# What Your Mother Never Told You About Agile Development

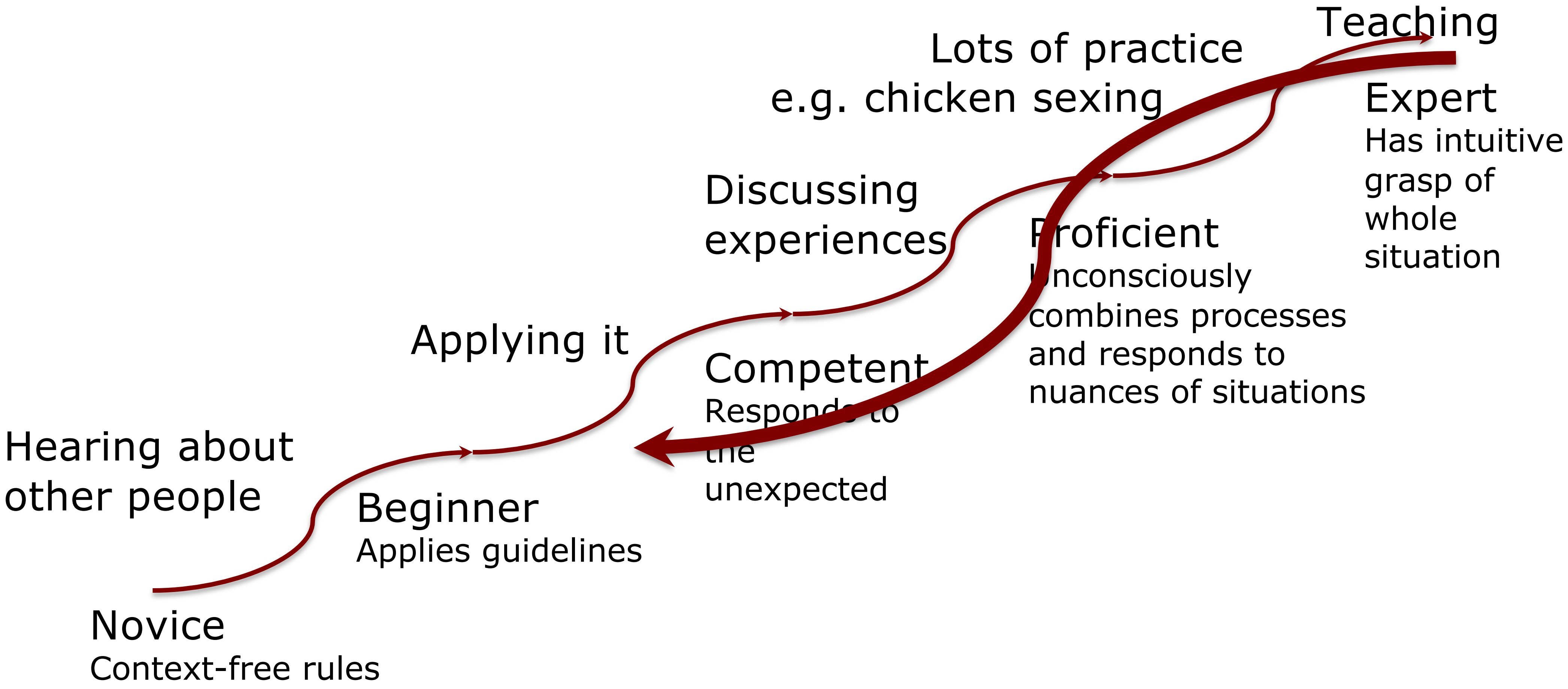
Aino Vonge Corry, ph.d.
Metadeveloper
@apaipi

YOW! London 2022

# Once upon a time in Denmark

# But there be Dragons

# Dreyfus model of skill acquisition



Novice
Context-free rules

Hearing about other people

Beginner
Applies guidelines

Applying it

Competent
Responds to the unexpected

Discussing experiences

Proficient
Unconsciously combines processes and responds to nuances of situations

