expo QA 18

MADRID 4th - 6th JUNE 2018

Huib Schoots

Improve
QUALITY SERVICES

Exploratory Testing explained

Introduction

Who am I?

- Context-driven software tester
- Rapid Software Testing teacher
- Agile humanist, Scrum master, agile coach
- Curious & lifelong learner
- Passionate & energetic people lover
- Trainer, coach, writer, speaker, leader
- Books & Apple gadget collector
- Trombone and guitar player
- Aspiring magician
- Gamer
- Beer brewer
- STAR WARS & LEGO freak







Goals for today

- Explain what I think (exploratory) testing is
- Learn to become a better tester, not by teaching you how to test, but by letting you recognize what you do when testing!
- You can use that as a learning tool to get better at testing

Software Development

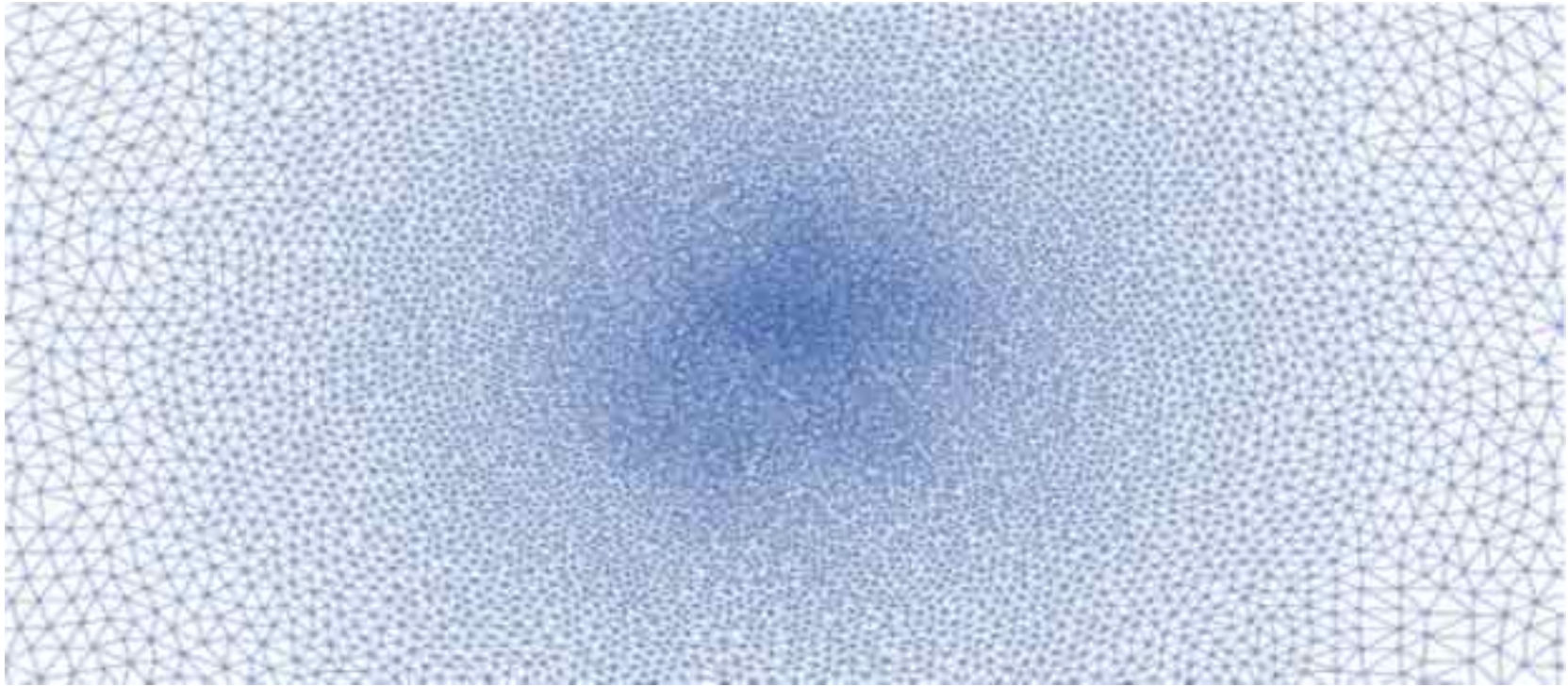
Software development

Factory

or

Research & development

Software is complex



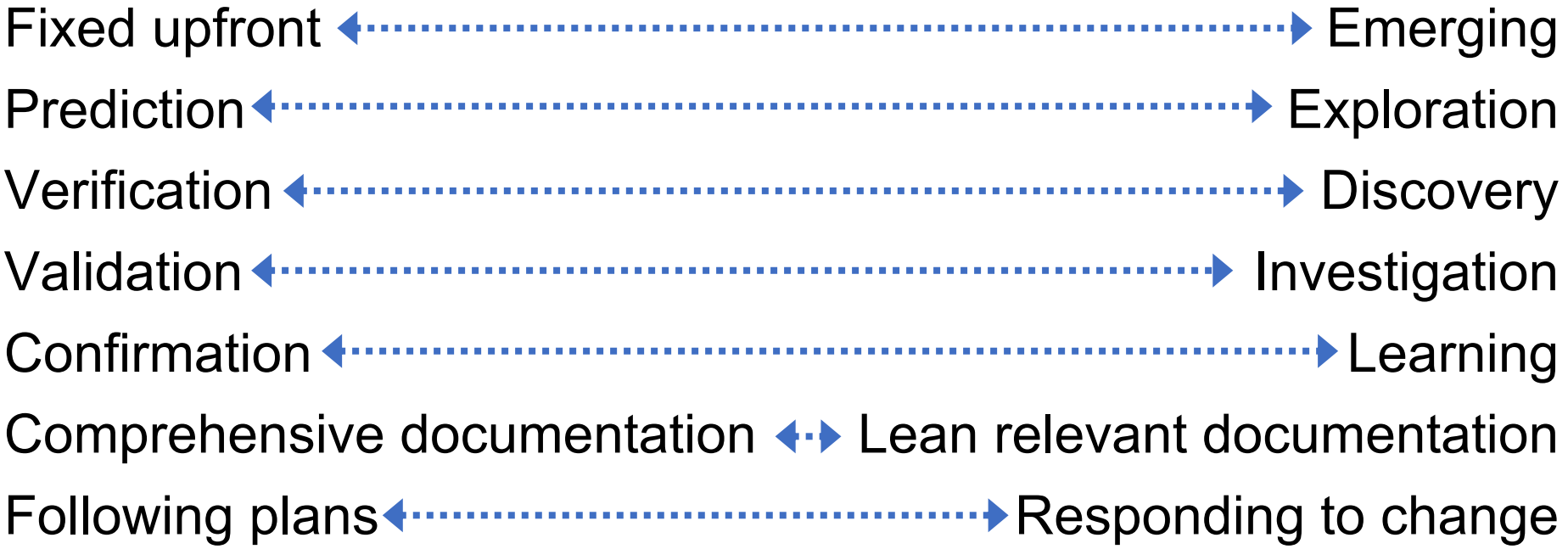
What is the biggest risk in software development ??

The risk of overlooking or missing important risks!!

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" get in the way of insight and serendipity

Why learning is important!



How do you learn?

If you don’t know what activities, tactics or skills you are using (or could use), how can you learn or train them?

Learn by experience:

- Concrete, challenging & achievable tasks
- Realistic application, processing & reflection
- Personal interpretation, exchange with others & constructive feedback
- Safe environment to experiment & make mistakes

What do you do when you test?

