



# Testing in modern times

*"a story about Quality and Value"*

De Agile Testers



# Software Development = R&D



## Research & Development



# Research & Development: Dealing with unknown unknowns

- ▶ Capture everything upfront vs. building new insights
- ▶ Customers don't know what they want
- ▶ Cope with complexity, confusion, change, new insights and half answers

So: We need to **learn** and to deal with **risks**!



# Testing and quality?

- ▶ **Testing** informs decisions about quality & risk
- ▶ **Quality** is value to people who matter

Quality is **NOT**: conformance to requirements

Quality is **NOT**: the best product possible



Quality products **solve the problem** and are “**good enough**”

# Business case of testing?





# Business case of **LEARNING?**

- ▶ The question is: do you value learning? And if so: how much are you willing to pay?
- ▶ If you value it enough, you won't discuss the price...
- ▶ But are the costs justified?

Or learn to live without it ... with all risks involved.





Reduce cost of testing?

Focus on **Testability!**





Reduce cost of ~~testing?~~  
learning

Focus on ~~Testability!~~  
Learnability (\*)



(\*) Learnability: not the old ISO9126 definition

My definition: how easy is it to learn about the value and risks of a product/solution





# Testability you say?

Testability of a product is how easy it is to learn (read: test) by a particular team in a given context.





# 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>)



# Automate everything?

- ▶ **Automated checks** provide binary answers to a question to which you already know the answer.
- ▶ We need a lot of checks to **speed up development** by detecting regression problems.
- ▶ **Exploring** can provide new insights into the product. Only people can provide this insight.





# Can everybody test?

- ▶ Sure. The question is: how good do you want it to be?
- ▶ Most people **do not like testing** at all! And that is why they will never be good at it. Nor do they have time to learn these skills.
- ▶ We need smart people with "**critical distance**" to do skilled testing. People determined to find problems that matter.
- ▶ Problems as in: "**are there problems that threaten the value of the product or the on-time successful delivery?**"

**So are we still talking about testing? Or is it much broader?**



# About critical distance

We need a diversity in thinking: different mindsets

## Opportunity mindset

Solve problems

Ask how and when questions

## Problem mindset

Search for problems

Ask what if questions



# Learning in teams: it's all about loops!

## Create learning loops (Plan-Do-Check-Act) in everything you do

- ▶ Team collaboration and skills are key: optimize your SDLC and processes
- ▶ Risks are a whole team responsibility
- ▶ Test your requirements & assumptions continuously
- ▶ Help programmers go faster: dev and release pipelines, automated checks, code quality, measuring the right metrics
- ▶ Test enough (good is good enough): test & automation strategy! Or even better: an integrated quality strategy based on risks
- ▶ Use tooling & automation in your exploration
- ▶ Mitigate “appropriate” risks by not testing: monitoring, test in production, etc.



# Learning in teams: it's all about loops!

Focus on fast learning in our teams:  
**Whole team quality!**

So are we still talking about testing?



# Thank you!

---

De Agile Testers





**Comments  
or  
Questions ??**

# References and more info



- ▶ 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/mBC3ssLIJfQ>
- ▶ Anne-Marie Charrett - Screw Testing, Let's Talk Quality - <https://youtu.be/v4uw29pW73E>
- ▶ Team Guide to Software Testability - <http://leanpub.com/softwaretestability>
- ▶ Heuristics of Software Testability - <https://www.satisfice.com/download/heuristics-of-software-testability>
- ▶ Interview with Rob Meaney on Quality coaching roadshow - <https://www.spreaker.com/show/quality-coaching>
- ▶ Testability Ask Me Anything (Ash Winter) - <https://www.ministryoftesting.com/dojo/lessons/testing-ask-me-anything-testability-ash-winter>
- ▶ Quality Engineering Ask me Anything (Anne-Marie Charrett) - <https://www.ministryoftesting.com/dojo/series/testing-ask-me-anything/lessons/ask-me-anything-anne-marie-charrett-quality-engineering>
- ▶ Rapid software testing – <https://www.rapid-software-testing.com>
- ▶ Collected links on my website - <http://www.huibschoots.nl/links>



# Who am I?

- ▶ Context-driven software tester
- ▶ Rapid Software Testing teacher
- ▶ Scrum master, team coach & agile expert
- ▶ Humanist
- ▶ Curious & lifelong learner
- ▶ Passionate & energetic people lover
- ▶ Trainer, coach, consultant, writer, speaker
- ▶ Storyteller, leader, human, rebel, juggler
- ▶ Trombone & guitar player, photographer, gamer, beer brewer, magician





# Huib Schoots

---

- ▶ huib@deagiletesters.nl
- ▶ 06 – 24641033
- ▶ [www.linkedin.com/in/huibschoots/](https://www.linkedin.com/in/huibschoots/)
- ▶ [www.huibschoots.nl](https://www.huibschoots.nl)
- ▶ [www.deagiletesters.nl](https://www.deagiletesters.nl)

