If you read this, please answer these 11 short questions:

https://bit.ly/oredev2022

Testing in modern times

"a story about Quality and Value"

Quality Accelerators

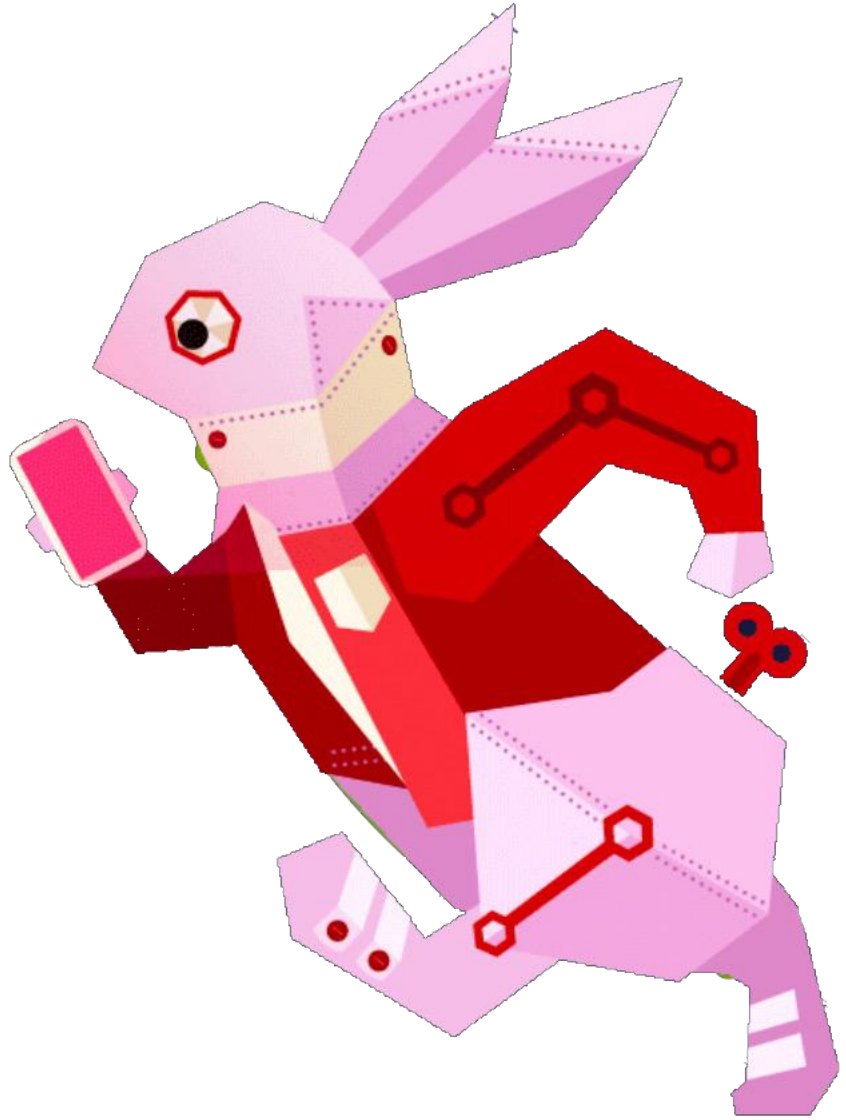
ØREDEV

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

Agenda

- Testing and quality
- Software Development
- Learning in general
- Testability
- Learning in teams



Poll

Please answer these 11 short questions:

https://bit.ly/oredev2022

Debrief poll

	TRUE	FALSE
1: Quality means the best possible product we can build	34%	66%
2: The goal of testing is finding bugs	31%	69%
3: Testers are responsible for the quality of product	7%	93%
4: Testing is a phase in our software development process	52%	48%
5: Testing is writing test cases and executing them	31%	69%
6: If the product conforms to requirements we can ship	42%	58%
7: We do explicit risk analysis in my team	32%	68%
8: We can automate all testing	10%	90%
9: Test Automation saves time and money	83%	17%
10: We don't need dedicated testers	40%	60%
11: Performance of teams is determined by the way we organize ourselves and the processes we use	85%	15%

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**”

Quotes

- Testing is not about assuring conformance to requirements; rather it is about understanding the requirements

-- James Bach

- A professional tester takes responsibility for interpreting the requirements with intelligence. Who tests, not only the system, but also (and more importantly) the assumptions of the programmers, and specifiers.

