

How to **NOT** waste time

Huib Schoots – OnlineTestConf 2022

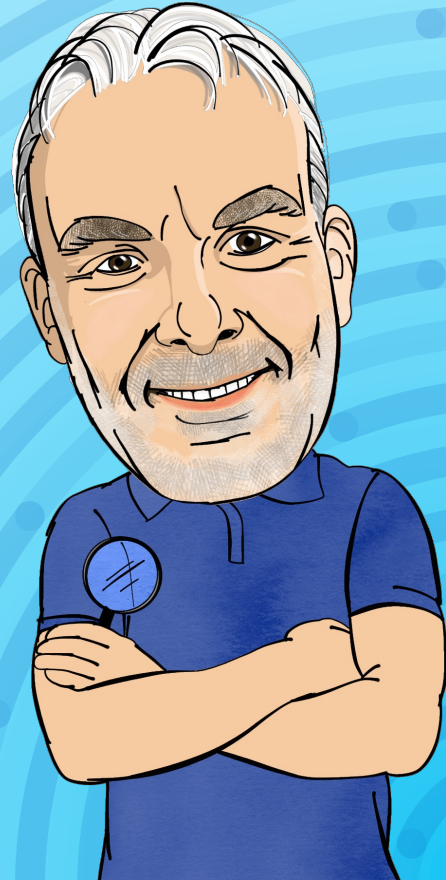


OnlineTestConf



Hi, my name is Huib

- Married to Karin, 2 bonus daughters
- I live in Den Bosch in the Netherlands
- 25+ years experience in IT: experience as developer, tester, consultant, manager, trainer & coach
- Managing consultant @ Quality Accelerators
- Senior consultant @ De Agile Testers
- Rapid Software Testing instructor
- Singer in a rock band



