



**Aalto University**  
School of Science  
and Technology

**Ohjelmistotestauksen Teemapäivä, TTY, 8.6.2010**

# **How Do Testers Do It? Exploratory and Experience Based Testing**

Juha Itkonen

Software Business and Engineering Institute (SoberIT)

[juha.itkonen@tkk.fi](mailto:juha.itkonen@tkk.fi)

+358 50 5771688



# Contents

- Introduction to experience based and exploratory testing
  - Intelligent manual testing
- Overall strategies and detailed techniques
- Selection of exploratory tester's pitfalls



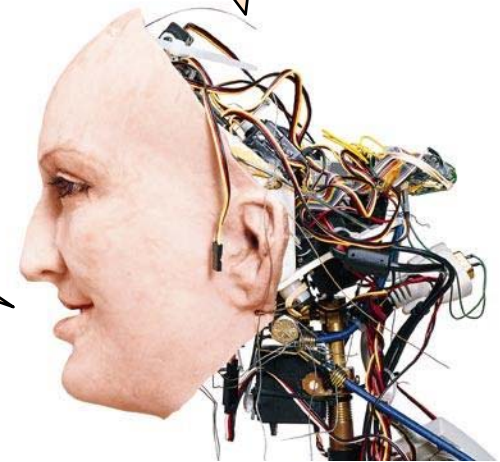
# Manual Testing

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

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

## Research has shown:

1. Individual differences in testing are high
2. Test case design techniques alone do not explain the results



# My viewpoint: Experience Based – Intelligent – Manual Testing

- Manual testing that builds on the tester's experience
  - knowledge and skills
- Some aspects of testing rely on tester's skills
  - during testing
  - e.g., input values, expected results, or interactions
- Testers are assumed to know what they are doing
  - Testing does not mean executing detailed scripts
- Focus on the actual testing work in practice
  - What happens during testing activities?
  - How are defects actually found?
  - Experience-based and exploratory aspects of software testing



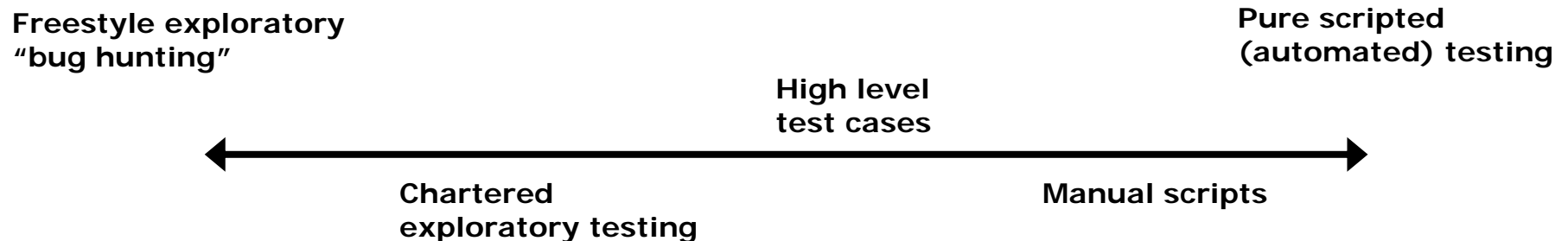
# Exploratory Testing is creative testing without predefined test cases