-- Uncle Bob

What is value? And for who?

- Value is in the eye of the beholder
- If you value it enough, you won't discuss the price... but are the costs justified?

Value is a **perception**, so who is doing the perceiving?

Value? For who?

- **Shareholder** value → high dividend and stock price
- **Business** value → stability, growth, making profit
- **Customer** value → the product or service they buy
- **Supplier** value → good terms, getting payed in time
- **Organisational** value → smoother process, better wow
- **Employee** value → stability, engaging work, healthy environment
- **Community** value → socially engaged, environmentally responsible

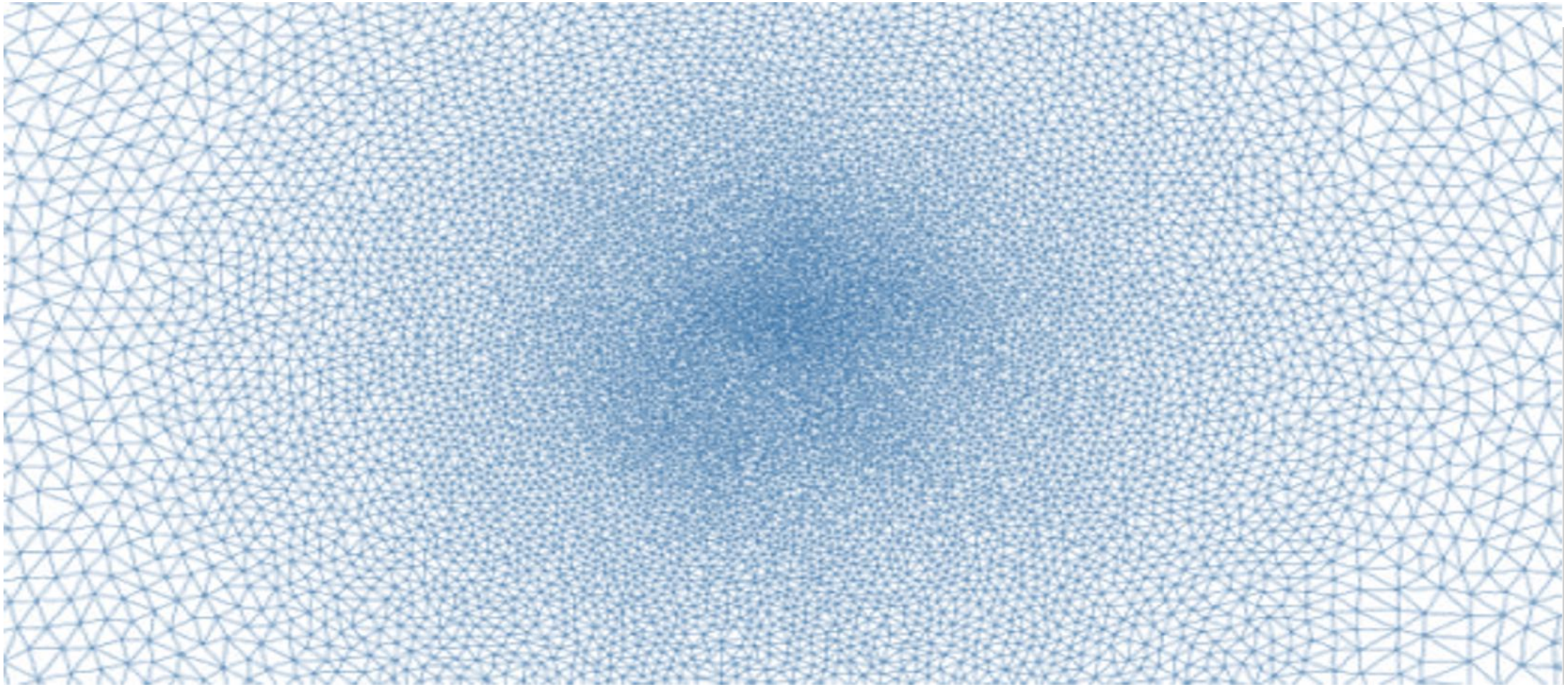
Value to our customers

- A customer does not want to buy a product,
  but to achieve a goal or solve a problem
