

Thought Leadership

Quality Acceleration

“Delivering quality software at speed”



 @Inflectra | #InflectraCon

Huib Schoots

Accelerator @ De Agile Testers



Quality Acceleration

"Delivering quality software at speed"



De Agile Testers



Main points

1. How can we create quality software fast?
2. How can your team manage risks and value?
3. Learn about great collaboration in teams: new ways of working to create valuable software!



Speed?



***Do you deliver higher quality
because you go faster?***

***Or do you go faster because you
deliver higher quality?***





Agenda

- ▶ Value & Quality
- ▶ What our customers want
- ▶ What is holding them back
- ▶ How to accelerate quality?

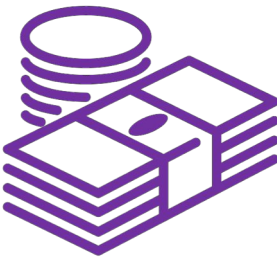




What is value?

- ▶ Value is in the eye of the beholder
- ▶ Something the customer is willing to pay for (Taiichi Ohno, Toyota)
- ▶ The importance or worth of something for someone (Cambridge dictionary)





What is it worth?

- ▶ 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 work
- ▶ **Employee** value → stability, engaging work, healthy environment
- ▶ **Community** value → socially engaged, environmentally responsible



Quality?



► **Quality** is value to some people (who matter)

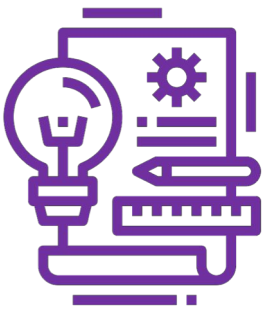
Quality is **NOT** conformance to requirements