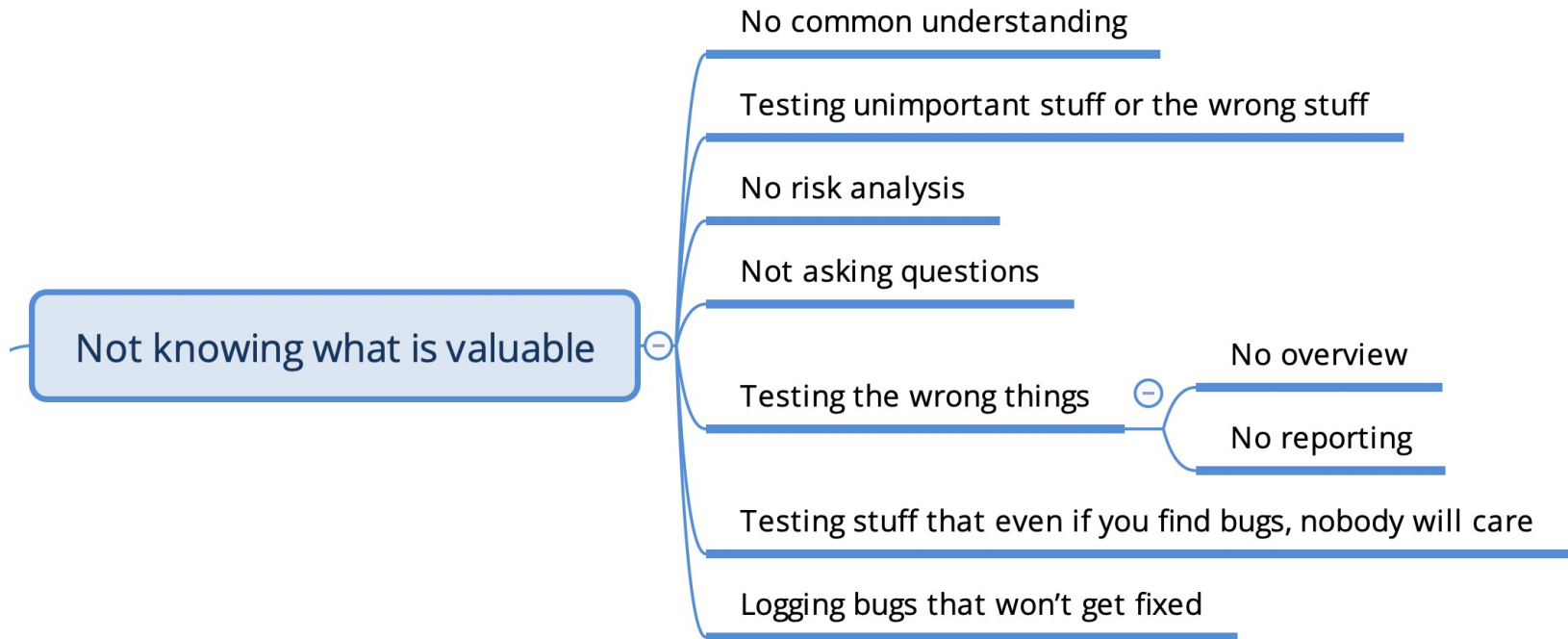


No missions for testing Using the wrong environment Automating everything UI testing all the time
No overview Writing reports no-one will ever read No risk analysis Context switching
No use of heuristics Siloes/no collaboration Spending too much time finding out why it is a bug
Getting into the flow takes time Testing too many variants of your testcase
Not having sufficient test data No focus on testability Not asking for help
Getting interrupted all the time Testing stuff that even if you find bugs, nobody will care
Shitty bug reporting Spending a lot of time estimating Testing when you are not in the right mental state
Let unskilled people do the work Using old testcases to create new ones Doing all the thinking yourself
No reporting No retrospectives
So much waste!!!
Testing unimportant stuff or the wrong stuff Testing stuff twice No wrap-up of debrief
Not knowing what is valuable Spending a lot of time creating test data Bad Unit Testing
Spending too less time creating a decent bug report No use checklists No common understanding
Testing too much Useless or bad meetings Not taking good notes No test strategy
Not asking questions Over engineering your testing Writing test cases all the time
Testing shitty software Using tools because you have to, not because they help you Mocking too much
No pairing/ensemble testing Writing brittle automation Logging bugs that won't get fixed

Ways to waste time

- Not knowing what is valuable
- No test strategy
- Testing too much
- Ineffective learning
- Testing shitty software
- Doing all the thinking yourself
- Bad testing
- Bad automation
- Brain waste
- Useless or bad meetings