- Customer value cannot always be expressed in monetary terms

Software Development = R&D

Research & Development

Software is complex



People are complex (and irrational)

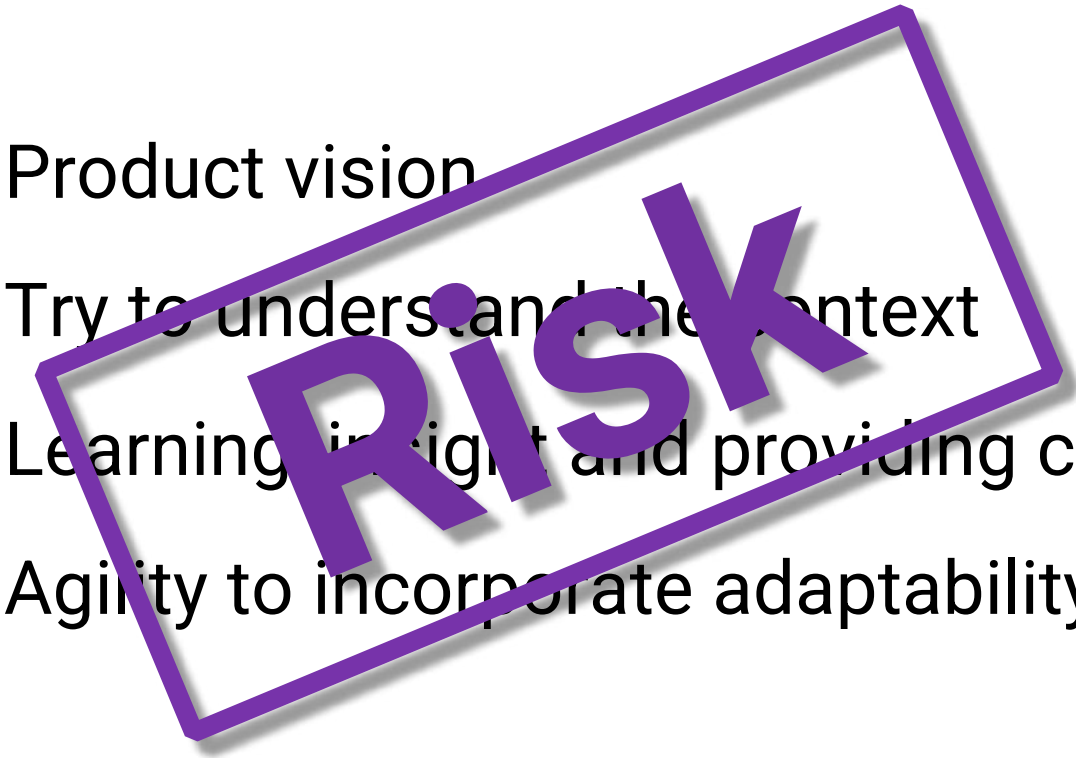
Dealing with this VUCA world

Volatile → Product vision

Uncertain → Try to understand the context

Complex → Learning insight and providing clarity

Ambiguous → Agility to incorporate adaptability



During software development we have to deal with unknown unknowns

- Quality is perception
- Customers and product owners don’t know or can’t imagine what they want
- Development team can’t image what customers will actually do
- Research: building new insights & evolutionary design
- Dealing with complexity, confusion, change, new insights and half answers

We have to **learn** and to deal with **risks**!

