

ThoughtWorks®

Get Faster Everywhere

SIMPLIFY THEN ADD LIGHTNESS

Kelsey van Haaster

kelseyvh@thoughtworks.com

@kelseyvh

**“Simplicity, the art of maximising
the work not done; is essential”**

The principles of the Manifesto for Agile Software Development

ABOUT ME

PAST

More than 20 years of total experience in Information Technology

More than 15 years experience working with Agile software development

PRESENT

TechOps - ThoughtWorks

PM/IM/PO leading a 100% distributed team

FUTURE

Completion (estimated mid 2017) of a PhD with Charles Sturt University

Investigating the consequences of various organisational approaches to transitions towards business Agility

OVERVIEW

- **Principle 10 - what does it all mean?**
 - *Simplicity what do we mean?*
 - *Why maximise the work NOT done, rather than minimise the work done?*
 - *Why is principle 10 essential?*
- **Anti-patterns for principle 10**
- **What does principle 10 look like in practice?**

SIMPLICITY

PHILOSOPHY

Occam's Razor

The idea that, in trying to understand something, getting unnecessary information out of the way is the fastest way to the truth or to the best explanation.

William of Ockham (1285-1349)

SIMPLICITY

INDUSTRIAL DESIGN

“A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away”

Antoine de Saint-Exupery

EXPERIENCE DESIGN

“Simplicity is the ultimate sophistication.”

Claire Booth Luce (1903 -1987)
(often misattributed to both Leonardo da Vinci and Steve Jobs)

**Simplicity is achieved by maximising
the work not done.**



MAXIMISING THE WORK NOT DONE



“Adding power makes you faster on the straights; subtracting weight makes you faster everywhere.”

Colin Chapman, Founder Lotus cars.

FROM A TECHNOLOGY PERSPECTIVE

Principle 10 expresses the connection between Agile and Lean:

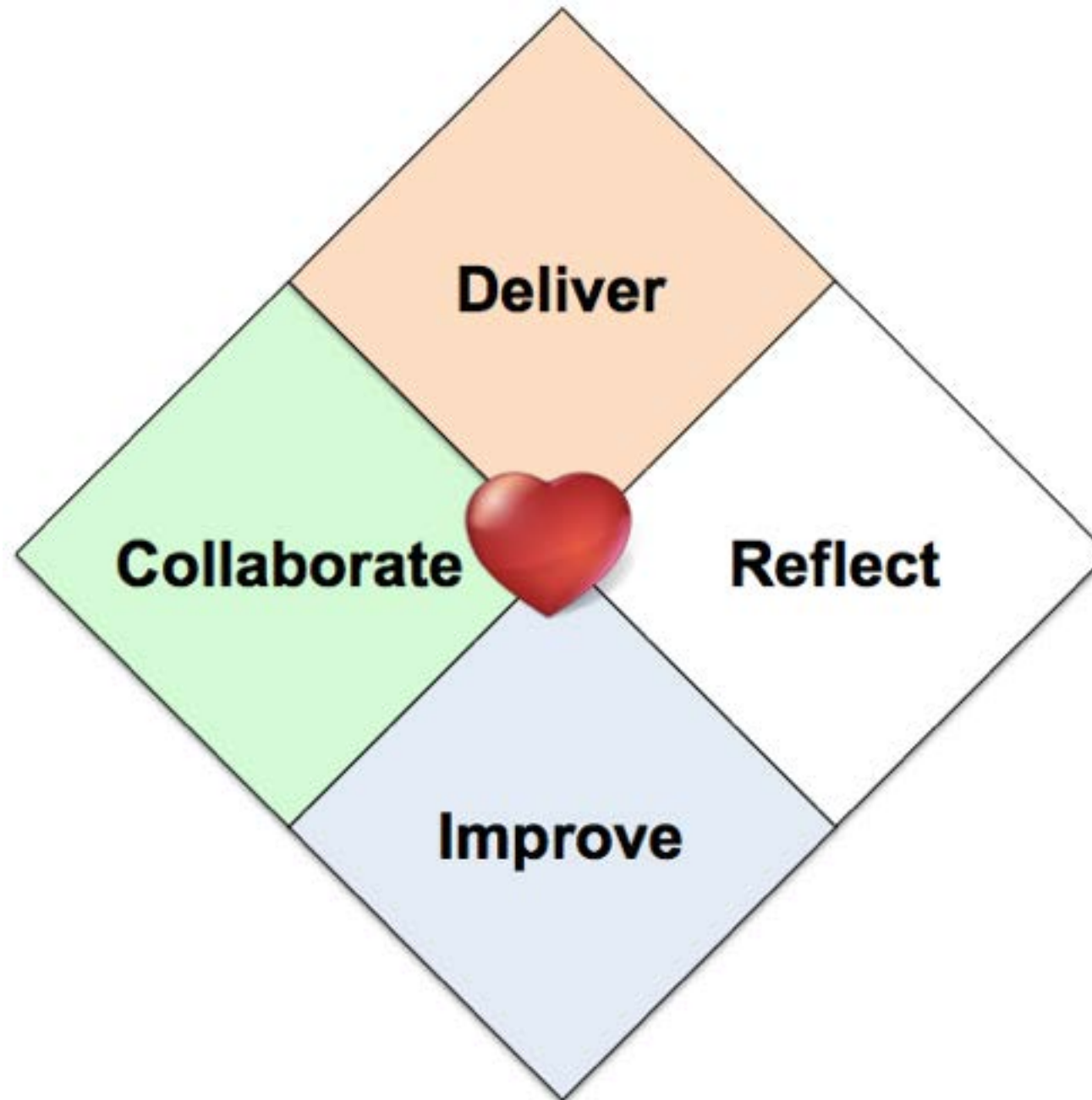
“Strive for perfection by continually removing waste”

