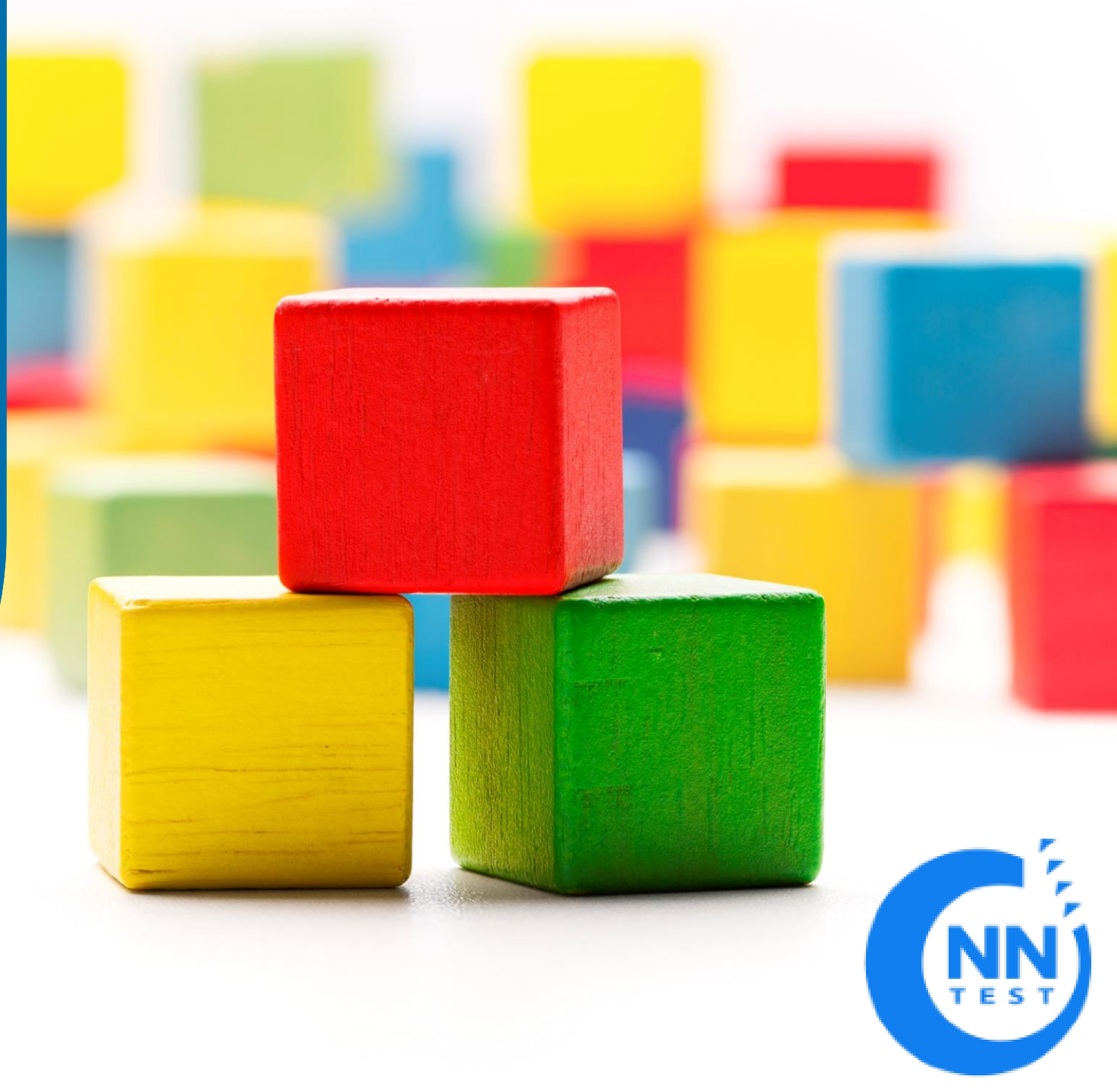




Back to Basics

Noordertest 2024

Huib Schoots





EXPERTS




RESEARCH



Back to BASICS



How testing often works

1. Read/discuss the user story
- 2.
3.  magic happens
- 4.
5. Write test cases
6. Execute test cases
7. All test cases passed
8. Story = done!

“ Most testing is driven by **implicit** questions about risk ”

“ **Nothing happens** here since we do not report ”

Elements of excellent Testing

1. Model the test space (product & context)
2. Identify risks
3. Consider coverage
4. Consider oracles
5. Determine test procedures
6. Test the system
7. Evaluate the test results
8. Report test results

“ Most testing is driven by questions about risk... ”