Not knowing what is valuable



No test strategy

No test strategy

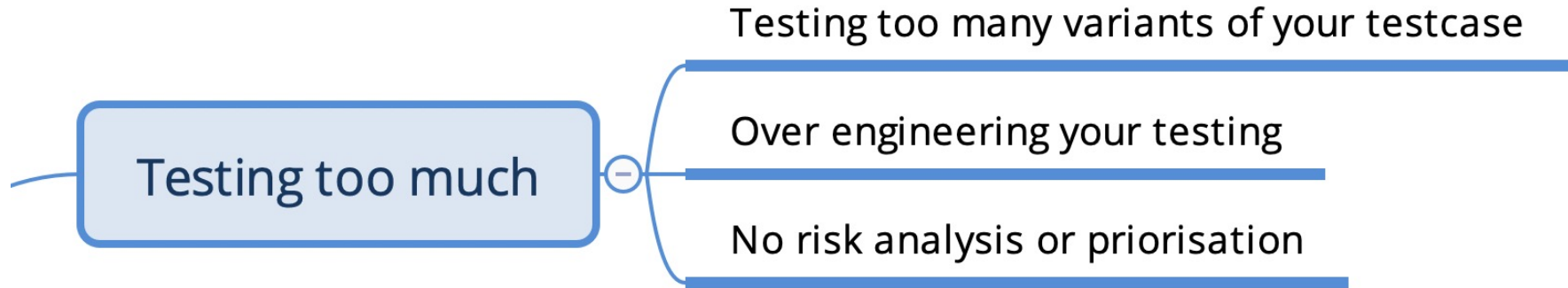
No missions for testing

Testing stuff twice

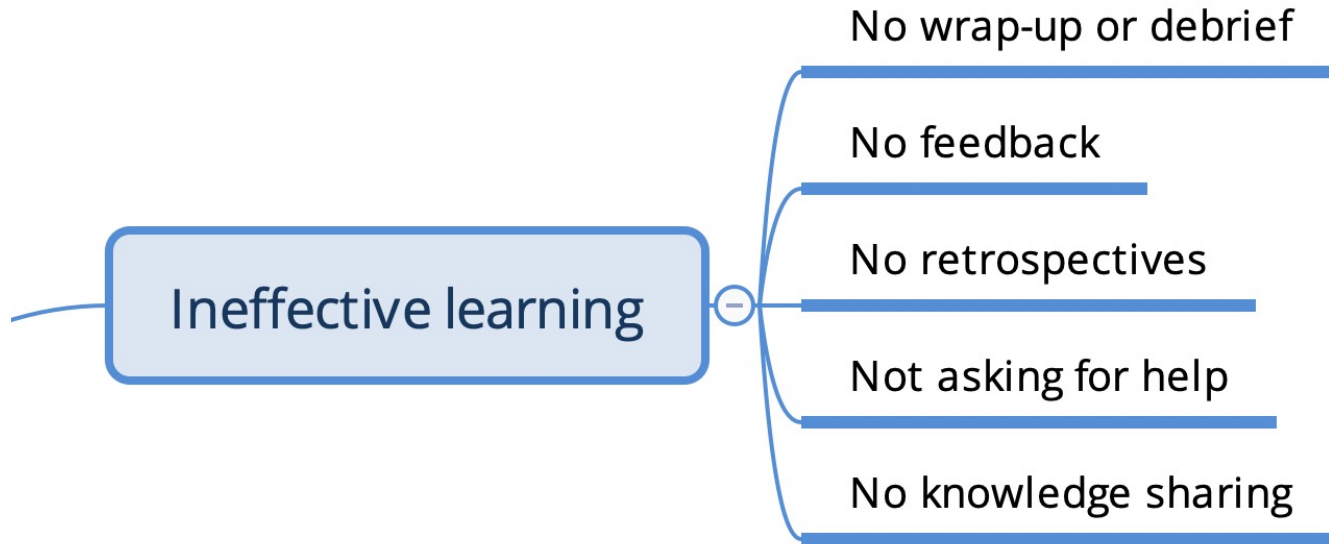
Testing stuff on the wrong level

No attention for testability

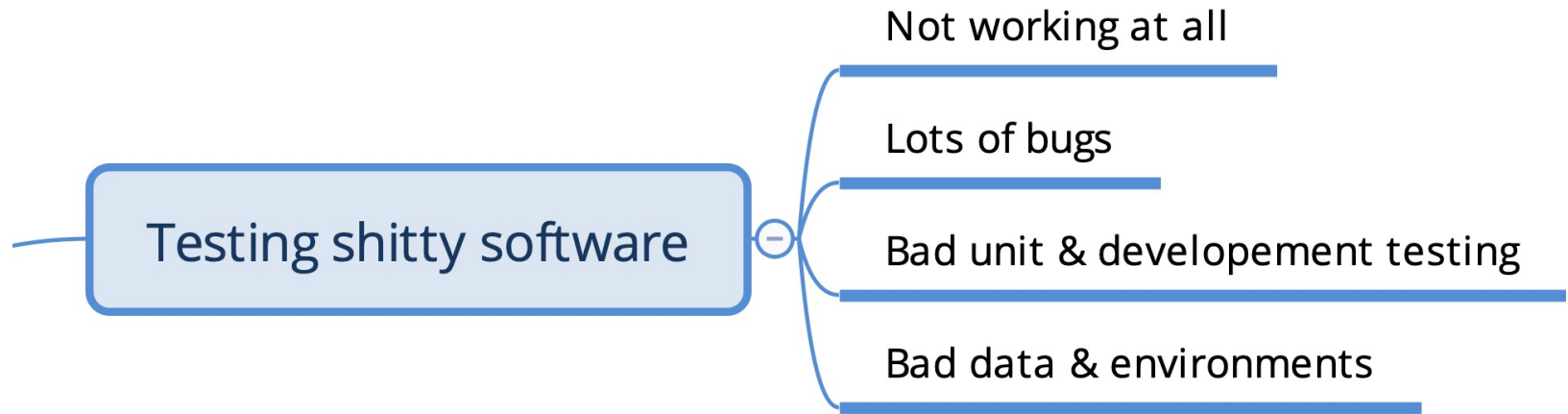
Testing too much



Ineffective learning



Testing shitty software



Doing all the thinking yourself

Doing all the thinking yourself

No use of heuristics

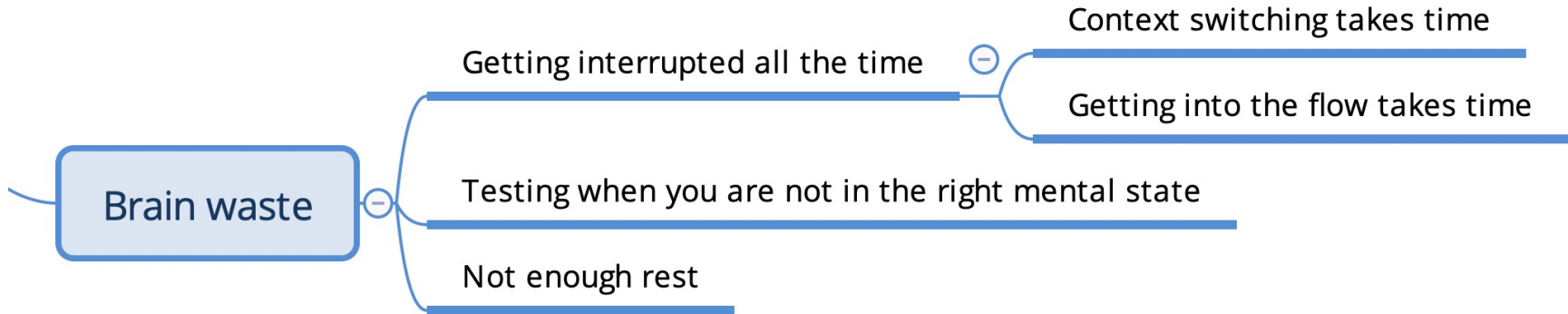
No use checklists

No pairing/ensemble testing

Siloes/no collaboration

Stay in your head

Brain waste



Bad testing

