

Exploratory Testing approach – Personal knowledge as a test oracle

Juha Itkonen Aalto University School of Science Department of Computer Science and Engineering juha.itkonen@aalto.fi 050 577 1688



Motivation

- Manual testing is a crucial practice for achieving software quality
 - Automation cannot replace the benefits of manual testing
- Research on testing focuses on theoretical optimizations and test
 case design techniques
 - Results, however, are inconclusive and contradicting.
- Experience-based and exploratory testing approaches are often applied in practice
 - Perceived to be effective and efficient.
- There is a gap between the testing research and industrial practice
 - Lack of research on how testing is done in the real world practice.



Manual Testing

Aalto University

School of Science

- Testing that is performed by human testers
- Stereotype of manual testing
 - Executing detailed pre-designed test cases
 - Mechanically following the step-bystep instructions
 - Treated as work that anybody can do

juha.itkonen@aalto.fi 31.1.2012 3

Research has shown:

- 1. Individual differences in testing are high
- 2. Testing techniques alone do not explain the results

In practice, it's clear that some individuals are better than others in testing and more effective at revealing defects...

Scripted vs. Exploratory Testing (ET)



Exploratory





Exploratory testing

- Is not based on pre-designed scripts
- Parallel test design, execution, interpretation of results, and learning
- Tester is in control
 - designs and improves new tests
 - based on the observed results
- Relies on the skills and knowledge of the tester
- Personal experience is applied directly to the testing



ET is efficient testing approach

Few studies comparing exploratory vs. scripted testing approach report:

- Exploratory testing reveals at least as many defects than scripted approach
- Exploratory testing is much more cost effective
 - Avoiding the expensive pre-design and documentation of the details of every test





Experimental Comparison of ET and Test Case Based Testing (TCBT)

Itkonen, J., M. V. Mäntylä and C. Lassenius. "**Defect Detection Efficiency: Test Case Based vs. Exploratory Testing**", in proceedings of the International Symposium on Empirical Software Engineering and Measurement, pp. 61-70, 2007.

- Effectiveness in terms of detected, reported defects
- Test execution time was fixed
- ET revealed little more defects
 - no statistical difference
- ET was much more efficient
 - TCBT required over five times more effort
- TCBT produced twice as many false reports than ET



Examples of efficiency of ET in our studies

Observations, round 1

- 4 organizations
- 2,9 defects / h

Student experiment

- 85 testers
- 4,7 defects / h
 (TCBT 0,75 defects / h)

Observations, round 2

- 4 organizations
- 6,0 defects / h

Industrial case study

- Case A: 4,8 defects / h
- Case B: 8,5 defects / session



Test oracle – How to recognize a failure when it occurs





The oracle problem

Expected results and recognizing a failure

- Oracle problem is one of the fundamental challenges in software testing
 - Oracle problem is a relevant challenge of all testing
 - A serious limitation and challenge in test automation
- Scripted testing aims at "solving" it by predocumenting the expected result in test cases
 - In practice, very challenging problem that cannot be solved simply by writing "the expected result" down



Personal knowledge as an oracle

- One aspect of exploratory testing is interpreting the test results and recognizing the failures
- Behaviour of systems is too complicated to predict
 - to describe comprehensively and precisely all that can go wrong
- Bugs are surprising and testers are able to recognize one when they see it
 - Human tester can identify problems without designing a check for that particular type of problem beforehand
- Partial oracles¹
 - Tester with experience can identify incorrect results that are not plausible without knowing the exactly correct result
 - E.g. a comptroller can differentiate incorrect values for financial figures
 - 300€, 1000€, 10 000€ and 250 000 € are clearly incorrect if correct figure is known to be around 1 000 000€, without knowing the correct figure exactly, e.g. 1 103 456,42 €

¹Weyuker, E.J., 1982. On Testing Non-Testable Programs. *The Computer Journal*, 25(4)



The role of knowledge in failure detection

Itkonen J., Mäntylä M. V., Lassenius C., "**The Role of Knowledge in Failure Detection During Exploratory Software Testing**", In review for *IEEE Transactions on Software Engineering*.

- Field observation study
 - Observing professionals performing testing
- Detailed analysis of 91 failure detections in real testing sessions from four organizations
- Analysed what type of *knowledge* is required for *detecting failures*?
- Analysed failure detection difficulty



How did we research ET in practice: Field observations in industry

- Field observations on testers' work in industry
 - Real testing work
 - Video taped
 - Several organizations, 10+ subjects, 20+ observed sessions





Identified knowledge categories

Domain knowledge

- Users' perspective
- Application domain perspective

System knowledge

- Interacting features and system perspective
- Individual features and functional perspective

Generic knowledge

- Generic correctness perspective
- Usability perspective
- Direct failure perspective









Spread of the knowledge

Spread	Domain knowledge	System knowledge
Focused	Application domain perspective	Individual features and functional perspective
Holistic	Users' perspective	Interacting features and system perspective

