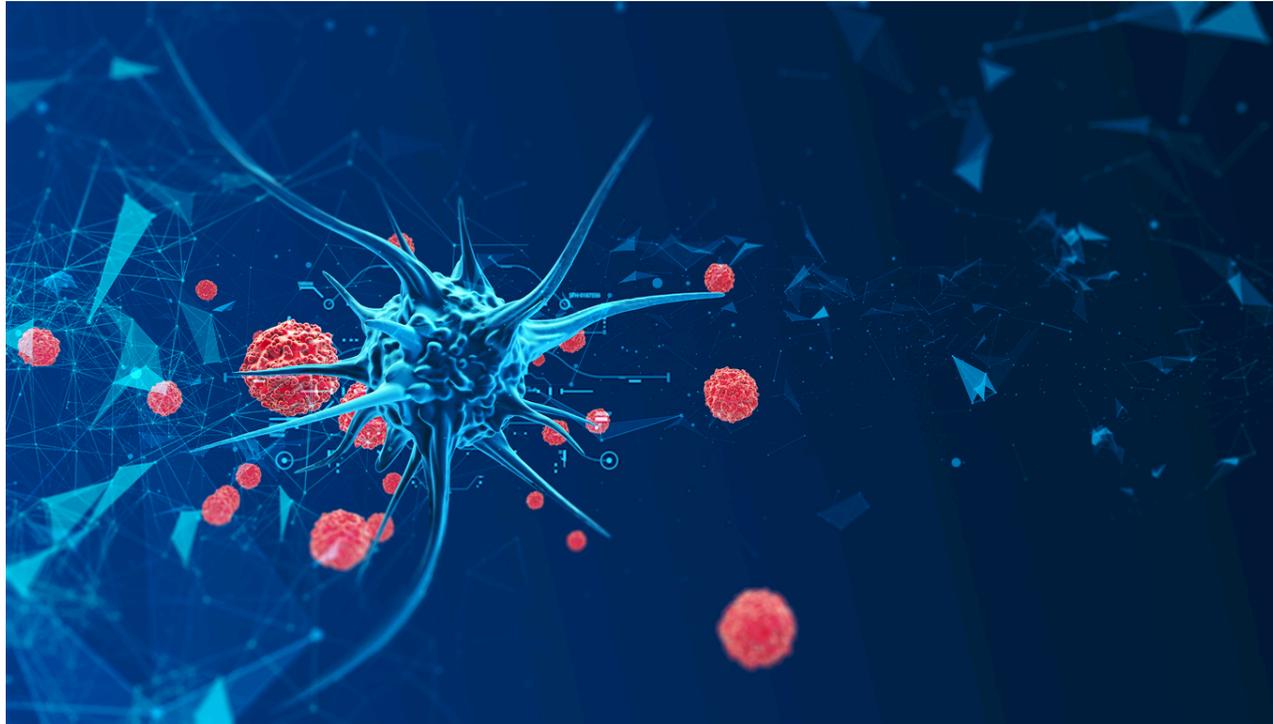


Radical Agile, Mild Agile and the Future

Allan Kelly - @allankellynet
allan@allankelly.net

Frug'Agile,
February 2021

First, the Virus



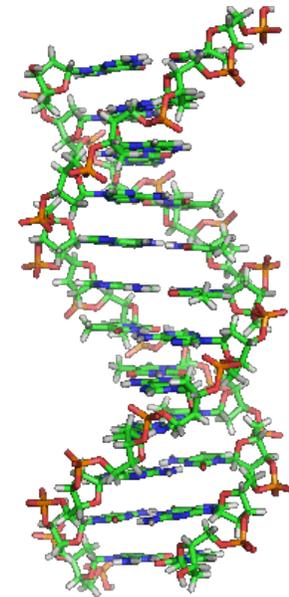
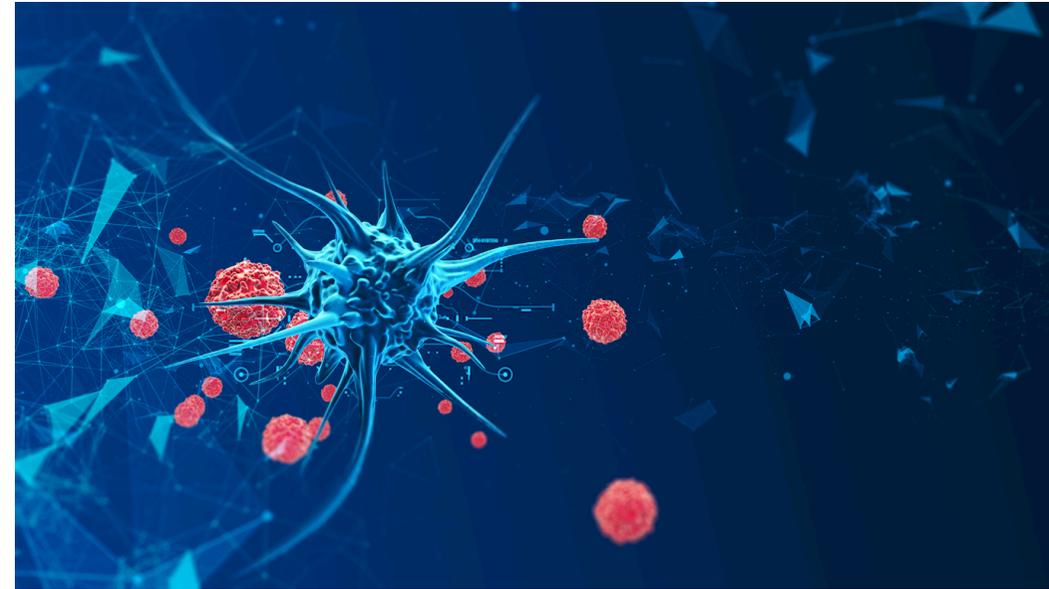
