



---

# Aiming Higher, Levelling Up Your Agile Improvements

by **Joep Schuurkes & Huib Schoots**

---

Who are **you**?

What are **your goals**  
for today?

# Way of working:

## Experiential workshop

(with some short lectures)



This is an **experiential workshop**. Why? You will learn more from an experience than a lecture. And the learning will stick.



Experiential means you participate in exercises and simulations rather than sit in a dark room watching a slide show and listening to a lecture. An exercise or simulation speeds up time: six minutes participating in a simulation may trigger you to have the same experiences as six months in a project. The shorter duration enables you to see your experience - what you did externally and what was happening internally. You can process your experiences and choose whether to continue making the same choices.

A thorough debrief of what happened during each exercise is part of its design. During the debrief, you will learn from us, the other participants, and -most of all- from yourself.

# Agenda

1. Introduction
2. Exercise: ideal company
3. Maturity & maturity models
4. Exercise: ideal standup
5. Complexity, learning & Shu-Ha-Ri
6. Our maturity model
7. Exercise: Serious game
8. Wrap up



# Mission

An investigation of strengths & weaknesses. A starting point for a discussion about potential problems and how to solve them. A journey into more mature people and teams.



# Goals for today

## Learn...

- about improvement in agile
- to create your own model of what is important
- to share what is important to you
- about dynamics to effectively work together (problem solving leadership)

---

# Exercise

# Ideal Company









# Your ideal company

You're about to start your own company, what will it look like?

- I like it when people ... / I want people to ...
- I hate it when people ... / I don't want people to ...
- What does the perfect team look like?
- What does the perfect office look like?
- What are the company's core values?
- What makes work fun?

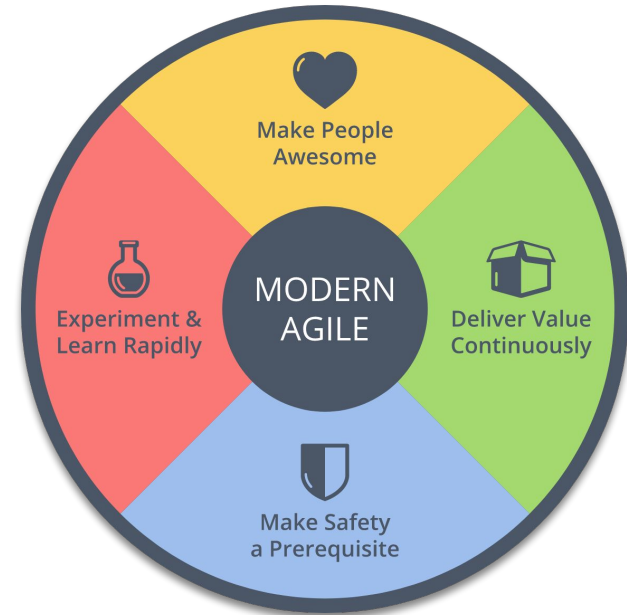
# Exercise - your ideal company

1. Write stickies
2. Cluster stickies
3. Forgotten anything?
4. Apply agile manifesto - modern agile principles



## 12 Principles of Agile Software Development

1. Satisfy the customer through early and continuous delivery.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently
4. Business people and developers work together daily
5. Build projects around motivated individuals.
6. Convey information via face-to-face conversation.
7. Working software is the primary measure of progress.
8. Maintain a constant pace indefinitely.
9. Give continuous attention to technical excellence
10. Simplify: maximizing the amount of work not done
11. Teams self-organize.
12. Teams retrospect and tune behavior



---

# Exercise

## Measuring Maturity



# What is maturity?

And why does it matter?