Lots of practice
e.g. chicken sexing

Expert
Has intuitive grasp of whole situation

Teaching

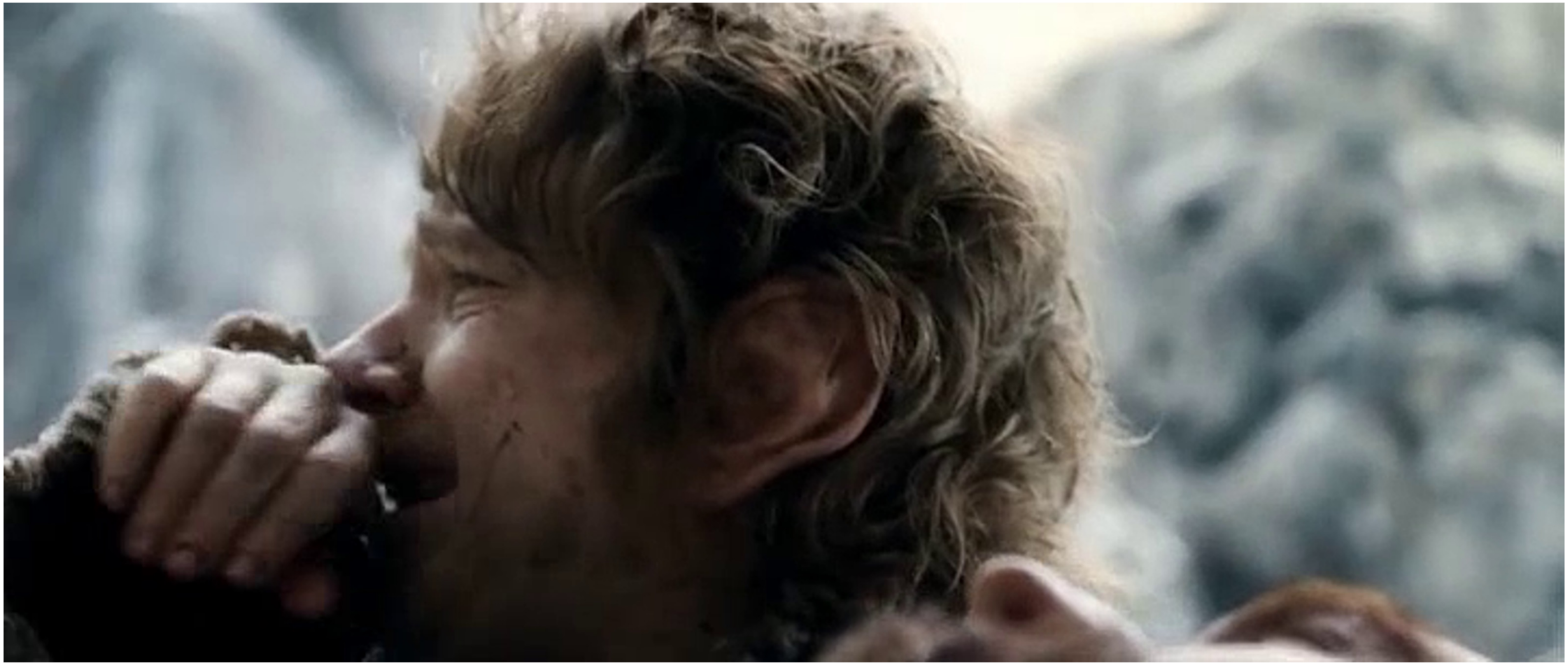
# Feedback cycles

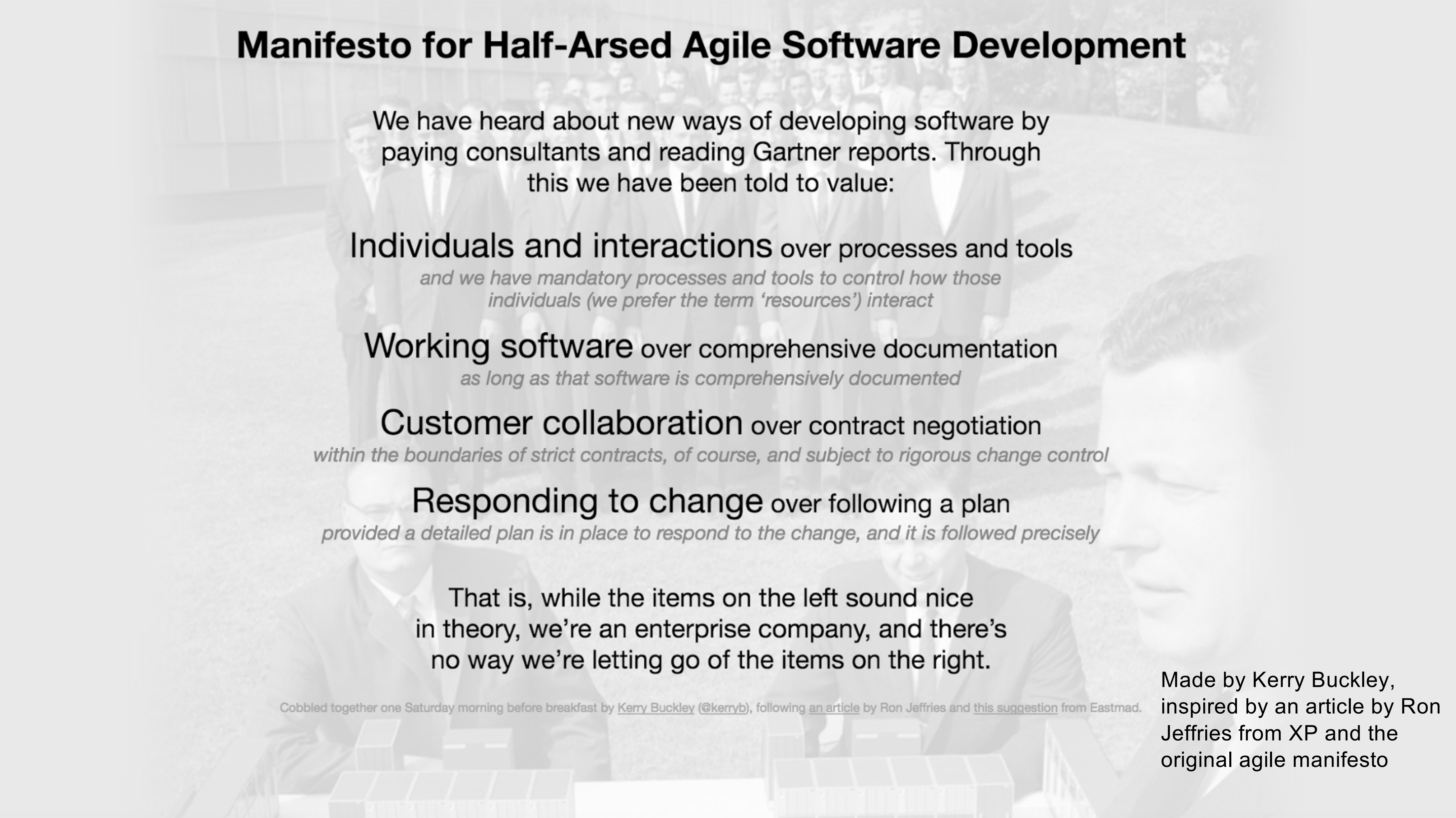
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Business people and developers must work together daily throughout the project.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

# Sad Hobbit





# Manifesto for Half-Arsed Agile Software Development

We have heard about new ways of developing software by paying consultants and reading Gartner reports. Through this we have been told to value:

Individuals and interactions over processes and tools
*and we have mandatory processes and tools to control how those individuals (we prefer the term ‘resources’) interact*

Working software over comprehensive documentation
*as long as that software is comprehensively documented*

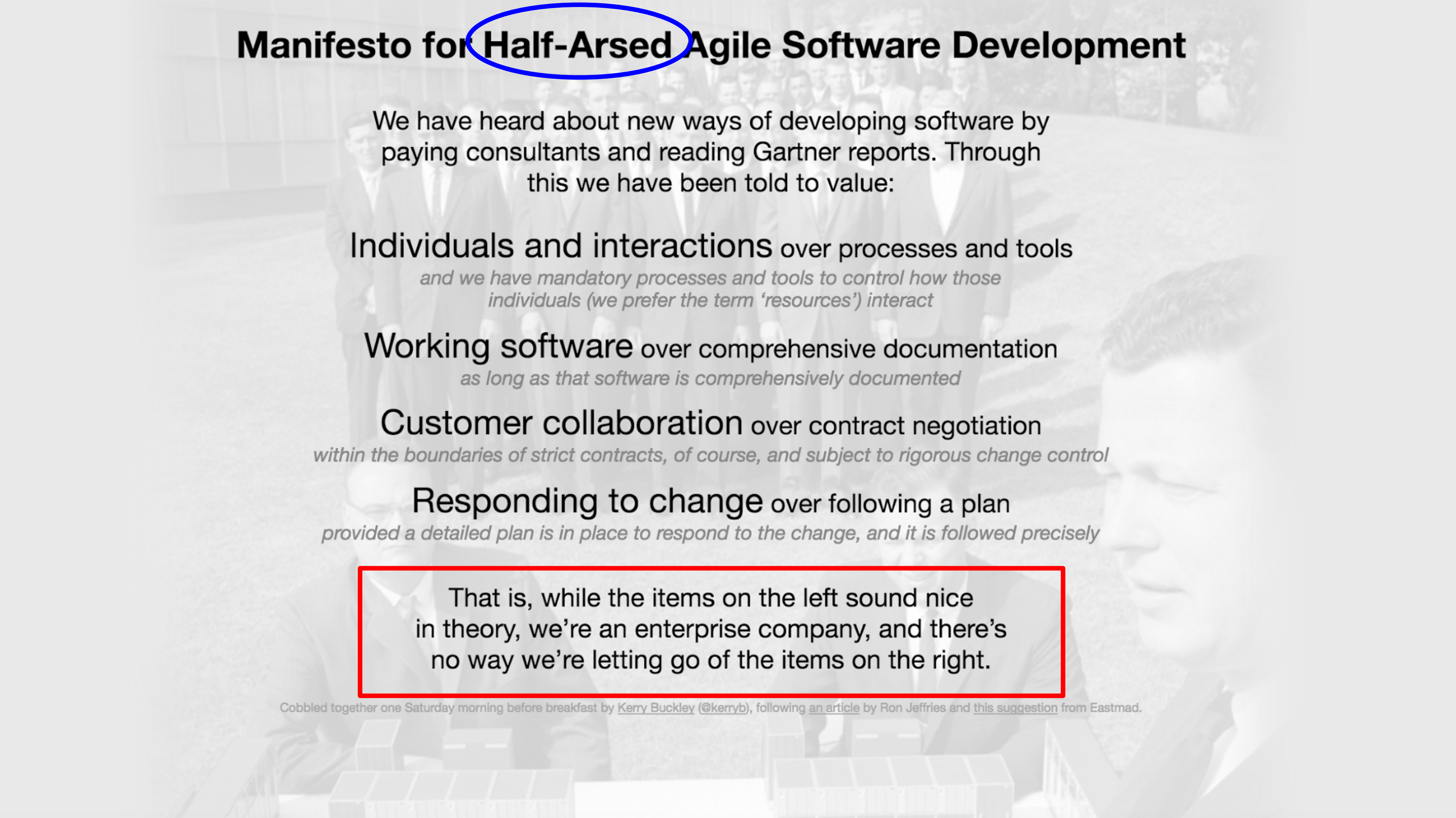
Customer collaboration over contract negotiation
*within the boundaries of strict contracts, of course, and subject to rigorous change control*

Responding to change over following a plan
*provided a detailed plan is in place to respond to the change, and it is followed precisely*

That is, while the items on the left sound nice
in theory, we’re an enterprise company, and there’s
no way we’re letting go of the items on the right.

Cobbled together one Saturday morning before breakfast by Kerry Buckley (@kerryb), following an article by Ron Jeffries and this suggestion from Eastmad.

