- 8 people participated + 1 referee and 1 coach.
- 15 minutes used for a short instruction to the Bug Hunt.
- 45 minutes were used for Bug Hunting,
- Test of a standard system with customer modification, ready for accept test according to supplier.
- 32 faults were identified during 45 minutes!
- Two pairs wanted more time, they were not done, and they were quite sure they could find more bugs!
- Everybody were fired up, this Bug Hunt was a different approach to testing, but it was also interesting, and test in a complete new way for all participants.

softwaretest.dk

Case 1

Experience from case 1 with Bug Hunting

- Testing in Pairs works:
  - Domain knowledge was actively shared between members of all pairs.
  - Test techniques were discussed and applied.
  - People influence each other, one example were a pair who identified an overflow error, and one of the testers suggested they used this as a thread to investigate other areas where overflow might exist.
- The sound of bells being used for each bug makes everybody more drawn in.
- All pairs want to find bugs, and the competition makes everyone focus even more on the task.

softwaretest.dk

Case 2

Bug Hunting in 2 hours

- 12 people participated + 1 referee and 1 coach, since a small it-department all participated.
- 15 minutes used for a short instruction to the Bug Hunt.
- 3 * ½ hour were used for Bug Hunting, see next slide.
- 15 minutes used to sum up experience.
- Test of a new developed system for a University for students to sign up for new classes each semester.

softwaretest.dk

Session based TM example

Case 2

	Pair 1 and 2 charter	Pair 3 and 4 charter	Pair 5 and 6 charter
First ½ hour	Student data (name, age, address ...)	Students selecting classes	Administration module
	Pair 1 and 2	**Pair 3 and 4**	**Pair 5 and 6**
Second ½ hour	All reports from system	Usability from all areas	Security areas
	Pair 1 and 2	**Pair 3 and 4**	**Pair 5 and 6**
Third ½ hour	Freestyle or go back to above areas	Freestyle or go back to above areas	Freestyle or go back to above areas

softwaretest.dk

Experience from case 2 with Bug Hunting

Case 2

- 77 faults were identified during 2 hours!
- Only 5 faults were duplicates.
- General disappointment with the quality of the delivery.
- Test Manager didn´t have to explain to management that the quality was not good enough.
- Management were part of the Bug Hunt and had already made there own conclusion.

softwaretest.dk

Case 3

Bug Safari

- In larger scale, a Bug Safari can be a way to catch up for lost time, and at the same time measure the quality of the project.
  - 3 days with participation of developers, designers, architects and testers, in total 20 persons:

✓ **DAY 1:**
Understand the system under test, read documentation, product exploration, test positive test cases, testing things expected to be working

✓ **DAY 2:**
Exploratory Testing with different attacks.