Unplanned… unanticipated… unspoken work…

(Exploratory) Testing

Myths & misconceptions

- Unstructured
- Undocumented
- Doesn’t provide accountability
- Cannot be measured
- For experienced people only
- Scripted testing is easier to do
- No review in advance
- Hard to repeat
- Difficult to report test coverage

Exploratory Testing

“ An approach to software testing that emphasizes the personal freedom and responsibility of each tester to continually optimize the value of his work by treating learning, test design and test execution as mutually supportive activities that run in parallel throughout the project. ”

“ Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next. ”

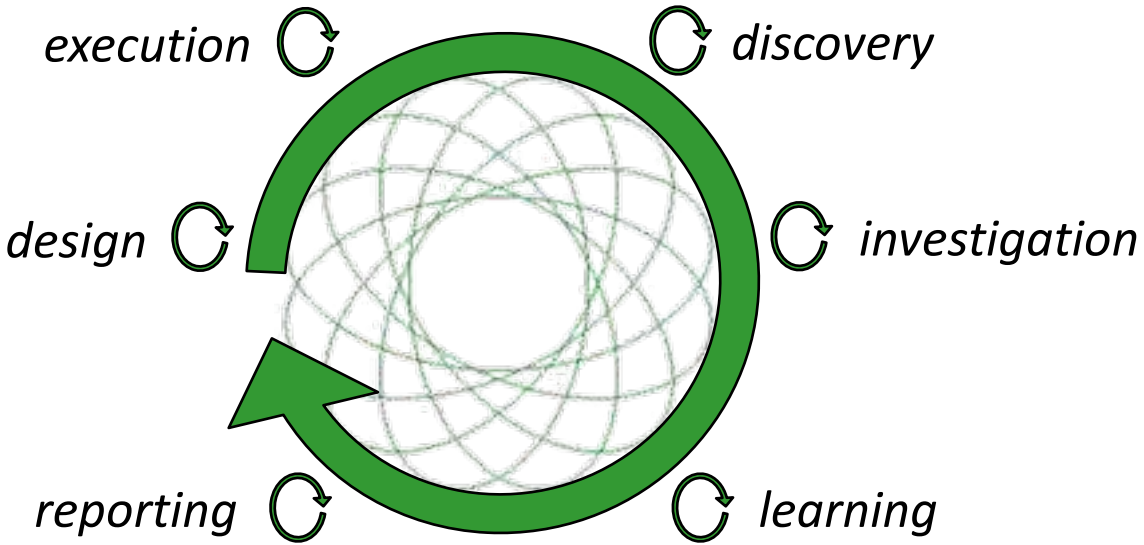
Exploratory Testing

“ An approach to software testing that emphasizes the personal freedom and responsibility of each tester to continually optimize the value of his work by treating learning, test design and test execution as mutually supportive activities that run in parallel throughout the project. ”

“ Simultaneously designing and executing tests to learn about the system, using your insights from the last experiment to inform the next. ”

Exploration: searching for value and risk

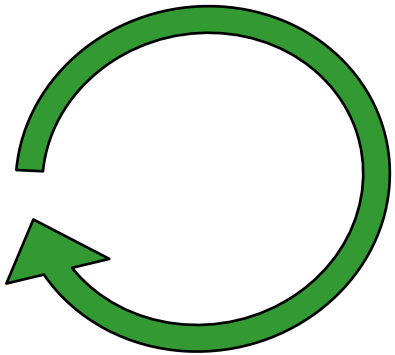
Testers help to defend the value of the product
by learning on behalf of our clients.



Exploratory Process

Self-Controlled Seeking in a Complex Environment.

Exploratory process acts on itself.
Scripted process does not.