~~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~~? 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?

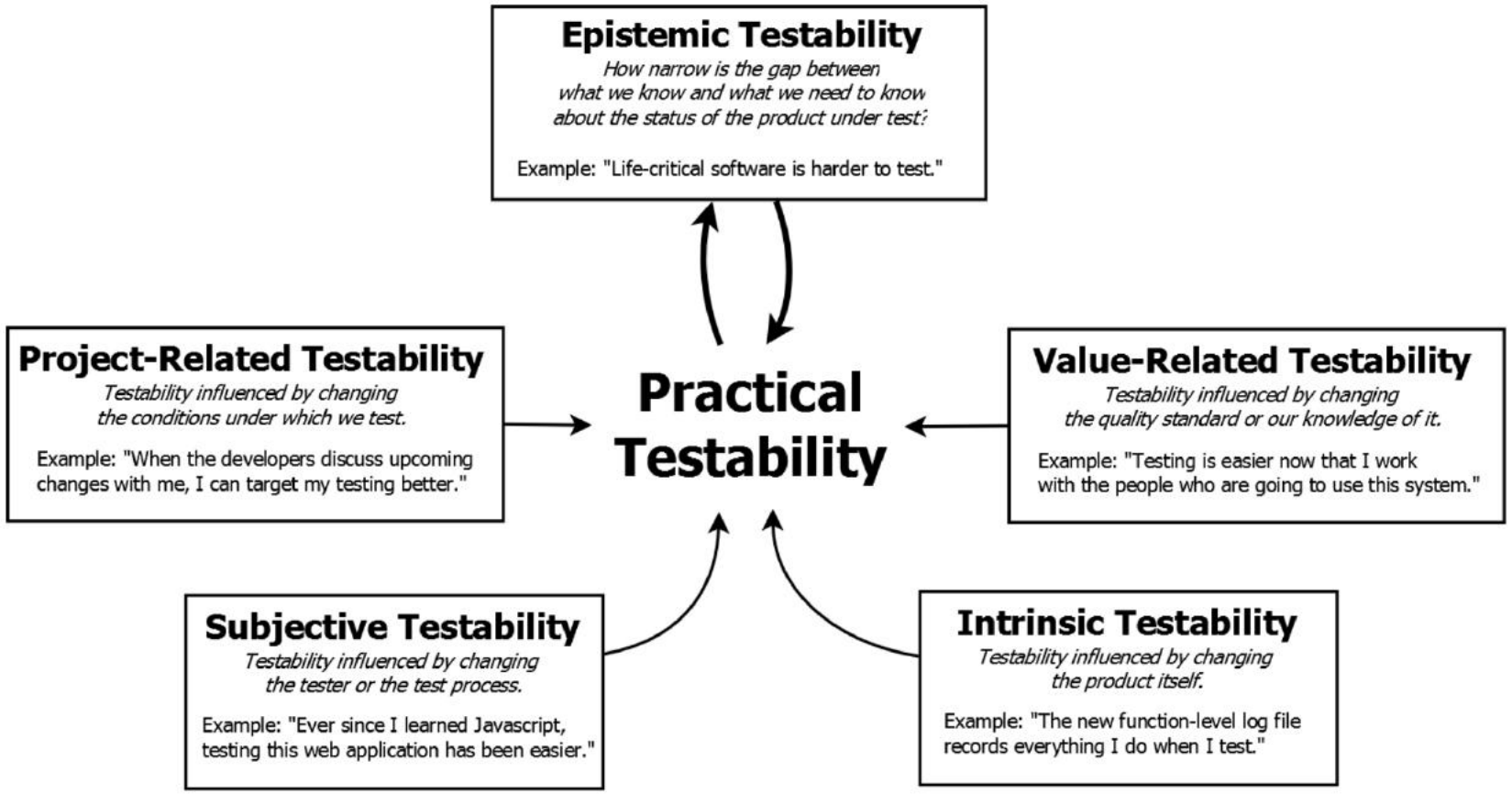
Testability of a product is how easy it is to learn (read: test) by a particular team in a given context.

