Software investigation and exploration

-

Supercharging your value

Huib Schoots

De Agile Testers

Alex Schladebeck

BREDEX

What do you what to learn?

What is ET anyway?

- What is exploratory testing?
- What is the goal of ET?
- Who does it?

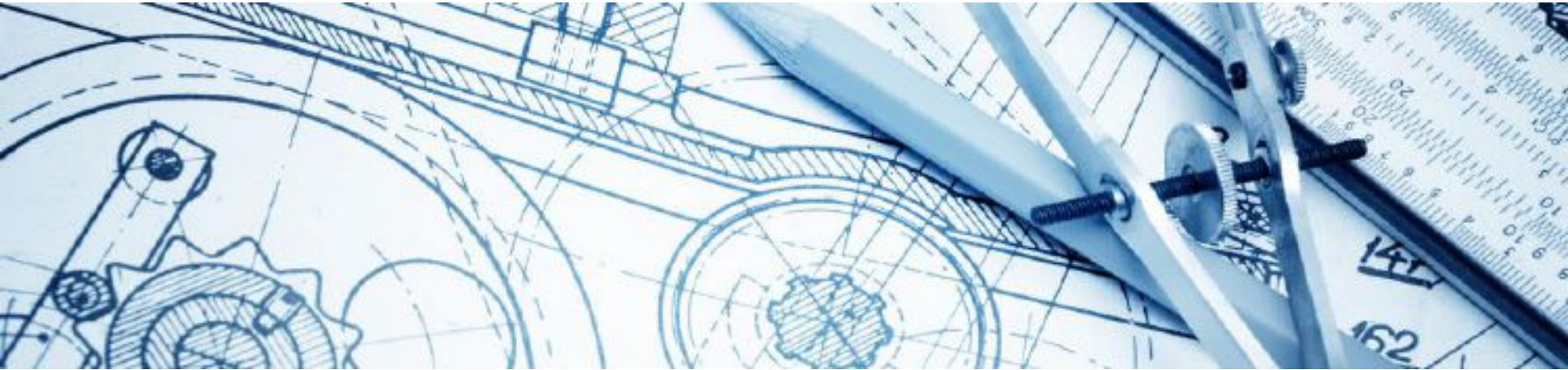
What makes ET hard?

Introduction

Goals for today

- Explain what we think (exploratory) testing is
- Learn by doing lots of fun exercises where we test real software.
- Learn essential skills like exploration, note taking, use of heuristics and chartering
- Learn to become a better tester, not by teaching you how to test, but by letting you experience what you do!
- You will learn to recognise what you do while you are testing. You can use that as a learning tool to get better at testing.

Software development



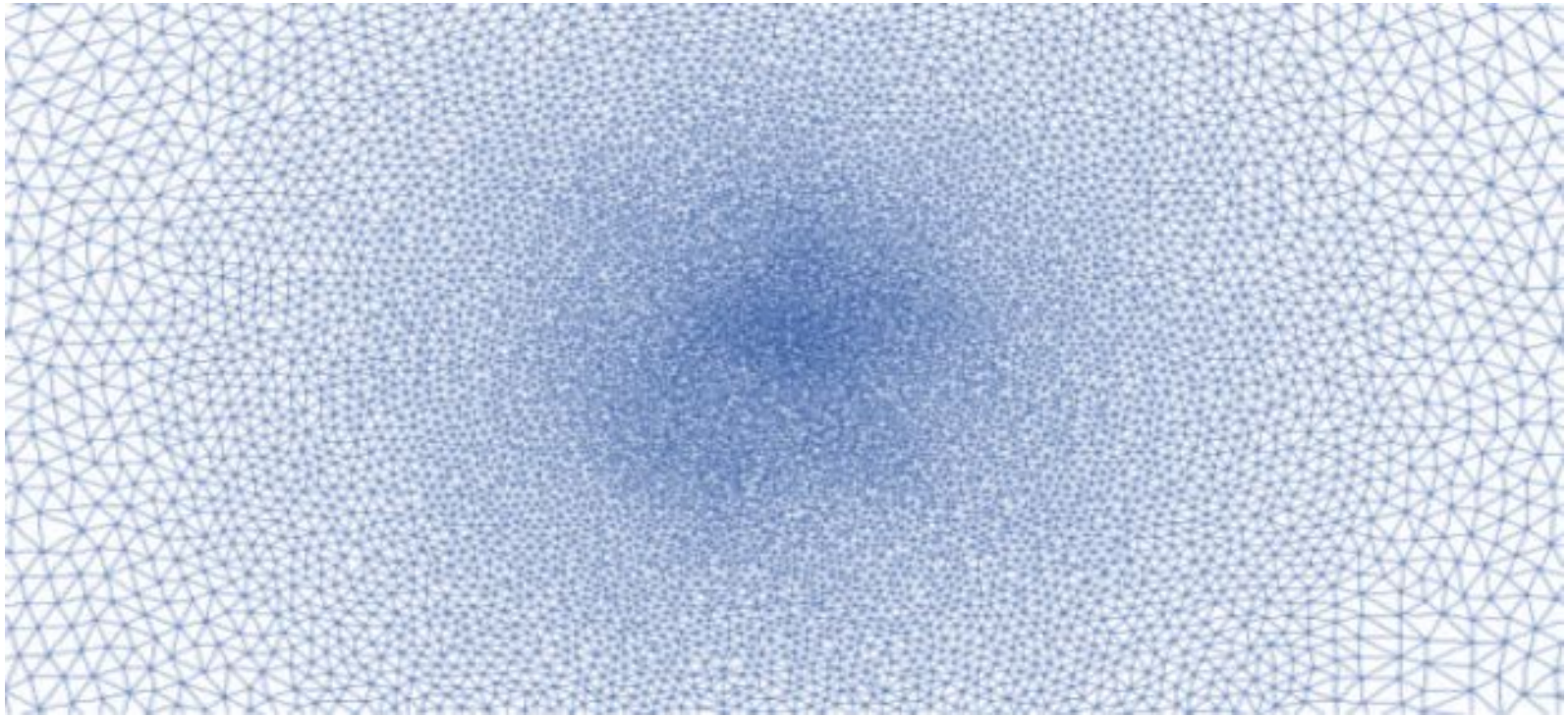
Software development

Factory

or

Research & development

Software is complex



What is the biggest risk in software development?