The Agile virus

First identified 2001

- Circulating earlier
- Spread aided by growth of digital technology

Mutates

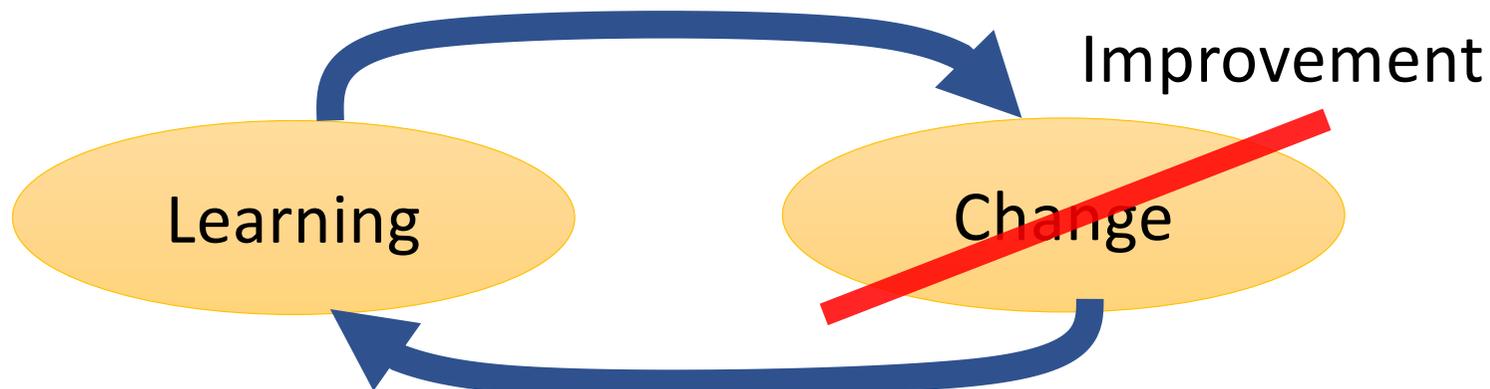
- Agile has self-modifying DNA
- For better
- For worse



How do you test for
the virus?