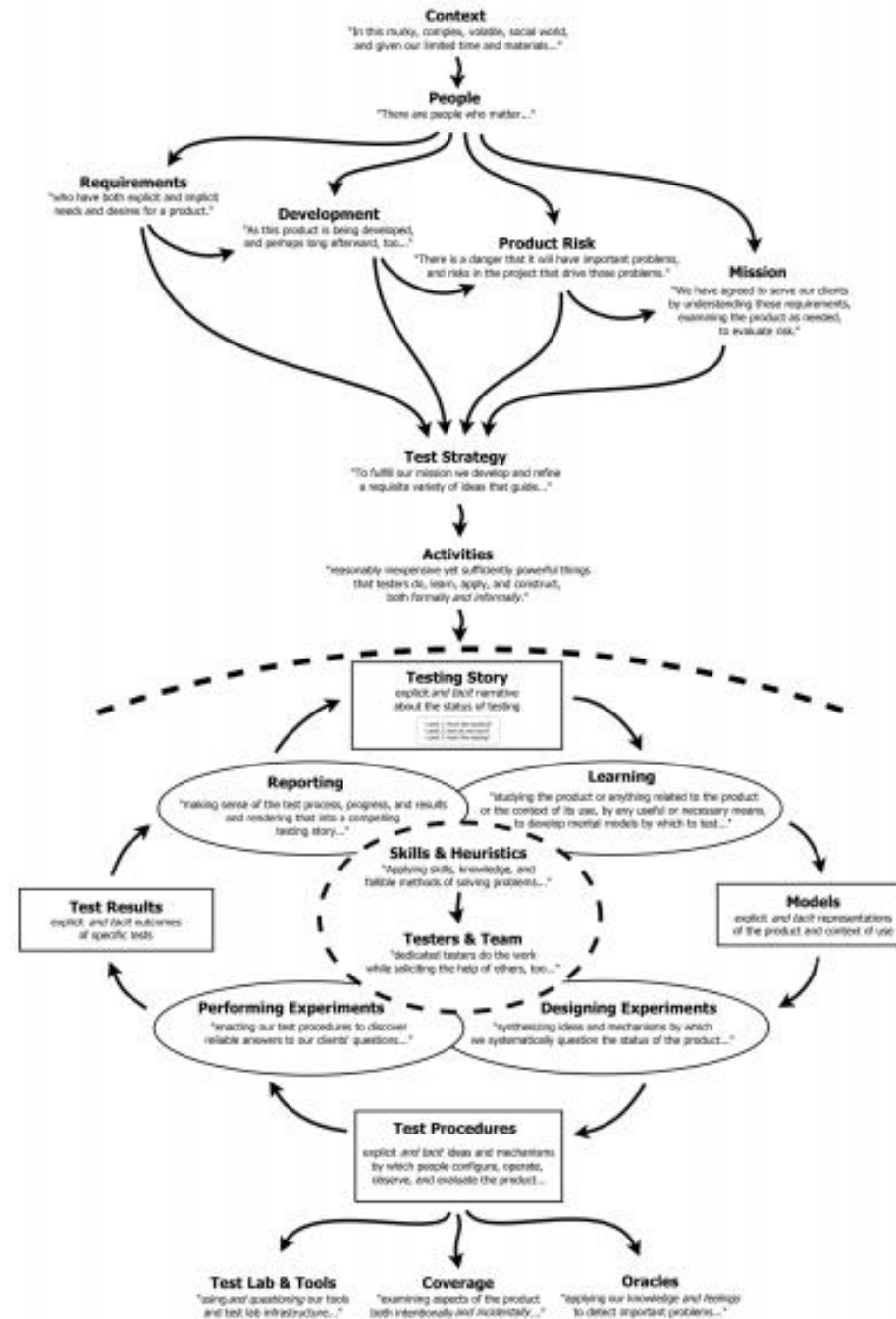
“ ... so, it helps to relate test results back to risk ”

Important questions about testing

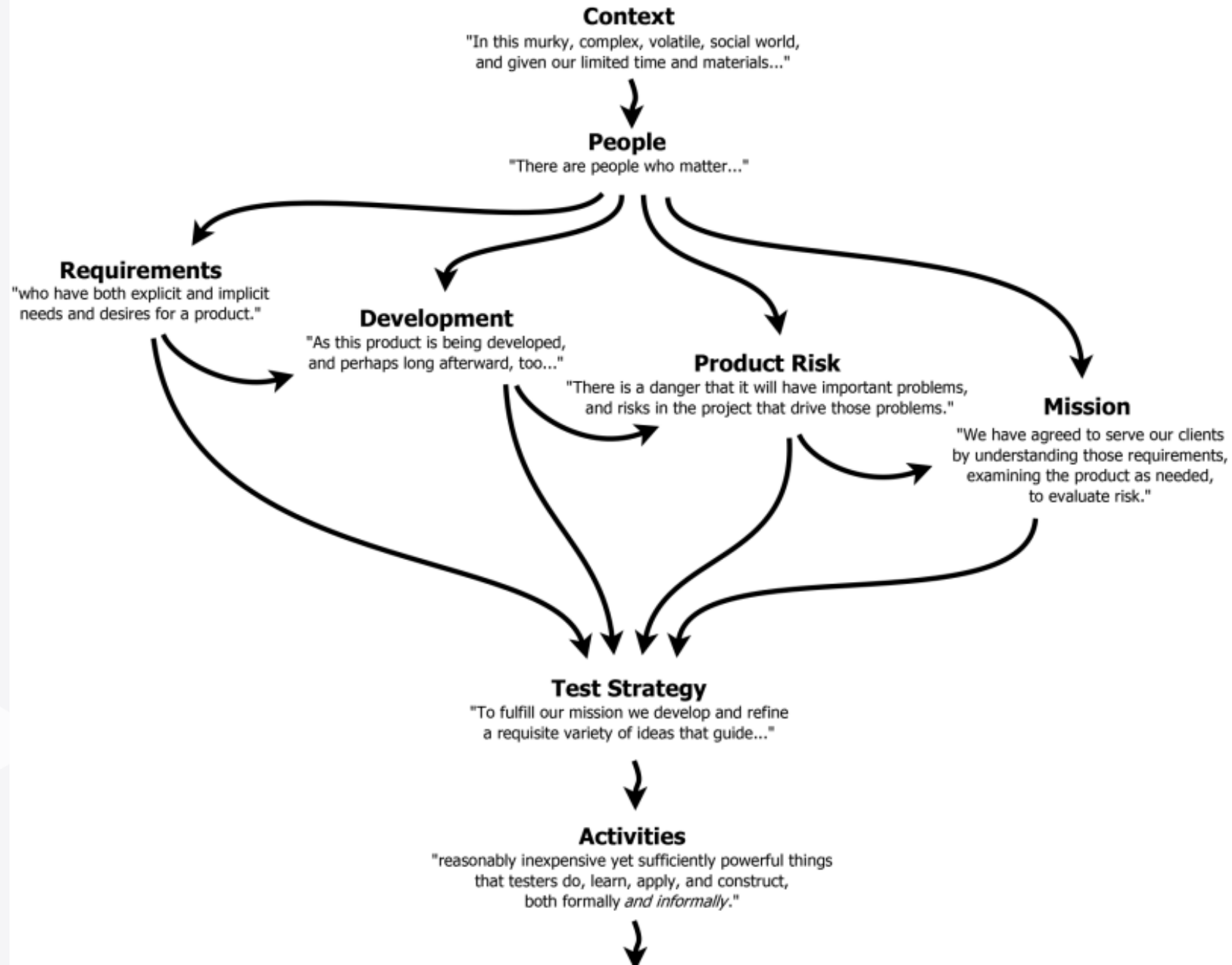
- How does testing work?
What is your testing paradigm?
- How do you decide what to test?
How do you make choices?
- What do you do?
How do you do it?
- How do you decide when you stop?
How do you know you are done?

