

Building High Quality Complex Software

Johan Hoberg



Introduction

- Johan Hoberg
- Where I have been
 - Sony Ericsson / Sony Mobile - 10 years
 - King - 4 years
- What I am passionate about
 - Quality
 - Software testing
 - Games
 - Agile / Leadership

What is this presentation about?

- Building high quality complex software is difficult, and it will only get harder and harder as complexity grows
- Outline of some of the principles that guide me in my everyday work and what practical implications they have

Overview

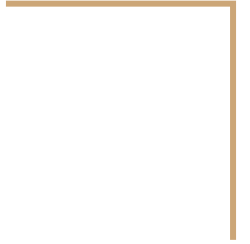
Understanding Value

Understanding Complexity

Understanding Motivation

Understanding Test

Understanding Value



Quality

“Quality is value to some person.”

Gerald Weinberg

Stakeholders & Customers

“High quality means high value to stakeholders and customers - this cannot be achieved without continuous, clear and open communication.”

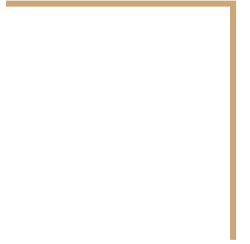
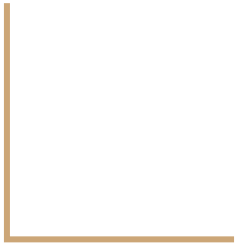
Metrics

“Metrics drive behaviour. Ask the right questions. Are your metrics value-driven?”

Wasteful artefacts

“Always question the value of artefacts you create - why are you creating them, and do they fulfill their purpose efficiently?”

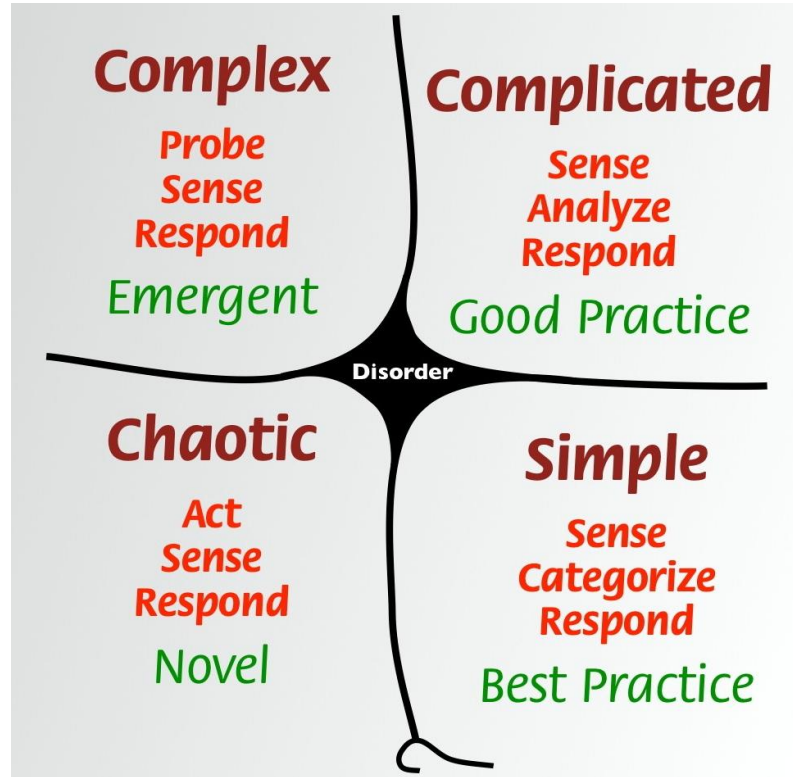
Understanding Complexity



Complexity

“Complexity is generally used to characterize something with many parts where those parts interact with each other in multiple ways, culminating in a higher order of emergence greater than the sum of its parts.”

Cynefin



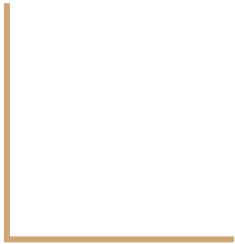
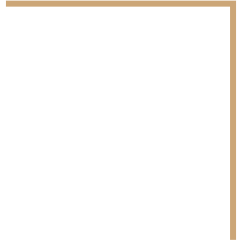
Planning complex problem solving

“You cannot plan for solving complexity up-front. It has to be done iteratively.
Only in retrospect do you know exactly how long it took.”

Solving Complexity

“Complex problems are best solved by intrinsically motivated experts, enabled by the structure they operate within.”