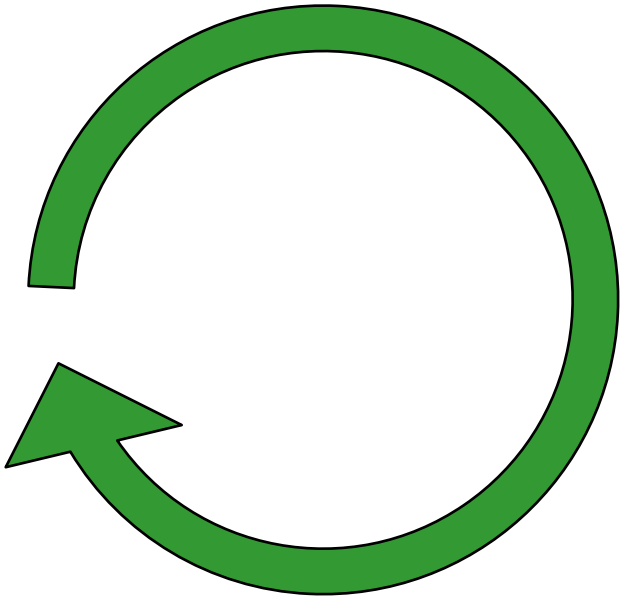


value seeking

task performing

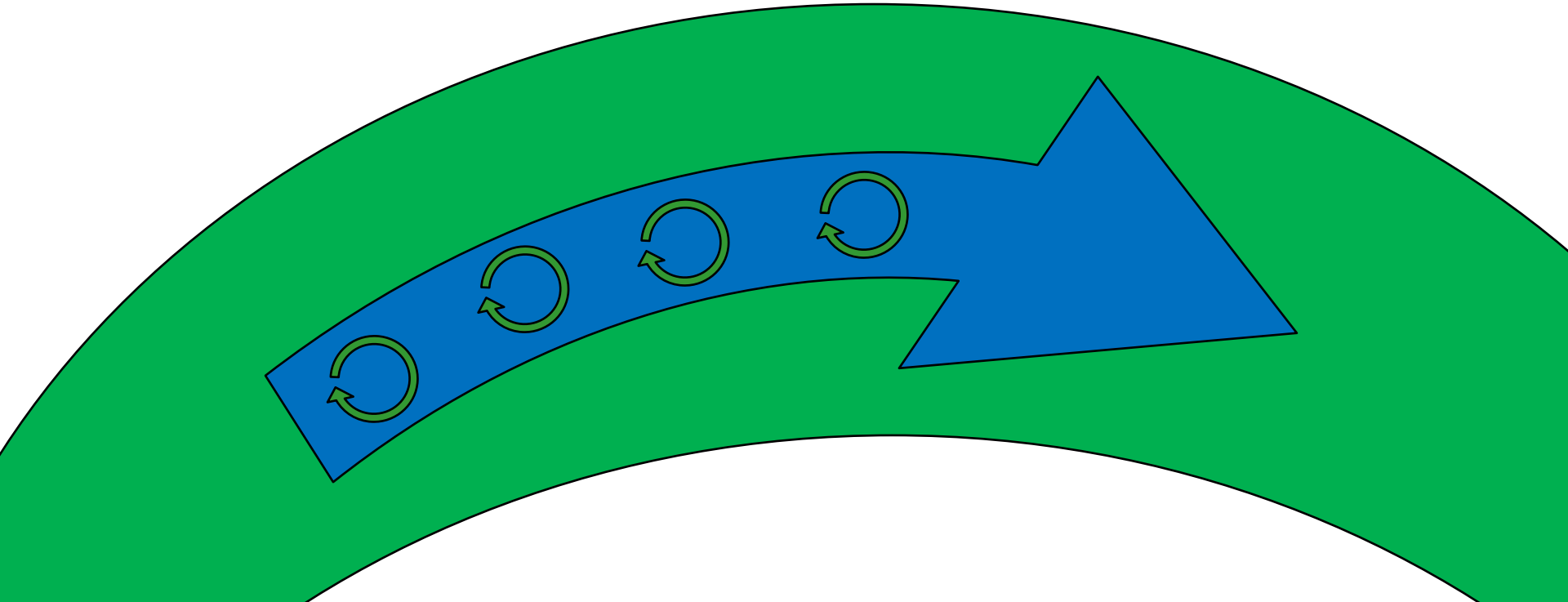
You can put them together!