The only thing you can do wrong in agile is to do things the same as you did 3 months ago

Allan Kelly

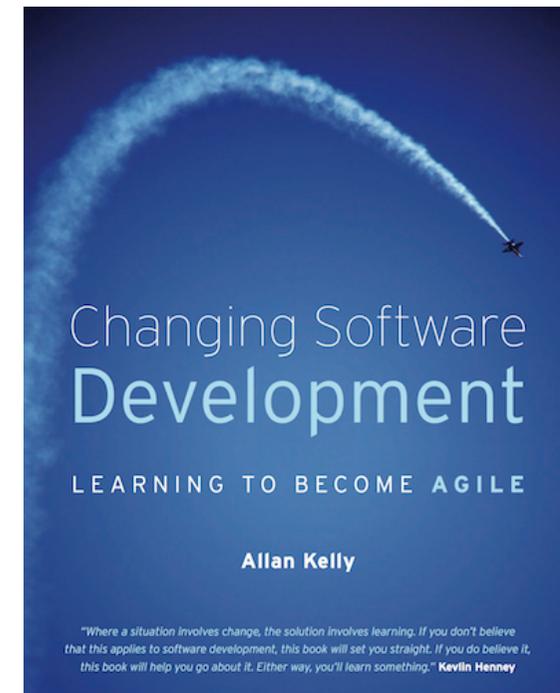


Agile is learning

Agile is *organizational learning*

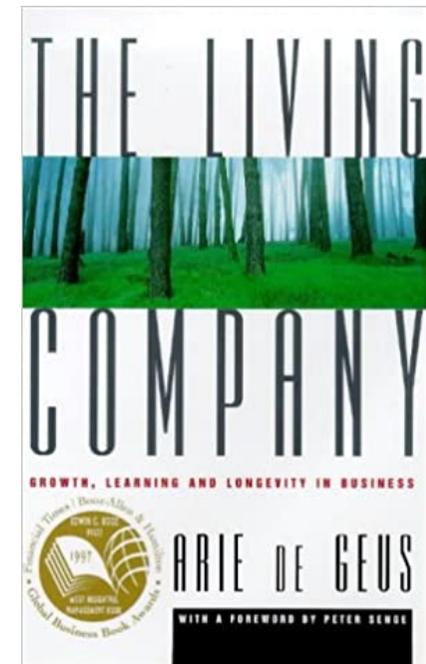
Agile is your change method

Alan Kelly
Changing Software Development
2008



“We understand that the only competitive advantage the company of the future will have is its managers’ ability to learn faster than then their competitors.”

Arie de Geus, *The Living Company*, 1997



“How many companies are doing agile?”

Agile-R?