- It seems that focused knowledge types were more often applied as a pure oracle
- Holistic types were applied also to test design
 - e.g. simulating user's goals and activities or attacking a known risk



Opportunity bugs

- Relatively high number (20%) of bugs were found by opportunity
 - Meaning that testers detected failures in other features than the primary target of the testing session in question



- as a result of exploring, as a "side effect"
- This finding supports the strength of ET in enabling more versatile testing
 - Testers are not working blinders on
 - Testers explore and investigate the system, and reveal bugs, when they see the opportunity





Conclusions: Personal knowledge as a test oracle

- Testers are able to apply varying types of knowledge as an oracle
- The most distinctive knowledge types seem to be
 - Users' perspective
 - Individual features perspective
 - Interacting features perspective
- Similar concepts have been identified also in studies of human competence at work on other fields
 - e.g., Sandberg, J., 2000. Understanding human competence at work: An interpretative approach. *Academy of Management Journal*, 43(1): 9-25.



Conclusions: Not all bugs are buried deep or masquerade cleverly

- Almost third of the failures could be identified based on generic knowledge
- Over 50% were obvious or straightforward to reveal in terms of interacting variables

This implies that it is possible to provide fast contribution without rigorous or sophisticated test design or deep knowledge...

... but the challenge is to know what remains under the surface.





Is there alternatives for experience based oracle?

- It seems that experience based oracles are often enough
- If documentation is needed it often does not provide the answer -> testers have to ask others
 - Many times they prefer to ask people without bothering to dig into the documentation at all
- In real testing the goal is not to check against the documentation, but to test and reveal new information



Conclusions: Contribution of domain experts

- Failures that required specific domain knowledge or users' perspective to be revealed were often straightforward to provoke
- People with right type of knowledge are useful for revealing defects and issues even if not very skilled in testing



Challenges – to distinguish obvious and straightforward from hidden and complicated

- Our results contradict the need for scripted approach for less experienced testers
- It is easy to see what is on the surface
- What lies below will probably determine the result at the end
- Managing different types of testing contributions is a challenge
 - Understanding the testing done by different testers and how much their efforts can be relied on
 - Interpreting the results and findings of different testers





Summary

- Much can be achieved without detailed pre-design or scripting
 - no need to have documented result to check against
- We suggest that exploratory testing is an effective testing approach even for less experienced testers
- The ET approach is an effective way of involving the knowledge of domain experts in testing activities
 - who are not experts in testing
- Next we need deeper understanding of the highly skilled exploratory testing
 - The advantages of truly devoted and passionate testers



Read more lessons and observations in the ESPA Guidebook

- Intelligent Manual Testing approach
- Descriptions of empirically observed testing practices
- Time-paced framework for analysing quality practices in iterative and incremental (agile) development

http://www.soberit.hut.fi/espa/seminar/



TESTERA Bringing software testing to a new era Research project under preparation right now!

- Preliminary research themes
 - Exploratory testing
 - Testability
 - Model-based test automation
 - New competences of testing
- Industry partners needed

If you are interested in these topics, please contact us! juha.itkonen@aalto.fi 050 577 1688





List of related publications

Itkonen J., "Empirical Studies on Exploratory Software Testing", Doctoral dissertation, Aalto University School of Science, Novembr 2011.

Itkonen J., Mäntylä M. V., Lassenius C., **"The Role of Knowledge in Failure Detection During Exploratory Software Testing"**, Submitted in review for *IEEE Transactions on Software Engineering*, 2011.

Mäntylä, M. V., Itkonen, J., Iivonen, J., "Who Tested My Software? Testing as an Organizationally Cross-Cutting Activity", Software Quality Journal, 2012.

Itkonen, J., M. V. Mäntylä and C. Lassenius. "How do Testers Do It? - An Exploratory Study on Manual Testing Practices", in proceedings of the International Symposium on Empirical Software Engineering and Measurement, 2009.

Vanhanen J, Itkonen J. Mäntylä M.V., "Lightweight Elicitation and Analysis of Software Product Quality Goals - A Multiple Industrial Case Study", IWSPM, August 2009.

Itkonen, J., M. V. Mäntylä and C. Lassenius. "**Defect Detection Efficiency: Test Case Based vs. Exploratory Testing**", in proceedings of the International Symposium on Empirical Software Engineering and Measurement, pp. 61-70, 2007.

Itkonen, J., K. Rautiainen and C. Lassenius. "**Toward an Understanding of Quality Assurance in Agile Software Development**", in *International Journal of Agile Manufacturing*, vol. 8(2), pp. 39-49, 2005.

Itkonen, J. and K. Rautiainen. "Exploratory Testing: A Multiple Case Study", in Proceedings of the International Symposium on Empirical Software Engineering, pp. 84-93, 2005.