arrows and cycles



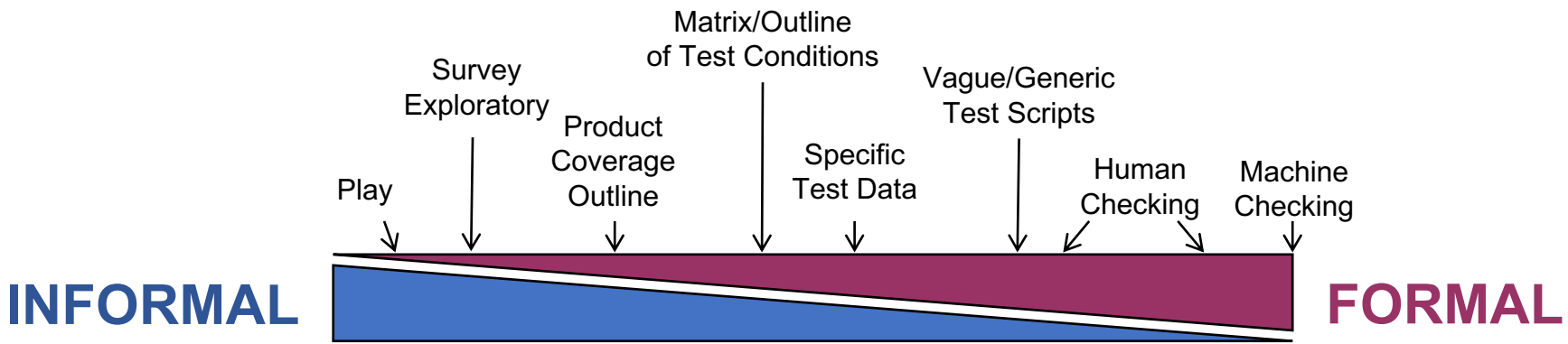
You can put them together!