Bad testing

Writing test cases all the time

Writing testcases you will never use

Not taking good notes

UI testing all the time

Not having sufficient testdata

Using old testcases to create new ones

Writing reports no-one will ever read

Using the wrong environment

Spending a lot of time estimating

Shitty bug reporting:

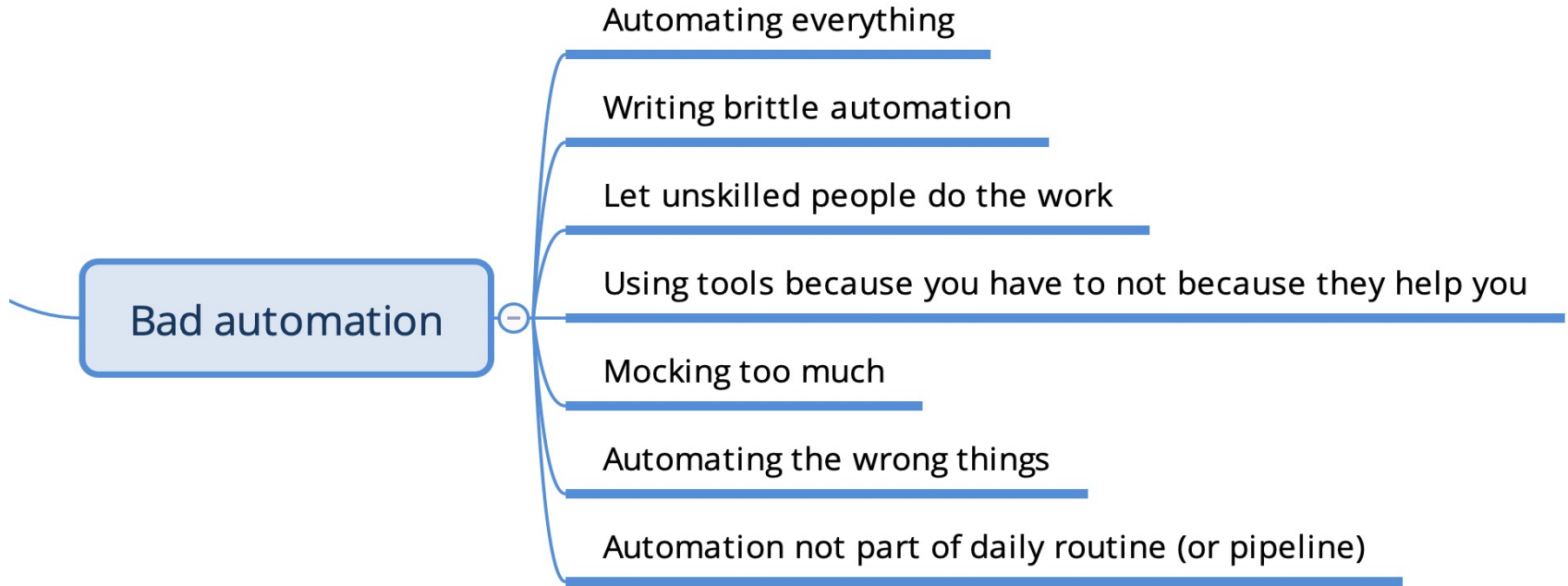
Spending too much time finding out why it is a bug

Spending too less time creating a decent bug report

Sunk cost fallacy: not getting rid of useless stuff

Ass covering instead of product coverage

Bad automation



Useless or bad meetings



Solutions?