Unknown unknowns

- Capture everything upfront vs. building new insights
- Customers don’t know what they want
- Knowledge comes from experience and making decisions based on what is known (empirical process control theory)
- Cope with complexity, confusion, change, new insights and half answers
- Classic waterfall and "command & control" gets in the way of insight and serendipity

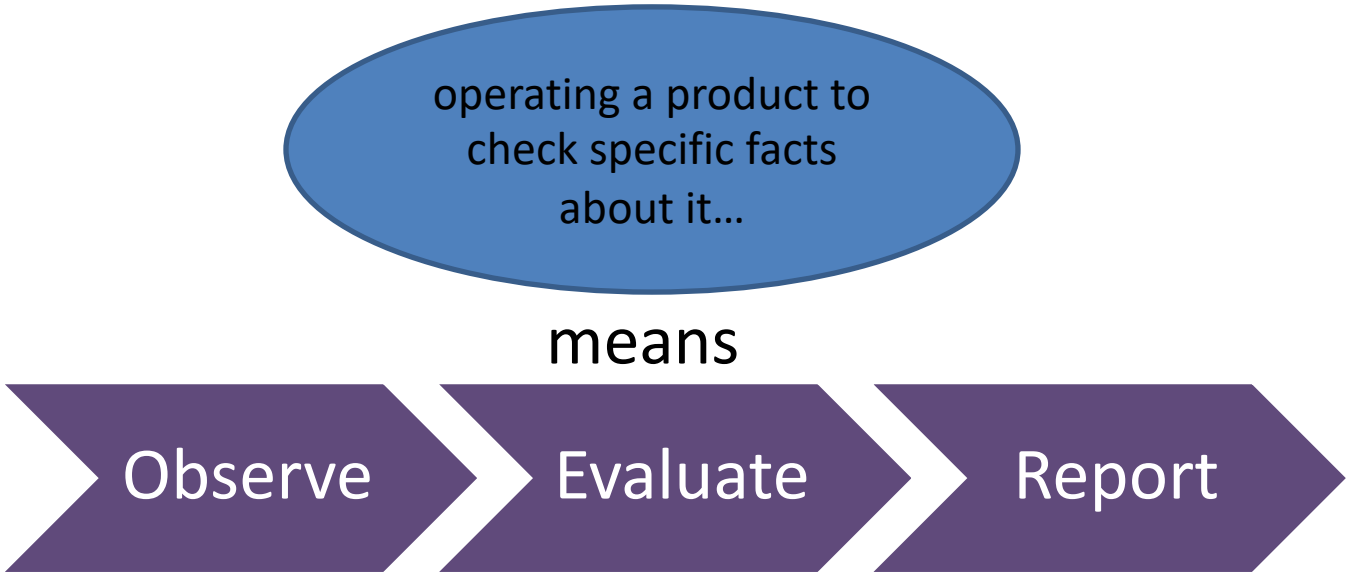
[EXPLORATORY TESTING] IS A THING THAT ALWAYS HAS TO HAPPEN BECAUSE WE CAN AUTOMATE AS MUCH AS WE LIKE BUT WE CAN ONLY AUTOMATE JUST RISKS WE HAVE SEEN AND THERE ARE ALWAYS GOING TO BE RISKS THAT WE HAVEN'T SEEN.

MY CURRENT FAVORITE EXAMPLE OF THIS IS: I WENT RUNNING IN A PARK VERY EARLY IN THE MORNING. OBVIOUSLY, I HAVE VARIOUS RISKS I CAN THINK ABOUT: I MIGHT INJURE MYSELF, I MIGHT GET LOST, I MIGHT GET ATTACKED. THOSE ARE THINGS I THOUGHT ABOUT. AS IT TURNED OUT, IN THE PARK THEY HAVE A PELICAN. I APPROACHED THIS PELICAN TO GET A PHOTO OF IT.

THE PELICAN MADE A THREATENING MOVE TOWARDS ME AND I RAN AWAY, TERRIFIED. ON THE LIST OF RISKS THAT I HAD IN THAT PARK ON THAT MORNING, BEING ATTACKED BY WILDLIFE WAS NOT ONE OF THEM. AND THAT IS ONE OF MY BEST EXPLANATIONS OF WHY WE NEED EXPLORATORY TESTING.
– ALEX SCHLADEBECK

(Exploratory) Testing

Call this “Checking” not Testing



Source: Rapid Software Testing by James Bach & Michael Bolton

A Check Has Three Elements

1. An *observation* linked to…
2. A *decision rule* such that…
3. both observation and decision rule can be applied algorithmically.

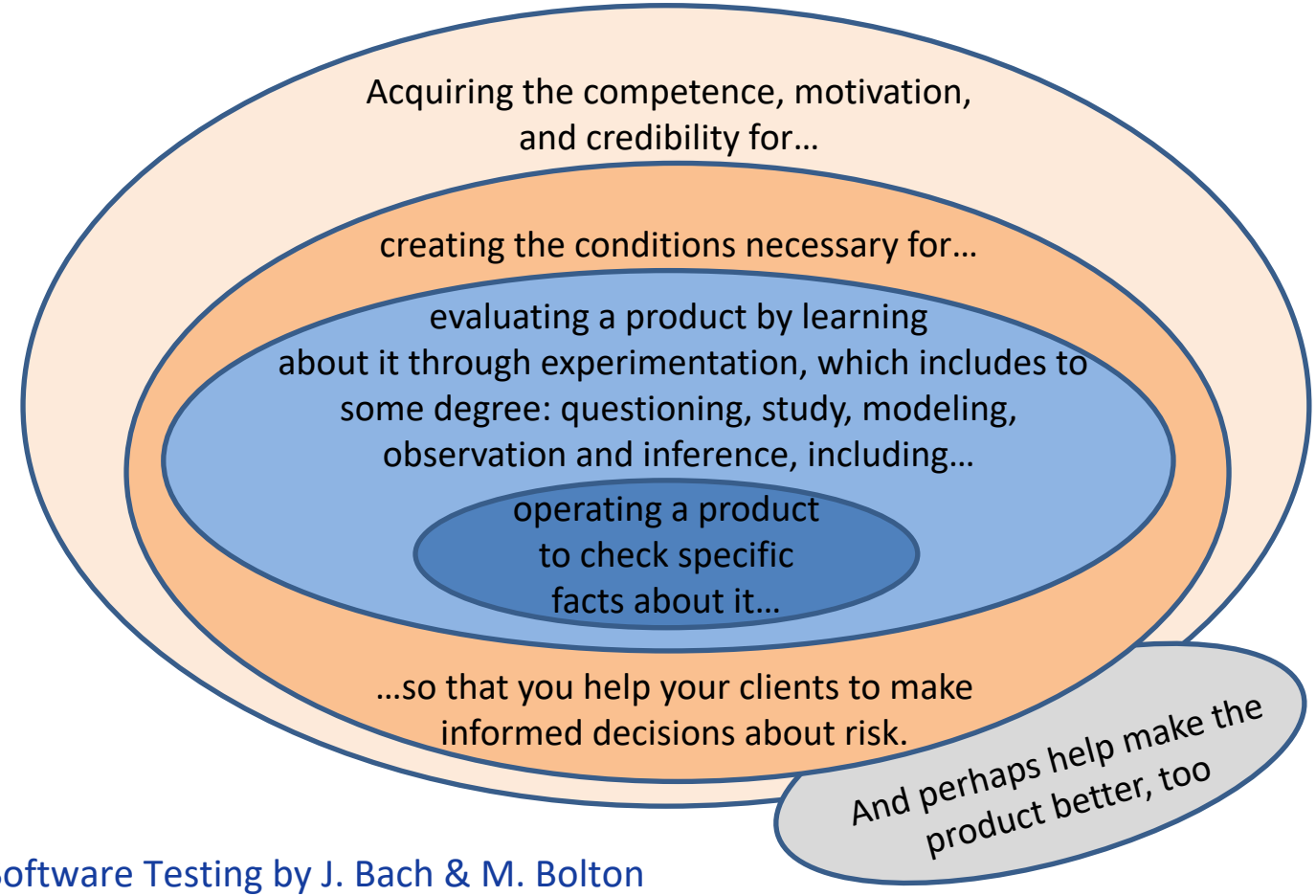
A ***check*** can be performed

by a machine
that *can't* think
(but that is quick and precise)

by a human who has been
instructed *not* to think
(and who is slow and variable)