Quality is **NOT** the best product possible



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





Software development is R&D

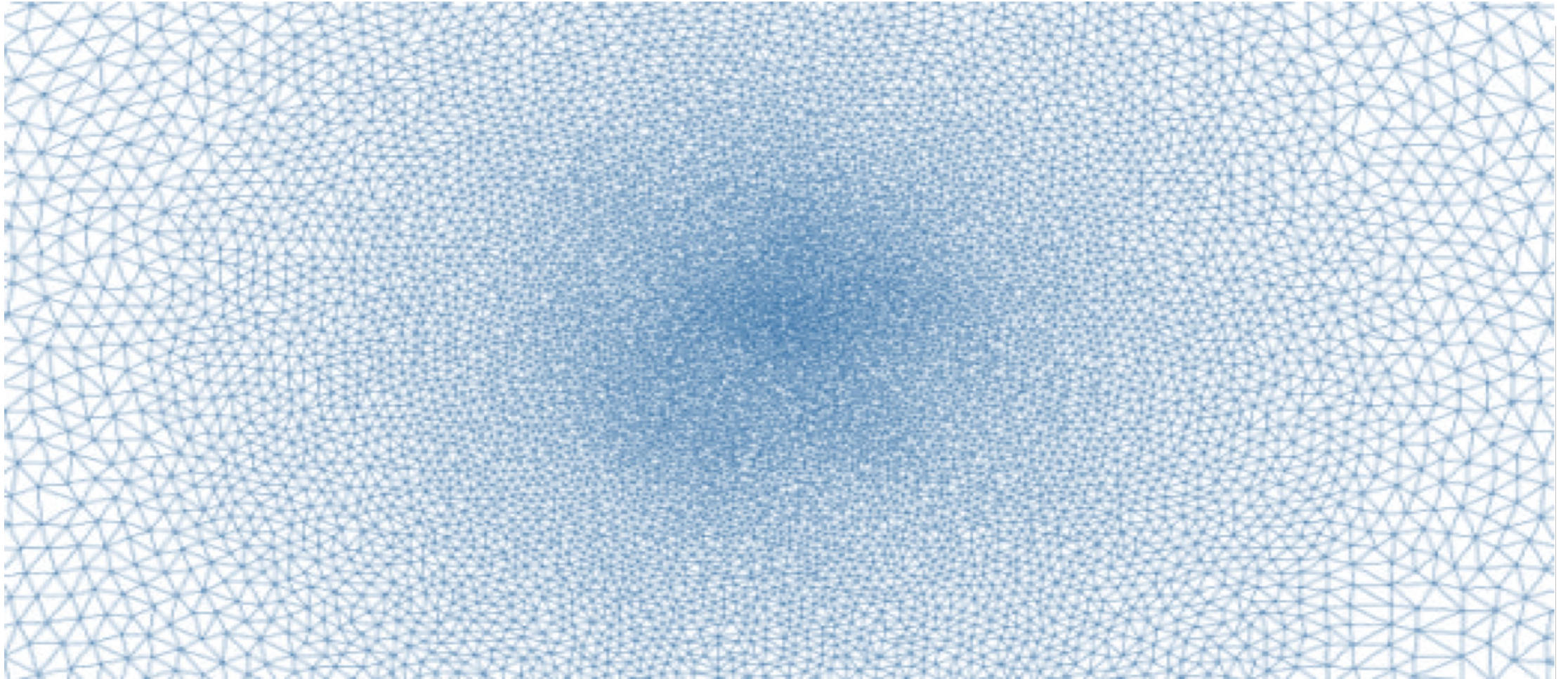
Dealing with unknown unknowns

- ▶ Research: building new insights & evolutionary design
- ▶ Customers don't know or can't imagine what they want
- ▶ Cope with complexity, confusion, change, new insights and half answers

Ergo: we need to **learn** and to deal with **risks**!



Software is complex



People are complex





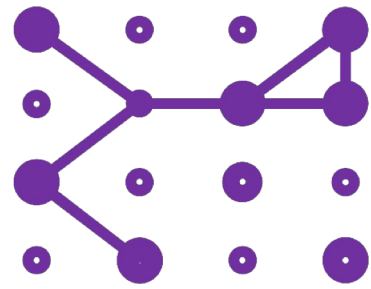
Dealing with this VUCA world

- Volatile** → Product vision
- Uncertain** → Try to understand the context
- Complex** → Learning insight and providing clarity
- Ambiguous** → Agility to incorporate adaptability

Risk



Dealing with complexity

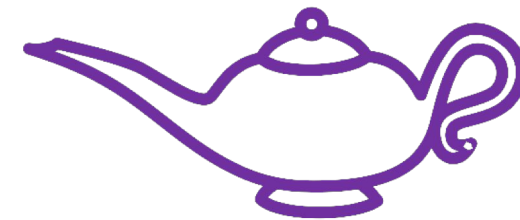


... creating complex products with people for people ...

Continuous learning is key here!

- ▶ 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





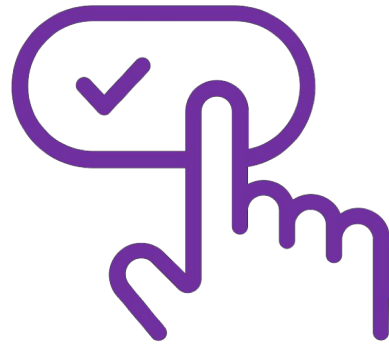
What do our customers want?

- ▶ Cost reduction
- ▶ Reduced lead times
- ▶ Higher quality
- ▶ Higher customer satisfaction
- ▶ Higher reliability
- ▶ New business opportunities
- ▶ Innovation
- ▶ Flexibility
- ▶ Collaboration



But how do we translate this into (IT) solutions?