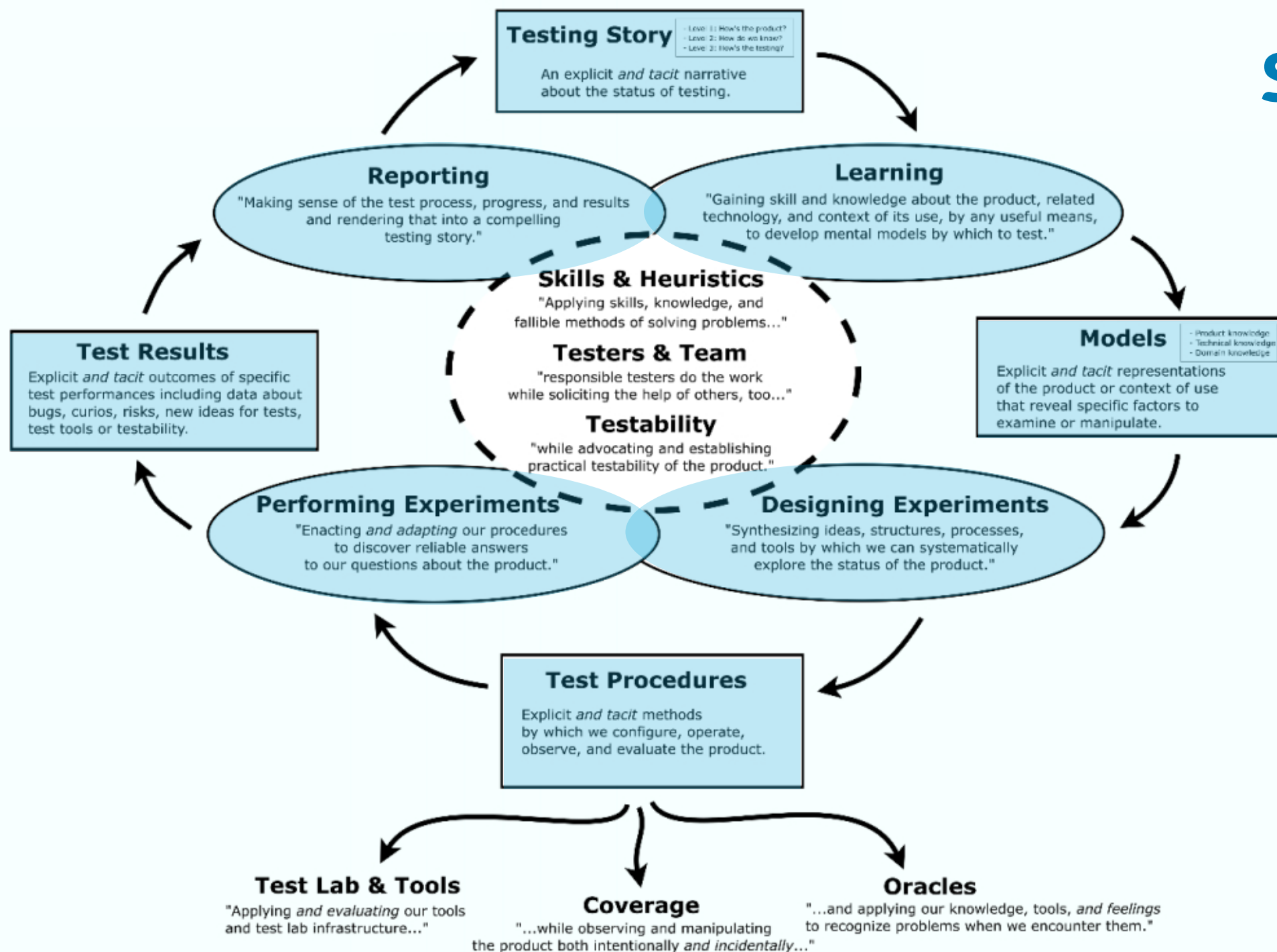


A Rapid Testing Framework



A Rapid Testing Framework



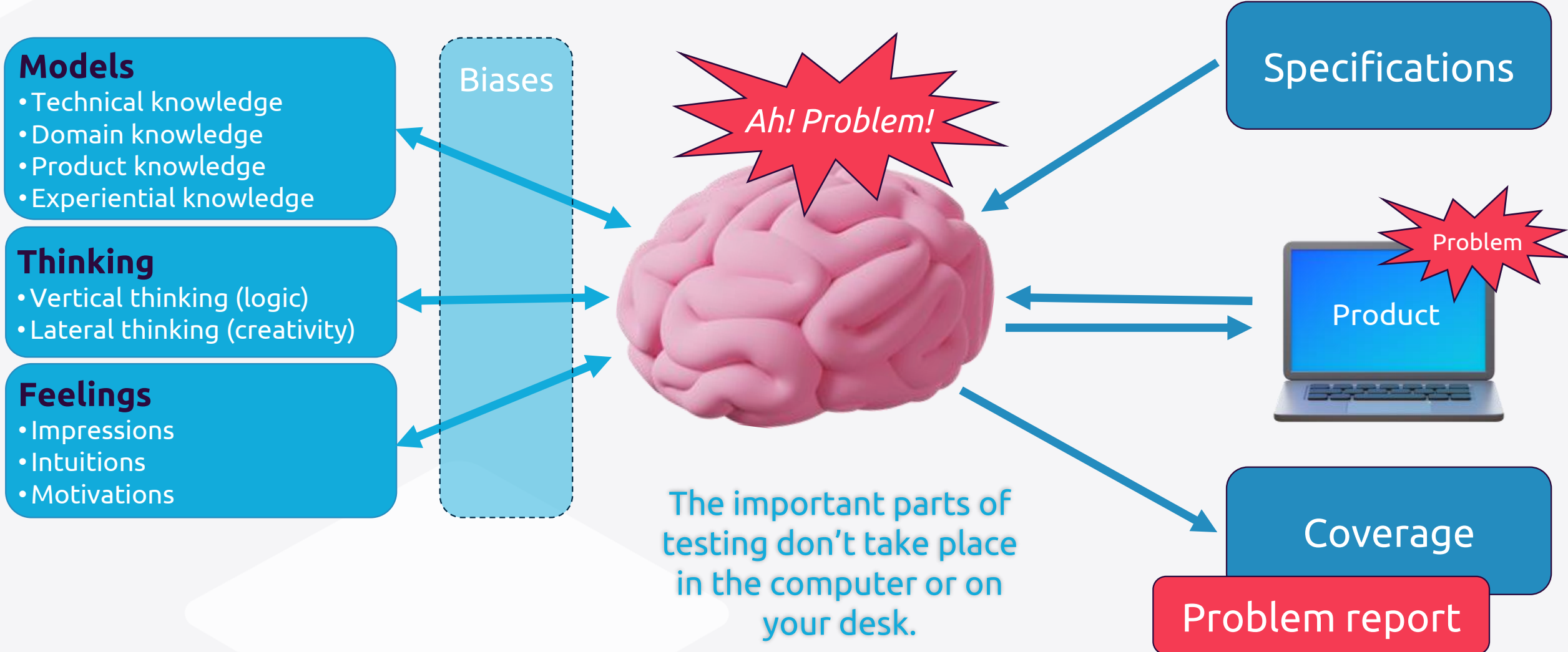


Testing is learning!

- Testing is so much more than creating & executing test cases!
- But what do I have to learn?
- The requirements might be a good start...
but how do you fully understand the product?



Testing is in your head

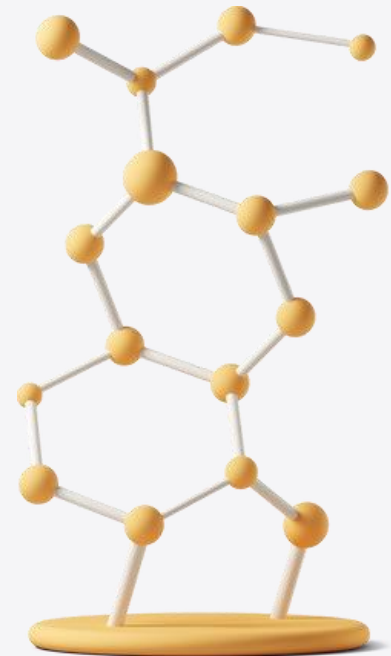


The better your model, the better your testing!