Based on knowledge and skills of the tester

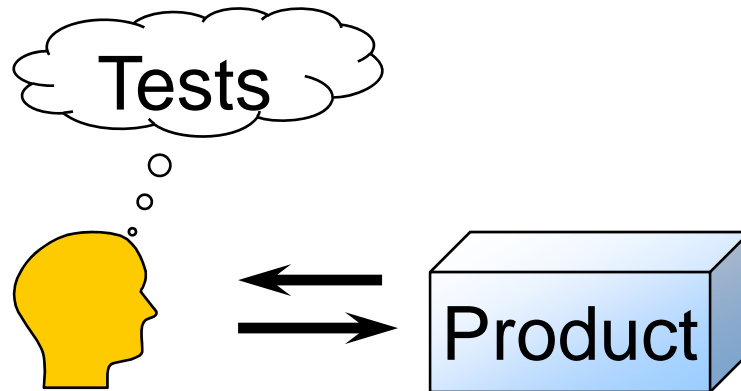
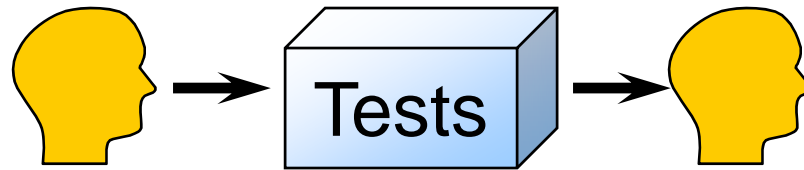
1. **Tests are not defined in advance**
  - Exploring with a general mission
  - without specific step-by-step instructions on how to accomplish the mission
2. **Testing is guided by the results of previously performed tests** and the gained knowledge from them
3. **The focus is on finding defects** by exploration
  - Instead of demonstrating systematic coverage
4. **Simultaneous learning** of the system under test, test design, and test execution
5. **Experience and skills of an individual tester** strongly affect effectiveness

# Exploratory and scripted testing are the ends of a continuum

- ET is an approach
- Many testing techniques can be used in exploratory way



# Scripted Model vs. Mental Model



# Strengths of IMT – Testers' skills

- Utilizing the knowledge and skills of the tester
  - Testers know how the software is used and for what purpose
  - Testers know what functionality and features are critical
  - Testers know what issues are relevant
  - Testers know how the software was built
- Enables creative exploring
- Enables fast learning and improving testing



# Strengths of IMT – Process

- Agility and flexibility
  - Easy and fast to focus on critical areas
  - Fast reaction to changes
  - Ability to work with missing or weak documentation
- Effectiveness
  - Reveals large number of relevant defects
- Efficiency
  - Low documentation overhead
  - Fast feedback when changes or new risks appear

# Challenges of IMT

- Planning and tracking
  - How much testing is needed, how long does it take?
  - What is the status of testing?
  - How to share testing work between testers?
- Managing test coverage
  - What has been tested?
  - When are we done?
- Logging and reporting
  - Visibility outside testing team
    - or outside individual testing sessions
- Quality of testing
  - How to assure the quality of tester's work
    - Detailed test cases can be reviewed, at least

# Some ways of exploring in practice

1. Session-based exploratory testing
2. Functional testing of individual features
3. Exploratory smoke testing
4. Freestyle exploratory testing
  - Unmanaged ET as part of other duties
  - Extending test cases
5. Outsourced exploratory testing
  - Advanced users, strong domain knowledge
6. Exploratory regression testing
  - by verifying fixes or changes

# Intelligent Manual Testing Practices

- Empirically observed practices from industry
- Testing, not test case pre-design
- Practices work on different levels of abstraction
  - Many practices are similar to traditional test case design *techniques*
  - Many practices are similar to more general testing *strategies*, heuristics, or rules of thumb