Made by Kerry Buckley, inspired by an article by Ron Jeffries from XP and the original agile manifesto



# Manifesto for Half-Arsed Agile Software Development

We have heard about new ways of developing software by paying consultants and reading Gartner reports. Through this we have been told to value:

**Individuals and interactions** over processes and tools
*and we have mandatory processes and tools to control how those individuals (we prefer the term 'resources') interact*

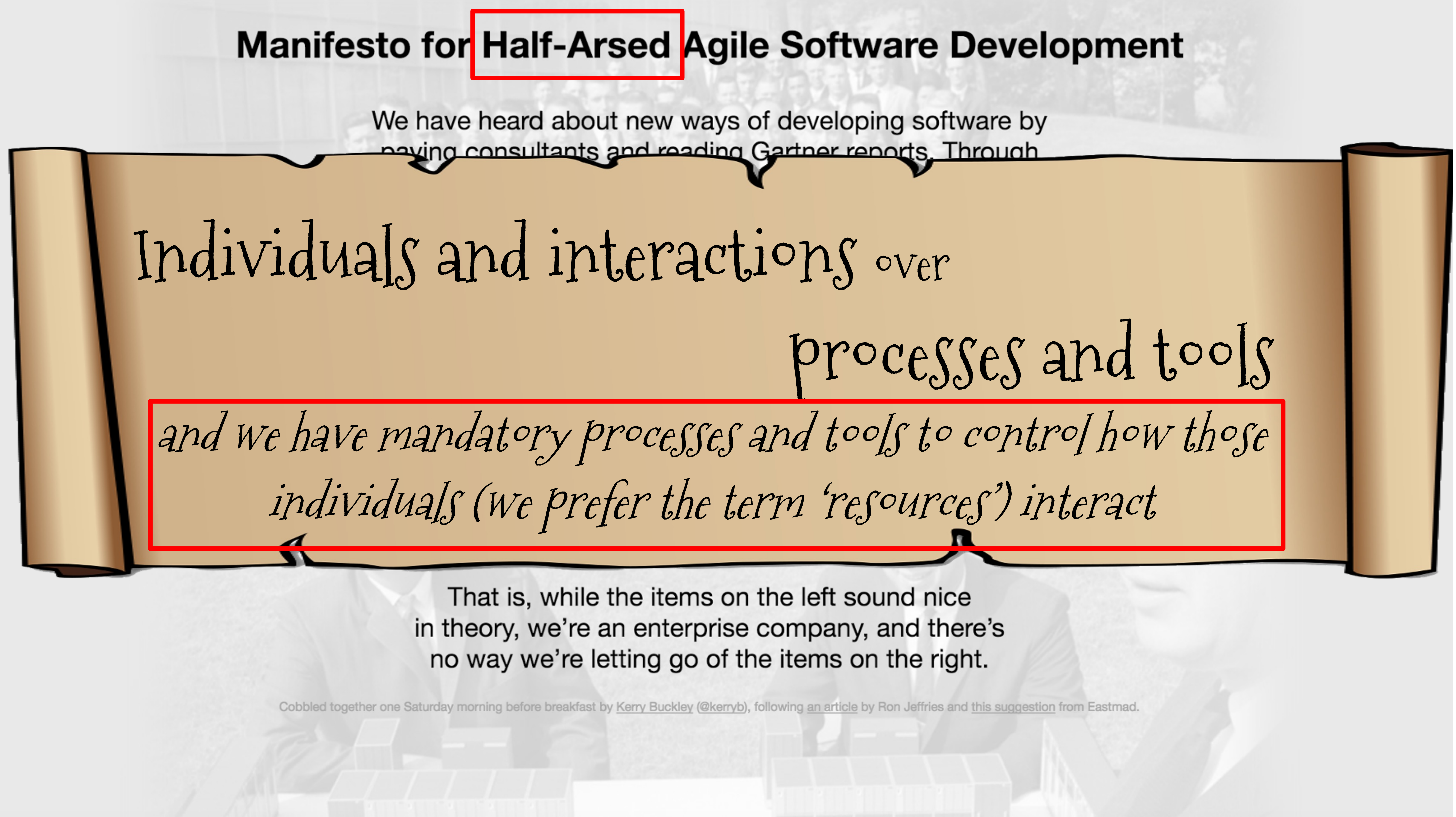
**Working software** over comprehensive documentation
*as long as that software is comprehensively documented*

**Customer collaboration** over contract negotiation
*within the boundaries of strict contracts, of course, and subject to rigorous change control*

**Responding to change** over following a plan
*provided a detailed plan is in place to respond to the change, and it is followed precisely*

That is, while the items on the left sound nice in theory, we're an enterprise company, and there's no way we're letting go of the items on the right.

Cobbled together one Saturday morning before breakfast by Kerry Buckley (@kerryb), following an article by Ron Jeffries and this suggestion from Eastmad.



# Manifesto for Half-Arsed Agile Software Development

We have heard about new ways of developing software by paying consultants and reading Gartner reports. Through

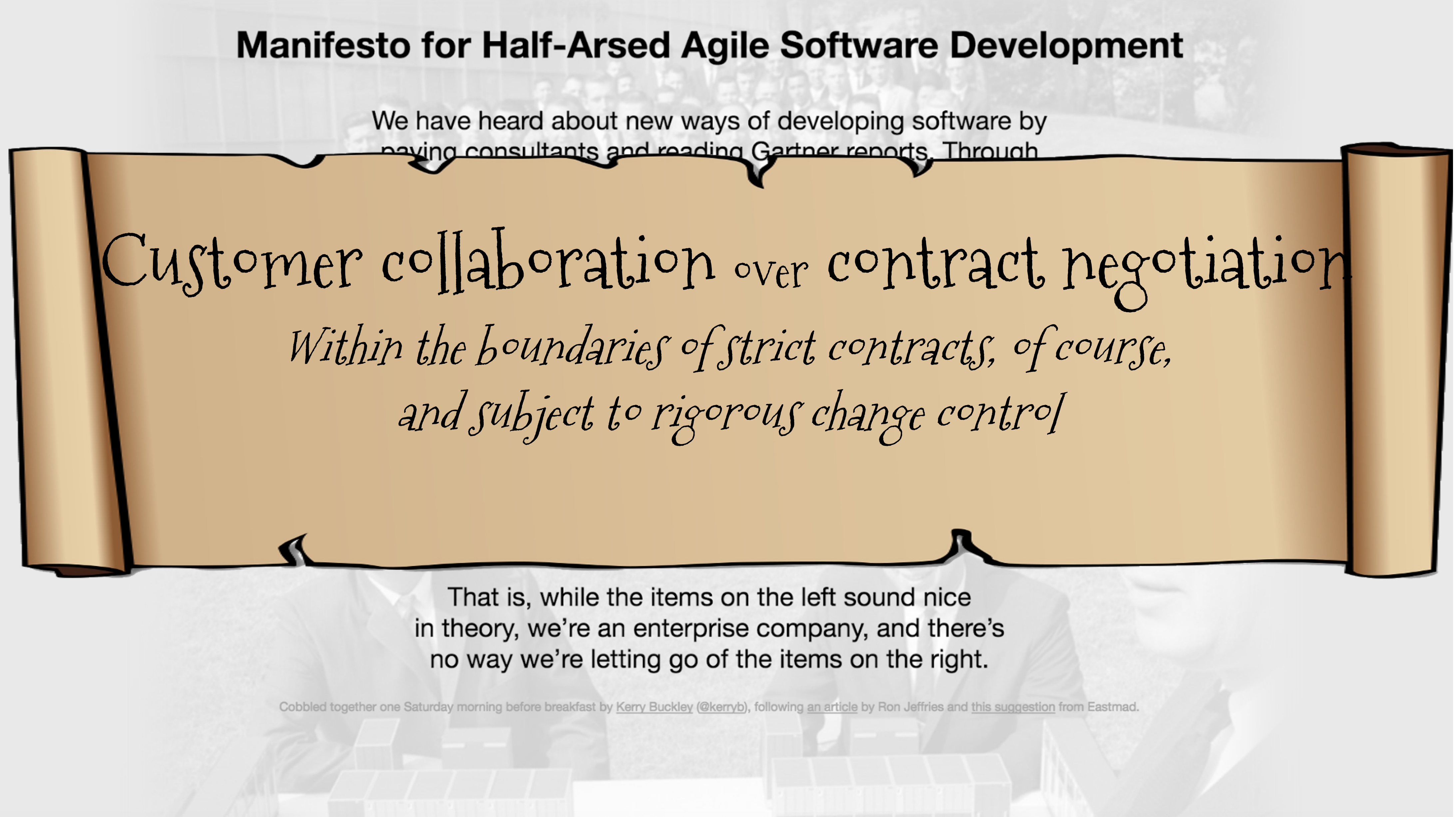
Individuals and interactions over

processes and tools

*and we have mandatory processes and tools to control how those individuals (we prefer the term 'resources') interact*

That is, while the items on the left sound nice
in theory, we're an enterprise company, and there's
no way we're letting go of the items on the right.

Cobbled together one Saturday morning before breakfast by Kerry Buckley (@kerryb), following an article by Ron Jeffries and this suggestion from Eastmad.