Source: Rapid Software Testing by James Bach & Michael Bolton

Testing is…



Source: Rapid Software Testing by J. Bach & M. Bolton

Myths & misconceptions

- ET is unstructured
- ET is undocumented
- ET doesn’t provide accountability
- ET cannot be measured
- ET is for experienced people only
- Scripted testing is easier to do than ET
- No review in advance
- Hard to repeat
- Difficult to report test coverage

Exploratory Testing

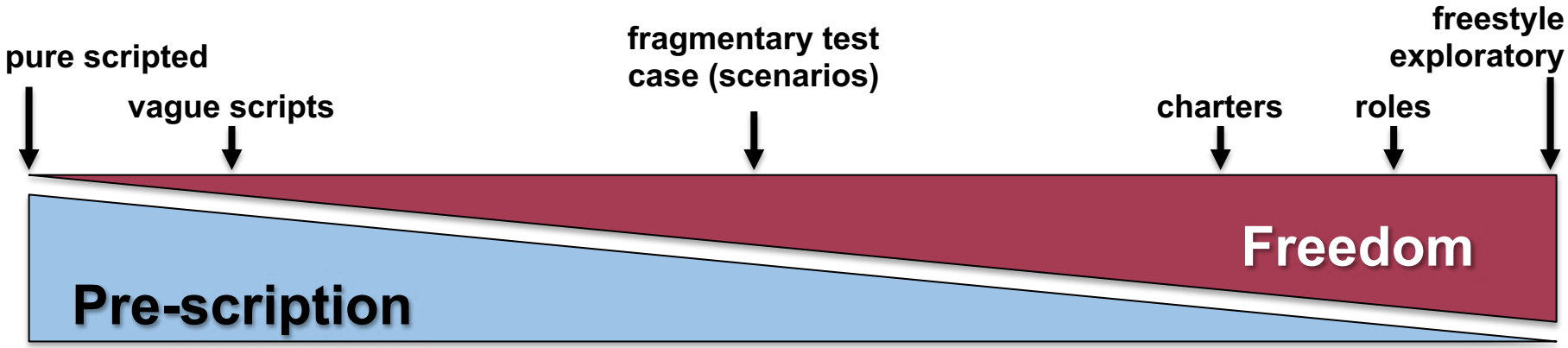
“a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of his/her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project”

“Exploratory testing is an approach to testing. It says whoever tests needs to be learning. Learning needs to change what you are doing.”

(Maaret Pyhäjärvi)

“If you can‘t define in advance what you must do – and first need to find out more about the problem to determine the next step, then you‘re exploring.“

(Alex Schladebeck)



Source: Telling Your Exploratory Story by Jon Bach

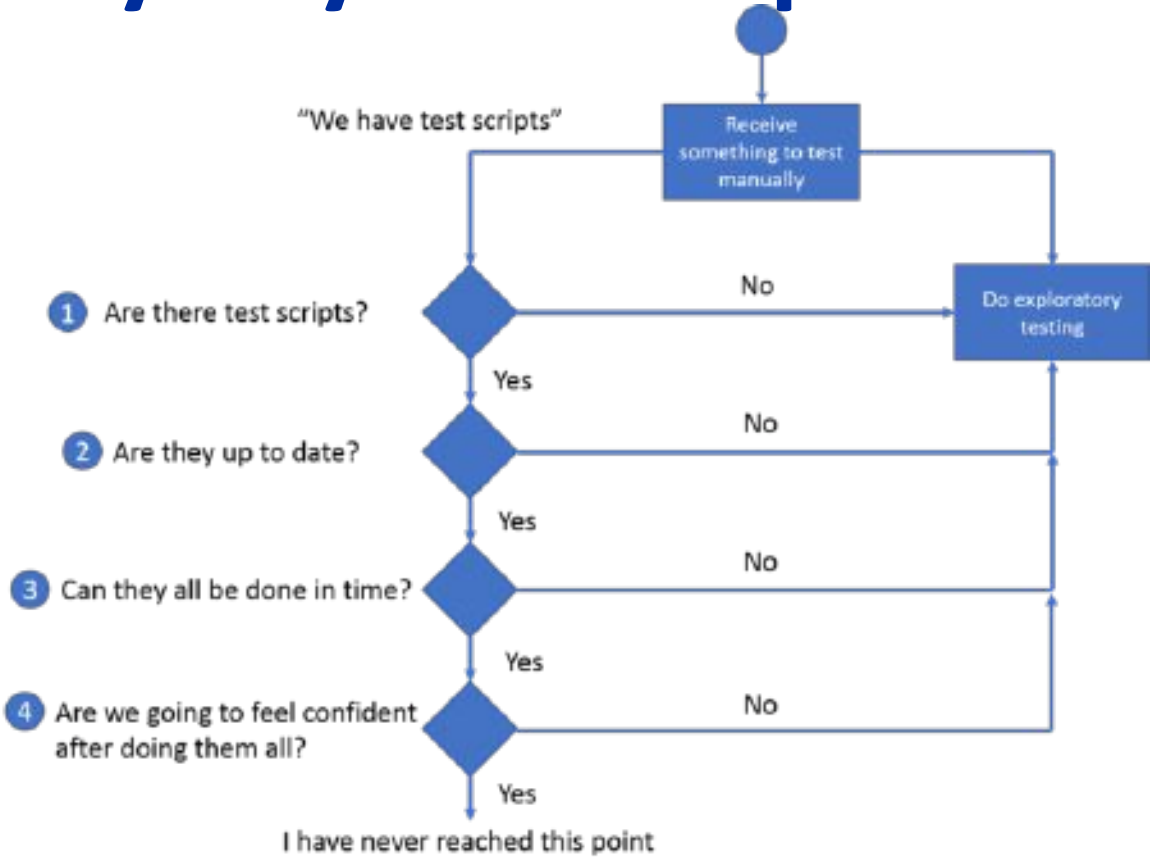
The structure of ET comes from many sources:

- Test design heuristics
- Chartering
- Time boxing
- Perceived product risks
- The nature of specific tests
- The structure of the product being tested
- The process of learning the product
- Development activities
- Constraints and resources afforded by the project
- The skills, talents, and interests of the tester
- The overall mission of testing

https://youtu.be/Ahg6qcgoay4

https://youtu.be/ubNF9QNEQLA

And anyway from experience…



Let’s Test

session 1

Session 1

Test the product for 30 min

Debrief

Structuring your testing

Product Coverage Outline

- Is an artifact (a map, list, diagram, sketch, table…) that identifies the dimensions or elements of a product that might be relevant to testing it

- The Product Elements section of the Heuristic Test Strategy Model (SFDIPOT) provides a point of departure for creating a coverage outline