# IMT practices

## Overall strategies

- Exploratory
- Documentation based

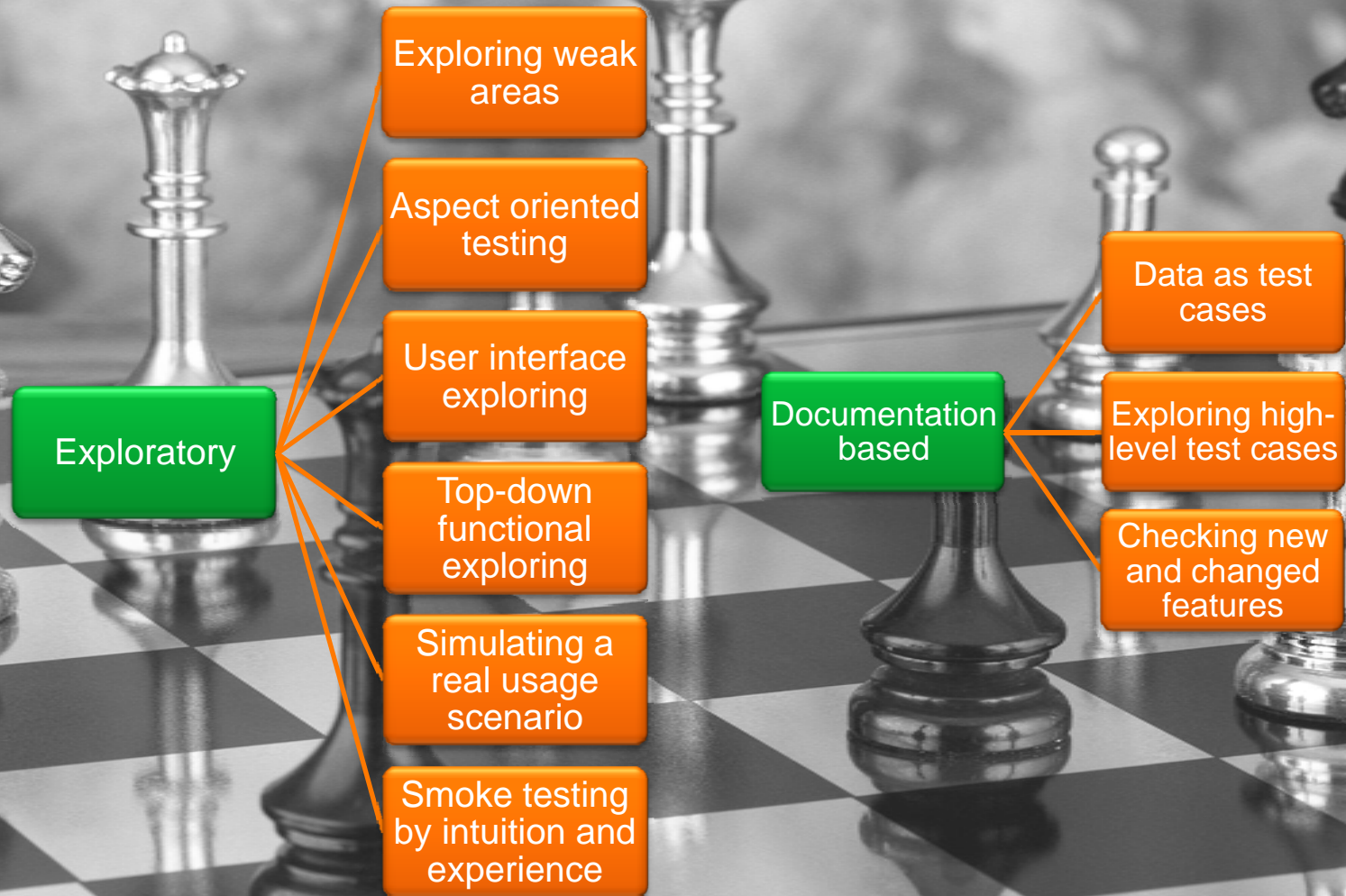


## Detailed techniques

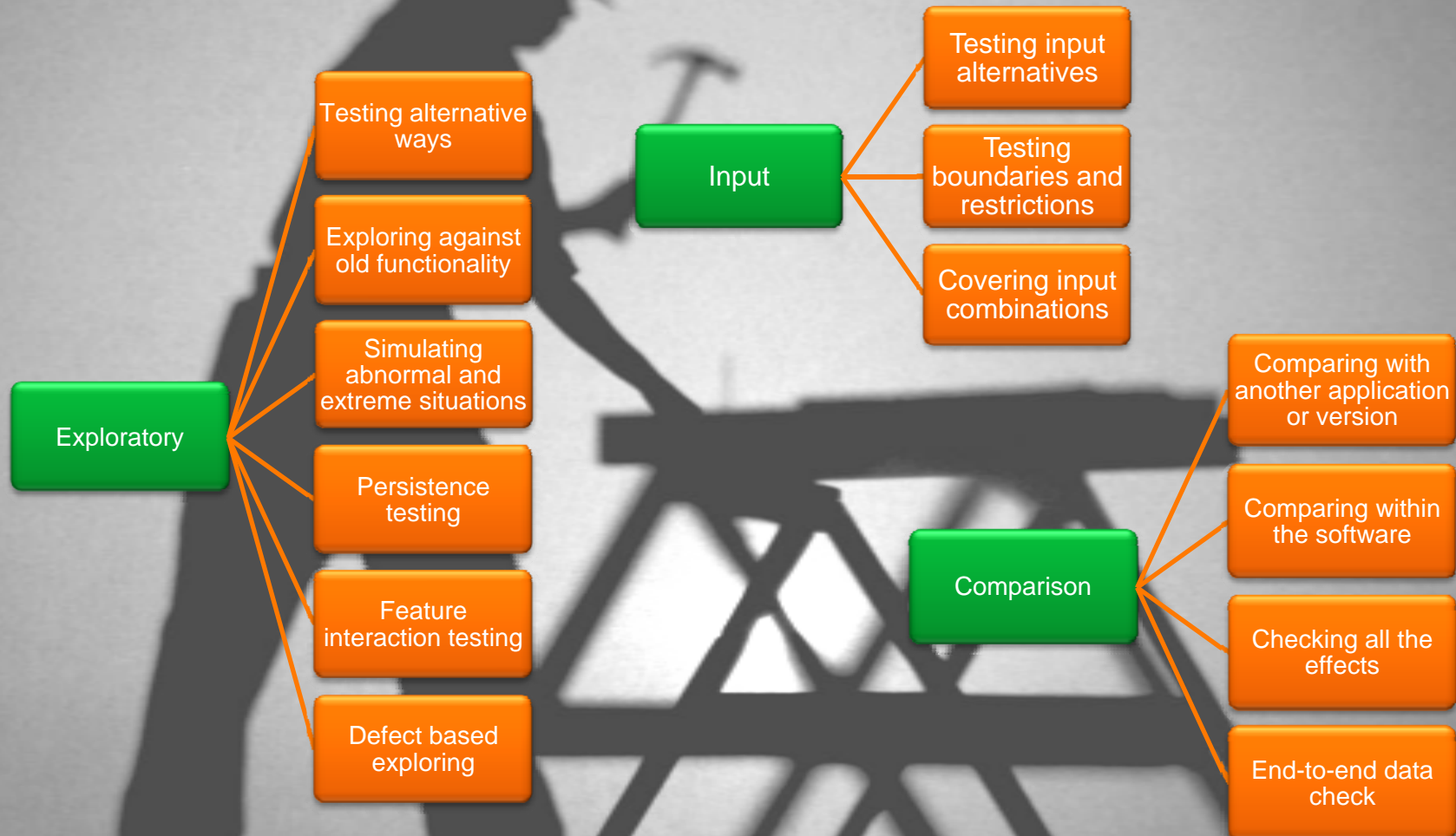
- Exploratory
- Input
- Comparison



# Overall strategies



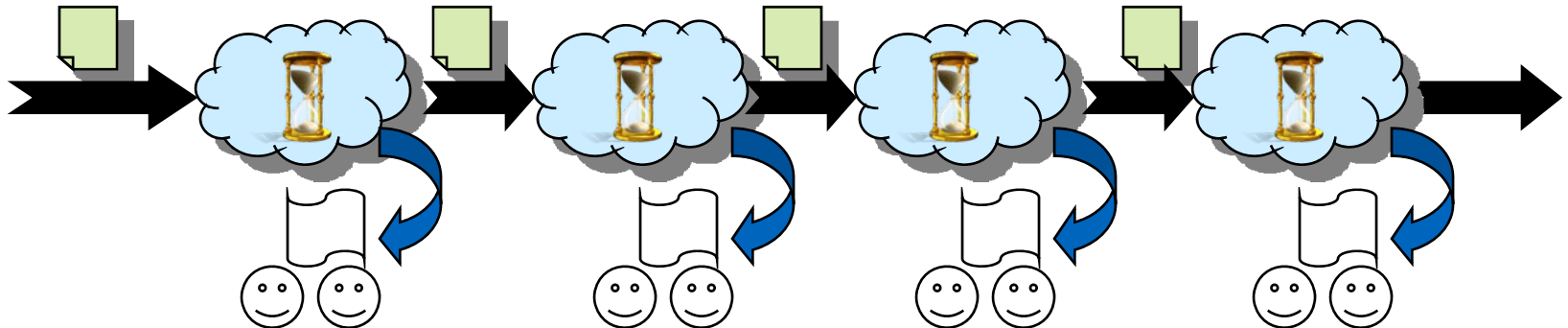
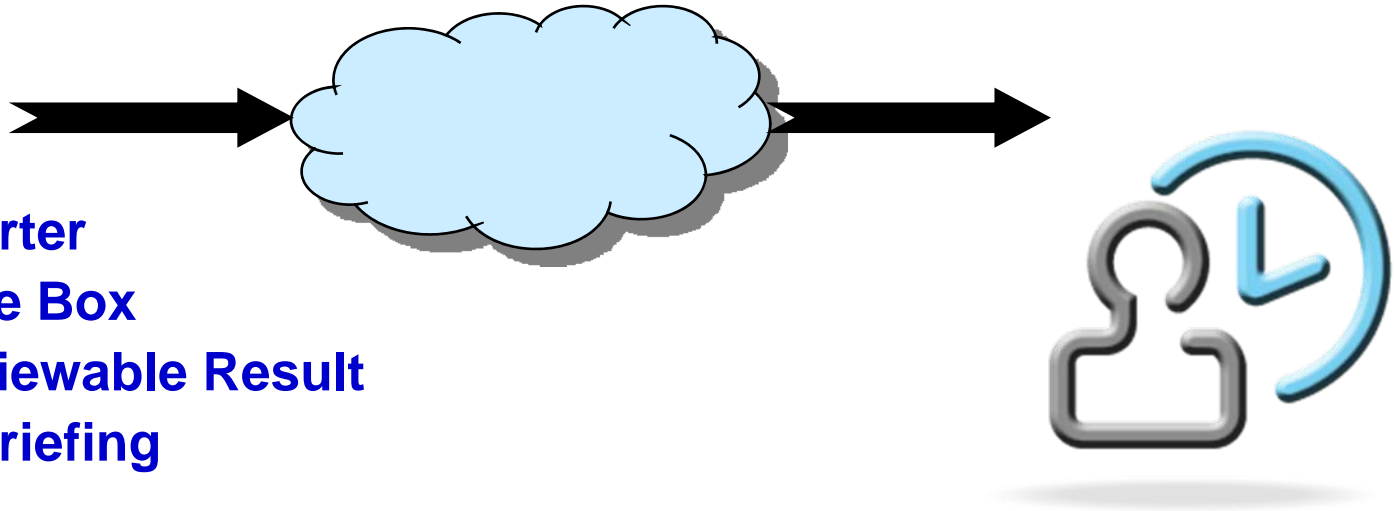
# Detailed techniques



# Session Based Test Management (SBTM)

Bach, J. "Session-Based Test Management", *STQE*, vol. 2, no. 6, 2000.

- Charter
- Time Box
- Reviewable Result
- Debriefing





# Session-Based Testing

## – Way to Manage ET

- Enables planning and tracking exploratory testing
  - Without detailed test (case) designs
  - Dividing testing work in small chunks
  - Tracking testing work in time-boxed sessions
- Efficient – no *unnecessary* documentation
- Agile – it's easy to focus testing to most important areas based on the test results and other information
  - Changes in requirements, increasing understanding, revealed problems, ...
- Can help getting testing done when resources are scarce

# Exploring like a tourist

## – The Tourist Metaphor

- Using different tours to guide testers' actions
  - Guidebook tour
  - Money tour
  - Landmark tour
  - Back alley tour
  - All-nighter tour
  - Coach potato tour
  - Lonely businessman tour
  - Saboteur tour
  - Antisocial tour
  - ...