Learning about testability

10 P’s of Testability

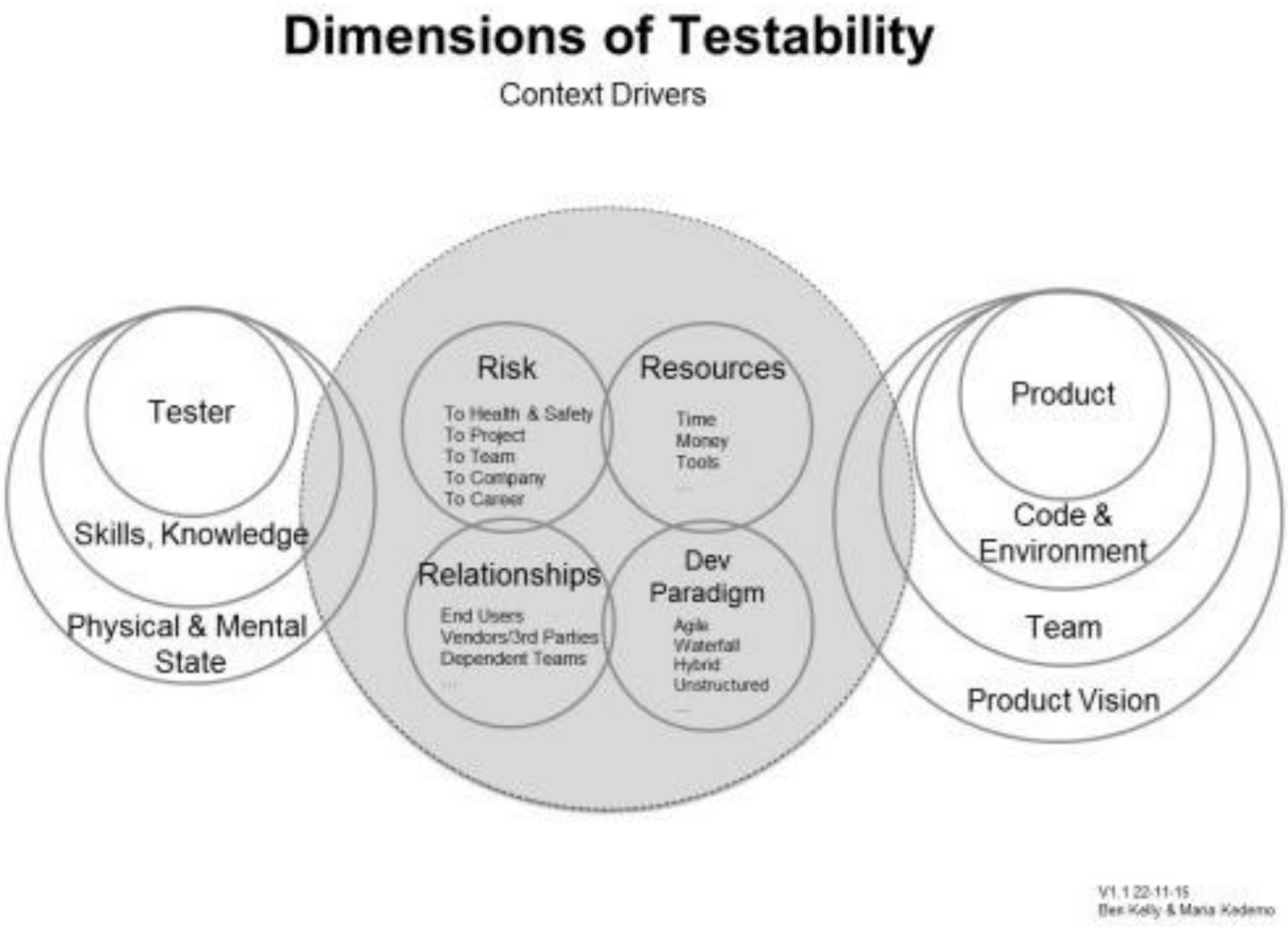
Read more about this here:
Team Guide to Software Testability - Better software through greater testability
By Ash Winter and Rob Meaney
(http://leanpub.com/softwaretestability)