The background of the slide features a series of concentric, semi-circular arcs in various shades of blue, ranging from light to a deeper blue. Scattered across these arcs are numerous small, solid blue dots. The overall effect is a dynamic, wave-like pattern that suggests movement or a field of data points.

Some suggestions

- Know what is risky and what is valuable
- Create and discuss test strategy
- Stop testing too much: seriously!
- Create a culture of learning
- Stop testing shitty software: work with developers
- Collaborate and don't do all the thinking yourself
- Get off the hamster wheel
- Learn how to test effectively
- Learn about good automation
- Learn how to do awesome meetings

Some suggestions

- Know what is risky and what is valuable
- Create and discuss test strategy
- Stop testing too much: seriously!
- Create a culture of learning
- Stop testing shitty software: work with developers
- Collaborate and don't do all the thinking yourself
- Get off the hamster wheel
- Learn how to test effectively
- Learn about good automation
- Learn how to do awesome meetings

Testability!

Learning about testability



10 P's of Testability

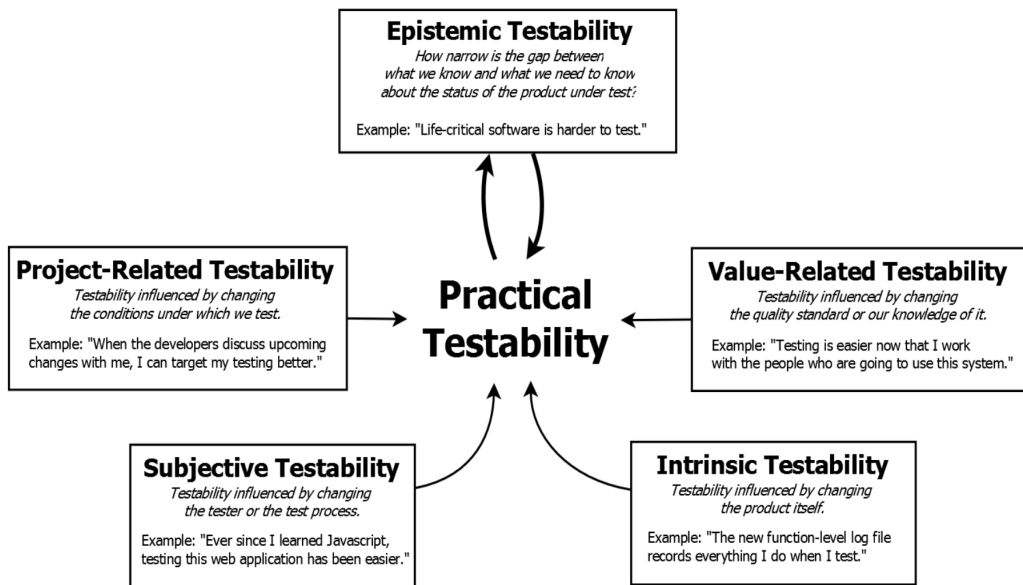
Read more about this here:

Team Guide to Software Testability - Better software through greater testability

By Ash Winter and Rob Meaney

(<http://leanpub.com/softwaretestability>)

More about testability



Read more about this here:

Heuristics of Software Testability

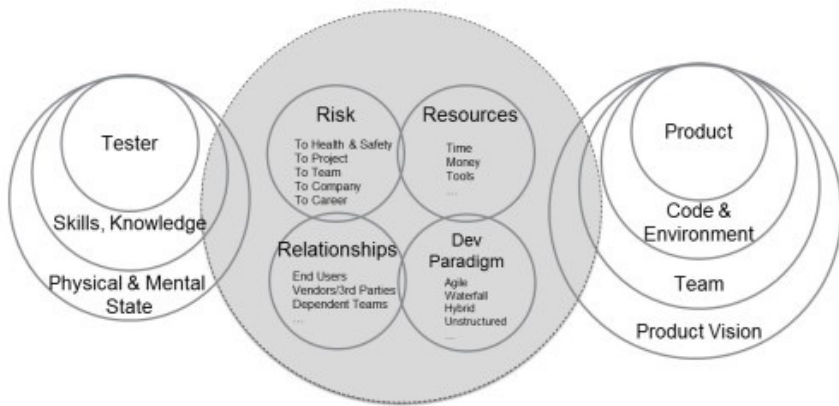
By James Bach

(<https://www.satisfice.com/download/heuristics-of-software-testability>)