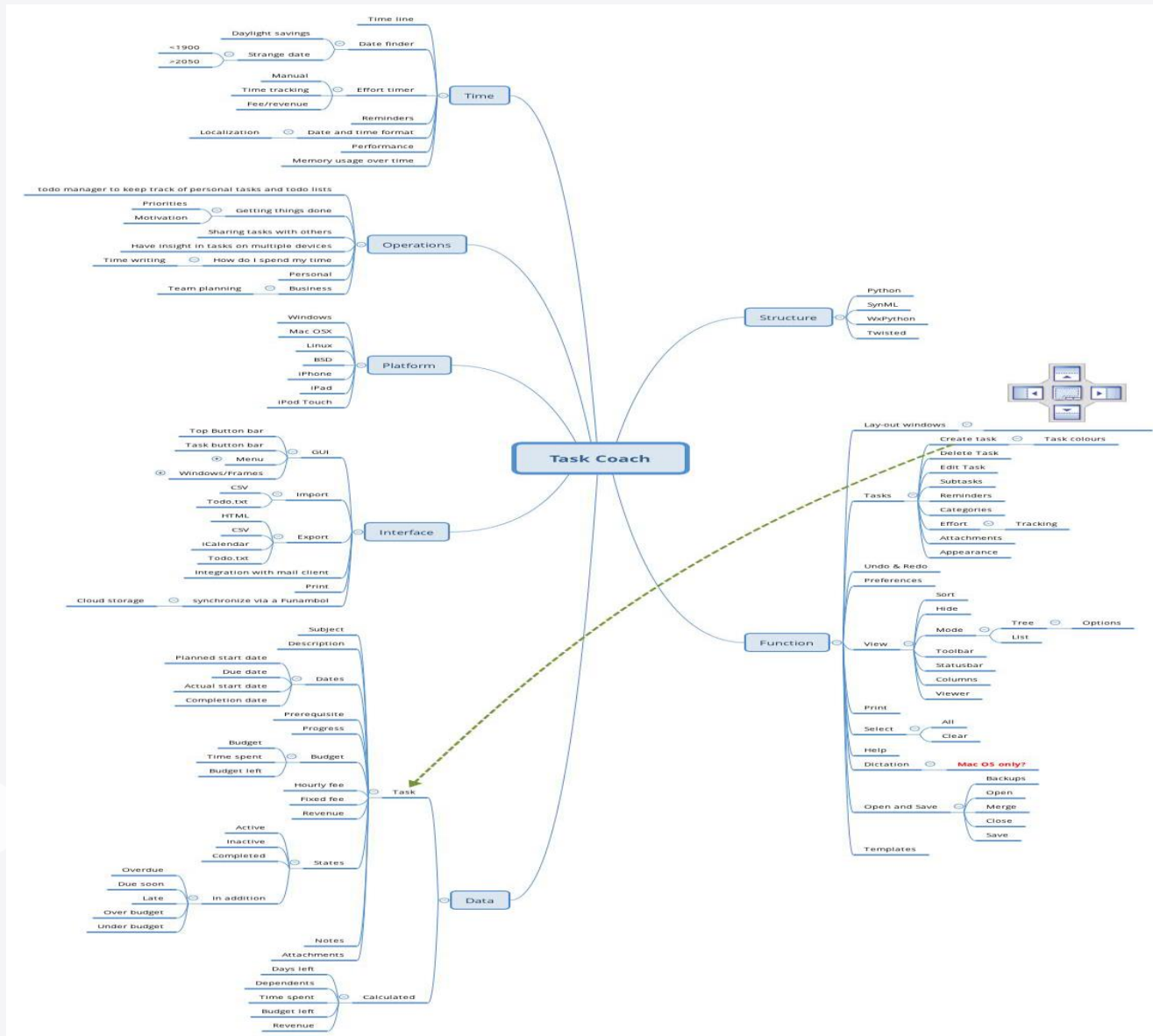
- Create a mental model*
- Question your model
- Design actions to investigate
- Apply critical thinking on your model & actions
- Work with the intention to find problems

* Since we deal with complex products, you might want to visualise your mental models

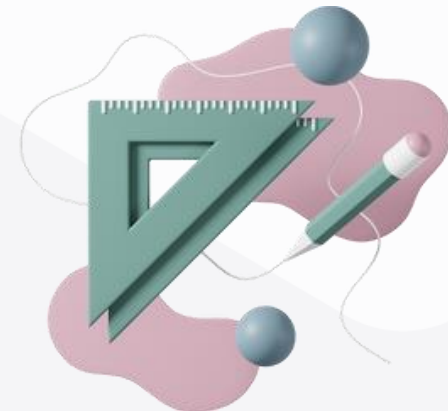
Bad model = bad testing!!



Example model: Product Coverage Outline



Watch my video in which I explain and create a Product Coverage Outline here:
<https://youtu.be/NUojNfDjljw>



Making choices about testing?

- Risks to drive choices
- Test strategy (**What** to test and **how** to test) to be able to discuss your choices about coverage
- Involve your team and product owner in making the choices!



An example risk analysis & test strategy

Risk	Task	Test type
Search functionality stations shows wrong stations ...	Check search algorithm	Unit test
Schedule information is wrong because API call go wrong ...	Check schedule API calls	Integration test
Search functionality not usable on different dates ...	Explore date select pop-up "search functionality"	Feature test
Search functionality not easy to understand and use ...	Test usability using usability heuristics	E2E
Regression after merge of new feature	Run regression checks	Automated regression checks

Collaborating & communication

- Work with your team :
 - Diversity of thinking
 - Content of unit/integration tests is important information
 - Creating feedback loops to improve your thinking
- Create insight & overview
 - Help answering the question:
"Are there problems that threaten the on-time, successful completion of the project/product??"
 - Help your team & Product Owner to make informed decisions

To be able to do this, you need to be able to explain & justify your testing!