IT should be an enabler



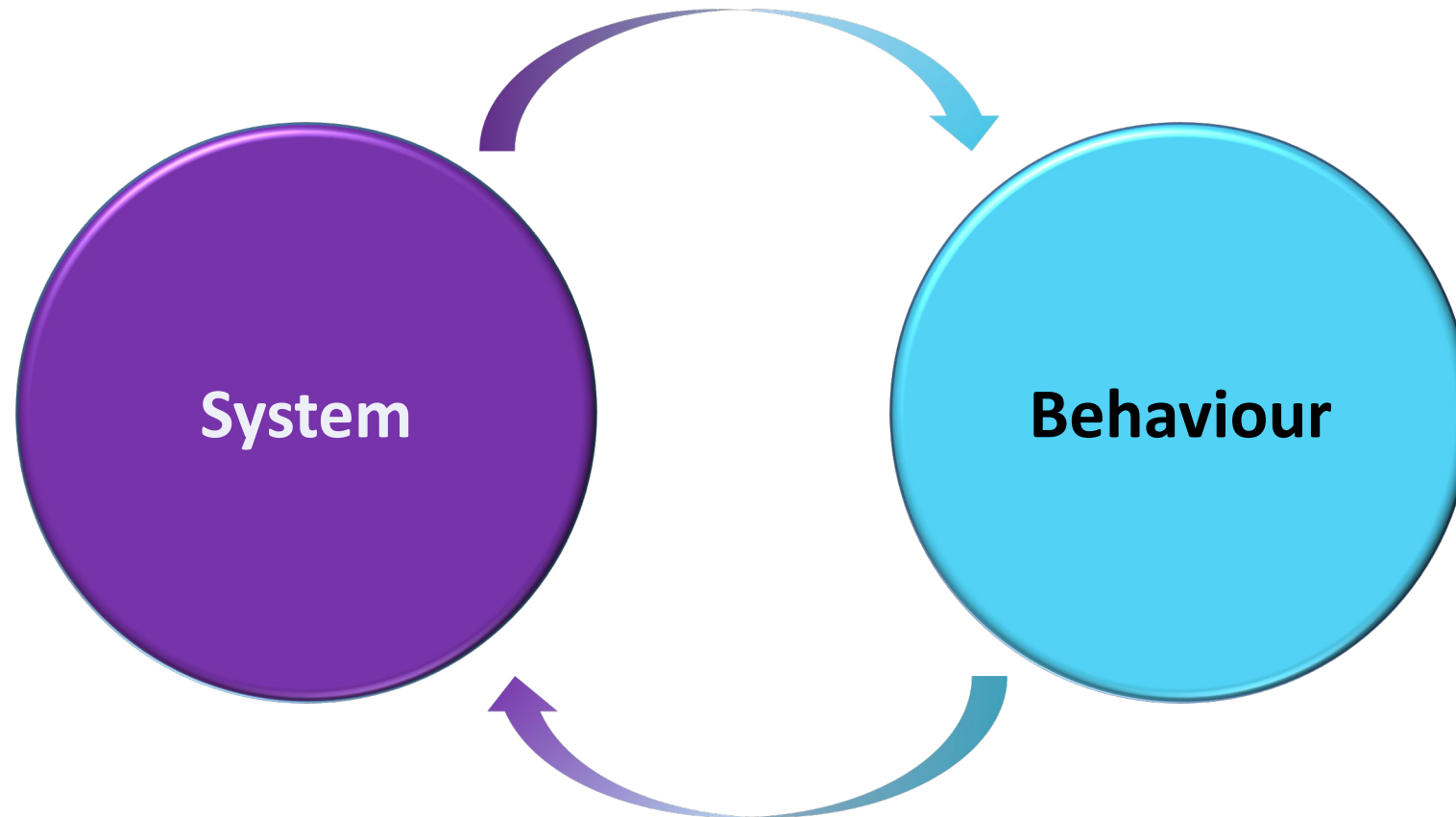
Often IT is a disabler:

- ▶ Complex landscapes
- ▶ Legacy systems
- ▶ Technical debt
- ▶ Sunken costs of systems & tools
- ▶ Not willing to pay for value
- ▶ Lack of skilled people
- ▶ Inhibited learning