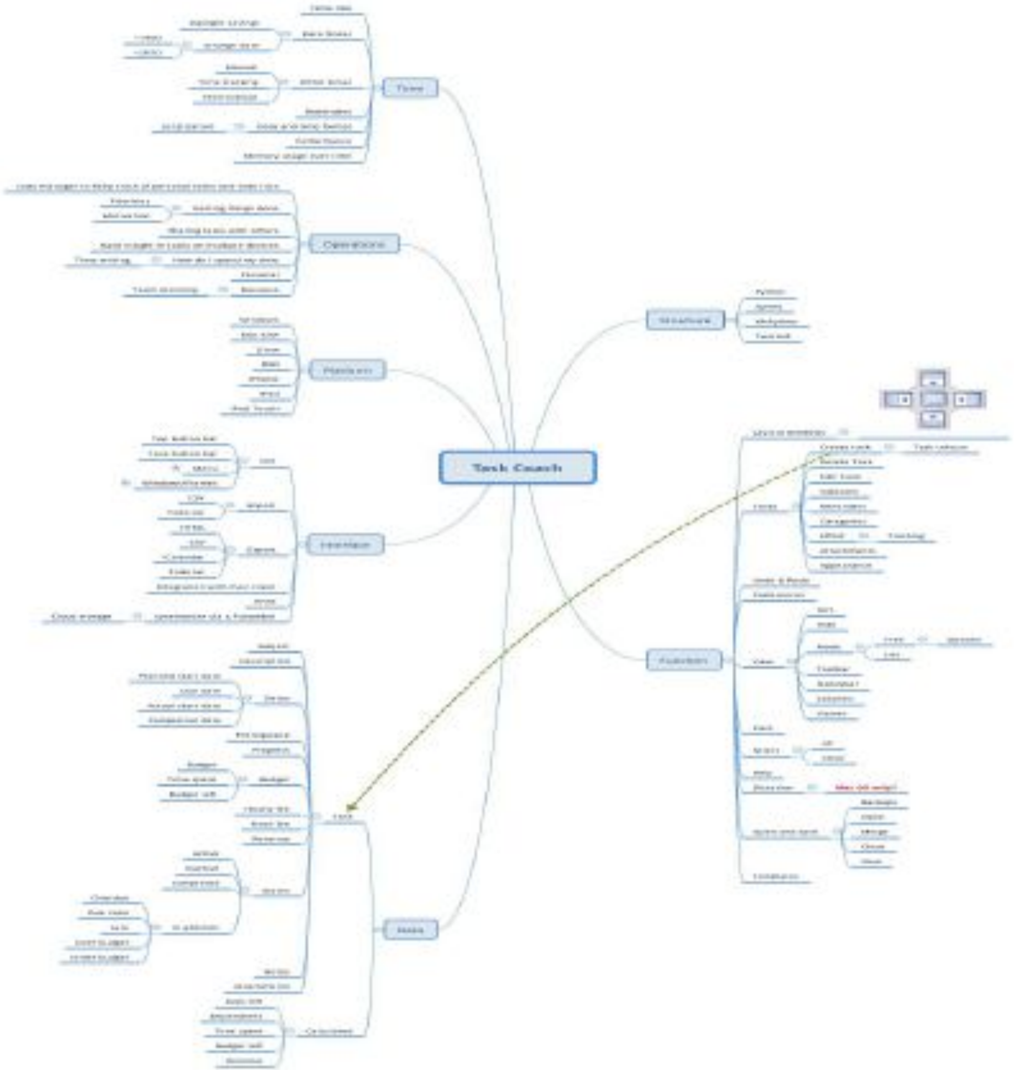
Product Elements: SFDIPOT

- Structure
- Function
- Data
- Interfaces
- Platform
- Operation
- Time

Source: https://www.satisfice.com/download/heuristic-test-strategy-model

What we did (and why)

- Explored the application with the intention to create an overview which is helpful for further testing
- Created a mind map based on set of ideas: SFDIPOT heuristics from the Heuristic Test Strategy Model
- Learned rapidly about the application by interacting with it and recording this in my mind map
- Creating a map has 3 goals:
    1. explore the application in a systematic way
    2. learn about useful aspects of the application
    3. A "map" with the elements of the product will help you with future test sessions and determine your test strategy.



Task Coach
Time
Operations
Platform
Interface
Data

Watch the video

http://youtu.be/NUojNfDjIjw

Session 2

Conduct a "survey session" to learn the product.

Tour the product to map every important function.

Using a mind map, model an overview of the product and start a provisional risk list.

Debrief

Sources & Heuristics

Sources & Heuristics

- How do we know what to test?
- How do we generate ideas?

Exercise

- What sources of information are there?
- What information can we get from them?

Debrief

LUNCH

Heuristics

- A heuristic is “a fallible method for solving a problem or making a decision”
- A heuristic is a mental shortcut that allows people to solve problems and make judgements quickly and efficiently. These rule-of-thumb strategies shorten decision-making time and allow people to function without constantly stopping to think about their next course of action.
- Snap judgement / rule of thumb

https://www.verywellmind.com/what-is-a-heuristic-2795235

Heuristics

- A product with more description is better than one without...
- Back pain → getting a cold?
- Red face → is there a problem?
- Pickle Jar (Katrina Clokie)
- In testing:
  - If something takes longer than expected → is there a problem?
  - Using back in web applications can cause problems
  - CRUD (create read update delete) is a heuristic for working with database applications
  - Interrupts in mobile testing are a good way of discovering problems
- Your examples...

Heuristics in testing

- **FEW HICCUPS** (Oracles: Consistency heuristics) http://www.developsense.com/blog/2012/07/few-hiccupps/
- **Testing Heuristics Cheat Sheet** (https://bit.ly/1or9Inn)
- **Goldilocks** (Elisabeth Hendrickson)
- **Bad strings** (https://github.com/minimaxir/big-list-of-naughty-strings)
- **Falsehoods programmers believe** (https://spaceninja.com/2015/12/08/falsehoods-programmers-believe/)
- **SFDIPOT** - Heuristic for product elements
- **RCRCRC for regression testing** (http://karennicolejohnson.com/2009/11/a-heuristic-for-regression-testing/)

Other tools

- Personas
- Test Tours: Heuristics to guide exploratory testing
  http://www.developsense.com/blog/2009/04/of-testing-tours-and-dashboards/ and
  http://test-able.blogspot.com/2013/11/web-testing-exploratory-tours.html
- Software Quality Characteristics
  http://thetesteye.com/posters/TheTestEye_SoftwareQualityCharacteristics.pdf

Exercise: use heuristics

Create 4 groups. Each choose a heuristic:

- Cheat sheet
- Personas
- Tours
- Quality Characteristics

Find out what the heuristic is and how you could use it in your testing (15 min)

Present your findings to the group (5 min)