## Manifesto for Half-Arsed Agile Software Development

We have heard about new ways of developing software by
paying consultants and reading Gartner reports. Through

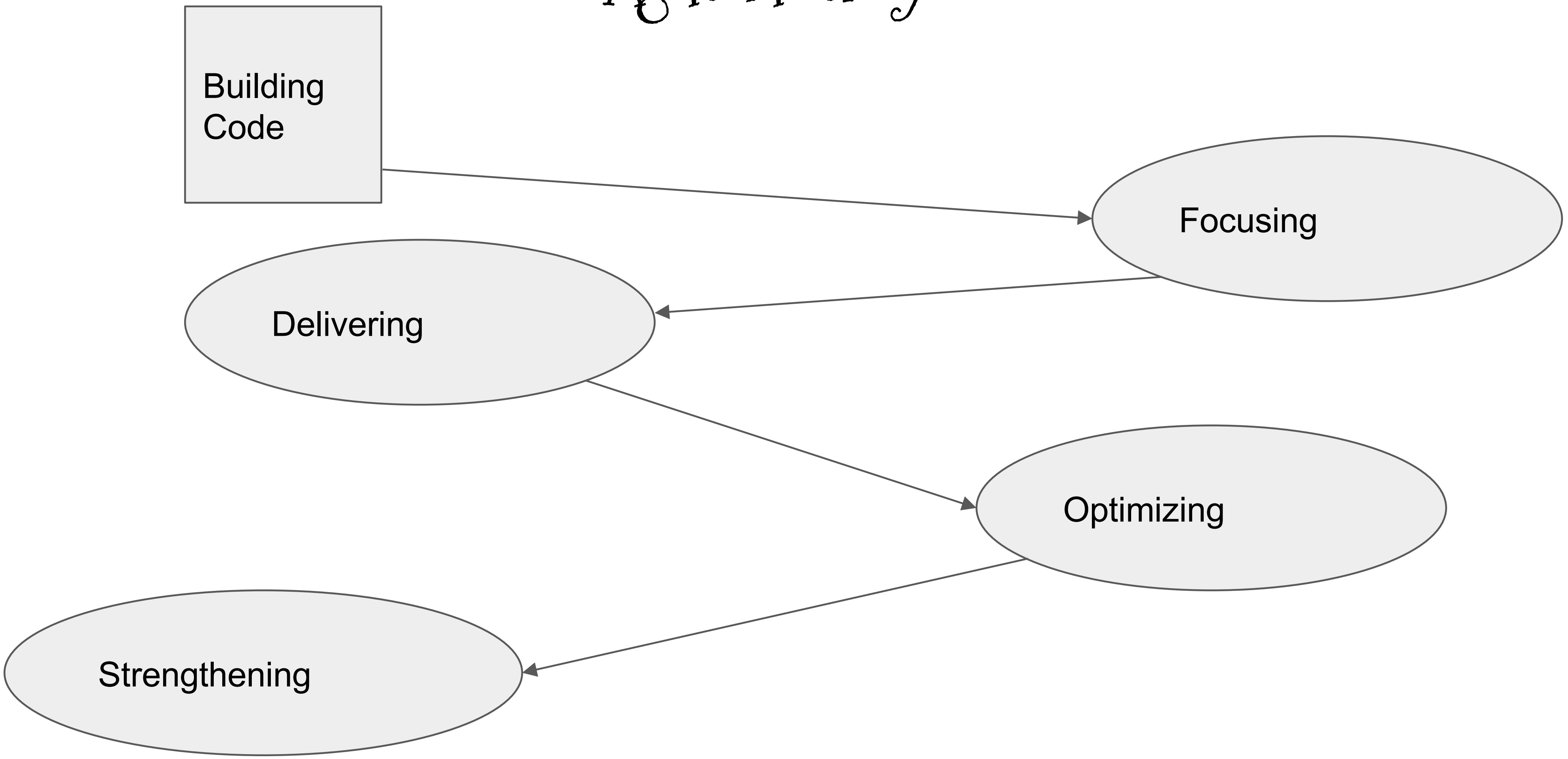
Customer collaboration over contract negotiation

*Within the boundaries of strict contracts, of course,*
*and subject to rigorous change control*

That is, while the items on the left sound nice
in theory, we're an enterprise company, and there's
no way we're letting go of the items on the right.

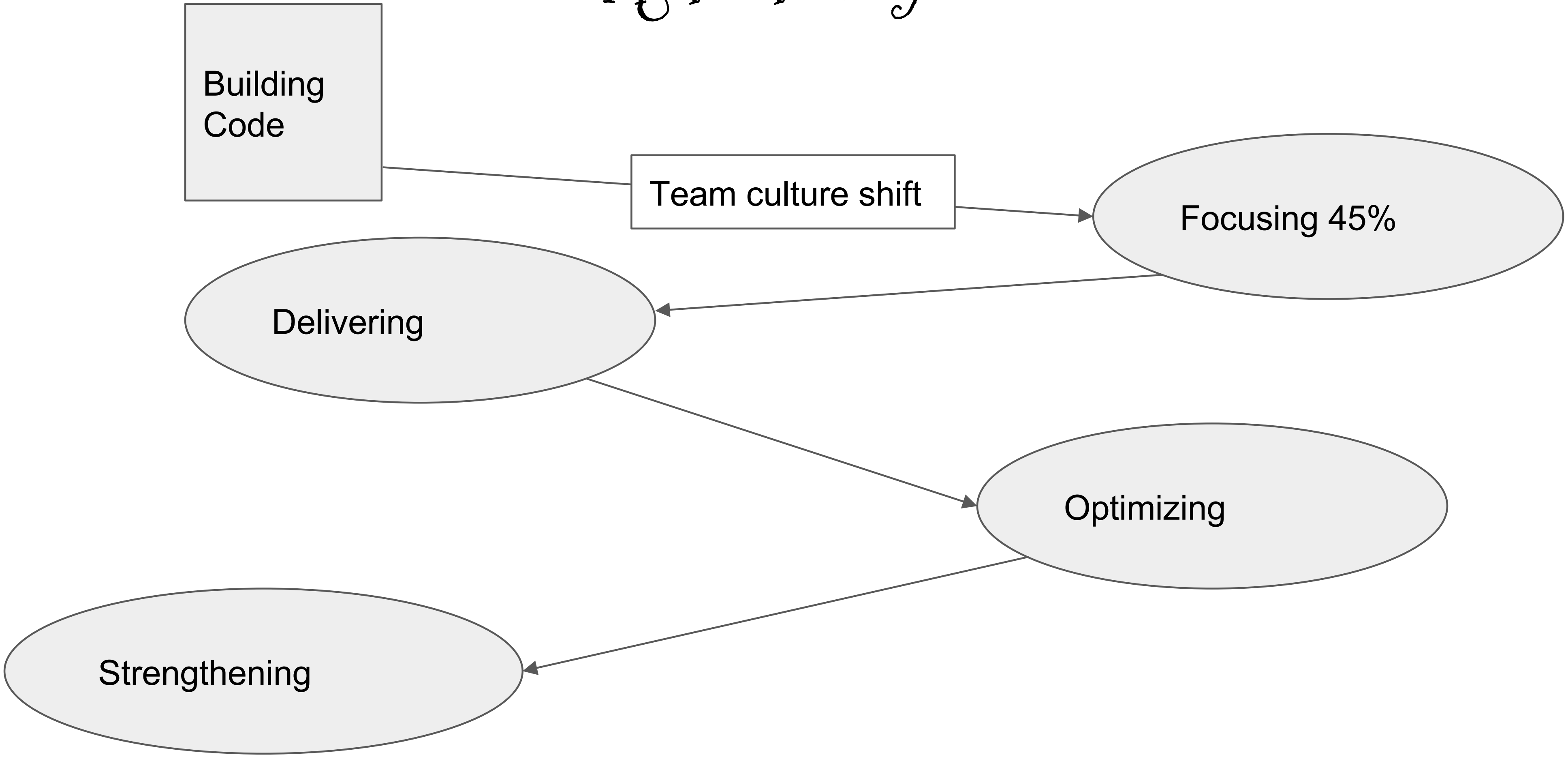
Cobbled together one Saturday morning before breakfast by Kerry Buckley (@kerryb), following an article by Ron Jeffries and this suggestion from Eastmad.