Introducing E

Effectiveness

Best are getting better

High E

Effectiveness

Elite agile

Best teams
seem Radical

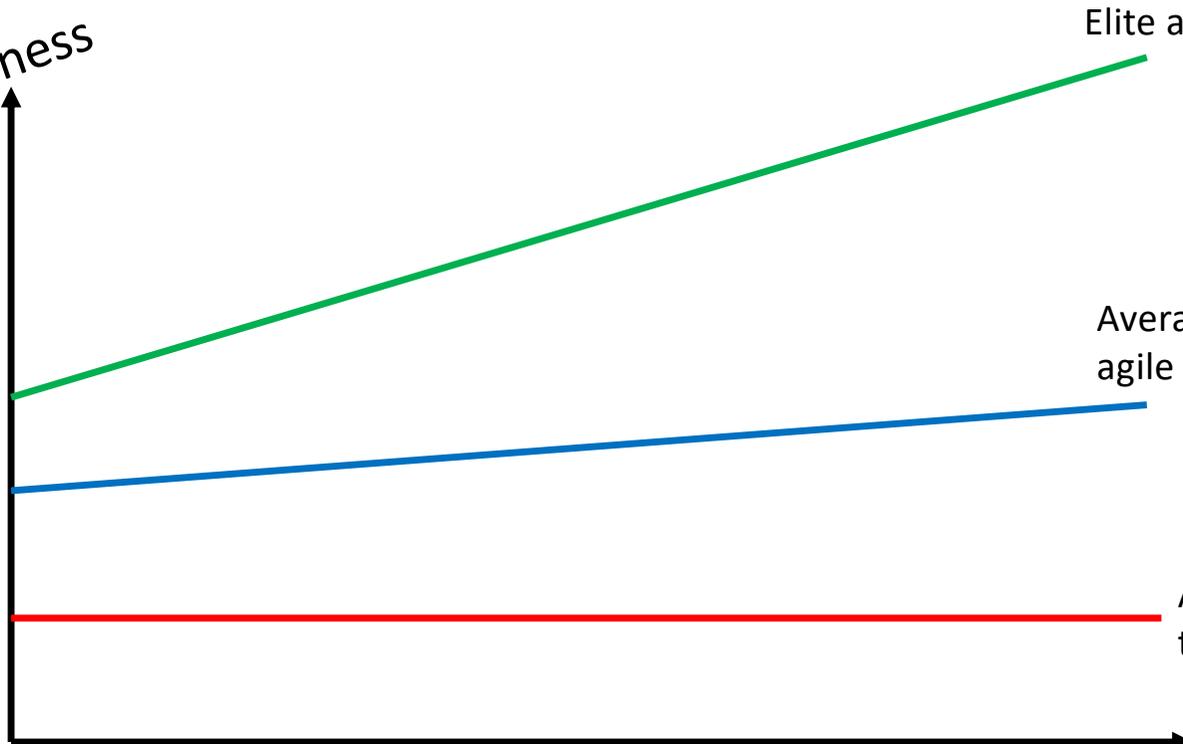
Average
agile

Average teams stay
safe – Mild Agile

Average
traditional

Low E

Time



2 strains of Agile virus



Mild Agile

- Standardization & processes
- Estimation hell
- Fear of coding
- Premature optimizations
- Dependency management
- Fear of failure
- Many hungry mouths (removed from code)

Low E?

Radical Agile

- Continuous Delivery
- #NoEstimates
- #NoProjects
- Mob Programming (& pairing)
- Beyond Budgets
- Purpose over backlogs
- Technical excellence
- More in close to code

High E?

Confirmation bias?

Grass is always greener on the other side

Do we see better because we want to see better?



Examples get magnified

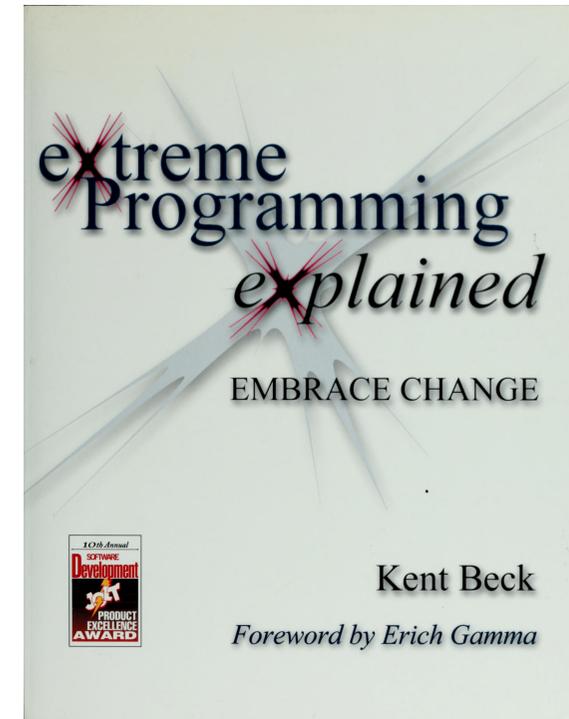


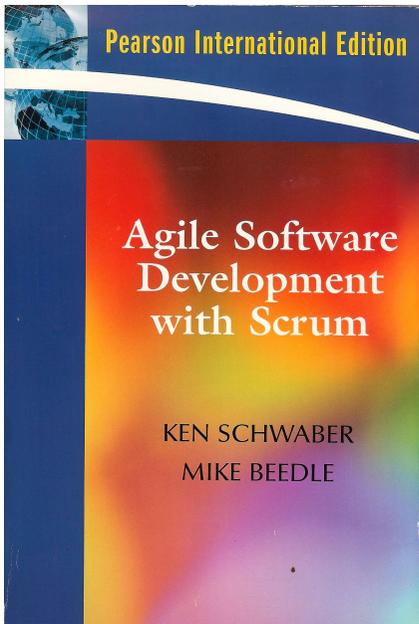
- 1 example does not equal success
- 1 in 1,000,000 teams tries mobbing and has success
- Team writes and present about it
- Everyone hears of mobbing success!
- 999,999 don't – nobody writes about success doing it the usual way