arrows and cycles

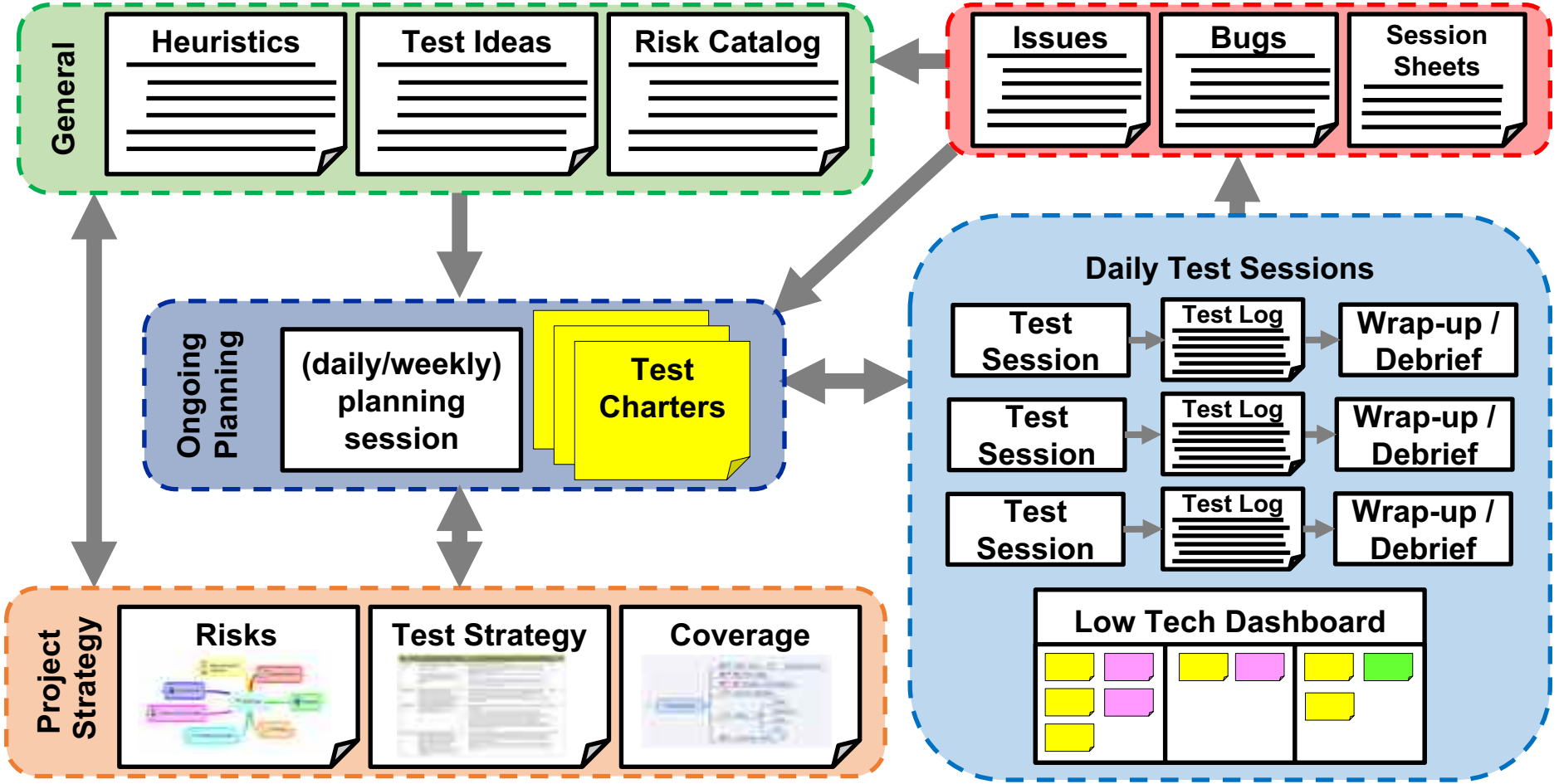


The Testing Formality Continuum

Mixing Scripting and Exploration



Informal = Not done in any specific way, nor to verify specific facts.
Formal = Done in a specific way, or to verify specific facts.



#expoQA18 - @huibschoots

Structure?

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

One dominating structure

Exploratory testers construct a compelling story of their testing. It is this story that gives testing a backbone.

The testing story has 3 levels:

1. A story about the status of the **product**
2. A story about **how you tested** it
3. A story about the **value of the testing**

Demo
Test Session

Demo & exercise

You are going to observe a demo:
Santosh is going to test an app for some time.

While observing:

- Try to see what is happening
- Take notes
- Try to be as detailed as possible


LIVE DEMO

Debrief

Testopsies

What?

Autopsy: a critical examination or dissection of a subject or work

Testopsy: 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.

* The term testopsy is coined by James Bach

The Basic Idea

- Observe a testing session (your own or somebody else's)
- 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 – Improve 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 / explaining / justifying
    - Narrating / framing
    - Observing
4. Understand (and even appreciate) yourself
5. Ability to explain what you're doing

How about a team exercise?

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

Wrap-up

Exploration will boost the value of your testing

- Any testing you do is exploratory is some way…
- Deal with unknown unknows
- Fast learning, more freedom, more fun
- Do what really needs to be done
- Create engagement: help people use their brains
- Take advantage of tacit knowledge and skill
- Use insights from experiments to inform the next

Talk better to think better!

Replace…	With…
• Verify that…	• Challenge the belief that…
• Validate	• Investigate
• Confirm that…	• Find problems with…
• Show that it works	• Discover where it doesn’t work
• Pass vs. fail	• “Is there a problem here?”
• Test cases	• Test conditions & test ideas
• Expected result/actual result	• Actual result/oracle
• Counting test cases	• Describing coverage

QUESTIONS, YOU HAVE, HMM?

#expoQA18 - @huibschoots

huib.schoots@improveqs.nl
@huibschoots
http://www.huibschoots.nl
http://www.improveqs.nl

Improve
QUALITY SERVICES

TESTBASH NETHERLANDS

Acknowledgements

- Some slides shown are taken from Rapid Software Testing and are used with permission.
- Rapid Software Testing is developed by James Bach and Michael Bolton. Also see: http://www.satisfice.com/info_rst.shtml

Rapid Software Testing (RST) is a mind-set and a skill-set focused on performing testing more quickly and less expensively while still completely fulfilling the mission of testing.

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/
- Checklists to help observation - bit.ly/testopsy1 & bit.ly/testopsy2

More references

- More on Exploratory testing - http://www.huibschoots.nl/links
- 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

Even more references

- Task Coach - http://taskcoach.org/
- Heuristic Test Strategy Model - http://www.satisfice.com/tools/htsm.pdf
- Rapid Software Testing - http://www.satisfice.com/info_rst.shtml
- 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/