More about testability



Read more about this here:
Heuristics of Software Testability
By James Bach
(https://www.satisfice.com/download/heuristics-of-software-testability)

More about testability



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)

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

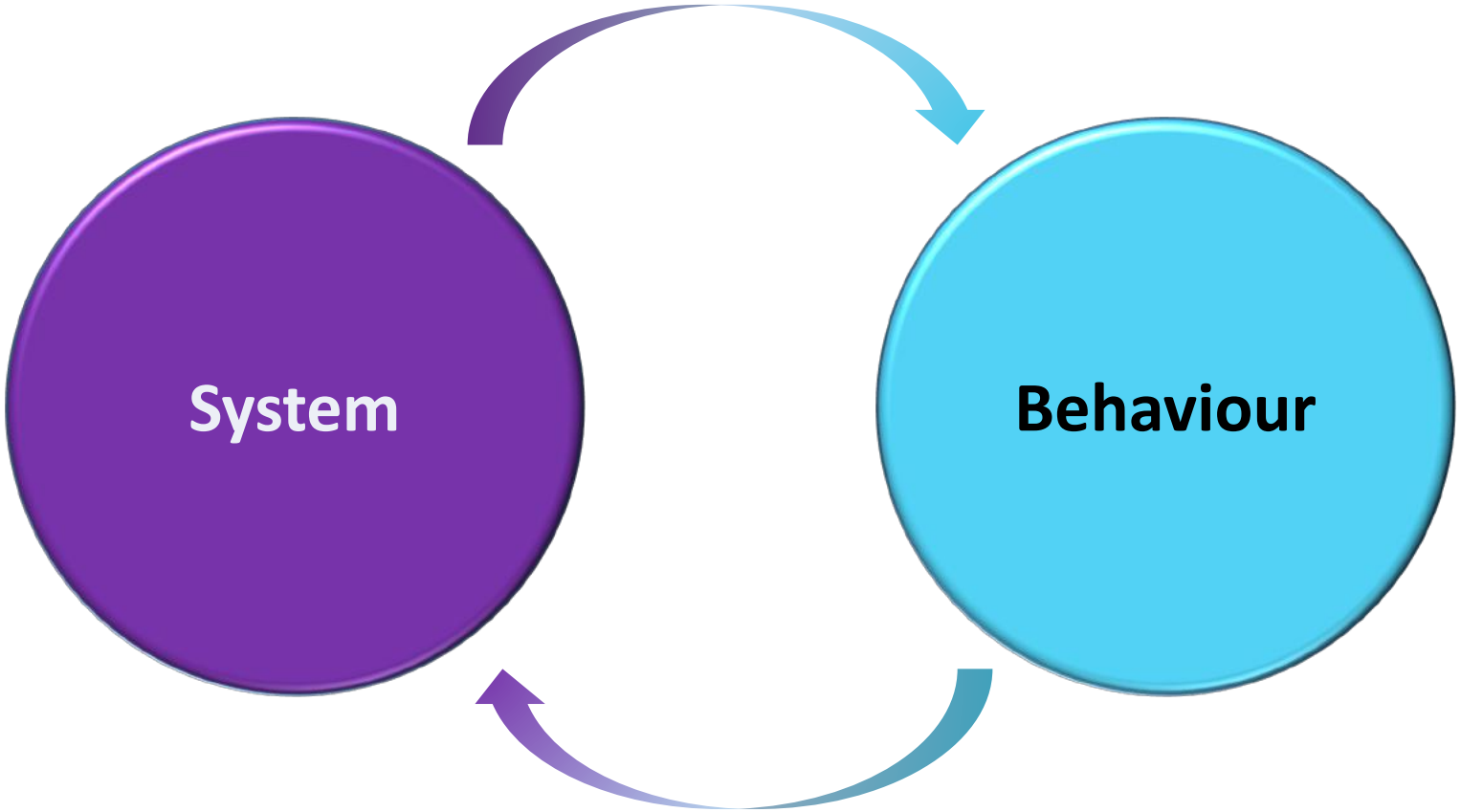
- Team collaboration and skills are key: optimize your SDLC and processes
- Risks are a whole team responsibility
- Test your requirements & assumptions continuously
- Help the team go faster: dev and release pipelines, automated checks, code quality, measuring useful metrics
- Test enough (good is good 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.