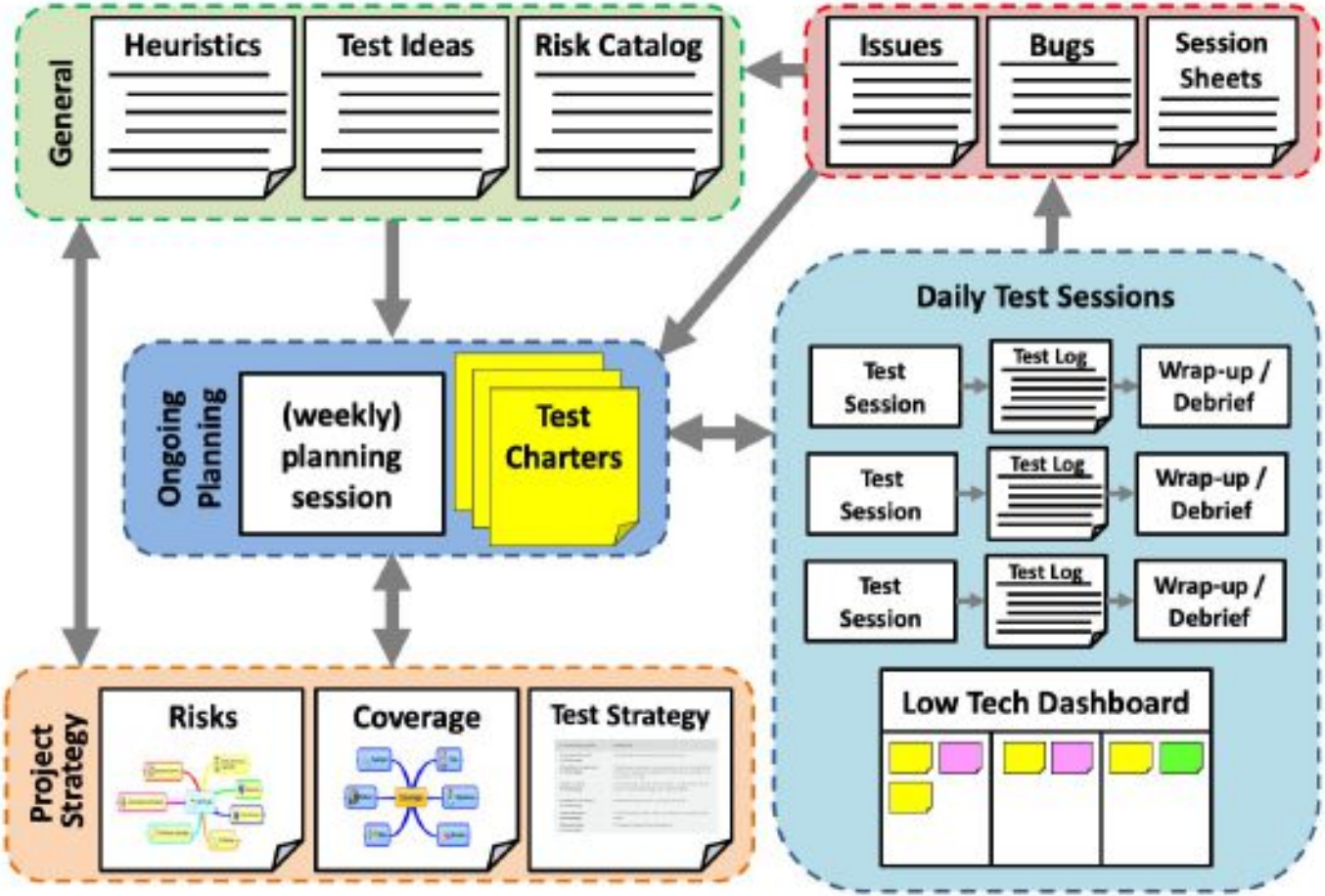
Debrief

TEST IDEAS

How to come up with Test Ideas?

http://erik.brickarp.se/2016/08/how-to-come-up-with-test-ideas.html

Session Based Test Management


General
Heuristics
Test Ideas
Risk Catalog
Issues
Bugs
Session Sheets
Daily Test Sessions
Ongoing Planning
(weekly) planning session
Test Charters
Test Session
Test Log
Wrap-up / Debrief
Test Session
Test Log
Wrap-up / Debrief
Test Session
Test Log
Wrap-up / Debrief
Low Tech Dashboard
Project Strategy
Risks
Coverage
Test Strategy

Test Charters

Explore (target)
With (resources)
To discover (information)

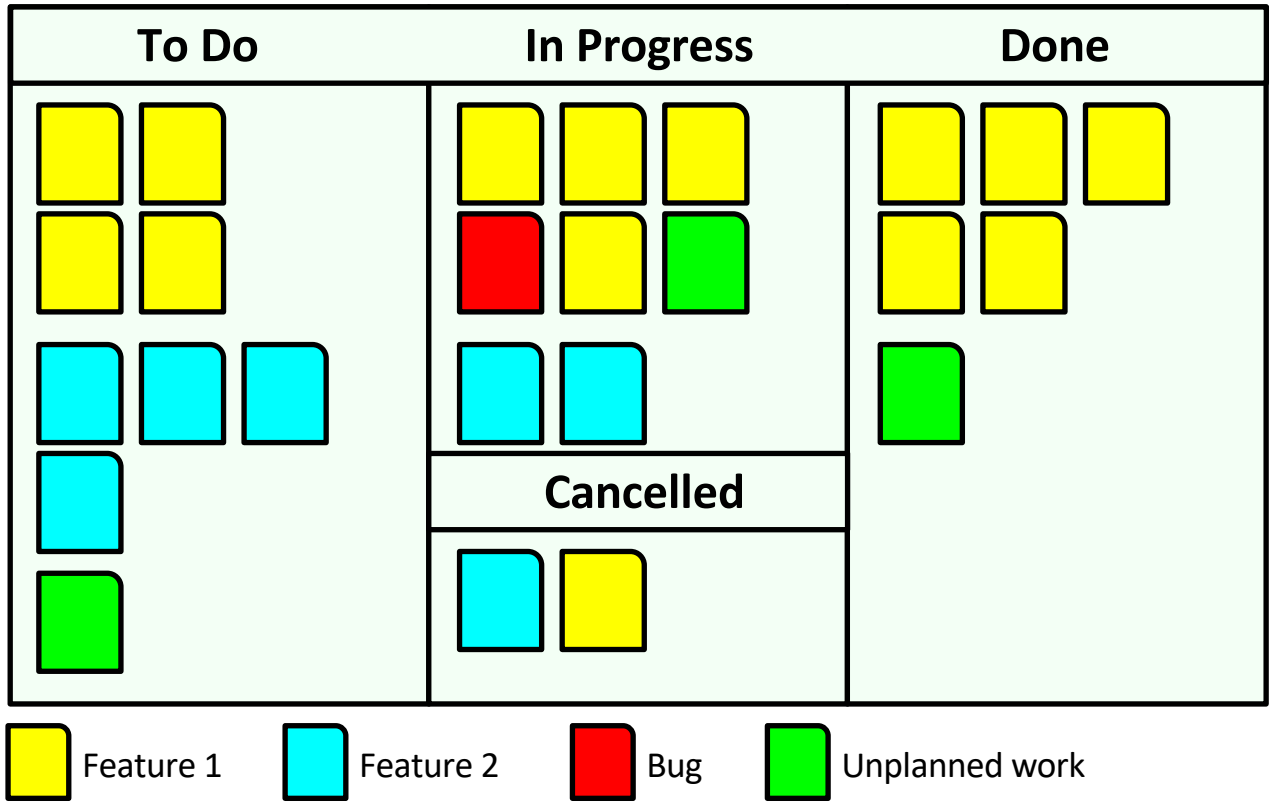
Explore Application X import menu. Identify important features with the goal of developing a coverage outline and a risk list.

My mission is to test <insert risk here> to <insert coverage here>

Read Chapter 4 of the product specification. Prepare a mind map, and discuss it with Peter (programmer) and David (architect).