James A. Whittaker. *Exploratory Software Testing, Tips, Tricks, Tours, and Techniques to Guide Test Design*. Addison-Wesley, 2010.

Examples

# SELECTED IMT PRACTICES

---

# Purpose clusters



*Exploring:* guiding exploratory testing



*Coverage:* managing test coverage



*Results:* deciding if the results were correct



*Defects:* detecting specific types of defects



*Prioritization:* selecting what to test

<exploratory strategy>

# Exploring weak areas



- **Description:** Exploring areas of the software that are weak or risky based on the experience and knowledge of the tester.
- **Goal:** Reveal defects in areas that are somehow known to be risky. Focus testing on risky areas.
  - complicated
  - coded in a hurry
  - lots of changes
  - coders' opinion
  - testers' opinion
  - based on who implemented
  - a hunch...



<exploratory strategy>

# Top-down functional exploring



- **Description:** Proceeding in testing by first going through typical cases and simple checks. Proceed gradually deeper in the details of the tested functionality and applying more complicated tests.
- **Goal:** To get first high level understanding of the function and then deeper confidence on its quality set-by-step.
  - Is this function implemented?
  - Does it do the right thing?
  - Is there missing functionality?
  - Does it handle the exceptions and special cases?
  - Does it work together with the rest of the system?
  - ...



<documentation based strategy>

# Using data as test cases



- **Description:** Pre-defined test data set includes all relevant cases and combinations of different data and situations. Covering all cases in a pre-defined test data set provides the required coverage.
  - Testing is exploratory, but the pre-defined data set is used to achieve systematic coverage.
  - Suitable for situations where data is complex, but operations simple. Or when creating the data requires much effort.
- **Goal:** To manage exploratory testing based on pre-defined test data. Achieve and measure coverage in exploratory testing.
- **Example:** Different types of customers in a CRM system.
  - User privileges
  - Situation, history, data, relationships
  - ...



<comparison technique>

# Comparing within the software



- **Description:** Comparing similar features in different places of the same system and testing their consistency.
- **Goal:** Investigating and revealing problems in the consistency of functionality inside software; help decide if a feature works correctly or not.





<input technique>

# Covering input combinations



- **Description:** Experience based identification of relevant combinations of inputs, outputs, data, etc. that affect the behaviour of the system.
- **Goal:** Systematic coverage of the combined effects of two or more related variables. Revealing defects related to the combined effects of variables.



# Ways of utilizing IMT Practices



Training testers



Guiding test execution



Test documentation and tracking



Test patterns for different situations

# Training Testers

- Testing heuristics and practices are good, experience based, knowledge for intelligent testers
- Named and documented
  - Give common terminology and names that can be used to discuss how the testing should be done
- By learning these practices a novice tester could do better job
  - Compared to just go and test around

# Guiding Test Execution

- Practices together with a high level test documentation can be used as a checklist
- A tester can choose applicable practices when doing exploratory testing
  - More conscious decisions
  - Better idea what the tester is actually doing
  - Easier to maintain focus – what am I going to achieve?

# Test Documentation and Tracking

- Testing practices can be used to write test specifications
  - No need for detailed descriptions for the tester
  - Tester knows what to do
  - Other people know what has been done
- Test planning and design can focus on high level structure and coverage issues
  - Not to teaching experienced tester how to test ;-)
  - Example:
    - Use *exploring high-level test cases* to cover the functionality
    - Apply *Testing input alternatives* and *Testing boundaries and restrictions* practices for each function
    - In addition, use *User interface exploring* practice to cover the functionality of all new dialogs

# Test patterns

- Testing practices could be further developed
- Testing pattern will provide set of good testing practices
  - For a certain testing problem and motivation
  - With a certain testing goal
  - Describing also the applicability (context) of the pattern

# Questions and more discussion



## Contact information

Juha Itkonen

[juha.itkonen@tkk.fi](mailto:juha.itkonen@tkk.fi)

+358 50 577 1688

<http://www.soberit.hut.fi/jitkonen>