Understanding Motivation



Intrinsic Motivation

- Dan Pink has shown in experiments that coming up with innovative solutions is best done by people who are intrinsically motivated to solve the problems
- Self-determination Theory
 - Autonomy
 - Relatedness
 - Competence
- Drive
 - Autonomy
 - Purpose
 - Mastery

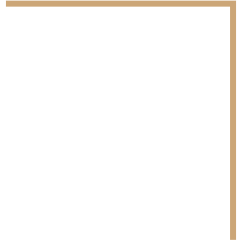
Building genuine teams

“A team that prevails over time, with a strong sense of ownership, and an open feedback culture is more efficient at building high quality software.”

Team Size

“Harvard research determined that optimal team size is 4.6 people. Experiments with high performing Scrum teams have repeatedly shown that 4 or 5 people doing the work is the optimal size.”

Understanding Test



Quality Intelligence, not Quality Assurance

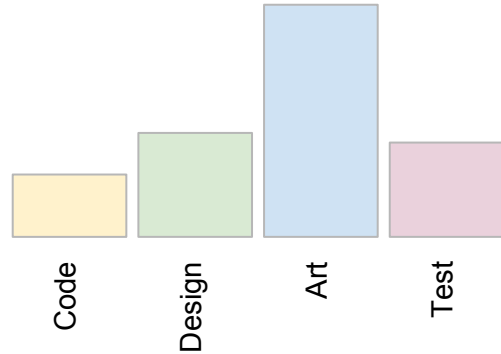
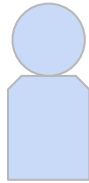
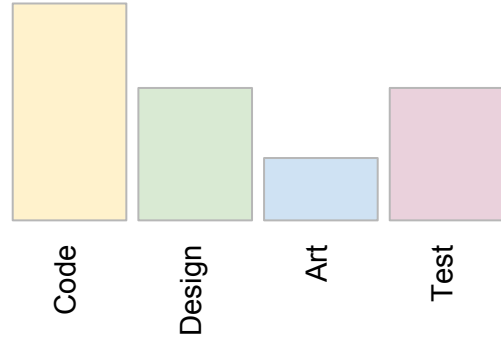
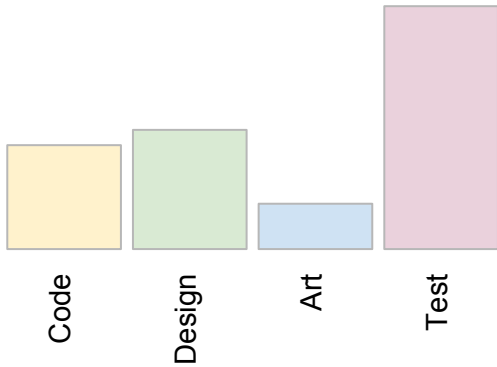
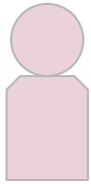
“A tester’s job at its core is to uncover product information, analyze this information, and provide that product intelligence to stakeholders - not to assure quality, mostly because this would be an impossible task.”

Everyone can test

“Everyone can and should test, but not everyone can solve complex test problems.”

Cross Functional Teams

- In an Agile game team, everyone is a game developer - but everyone has different competences



Test Competence

“Exploring and investigating complex systems efficiently, and setting up the right experiments, is far from trivial, and should not be treated as such.”

Quality is built in

“Quality is not something you assure at the end, it is something you build in from the start.”

Necessary risks

“When dealing with complexity you can never mitigate all risks - you must always analyze at the cost and value of further risk mitigation.”

The right tools for the right job

“Some tasks are better performed by machines, some better performed by humans - automated the former, not the latter.”

Some practical implications

- Developers responsible for basic testing
- Testers focus on complex test problems, and on coaching and supporting the team
- Everyone participates in exploratory testing sessions
- No scripted test cases, and no traditional test reports or plans
- All testing is risk-based
- A release process that enables high quality
- Metrics focused on value



Some practical implications

- Test and quality discussions are part of the entire development life cycle
- Test automation is owned by developers, and used only when it adds more value than it costs
- Close cooperation with Customer Support and stakeholders
- Acceptance for and trust in our test strategy throughout the organization
- No micromanagement
- Scrum



Contact & Stuff

Email: Johan.hoberg@king.com

LinkedIn: <https://www.linkedin.com/in/johanhoberg/>

Slideshare: <https://www.slideshare.net/JohanHoberg>

Gamasutra: <https://gamasutra.com/blogs/JohanHoberg/990003/>

King Tech Blog: <https://techblog.king.com/author/johan-hoberg/>