It is all about people!

No matter how it looks at first, it's always a people problem.

Organisations are social systems

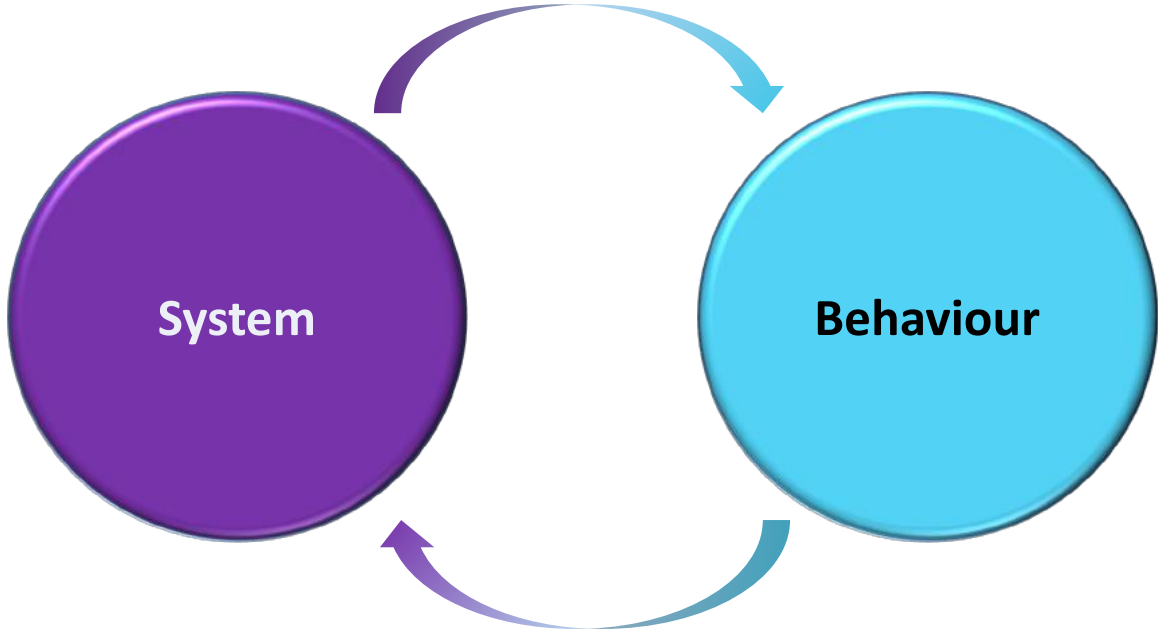
System evokes behaviour



Behaviour determines success of the system

Organisations are social systems

Performance of the organization is determined
1/3 by the system and 2/3 by behaviour



What is a learning organisation?

An organisation that encourages and facilitates learning in order to continuously transform itself to survive and excel in a rapidly changing business environment.

Source: https://study.com/academy/lesson/learning-organizations-characteristics-examples-quiz.html

Learning organisations create conditions for people to learn faster and more effectively so that the organisation and its people can thrive on change.