# Agile Fluency



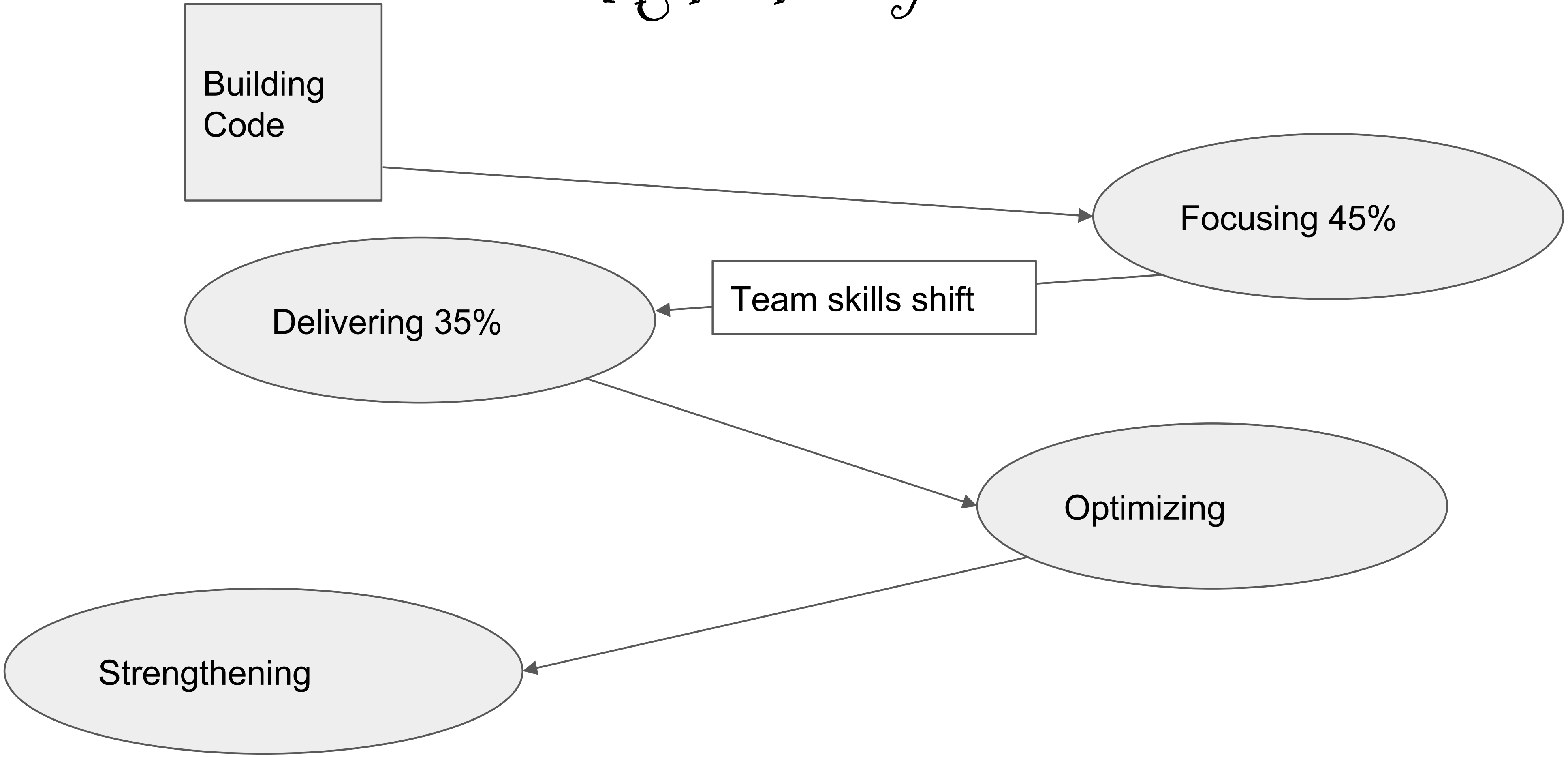
By Diana Larsen and James Shore

# Agile Fluency



By Diana Larsen and James Shore

# Agile Fluency



Building Code

Focusing 45%

Team skills shift

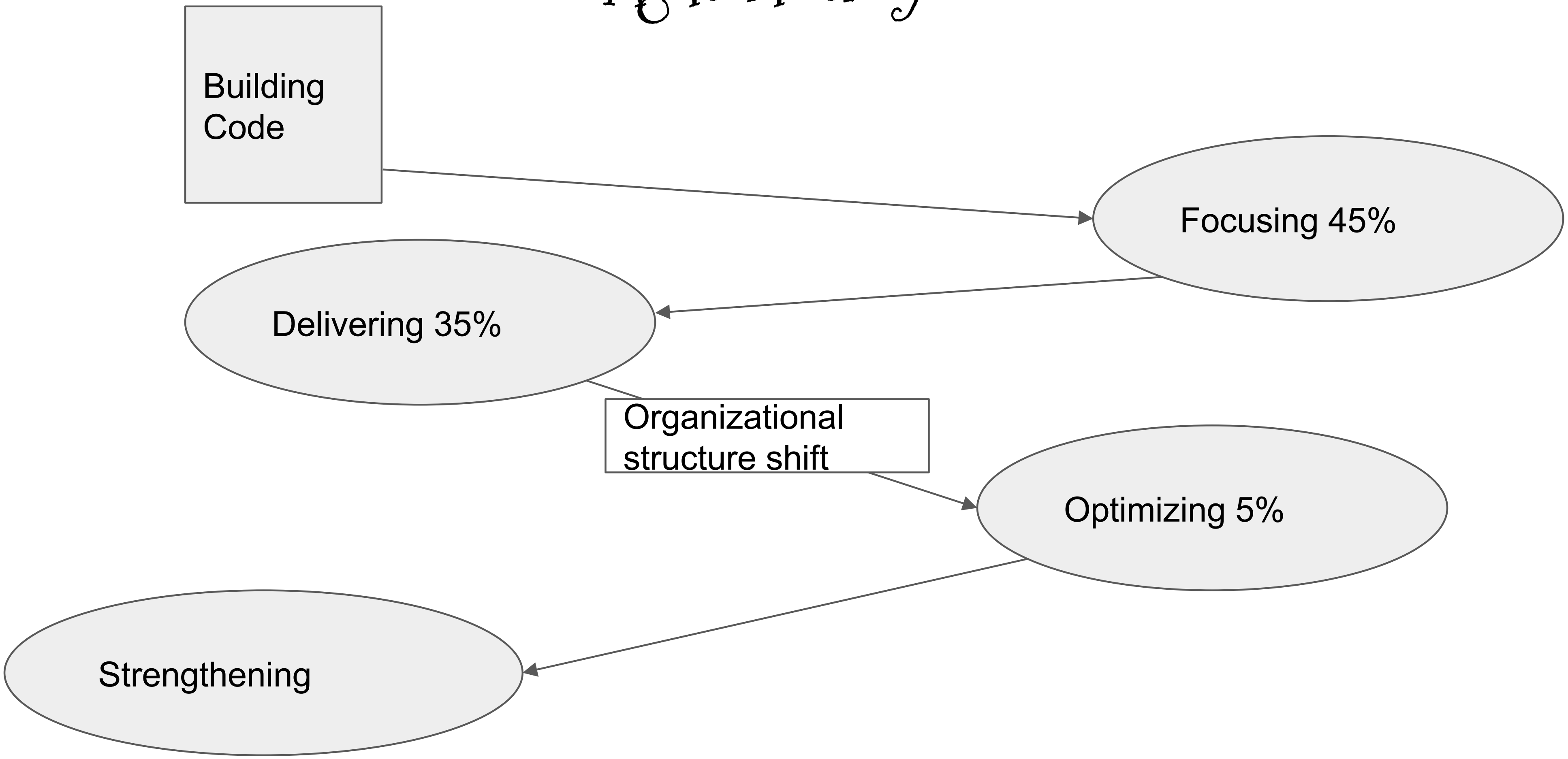
Delivering 35%

Optimizing

Strengthening

By Diana Larsen and James Shore

# Agile Fluency



Building Code

Focusing 45%

Delivering 35%

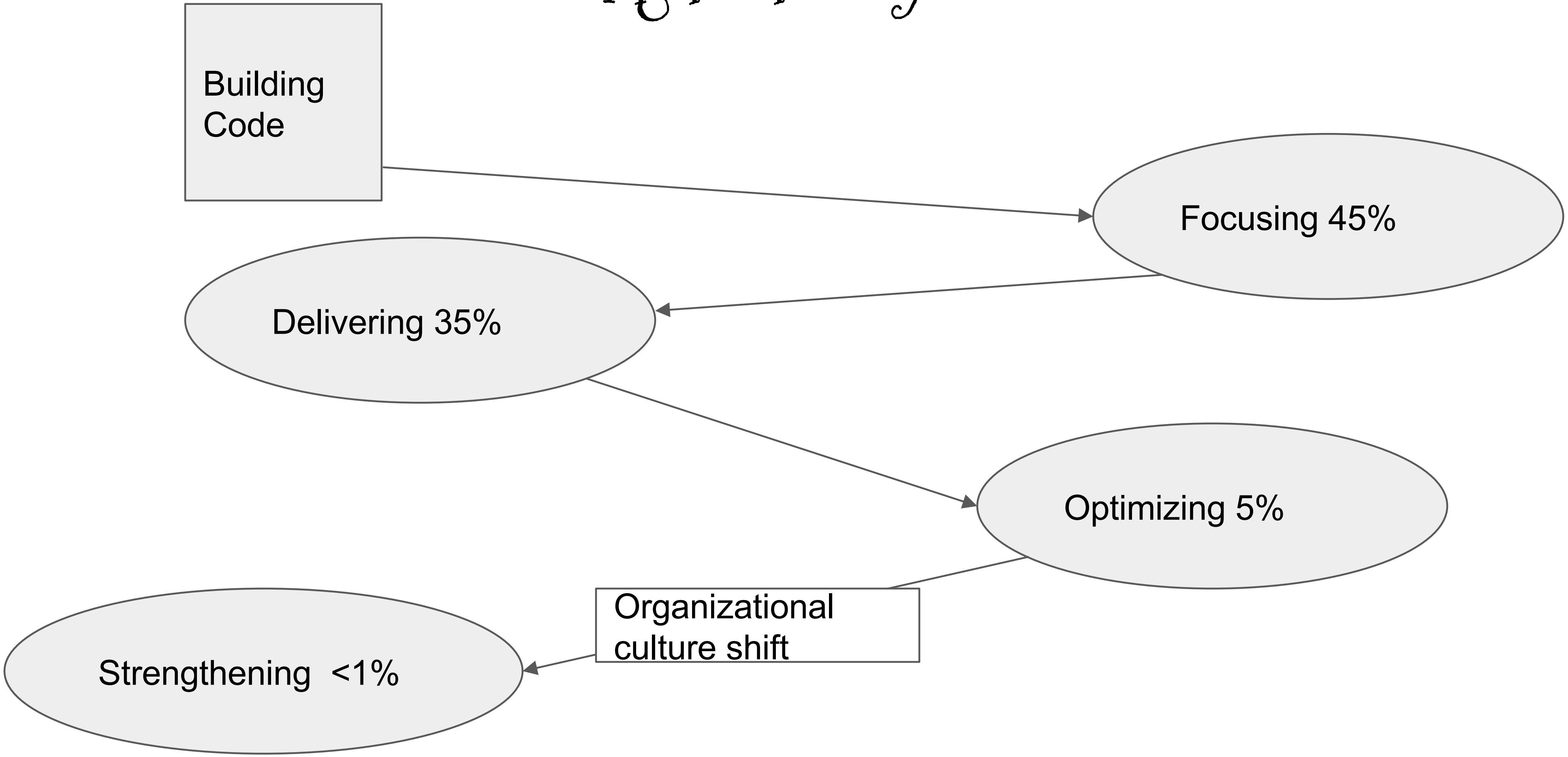