It was always so: XP, 2000

- Regular planning & small releases
- Simple design
- Automated Testing (TDD, ATDD)
- Refactoring
- Pair programming
- Continuous integration
- 40-hour work week
- On-site customer
- Face-to-face communication

XP is
still
Radical in 2021

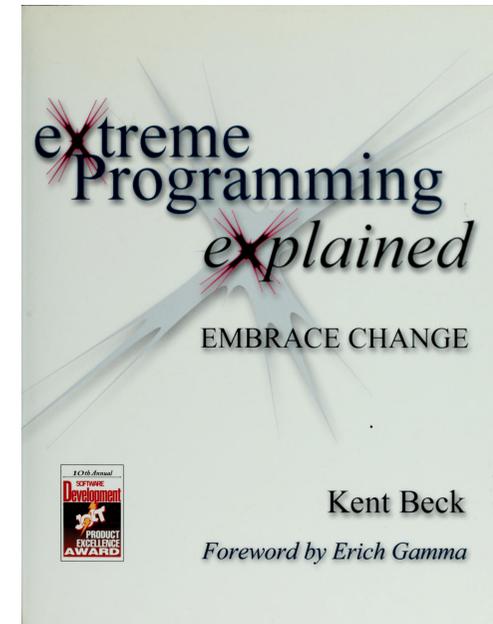




3 reasons why Scrum is better than XP

1. Scrum traces origins to *Harvard Business Review**
2. You can get certified in Scrum
3. Scrum does not contain the words “extreme” or “programming” in the title (2 words guaranteed to scare managers)

**New new product development game*,
HBR Jan 1986



Spring Surprise Scrum tension



Monty Python on YouTube <https://youtu.be/mK3B0gjEjTs>

Is radical better?

Why is radical better?

And what is “better” ?

Is better just “better than what went
before”?

What is wrong with mild agile?

But then, what is wrong with radical ideas?

It takes time

Its right to be cautious
Right to be skeptical
Right to think



Agile



Mild Agile

- Estimation hell
- Fear of coding
- Premature optimization
- Dependency management
- Fear of failure
- Hungry mouths (removed from code)

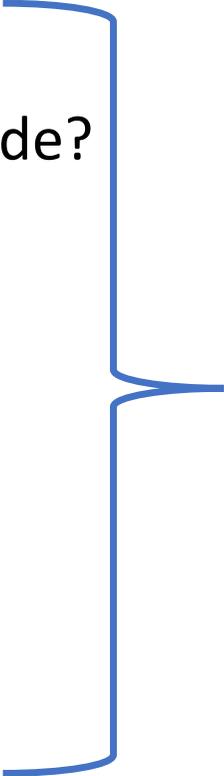
Radical Agile

- Continuous Delivery
- Beyond budgets
- Purpose over backlogs
- Technical excellence

Ideas need time to migrate from radical to safe to mainstream

Fear of techniques

- Do you really want someone watching you code?
- Tests might prove you code doesn't work
- If coders write tests will they be paid less?
- All that time spent in planning meetings
 - All the time in retrospectives, stand ups
- People need to be told what to do!