The charter is a one- to three-sentence mission for a testing session

Dashboard


To Do
In Progress
Done
Cancelled
Feature 1
Feature 2
Bug
Unplanned work

Charter Patterns:
Evolving test strategy

- **Intake Sessions** (Goal: negotiate mission)
  - "Interview the project manager about testing Xmind."
- **Survey Sessions** (Goal: learn product)
  - "Familiarize yourself with Xmind."
- **Setup Sessions** (Goal: create testing infrastructure)
  - "Develop a library of mindmaps for testing Xmind."
- **Analysis Sessions** (Goal: get ideas for deep coverage)
  - "Identify the primary functions of Xmind."
  - "Construct a product coverage outline."
  - "Brainstorm test ideas."
  - "Prepare a state model for state-based testing."
  - "Perform a component risk-analysis to guide further testing."
  - "Discover all the error messages in Xmind."

Source: Rapid Software Testing by James Bach & Michael Bolton

Charter Patterns:
Evolving test strategy

- **Deep Coverage Sessions** (Goal: find the right bugs)
  - “Perform scenario testing based on the scenario playbook.”
  - “Perform a tour that achieves double-transition state coverage.”
  - “Perform steeplechase boundary testing on the major data items.”
  - “Test each error message in Xmind.”
  - “Perform a function tour using the 2300 node mindmap.”
- **Closure Sessions** (Goal: get ready to release)
  - “Verify the latest fixes.”
  - “Re-test tutorial with the latest build.”
  - “Review help files and readme.”
  - “Go over deferred bugs with Customer Support people.”
  - “Perform clean-machine install test.”

Source: Rapid Software Testing by James Bach & Michael Bolton

Let’s Test

Create charters & Test them

Exercise: create charters

Create a list of at least 5 charters

Debrief

Exercise: execute charter

Select a charter for deep testing

Execute a deep testing session on this charter

Debrief - PROOF

- **Past**: What happened during the session?
- **Results**: What was achieved during the session?
- **Outlook**: What still needs to be done?
- **Obstacles**: What got in the way of good testing?
- **Feelings**: How does the tester feel about all this?

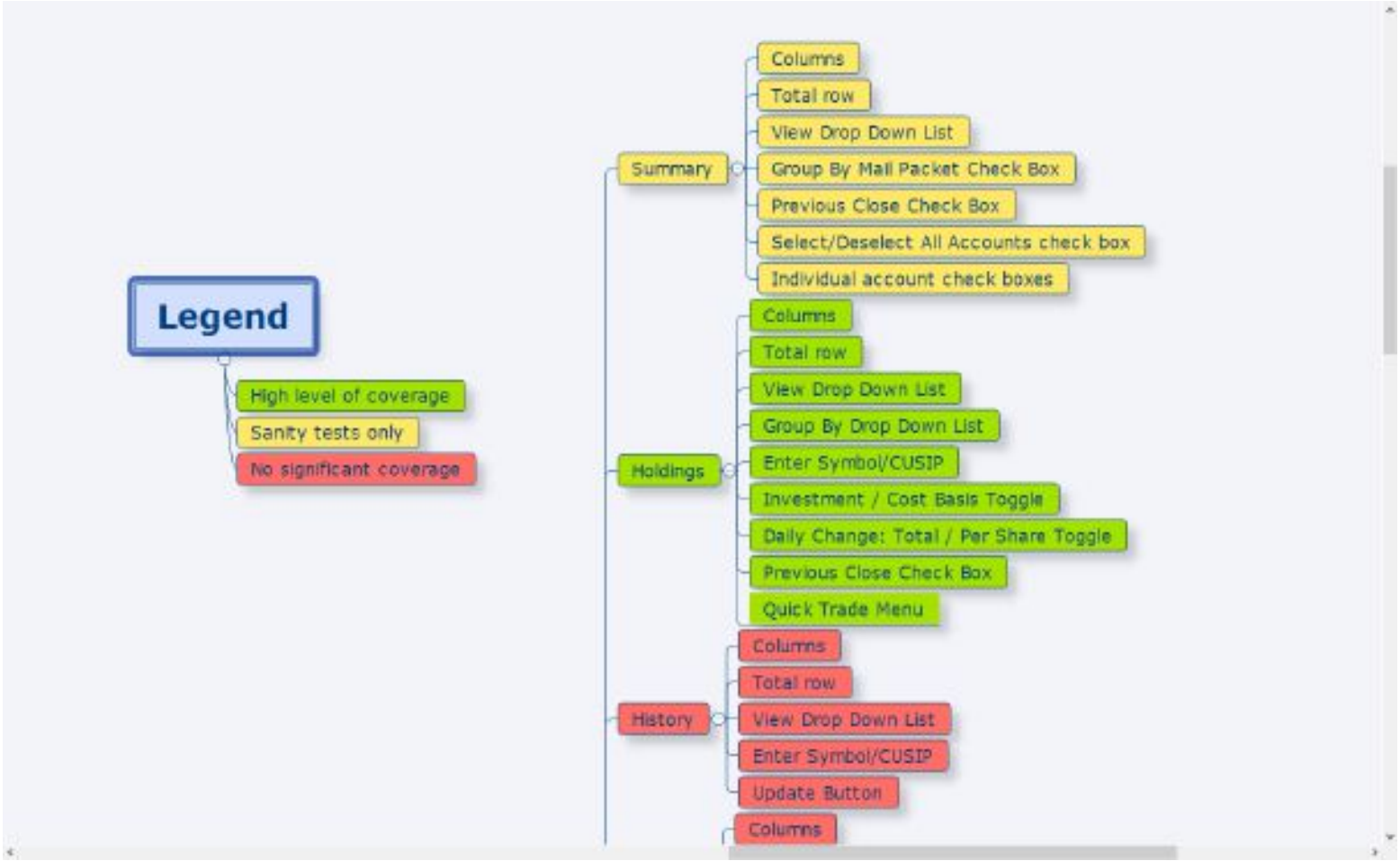
- Discuss session and ask questions: new sessions may be chartered
- Discuss session sheet to assure that both understand it.
- Coach & learn!