Talk better to think better!

Replace...

Verify that
Validate
Confirm that
Test cases
Pass vs. fail
Show that it works
Expected result/actual result
Automated testing
We don't have good requirements documents!
Environment down: I guess we're just hosed.
Counting test cases
We don't have enough time to test!
We have to...!

With...

Challenge the belief that
Investigate
Find problems with
Test ideas / test factors
Is there a problem here?
Discover where it doesn't work
Actual result/oracle
Programmed checking
Let's test this, find out what it does & document that
Are you okay with this situation, dear client?
Describing coverage
How can we add testability to product & project?
We choose to...

Reporting examples

- Test Strategy & coverage
- Debriefs
- Product & Release Coverage Outlines
- Testing Story



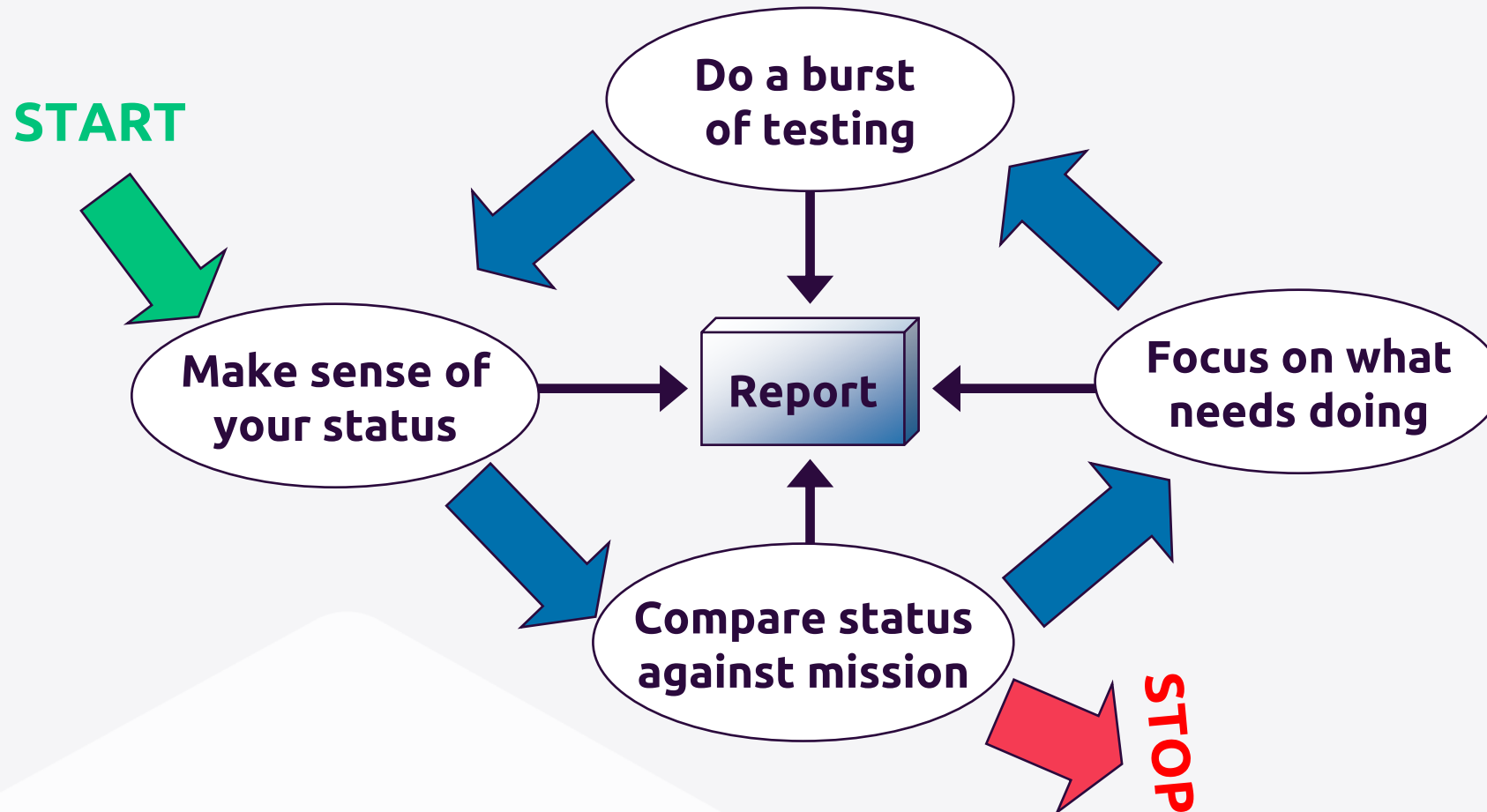
Talking about evolving test strategy

Discuss your test strategy (**what** are you going to test and **how**?) with your team and Product Owner to prevent:

- Testing too much
- Testing too deep
- Forgetting important risks



Reporting is a concept, not necessarily a document



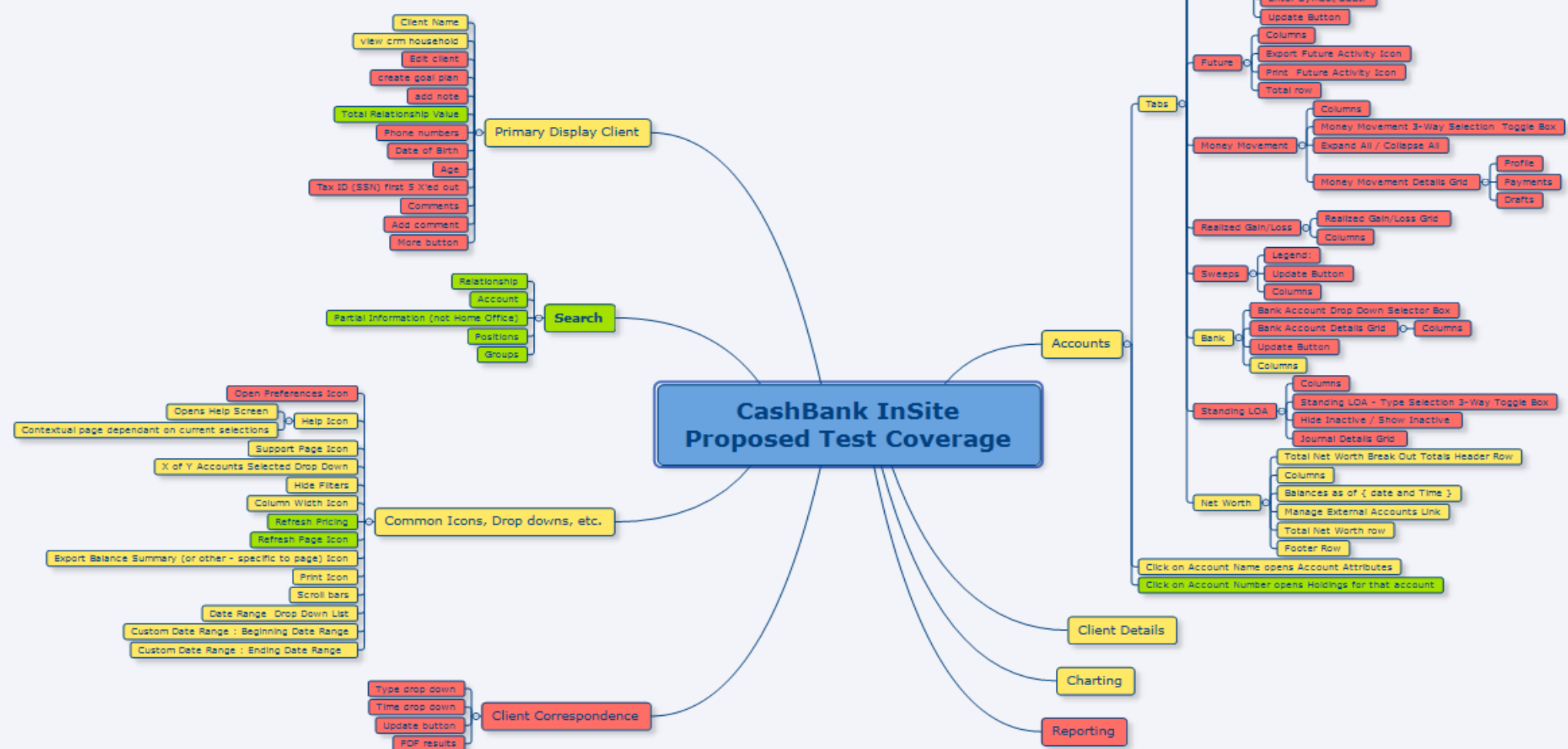
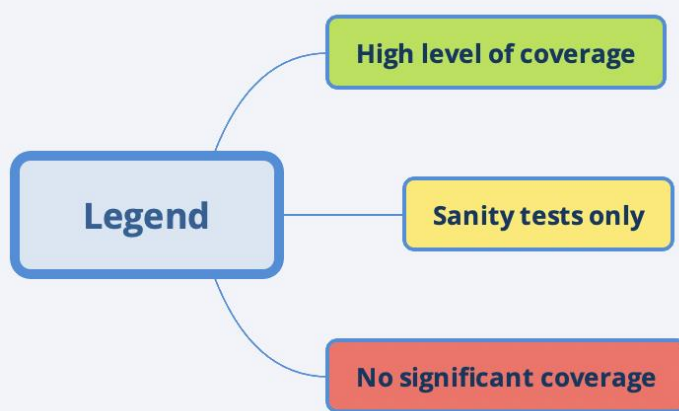
Debrief

- 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?

- Learning, knowledge sharing & coaching
- Deeper understanding of the product and better testing
- Better understanding of the testing your team member did
- New ideas on how to test the product



Coverage Reporting



Testing Story

Level 1: A story about the status of the PRODUCT...

- ...about how it failed, and how it might fail
- ...in ways that matter to your various clients

Level 2: A story about HOW YOU TESTED it...

- ...how you configured, operated and observed it
- ...about what you haven't tested, yet
- ...and won't test, at all

Level 3: A story about the VALUE of the testing...

- ...how testable (or not) the product is
- ...what the remaining risks are
- ...what the risks and costs of testing are
- ...things that make testing harder or slower
- ...what you need and what you recommend

Product any good?

How do you know?

Why should I be pleased
with your work?