Just plain scary

Reluctance is
natural

Fear of failure

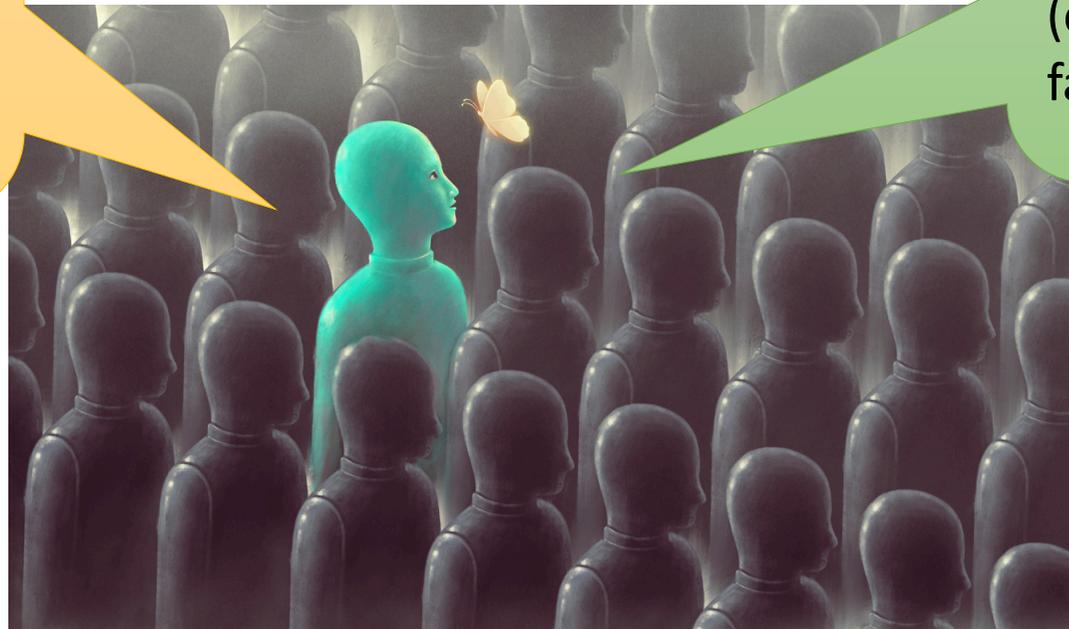


Past success

Incentive to conform

If my team works the same way as other teams and we all fail then I did nothing wrong

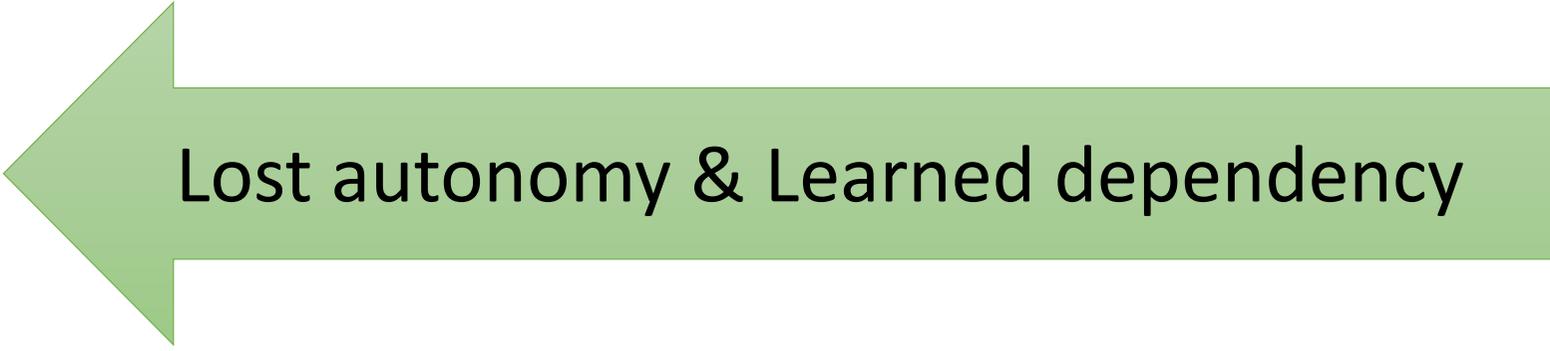
If my team does things differently and fails, then I fail (even if other teams fail)



Prisoner's dilemma



Just tell me what to do



Lost autonomy & Learned dependency

Blocks to learning & improvement

Structural inhibitors

- Define processes & corporate standards
 - Agile or SDLC
- Governance model & reporting standards
- Learning resource, time, space
- Past selection criteria
 - Explicit & implicit

Mental inhibitors

- Conformity
- Seniority & Hierarchy
- Psychological safety (lack off)