Even more about testability

Dimensions of Testability

Context Drivers



Read more about this here:

Testability awakens: moving testability into new dimensions

By Maria Kedemo

(<https://mkedemo.files.wordpress.com/2015/12/testability-awakenstestingtrapeze-2015-december.pdf>)

**Comments, feedback,
stories or questions ??**

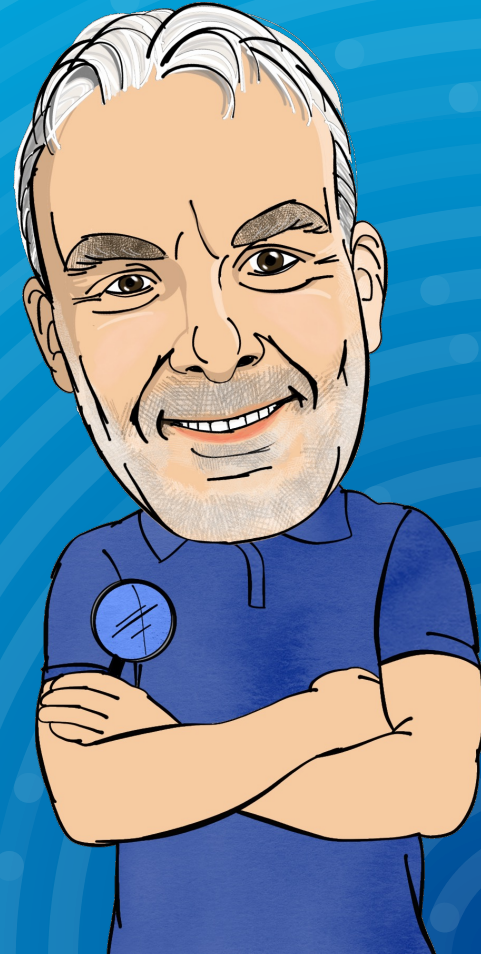
Thank you!

The background of the slide features a series of concentric, semi-circular arcs in various shades of blue, ranging from light to a medium blue. These arcs are centered on the right side of the image and curve towards the left. Scattered across these arcs are numerous small, solid blue dots, creating a pattern reminiscent of a fingerprint or a stylized sunburst.

Huib Schoots

- huib@qualityaccelerators.nl
- @huibschoots
- +31 6 – 24641033
- www.linkedin.com/in/huibschoots/
- www.huibschoots.nl
- www.qualityaccelerators.nl
- www.deagiletesters.nl

Everything I do comes with lifetime technical support.
So contact me if you have questions.
Even in ten years or later!



References & more info

- Huib Schoots - Creating a Test Strategy
<https://www.huibschoolts.nl/wordpress/?p=2726>
- Huib Schoots – Test Strategy Ask me Anything
<https://www.ministryoftesting.com/dojo/lessons/testing-ask-me-anything-test-strategies-huib-schoolts>
- James Bach & Aaron Hodder - Test cases are not testing: towards a culture of test performance
<https://www.satisfice.com/download/test-cases-are-not-testing>
- Ash Winter & Rob Meaney - Team Guide to Software Testability - Better software through greater testability
<http://leanpub.com/softwaretestability>
- James Bach - Heuristics of Software Testability
<https://www.satisfice.com/download/heuristics-of-software-testability>
- Maria Kedemo - Testability awakens: moving testability into new dimensions
https://bit.ly/testability_mk
- Ash Winter - Testability Ask Me Anything
https://bit.ly/askmeanything_testability
- Huib Schoots & Vincent Wijnen – Creating a culture of learning
https://bit.ly/learningculture_atd2022
- Huib Schoots & Paul Holland – Automation Addiction
<https://youtu.be/ZcwxyzvGWFc0>
- Stephen Altrogge - How to stop wasting time and start doing awesome work
<https://freedom.to/blog/stop-wasting-time/>
- How to run great meetings
<https://www.vorne.com/learn/accelerants/great-meetings/>
- Testing heuristics & test ideas
https://www.huibschoolts.nl/wordpress/?page_id=441#Heur & https://www.huibschoolts.nl/wordpress/?page_id=441#ideas
- More links on my website
<https://www.huibschoolts.nl/links>



Free templates for all your presentation needs



For PowerPoint and
Google Slides



100% free for personal
or commercial use



Ready to use,
professional and
customizable



Blow your audience
away with attractive
visuals