Source: https://talenttalks.net/enabling-talent-learn-15/

Characteristics of learning organisations

- Senior management encourages learning
- Growth mindset
- Psychological safety
- People-orientated and talent focus (mastery)
- Part of way of working
- Open communication
- Teamwork
- Inspired leadership
- Empowerment
- Learning opportunities
- Customer-orientation

Building a learning organisation

1. Systematic problem solving
   - Plan, Do, Check, Act
   - Fact-based management = using data, measuring improvements
2. Experimentation
3. Learning from past experience
4. Learning from others
5. Transferring knowledge

Continuous learning is a given...

… creating complex products build by and with people for people …

Continuous learning is key in (agile) organisations!

- Learning what our customers need
- Learning what the product actually is
- Learning how to work together
- Learning how to make technology work
- Learning to keep up with constant change
- Learning how to get better at <fill in current need>

Learning

What do you need to learn effectively?

- Showing up
- Focus & attention
- Time
- Curiosity
- Play
- Not afraid to fail
- No limits: no more “learned helplessness”
- Reflection (see my blogpost “the art of reflection”: https://www.huibschoots.nl/wordpress/?p=2824)

Source: Unleash Your Super Brain To Learn Faster by Jim Kwik (https://youtu.be/uT_GcOGEFsk)

Change behaviour (which is learning too)

Behaviour is doing. Not *'knowing'* or '*understanding'* or '*being able to*' but

DOING!

So changing behaviour is changing doing.

Not '*understanding that things must be done differently*' or '*knowing how it can be done differently*', but actually doing it differently.

Behaviour

Thoughts

Believes

Experiences

Convincing

Lure to experiment

Learning in teams:
it’s all about people & loops!

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

So are we still talking about testing?

Comments, feedback, stories or questions ??

Thank you!

Quality Accelerators

ØREDEV

Huib Schoots

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

Who am I?

- Context-driven software tester
- Rapid Software Testing teacher
- Scrum master, team coach & agile expert
- Quality advocate & question asker
- Humanist
- Curious & lifelong learner
- Passionate & energetic people lover
- Fascinated by mindset & behaviour
- Trainer, coach, consultant, writer, speaker
- Storyteller, leader, human, rebel, jiggler
- Lego & Star Wars freak
- Trombone player, photographer, (board) gamer, beer brewer, magician

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/mBC3ssLlJfQ
- 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
- Testability awakens: moving testability into new dimensions - https://bit.ly/testability_mk
- Test Eye Software Quality characteristics - http://thetesteye.com/posters/TheTestEye_SoftwareQualityCharacteristics.pdf
- Interview with Rob Meaney on Quality coaching roadshow - https://www.spreaker.com/show/quality-coaching
- Testability Ask Me Anything (Ash Winter) - https://bit.ly/askmeanything_testability
- Quality Engineering Ask me Anything (Anne-Marie Charrett) - https://bit.ly/askmeanything_qualityengineering
- Rapid software testing - https://www.rapid-software-testing.com
- Building a Learning Organization - https://hbr.org/1993/07/building-a-learning-organization
- Is Yours a Learning Organization? - https://hbr.org/2008/03/is-yours-a-learning-organization
- Collected useful links on my website - http://www.huibschoots.nl/links

Recommended books on (behaviour) change

- Nudge by Cass Sunstein & Richard Thaler
- Atomic habits by James Clear
- Power of habit by Charles Duhigg
- Turn the ship around by David Marquet
- Mindset by Carol Dweck
- Influence by Robert Cialdini
- Redirect: changing the stories we live by by Timothy Wilson
- Switch by Chip & Dan Heath
- Predictably Irrational by Dan Ariely
- The 7 Habits of Highly Effective People by Stephen Covey
- Start with why by Simon Sinek
- Thinking, fast and slow by Daniel Kahneman
- 7 rules for positive, productive change by Esther Derby
- Becoming a technical leader: An organic problem-solving approach by Jerry Weinberg
- Drive by Daniel Pink