Important skills

- Collaboration & communication
- Learning
- Modelling
- Analyzing & questioning
- Critical & creative thinking
- Risk analysis
- Test strategy (deliberate coverage choices)
- Talking about testing & evolving test strategy
- Creating insight & overview
- Mindset: curious, problem-solver, service provider



So how do I learn these skills??

- Deliberate Practice
- Continuous Learning







What will you do differently tomorrow?



**Thank
you!**

Questions?
Remarks?
Feedback?
Stories?

Huib Schoots

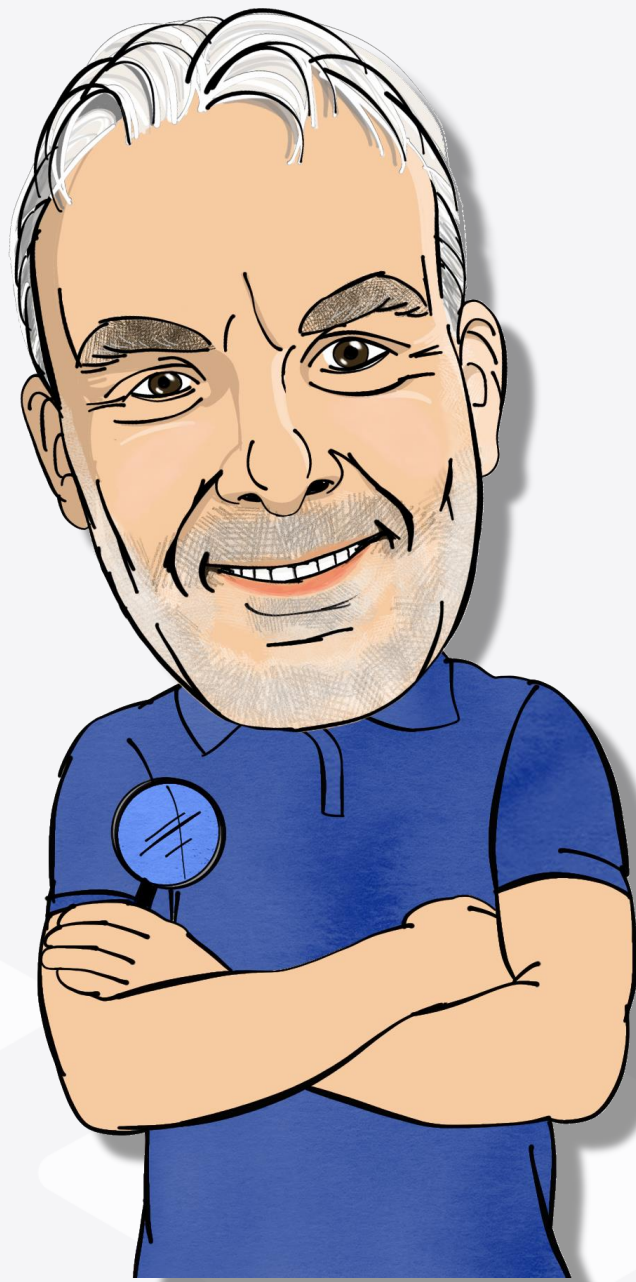
huib.schoots@sogeti.com

+31 6 – 24 64 10 33

www.linkedin.com/in/huibschoots/

www.huibschoots.nl

www.sogeti.nl



**Everything I do comes with lifetime technical support.
Contact me if you have questions or need help!
Even in ten years or more...**

Thank you

Slides:

Some of the slides are taken from Rapid Software Testing:

<https://www.rapid-software-testing.com>

Dennis Bergkamp: for being a beautiful player

Pictures:

- Businessman, Businesswoman, R&D, Learning, Brain, Laptop, Model, boy, girl, young woman, young man - <https://icons8.com/illustrations/author/zD2oqC8lLBBA>
- Design - <https://icons8.com/illustrations/author/CkHJmwURlxnt>
- Collaboration & thinking - all pics from www.pexels.com
- Risk - <https://icons8.com/illustrations/author/lZpGtGw5182N>
- Thank you - <https://icons8.com/illustrations/author/ODexzOcCgAMh>



More info & references

- Tanja Vos - We all need to become critical testers
<https://www.youtube.com/watch?v=CGDNczZmGB0>
- Rapid Software Testing:
<https://www.rapid-software-testing.com>
- Huib Schoots - Creating a Test Strategy:
<https://www.huibschoots.nl/wordpress/?p=2726>
- Huib Schoots - Test Strategy Ask me Anything:
<https://www.ministryoftesting.com/dojo/lessons/testing-ask-me-anything-test-strategies-huib-schoots>
- Considerations when testing a software application in a context-driven way:
<http://www.huibschoots.nl/wordpress/?p=2781>
- Huib Schoots & Vincent Wijnen – Creating a culture of learning:
https://bit.ly/learningculture_atd2022
- Huib Schoots & Paul Holland – Automation Addiction:
<https://youtu.be/ZcwxyzvGWFc0>
- Huib Schoots & Alex Schladebeck – Let's stop talking about testing, let's start thinking about value:
<http://www.huibschoots.nl/wordpress/?p=2763>
- Anne-Marie Charrett - Quality is a Team Responsibility:
<https://youtu.be/mBC3ssLlJfQ>
- Collected useful links on my website:
<http://www.huibschoots.nl/links>