## Discussion



# What is maturity?

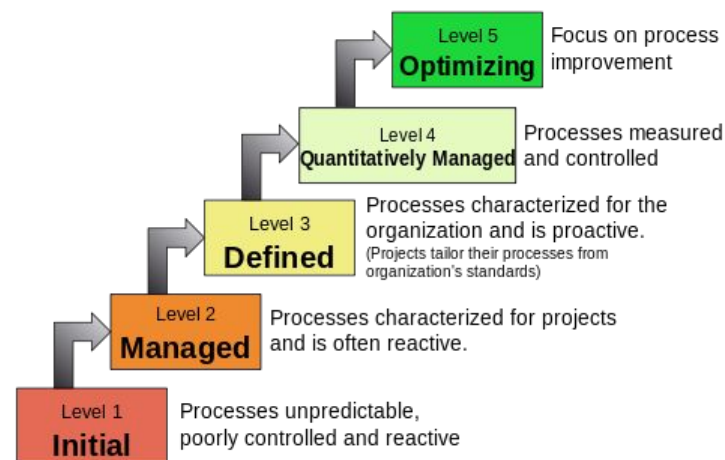
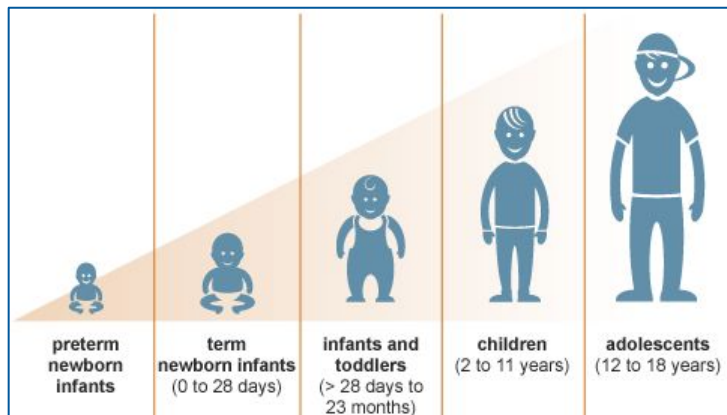
- The quality of behaving mentally and emotionally like an adult  
a very advanced or developed form or state  
(Cambridge Dictionary)
- In psychology, maturity is the ability to respond to the environment  
in an appropriate manner.  
(Wikipedia)
- Maturity is the degree to which a system has realized its potential  
and adapted to its context.  
(James Bach)



# What is a maturity model?

- A maturity model is a business tool used to assess people/culture, processes/structures, and objects/technology  
(wikipedia)
- A maturity model is plan for achieving maturity  
(James Bach)

# How is it done?



Key Area	Initial	Controlled				Efficient				Optimizing		
1 Stakeholder commitment		A	B	B	C	F	H	H		K	M	M
2 Degree of involvement		A	B	C	E	H	H	J		L	L	
3 Test strategy		A	A	B	E	F	F	H		K	L	
4 Test organization		A	D	D	E	I	I	J	J	K	L	L
5 Communication		B	C	C	D	F	F	J		M	M	
6 Reporting		A		C	C	F	G	G		K	K	
7 Test process management		A	A	B	B	G	H	J		K	M	
8 Estimating and planning		B	B	C	C	G	H	I	I	K	L	L
9 Metrics		C	C	C	D	G	H	H	I	K	K	
10 Defect management		A	A	B	D	F	F	H	J	K	L	L
11 Testware management		B	B	D	E	I	I	J		L	L	L
12 Methodology practice		C		D	E	F	H	J	J	M	M	
13 Tester professionalism		D	D	E	E	G	G	I	I	K	K	M
14 Test case design		A	A		E	F	I	I	J	K	K	M
15 Test tools		E	E		E	F	G	G	I	L	M	M
16 Test environment		C	D	D	E	G	H	J	J	L	M	M

---

# Exercise

# Ideal Company



# Your ideal company - part 2

Your company has been running for six months,  
will you have all the things you mentioned in place?

How do you know?

Do you still want the same things?

What will you do next?

# Exercise - your company 6 months later

Think about evaluating your new company after 6 months...

Would you use a maturity model?



---

# Experience report & lecture...

## Maturity Models

# A story about testing maturity

- Manager focused on skills, but still... what about maturity?
- TPI Next or TMMi?
- We'll have to figure something out ourselves.