Organizational structure shift

Optimizing 5%

Strengthening

By Diana Larsen and James Shore

# Agile Fluency



Building Code

Focusing 45%

Delivering 35%

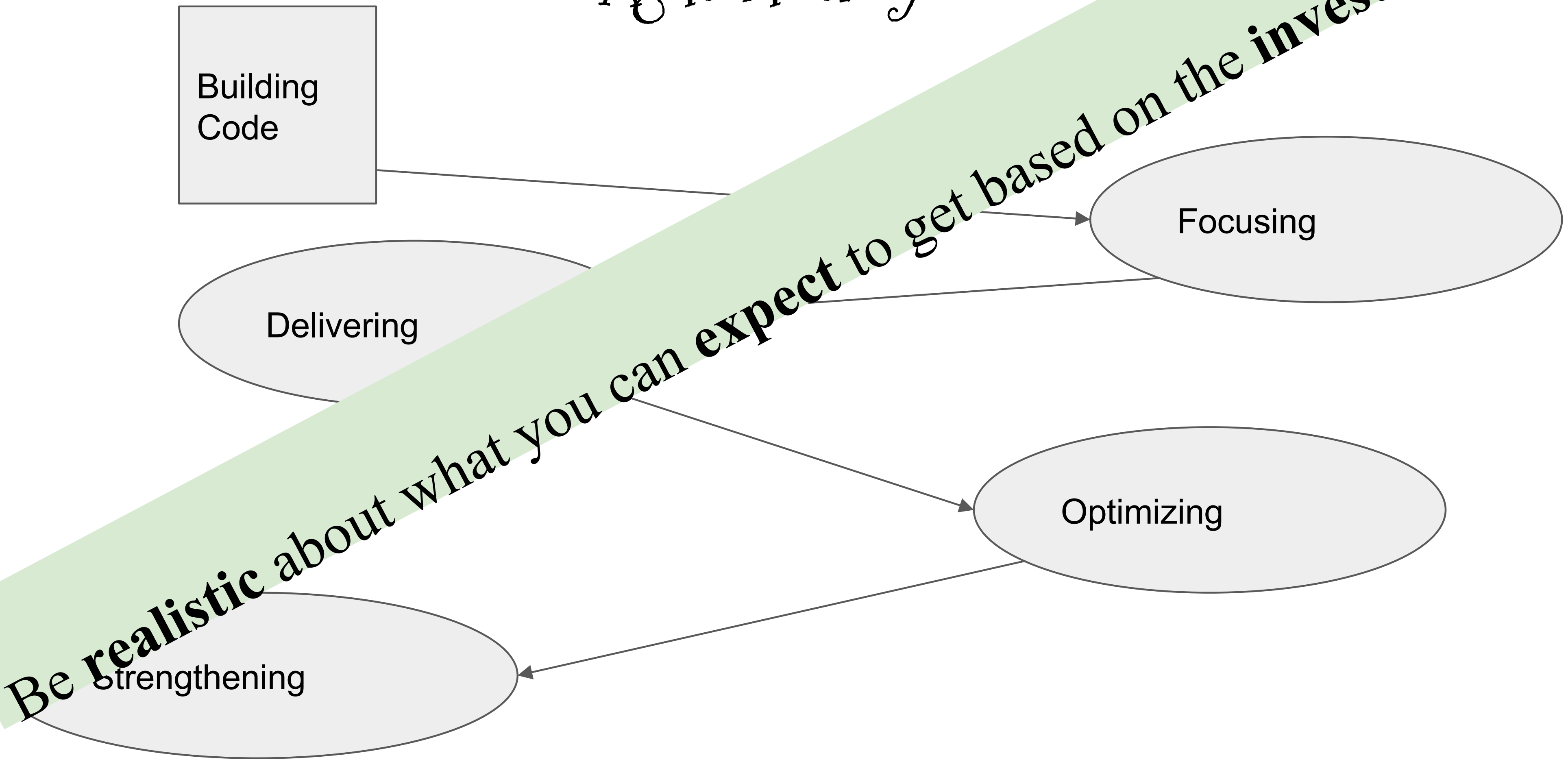
Optimizing 5%

Organizational culture shift

Strengthening <1%

By Diana Larsen and James Shore

# Agile Fluency

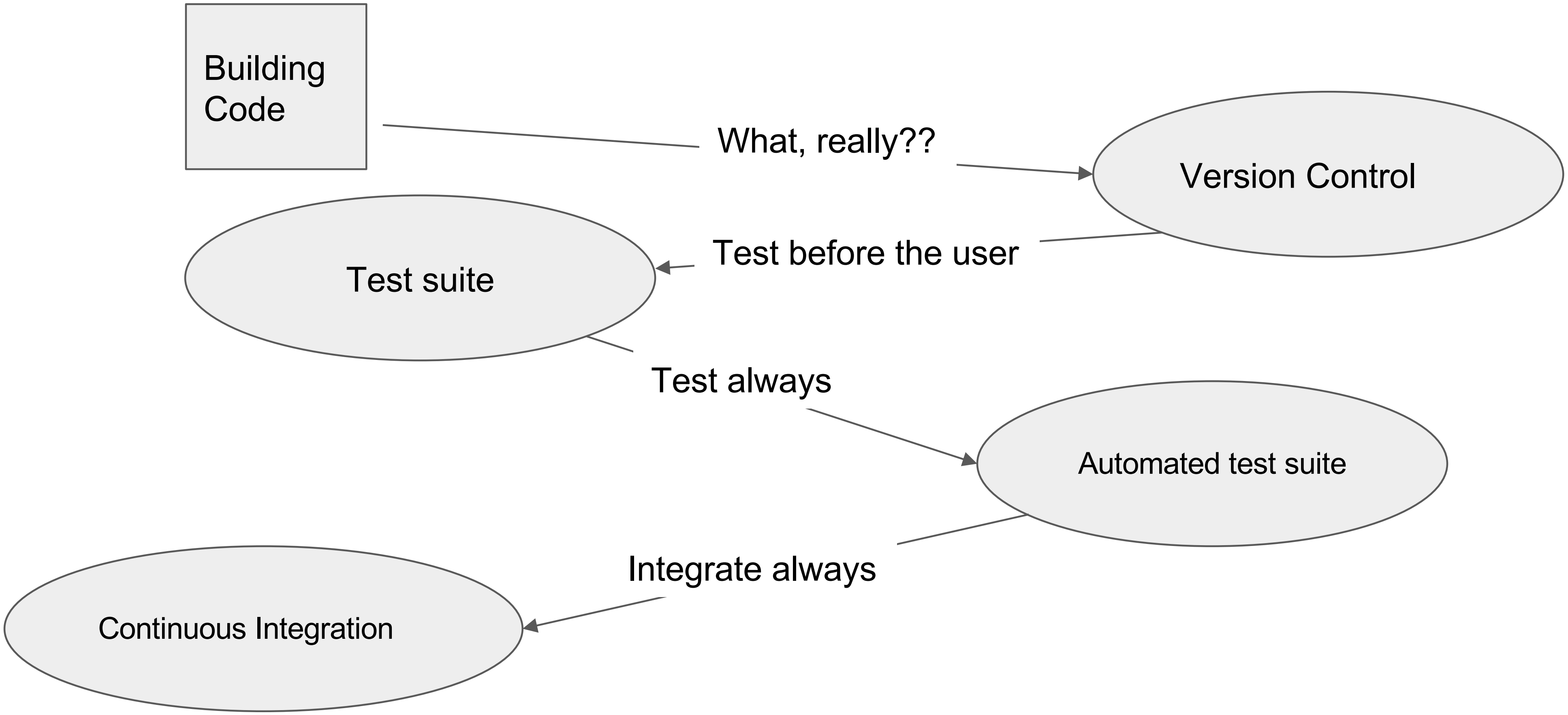