Organisations are social systems

System evokes behaviour

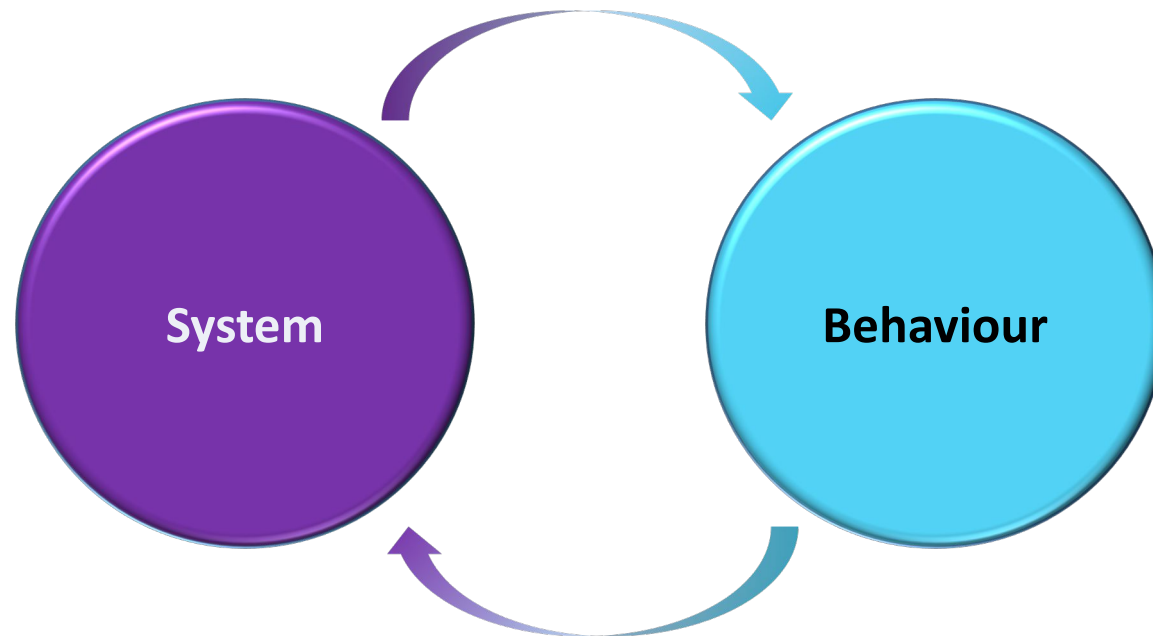


Behaviour determines success of the system

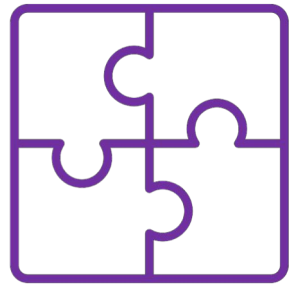


Organisations are social systems

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



Problems → Solutions



It is all about people!



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



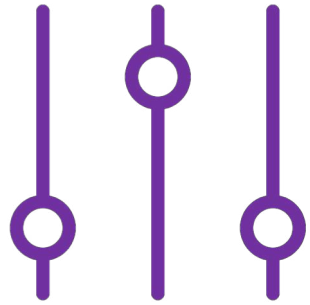


But how?

- ▶ Attune behaviour and mindset
- ▶ Improve communication and collaboration
- ▶ Technological innovation
- ▶ Feedback loops between people
- ▶ Feedback loops between technology and people
- ▶ Solve 'integration hell'
- ▶ Improve PDLC



Fine-tuning



In the past there was a separation between change managers/coaches on one hand and technicians on the other hand...

We need fine-tuning between system, behaviour and technology.

“Nerds who can explain IT”

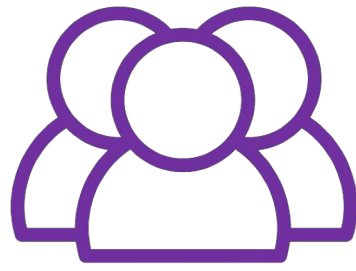




Improve way of working

- ▶ Smooth processes
- ▶ Collaboration
- ▶ Continuous learning
- ▶ Transparency
- ▶ Experimentation
- ▶ Early & fast feedback loops
- ▶ Measuring if you are on the right track