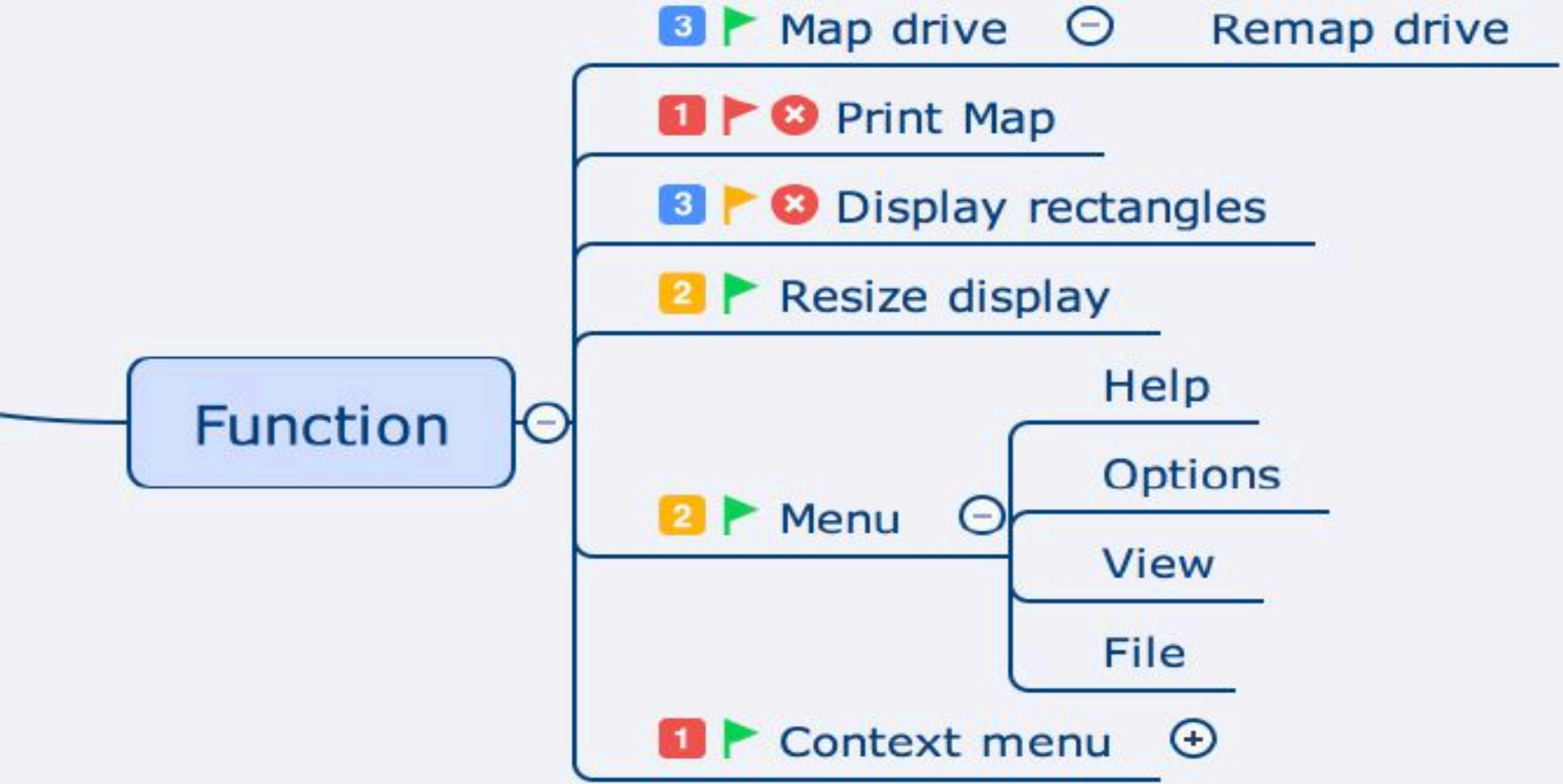
Exercise: debrief each other

Do a debrief of the last session and use PROOF.

Debrief

Feature Area Coverage




Function
3 Map drive
Remap drive
1 Print Map
3 Display rectangles
2 Resize display
2 Menu
Help
Options
View
File
1 Context menu

Wrap-up

Why ET is powerful

- Do what really needs to be done
- It facilitates experimentation, serendipity, discovery and learning
- It finds more bugs
- Create engagement: help people use their brains
- Take advantage of tacit knowledge and skill
- Use insights from experiments to inform the next
- Using the full creative power of exploration

Challenges?

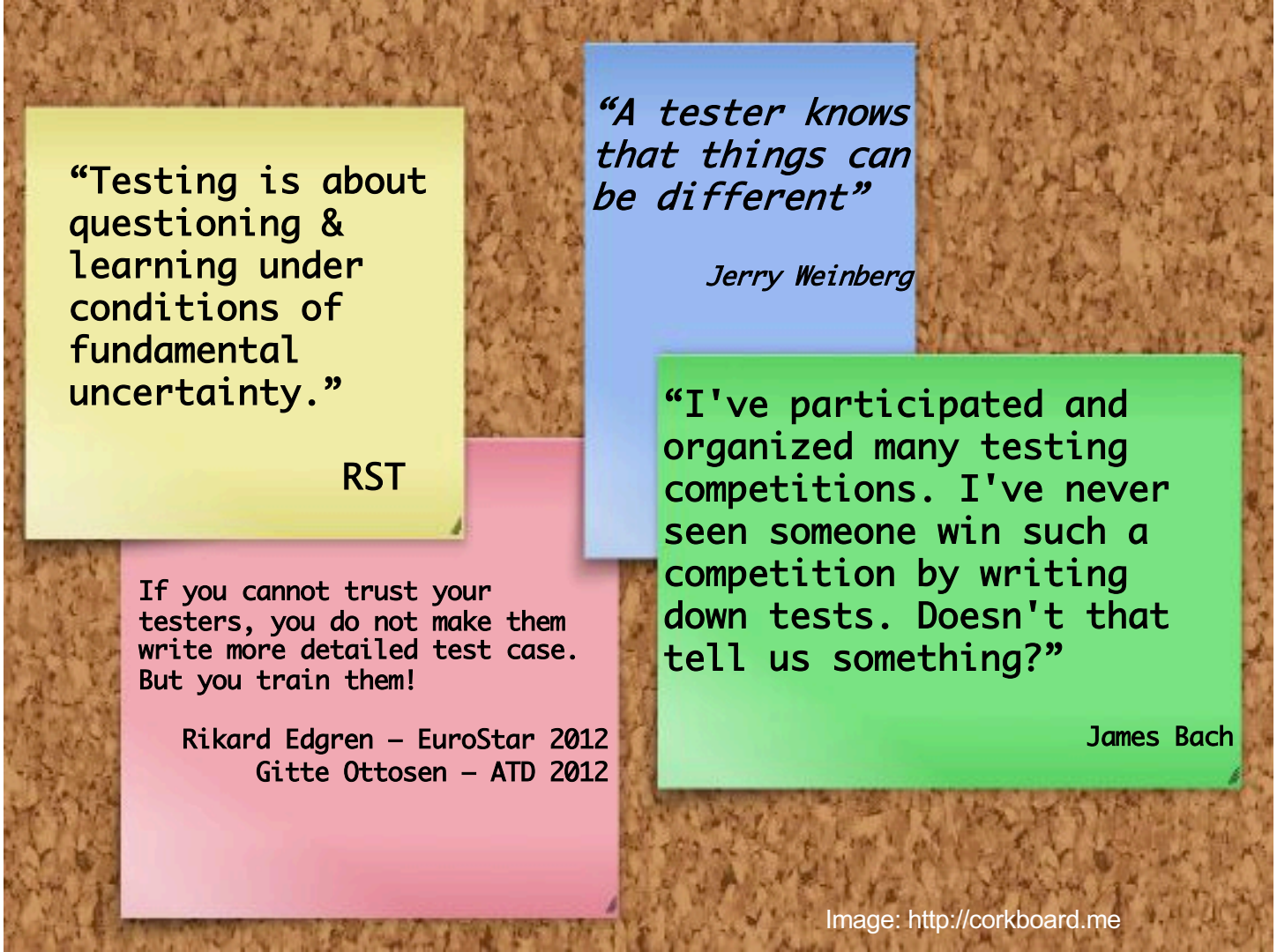
Exploratory Testing = (like testing in general) is not easy and needs (a lot) training and practice