WHY IS PRINCIPLE 10 ESSENTIAL?

- Some recent comments
 - Alistair Cockburn - Heart of Agile
 - Dave Thomas - Agile is dead

HEART OF AGILE - ALISTAIR COCKBURN

“Agile has become overly decorated. Let’s scrape away those decorations for a minute, and get back to the centre of agile.



AGILE IS DEAD - DAVE THOMAS

“Long Live Agility”



What to do:

1. Find out where you are
2. Take a small step towards your goal
3. Adjust your understanding based on what you learned
4. Repeat



How to do it:

When faced with two or more alternatives that deliver roughly the same value, take the path that makes future change easier.



PRINCIPLE 10 - ANTI-PATTERNS

How did we get into this state in the first place?

BEYOND FEATURES - DAN NORTH

We have based our model of software engineering on civil engineering

When we consume Civil Engineering -

- Bigger is better, more is better (more features)
- We front load our risks to avoid surprises (not enough bricks)

When we consume surgery

- We want the minimum possible amount (nobody wants more surgery)
- But we don't want to front load our risk to avoid surprises (more surgery)

MODERN TAYLORISM

“We will win, and you will lose. You cannot do anything about it because your failure is an internal disease. Your companies are based on Taylor’s principles. Worse, your heads are Taylorized, too.”

1986, Konosuke Matsushita the founder of Panasonic

THE NEW FRANKENSTEIN

Scrumbutt
Cowboy-Agile
Water-Scrum-Fall
Wagile
Agilefall

IN PRACTICE...

PRINCIPLE 10 + SOFTWARE DEVELOPMENT

Principle 10 Opportunity:

Write Less Code

PRINCIPLE 10 + SOFTWARE DEVELOPMENT

Principle 10 opportunity

Use a DevOps model - You build it, you run it

PRINCIPLE 10 + SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Minimise the time between comprehension and completion

PRINCIPLE 10 + SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Use Technical Debt to your advantage

PRINCIPLE 10 + SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Combine the paradigm of TDD with the customer focus of BDD

PRINCIPLE 10 - BEYOND SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Use Product Teams not Project Teams

PRINCIPLE 10 - BEYOND SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Lean Governance

PRINCIPLE 10 BEYOND SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Use Continuous Feedback

PRINCIPLE 10 - BEYOND SOFTWARE DEVELOPMENT

Principle 10 Opportunity

Track Failure Demand

“Maximize the outcome and minimize the output.”

Jeff Patten