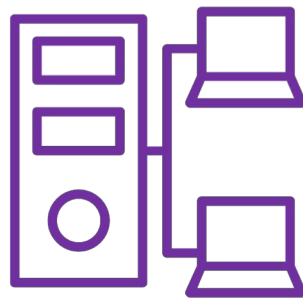




Improve people

- ▶ Right mindset
- ▶ Leadership
- ▶ Diversity in thinking
- ▶ Excellent knowledge and skills
- ▶ Rapid learning & common understanding





Improve technology

- ▶ Intelligent tool chain
 - ▶ Adopt CI/CD
- ▶ Modern architecture
 - ▶ Micro-services, API
- ▶ Automate wisely
- ▶ Tooling:
 - ▶ Modern tooling: e.g. IDE giving direct feedback
 - ▶ Collect data, AI/ML solutions





Improve PDLC

- ▶ Being in control for regulatory bodies
- ▶ Quality of product
- ▶ Quality of process
- ▶ Show desired behaviour
- ▶ Collaboration
- ▶ Pleasant working

Accelerate: ensure a culture and system of focus on an continuous and ever-increasing acceleration



Improve learning: it's all about loops!



Create learning loops (plan-do-check-act) in everything you do

- ▶ Help people understand: test your requirements & assumptions continuously
- ▶ Help programmers go faster: development and release pipelines, automated checks, code quality, measuring the right metrics
- ▶ Help testers go faster: test enough (good is good enough): an integrated quality strategy based on risks
- ▶ Help teams/departments go faster: collect data to make decisions
- ▶ Mitigate risks continuously: people, process and product risks!





Improve balance

- ▶ Speed and risk
- ▶ Exploration and automation
- ▶ You need common sense....
- ▶ Fine-tune. There is no “standard” or “best practice”
 - ▶ Find what works for you/your team/your organization
 - ▶ Adapt methods and frameworks to make them work for you





Improve metrics

- ▶ Is the solution successful for the business?
- ▶ Team performance
 - ▶ DORA metrics
 - ▶ Team health

Find metrics that help you improve...





Take aways

1. Deal with VUCA
2. Fine-tune system, behaviour and technology
3. Continuous Improvement
4. Understanding
5. Not matter how it looks, it is always a people problem



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



Thank you!

De Agile Testers



Thank you

- ▶ Joost Voskuil, Bart Knaack and Peter Schrier for helping me create and reviewing this presentation

Credits

- ▶ Photos by Nathan Cowley, Andrea Piacquadio, Juan Pablo Serrano Arenas, Ketut Subiyanto, Cameron Casey, Fauxels, Daria Shevtsova from <https://www.pexels.com/>
- ▶ Icons by Justin Blake, Ben Davis, Erik Arndt, SBTS, Magicon, Thomas Deckert, Adrien Coquet, Iconsphere, Eucalyp, Verry, Nhor, DinosoftLab, Luketaibai, Ruben Semedo, MRFA, Vectors Market, ProSymbols, Egidio Filippetti, Nithinan Tatah, Joel Avery, Adrien Coquet, Iconsmind.com, Tulpahn, Alice Design, Ainsley Wagoner, Sam Maulidna, Maxim Kulikov, , Iconmark from <https://thenounproject.com/>



Who am I?

- ▶ Accelerator & Quality Coach
- ▶ Product Improver & Problem Solver
- ▶ Team coach & agile expert
- ▶ Scrum master & tester
- ▶ Humanist
- ▶ Curious & lifelong learner
- ▶ Passionate & energetic
- ▶ Trainer, coach, consultant, writer, speaker
- ▶ Storyteller, leader, human, rebel, juggler
- ▶ Trombone & guitar player, photographer, gamer, beer brewer, magician



Huib Schoots

Accelerator: Quality Coach, Agile Expert, Change Maker,
Product Improver, Problem Solver, Tester, Scrum Master

- ▶ huib@deagiletesters.nl
- ▶ [@huibschoots](#)
- ▶ +31 6 24 64 10 33
- ▶ www.linkedin.com/in/huibschoots/
- ▶ www.huibschoots.nl
- ▶ www.deagiletesters.nl