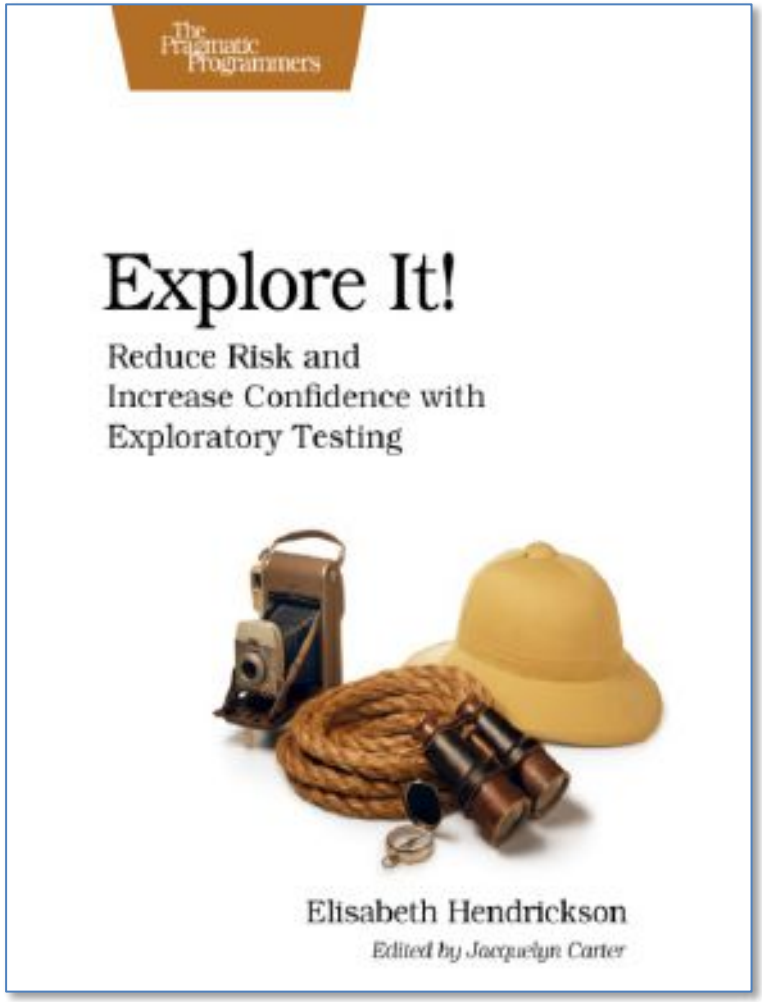
Often seen as most difficult:

- Note taking
- Test ideas
- Coverage reporting
- Managing ET

How to master ET?

- Just do it!
- Practice, practice and practice some more…
- Pair with colleagues
- Debrief & retrospect
- Testing Dojo
- TestOpsy
- Train creating test ideas fast
- Learn more about observation, experiments, biases, social science, ...


"Testing is about questioning & learning under conditions of fundamental uncertainty."
RST
"A tester knows that things can be different"
Jerry Weinberg
"I've participated and organized many testing competitions. I've never seen someone win such a competition by writing down tests. Doesn't that tell us something?"
James Bach
If you cannot trust your testers, you do not make them write more detailed test case. But you train them!
Rikard Edgren – EuroStar 2012
Gitte Ottosen – ATD 2012
Image: http://corkboard.me


The Pragmatic Programmers
Explore It!
Reduce Risk and
Increase Confidence with
Exploratory Testing
Elisabeth Hendrickson
Edited by Jacquelyn Carter

Value

QUESTIONS, YOU HAVE, HMM?

Testopsies

What?

autopsy (n): a critical examination or dissection of a subject or work

testopsy (n): an autopsy of a testing session

Testopsy

A testopsy* is an examination of testing work, performed by watching a testing session in action and evaluating it.

Testopsies can help in training, assessment, and developing testing skill for novices and experienced testers alike.

* The term testopsy is coined by James Bach

The Basic Idea

- Observe a testing session (your own or somebody else)
- Become aware of something interesting or complicated
- Name it and make it explicit
- Analyze it
  - When do you need to do it? When do you need to avoid it?
  - Do we like it? Do we want it?
- Close the loop intentionality:
  - Intend it - Do it - Explain it - Justify it

How to do a Testopsy at home?

1. Record a session of your testing
2. Go through the recording and note every single activity that you did. Put specific words to each activity
3. Explain why you did what you did

You can do this for a 10 minute session or a two-hour session. We feel that very short sessions that are rich in product learning and test design are the most interesting to study.

What does a Testopsy bring?

1. Learn new skills
2. Discover what you need to practice
3. Improve your skills like:
    - Teaching
    - Narrating/framing
    - Observing
4. Understand (and even appreciate) yourself
5. Ability to explain what you’re doing

How about a team exercise?

1. Show off your amazing skills :-)
2. Have more fun while pairing (coaching colleague tester)
3. Teach testing to your colleagues
4. Learn from your colleagues
5. Promote deeper understanding of testing
6. Finding commonalities in thinking patterns

References ET

- RST Appendices
  http://www.satisfice.com/rst-appendices.pdf
- “Evolving Understanding of Exploratory Testing” en “Structures of Exploratory Testing”
  http://www.developsense.com/resources.html
- Test cases are not testing: towards a culture of test performance by James Bach & Aaron Hodder
  http://www.testingcircus.com/testing-trapeze-2014-february-edition/
- Collected resources:
  http://www.huibschoots.nl/links
- Considerations when testing a software application in a context-driven way
  http://www.huibschoots.nl/wordpress/?p=2781

References Test Ideas

- Little black book on Test Design
  http://www.thetesteye.com/papers/TheLittleBlackBookOnTestDesign.pdf
- Collected resources on test ideas:
  http://www.huibschoots.nl/wordpress/?page_id=441#ideas
- Test Insane – Testing Mind maps
  http://apps.testinsane.com/mindmaps/

References Testopsy

- Exploratory Testing Skills & Dynamics (in RST Appendices) – http://www.satisfice.com/rst-appendices.pdf
- Skills mind map - http://goo.gl/VCQ0IN
- Podcast explaining Testopsy - http://www.qualitestgroup.com/The-Testing-Show/testopsies/
- Report of a Testopsy - http://patternsofproof.wordpress.com/2015/03/07/on-performing-an-autopsy/
- Report of a Testopsies workshop - http://www.brendanconnolly.net/testopsies/

References PCO

- Task Coach - http://taskcoach.org/
- Heuristic Test Strategy Model - http://www.satisfice.com/tools/htsm.pdf
- Rapid Software Testing - https://rapid-software-testing.com/
- Presentation on Test Coverage Outline - http://www.stickyminds.com/conference-presentation/test-coverage-outline-your-testing-road-map
- Experience report on using a Product Coverage Outline - http://prairietester.blogspot.nl/2013/09/monday-product-coverage-outlines.html
- Testing Story: http://www.developsense.com/blog/2012/02/braiding-the-stories/
- Example PCOs in Xmind: https://www.dropbox.com/sh/r769hj8dh498hfw/AAA48jaUb6moTB0rzucgSWpsa?dl=0

Other references

- Tacit and Explicit Knowledge and Exploratory Testing - http://steveo1967.blogspot.nl/2013/06/tacit-and-explicit-knowledge-and.html
- Shapes of Actions - http://www.developsense.com/blog/2011/12/shapes-of-actions/
- Testing Unexplained - http://goo.gl/nG0RZ6
- ET with Subtitles (video) - http://youtu.be/Vy0I2SB5OLo

alex.schladebeck@bredex.de
@alex_schl
http://www.schladebeck.de/
http://www.bredex.de

huib@deagiletesters.nl
@huibschoots
http://www.huibschoots.nl
http://www.deagiletesters.nl