***Rabobank***

# Why TPI and TMMi don't work for us.

- Models like TPI/TMMi don't focus on the people doing the work
- Fixed model: everybody uses the same model
  - The model is pre-defined, doesn't take into account the context
  - The authors of the model have determined what is important and how to measure the areas
  - Growth is predefined: this implies that all organizations have the same growth model
- The model itself is not part of the evaluation
- Scores are the only way to express what is important and what needs to improve



# Quotes from TMMi

- A maturity level is a well-defined evolutionary plateau towards achieving improved organizational processes.
- The evolutionary testing model of Gelperin and Hetzel has served as a foundation for historical-level differentiation in the TMMi ... Testing has, according to Gelperin and Hetzel, since progressed to a “prevention-oriented” period, which is associated with current best practices and reflects the highest maturity level of the TMMi.
- The achievement of a specific maturity level must mean the same thing for different assessed organizations.
- Degree of process improvement across a predefined set of process areas in which all goals in the set are attained.

*TMMi Reference Model, release 1.0, 2012*

# Gelperin and Hetzel

*SPECIAL ARTICLE*

## **THE GROWTH OF SOFTWARE TESTING**

*We can trace the evolution of software test engineering by examining changes in the testing process model and the level of professionalism over the years. The current definition of a good software testing practice involves some preventive methodology.*

**DAVID GELPERIN and BILL HETZEL**

*June 1988 Volume 31 Number 6*

# Our view on maturity assessments

*The end result of a “maturity assessment” should be valuable information on what is your vision on good testing and to what degree you live up to that vision.*

- Facilitating discussion
  - Creating insight in what is important to you.
  - Creating insight in how good you are at this.
- Helps discovering possible areas of improvement
- A heuristic model

---

# Exercise

## The perfect stand-up



# Let's look at a simple practice

*What is a  
perfect  
stand-up?*



# Your perfect stand-up (daily scrum)

What is a perfect stand-up?

How can you tell if a stand-up was excellent?

What's the purpose of having a stand-up?

What's the end result of a perfect stand-up?

When would you decide not to do one?

# Exercise - the perfect stand-up

1. Write stickies
2. Create a shared vision
3. Share vision with other group(s)



# Debrief (1)

Why are those things important to you?

Do you want to make your stand-up perfect?

How much effort do you want to put into improving it?

Do you ever observe your own team? How?

Do you need to define any actions? Or is a shared vision sufficient?



# Debrief (2)

Do you all agree on what makes a perfect stand-up?

How do you know?

How did you get there?