Be **realistic** about what you can **expect** to get based on the **investments**

By Diana Larsen and James Shore

But…we are doing everything right?….

# Technical Agile Fluency



Building Code

What, really??

Version Control

Test before the user

Test suite

Test always

Automated test suite

Integrate always

Continuous Integration

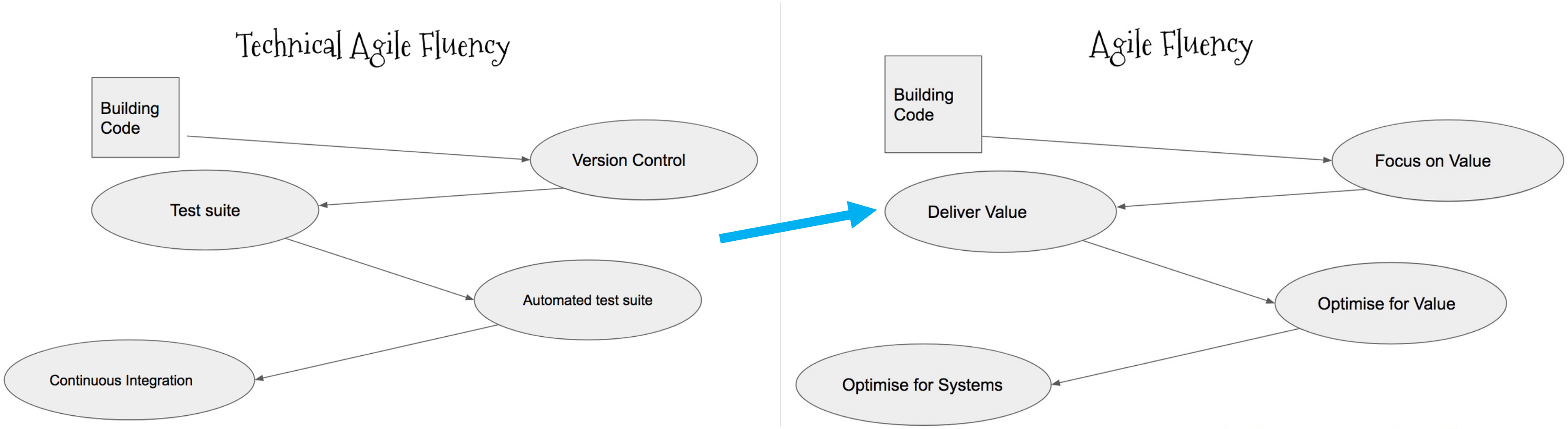
# And you?

Reflect on, whether agile is possible where you are?