✓ **DAY 3:**
Exploratory Testing with use of soap opera scenarios

softwaretest.dk

Case 3

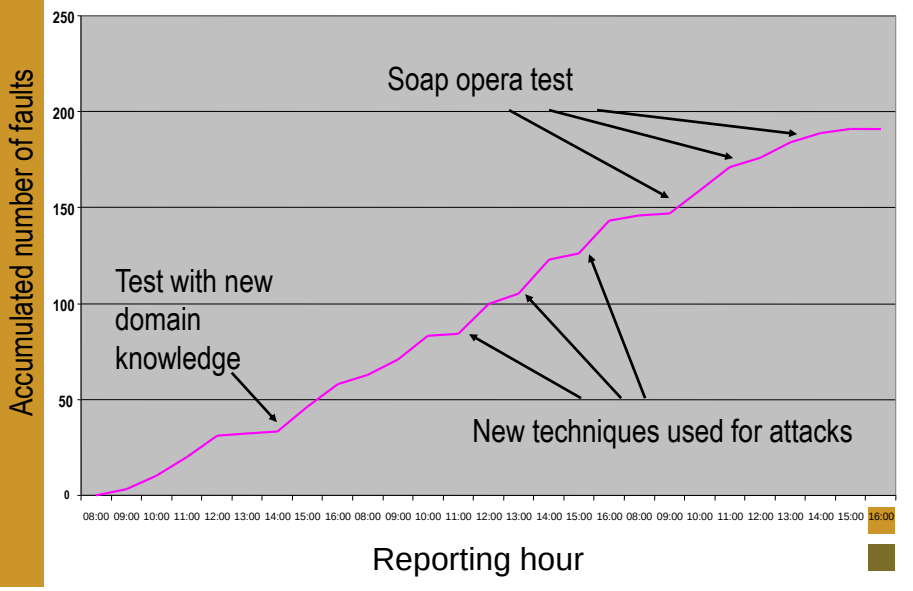
Good planning was a big advantage

- One room, where all people from the project were present, and all questions arised could be answer by at least one person.
- One person controlled all faults in order to make sure that only new faults were reported.
- Faults were reported on paper, using a template handed out prior to the bug hunt.
- A secretary was assigned to log all faults in a fault-log tool in order to keep the database with faults updated at all times, this was used to extract a fault curve.
- The result were a higher “E-factor”. The actually 60 man days, was judge to be closer to a normal effort of 75 man days, when measuring the effectiveness. This is an increase of 25%!

softwaretest.dk

Case 3

Fault reporting hour by hour in a Bug Safari



softwaretest.dk

Case 3

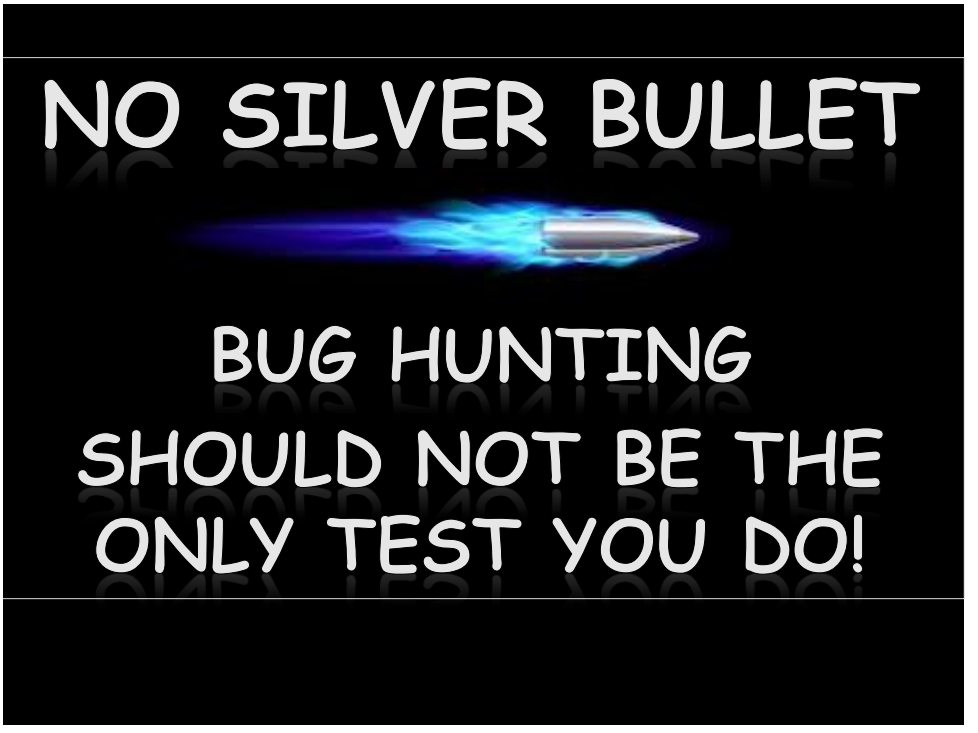
Experience from case 3 with Bug Hunting

- Knowledge sharing between all members of the project:
  - On domain, how should the product be used
  - On design ideas as they were thought out original, what was the thought behind the design, at the time it was created
  - On programs, why have they been developed as they have, told by the programmers who did it
- The test team gained respect form all colleagues, the number of bugs identified in 3 days showed test was very necessary.
- The curve of faults reported was slowing down end of day 3, this was viewed as a sign of less defect density.
- Signal value, quality is important, test works.

softwaretest.dk

More information

- **How to Break Software**
  - James A. Whittaker
  - ISBN 0-201-79619-8
- **More on Exploratory Testing see**
  - James Bach and his web-site:
  - www.satisfice.com



softwaretest.dk

Thank You

Contact Klaus
by email at klaus@softwaretest.dk