---

# Lecture...

# Wholeness

# &

# Shu-Ha-Ri

# Continuous Improvement

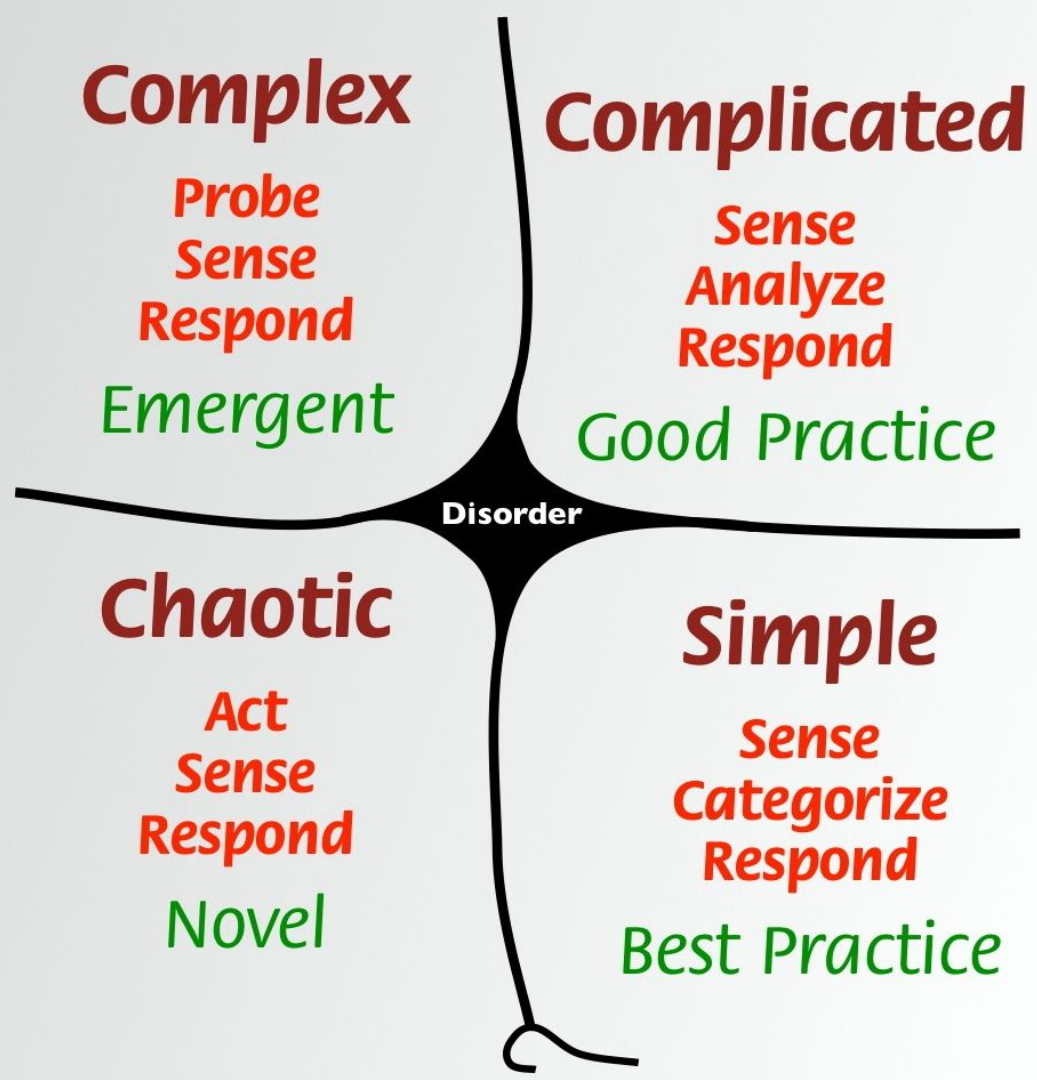
## Retrospectives & sprint reviews!

- Often focused on the last sprint
- Looking at what we already do
- Retro “full” with stuff to fix, no time for bigger picture
- Teams almost never walk through the whole way of working