## One minute challenge

Share with your neighbour

Listen to your neighbour



Technical Agile Fluency

Building Code

Version Control

Test suite

Automated test suite

Continuous Integration

Agile Fluency

Building Code

Focus on Value

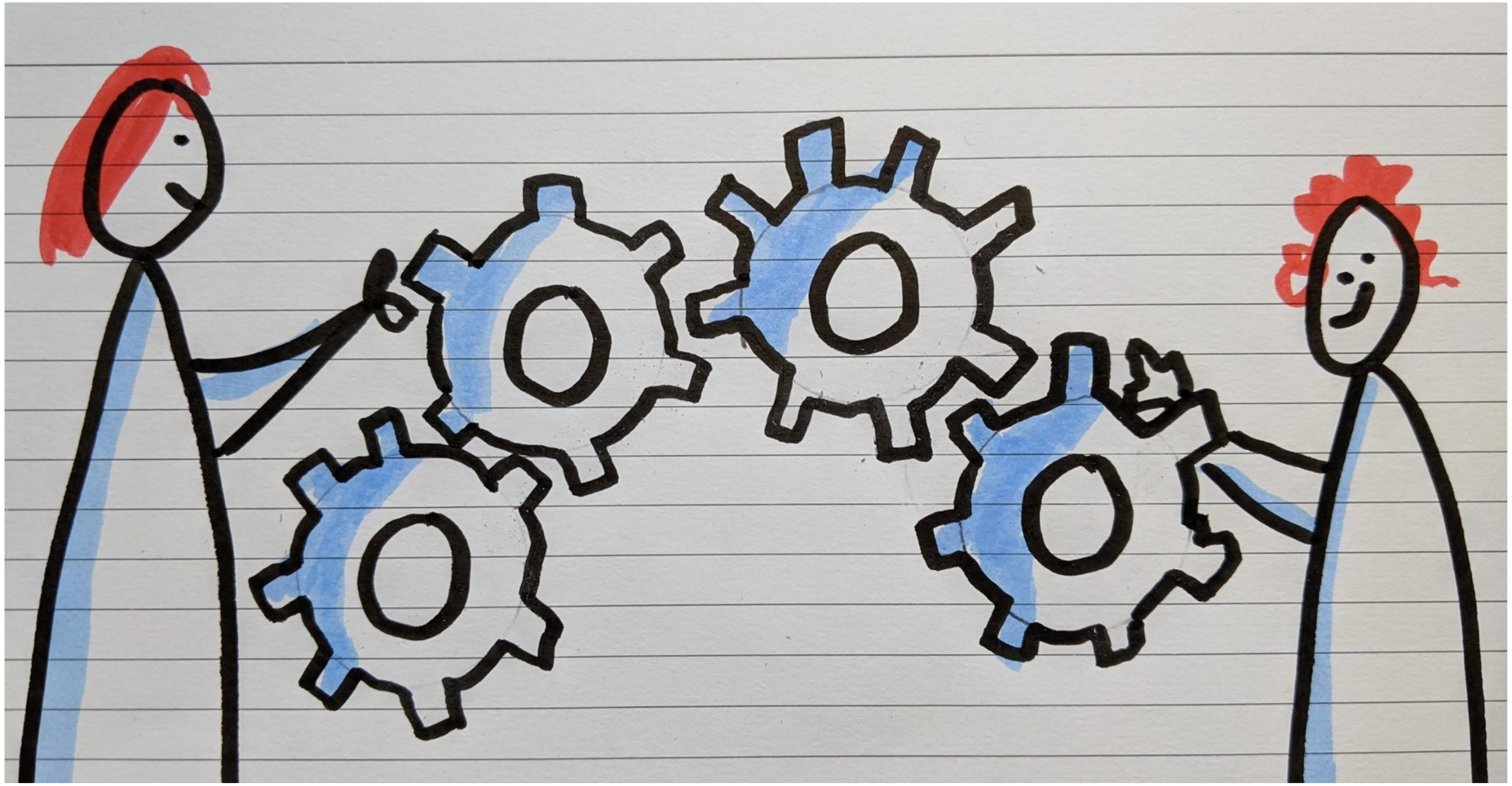
Deliver Value

Optimise for Value

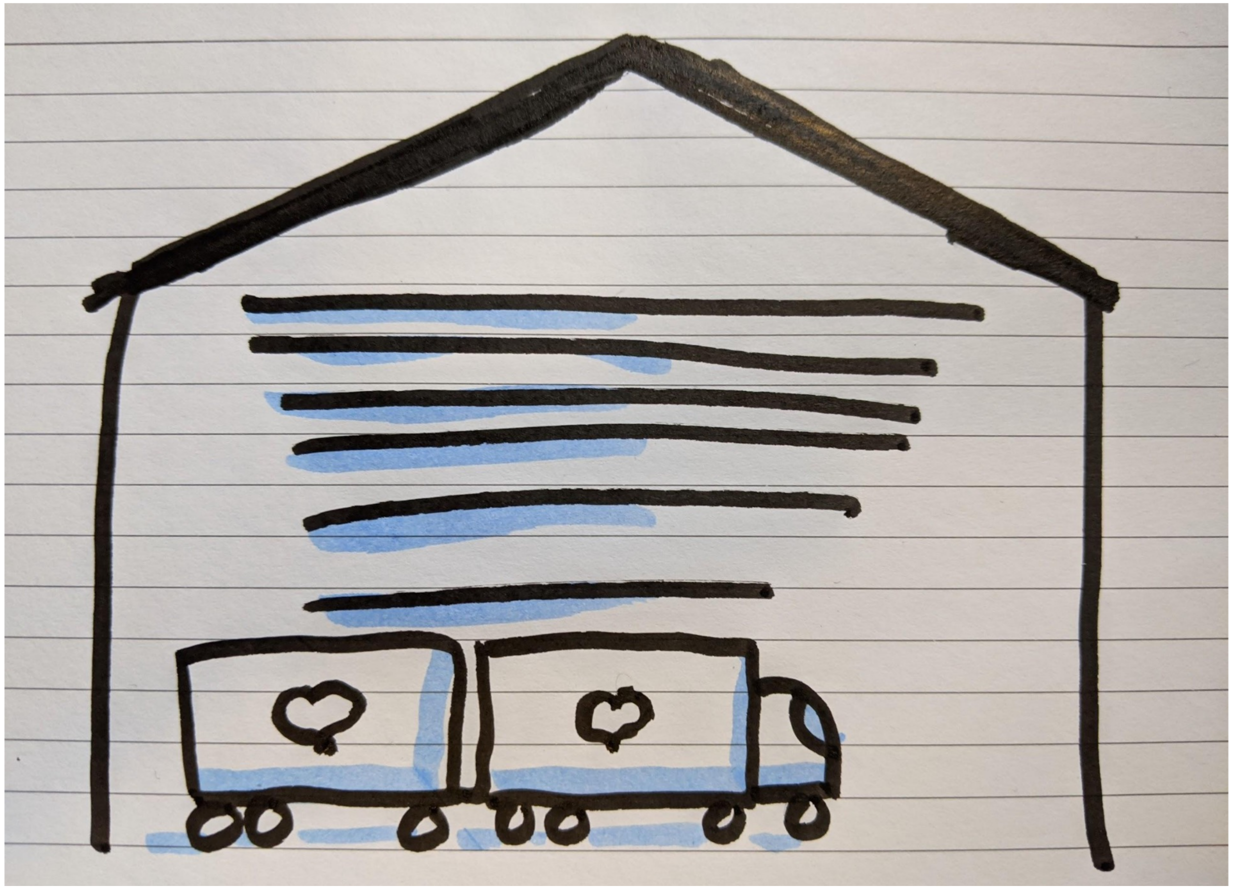
Optimise for Systems

By Diana Larsen and James Shore

# Continuous integration
# - for everyone

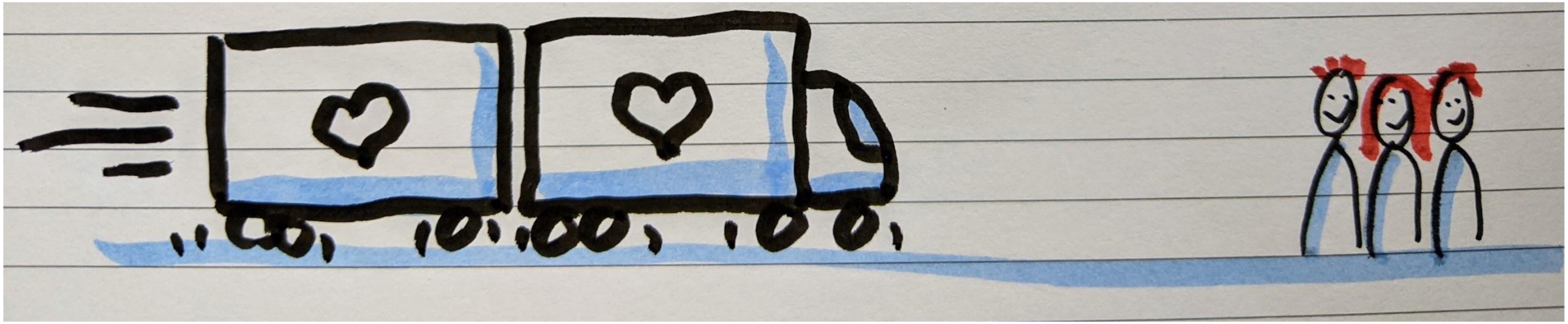


# Continuous delivery - for most

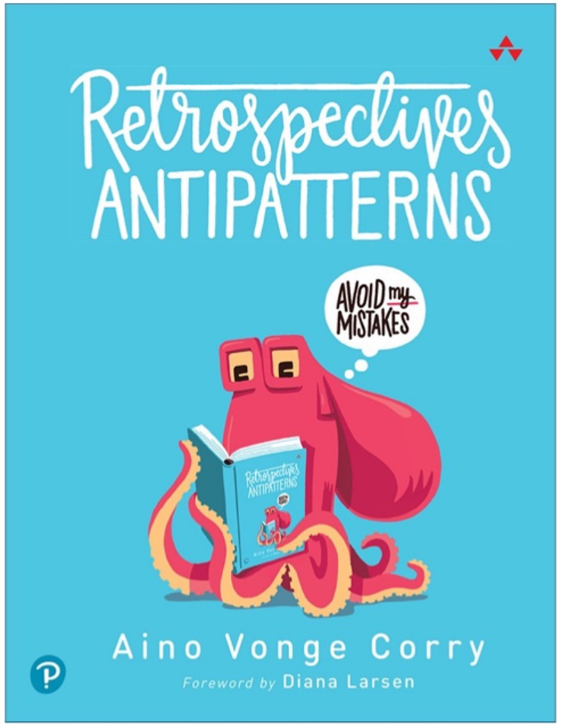


Continuous deployment

- impossible for some, and inappropriate for others



# Happy Hobbit



## Order & Save 35%*

## informit.com/corry

- Use code **CORRY**
- Available as Book and/or eBook

*(eBook files include PDF, EPUB, and MOBI)*

*Book is also available via booksellers including Amazon and Barnes & Noble and in O'Reilly's Online Learning Service (Safari).*

**Discount code CORRY is only good at informit.com and cannot be used on the already discounted book + eBook bundle or combined with any other offer.*

Pearson

Addison Wesley

informIT
the trusted technology learning source

# Takeaway message

My story of Agile Development

The promise of Agile

The misconceptions of Agile

What now?