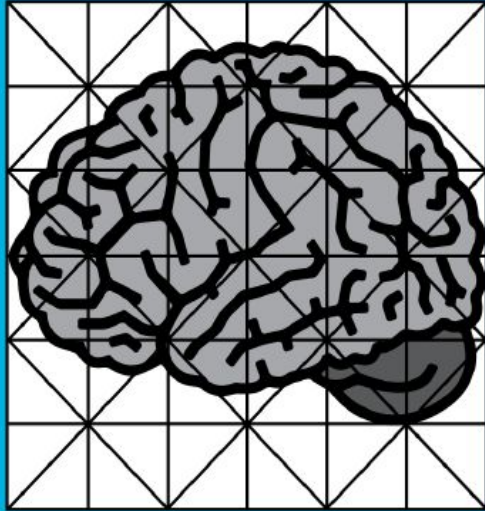


# Cynefin

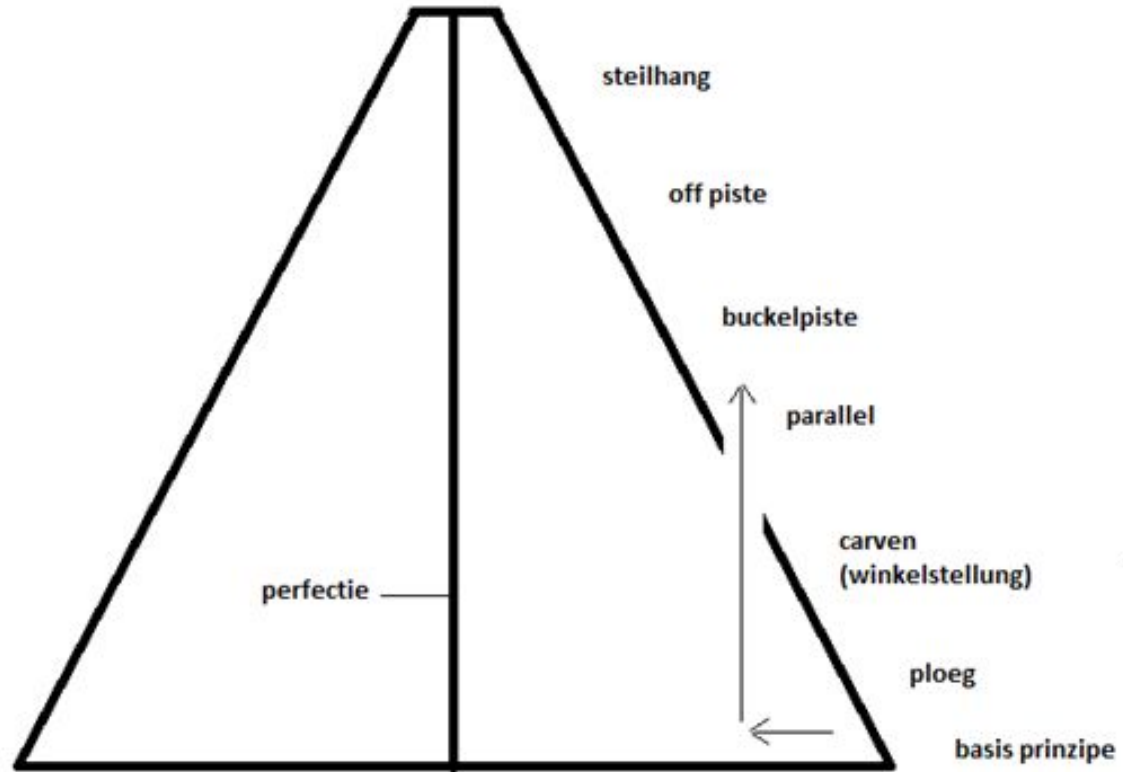
A sensemaking  
framework



# Our Complexity Thinking Guidelines



1. Address complexity with complexity
2. Use a diversity of perspectives
3. Assume subjectivity and coevolution
4. Steal and tweak
5. Expect dependence on context
6. Anticipate, explore, adapt
7. Shorten the feedback cycle
8. Keep your options open



守破離

**Shu - follow the rule (*obedience*)**

**Ha - break the rule (*divergence*)**

**Ri - be the rule (*transcendence*)**

---

# Our 'Method'

## AQuA







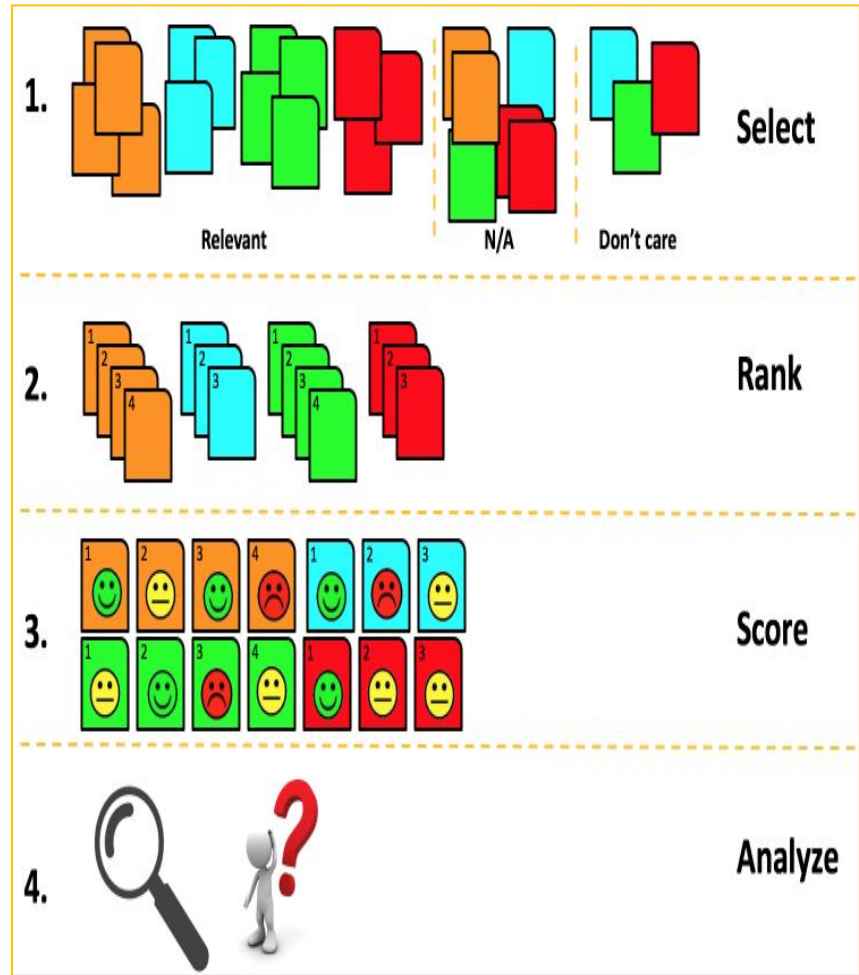
# AQUA: Improving in agile

## Lessons:

- Struggle with model: lists & categories
- Discovery by experimenting
- Holistic view

# Approach

1. Decide on the relevant of the criteria
2. Stack rank the relevant criteria
3. Score the relevant criteria
4. Analyze the results



# Five categories

Process

Team

Skills & Knowledge

Practices

Culture & Context

# The cards



---

# Exercise

## AQuA



# Exercise - AQuA

1. Everyone gets 5 cards from **'Team'**  
Play AQuA applied to your own context (use post-its as cards)
2. Per group: one person shares their result  
Others ask critical questions and offer feedback
3. Compare the results: what do you notice?



# Debrief

- **What did you notice?**
  - individual part
  - sharing part
  - comparison part
- **Our goal: your experience**
  - usefulness
  - difficulties

---

# Sprint review





# Sprint review

1. What do you guys think of AQuA?
2. How are you going to use this?
3. How can we improve this?

What would AQuA 2.0 look like?

---

# Wrap-up



# What did we do today?

## Exercises:

- Ideal company
- Ideal company after 6 months
- Perfect standup
- AQuA - Team

## Topics:

- Maturity
- Improvements
- Complexity
- Shu - Ha - Ri

# So what does that mean?

Insights

Confirmations

Questions

# Now what will you do?

What would you like to apply?

What will you apply?

What requires more thought?

# Our biggest lesson

## Sharing stories and vision

- What is important?
- How are we doing?
- What are we aiming for?





# Please help!

<https://www.savinglinnea.com/en/home/>

# Thank you!



**AGILE**  
TESTING DAYS

**EUROPE'S GREATEST  
AGILE SOFTWARE  
TESTING FESTIVAL!**

November 13-17, 2017 Potsdam, Germany





# Huib Schoots

@huibschoots

huib.schoots@improveqs.nl

www.huibschoots.nl

---



# Joep Schuurkes

@j19sch

j19sch@gmail.com

testingcurve.wordpress.com

---

# References

- Testing maturity in an agile/CDT environment  
<http://www.huibschoots.nl/wordpress/?p=2543>
- Testing maturity in an agile/CDT context  
<http://testingcurve.wordpress.com/2017/02/05/testing-maturity-in-an-agilecdt-context/>
- Blogs by Michael & James:  
<http://www.developsense.com/blog/2009/10/maturity-models-have-it-backwards/>  
<http://www.developsense.com/blog/2011/10/xmmwhy/>  
<http://www.satisfice.com/blog/archives/581>

# Retrospective ideas

- Book: Agile Retrospectives: Making Good Teams Great
- Book: Getting Value Out of Agile Retrospectives - A Toolbox of Retrospective Exercises
- Fun Retrospectives (book & website)  
<http://www.funretrospectives.com/>
- Retrospective Exercises Toolbox  
<https://www.benlinders.com/exercises/>
- 8 tips for better retrospectives  
<http://properosolutions.com/2009/08/eight-tips-for-better-retrospectives/>
- 7 cardinal sins of retrospectives  
<https://medium.com/@sskorc/7-cardinal-sins-of-retrospectives-419364a258c3>