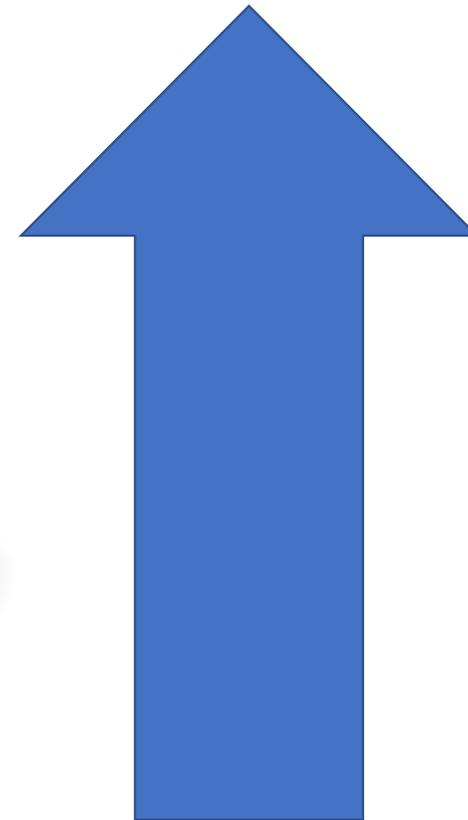


To increase E accelerate learning

- Remove blocks to learning
- Try new (radical) ideas
- Reflect
- Learn
- Repeat



E



The problem with scaling...

Usually

- Increasing R
 - Scaling out
- Standardization
 - Reduces variation
 - Inhibits learning
 - Disempowers



Should

- Focus on increasing E
- Remove blocks to learning
- Increase experiments in teams
- Increase learning across teams
 - Learn & steal from others

Learning is not enough

Unlearning

Out with the old

Make space for the new

Increase the speed of unlearning

$E = (\text{learning} + \text{unlearning}) \times \text{action}$

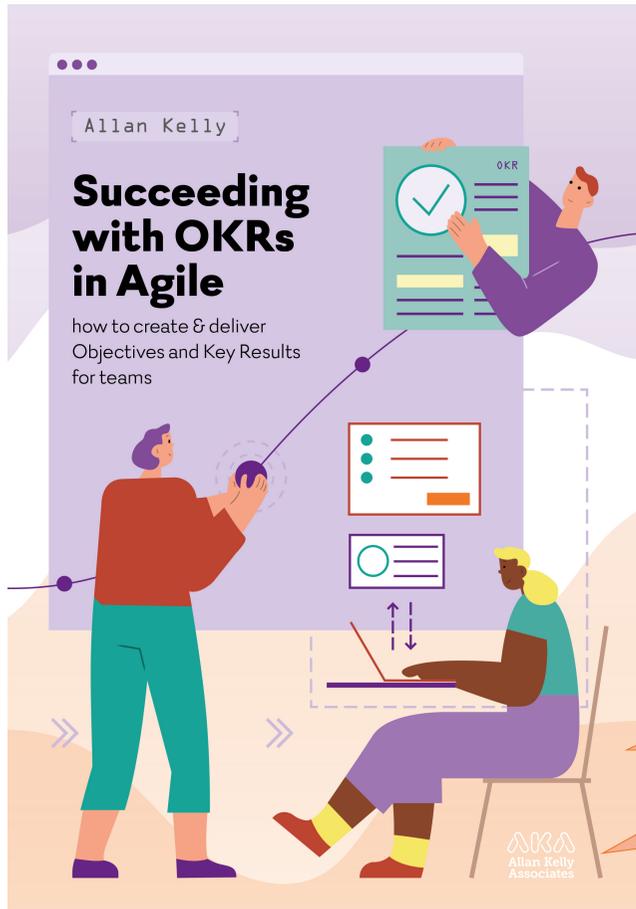
"The future is already here — it's just not very evenly distributed."

William Gibson,
novelist

Changes we need to see

- Reform Governance
- Emphasis Value Delivered
- Broad Value definition: social value
- Purpose over backlogs
- Swap BAs and Testers
- Enhance understanding & use of statistics

Excited by OKRs



Radical potential?

**Published today
on Amazon**

E over R – Radical Learning

- Its all about learning
 - And unlearning
- Try something radical today

Allan Kelly @allankellynet

allan@allankelly.net

<https://www.allankelly.net>

<https://www.linkedin.com/in/allan-kelly-19b16/>



The End

Failure to learn

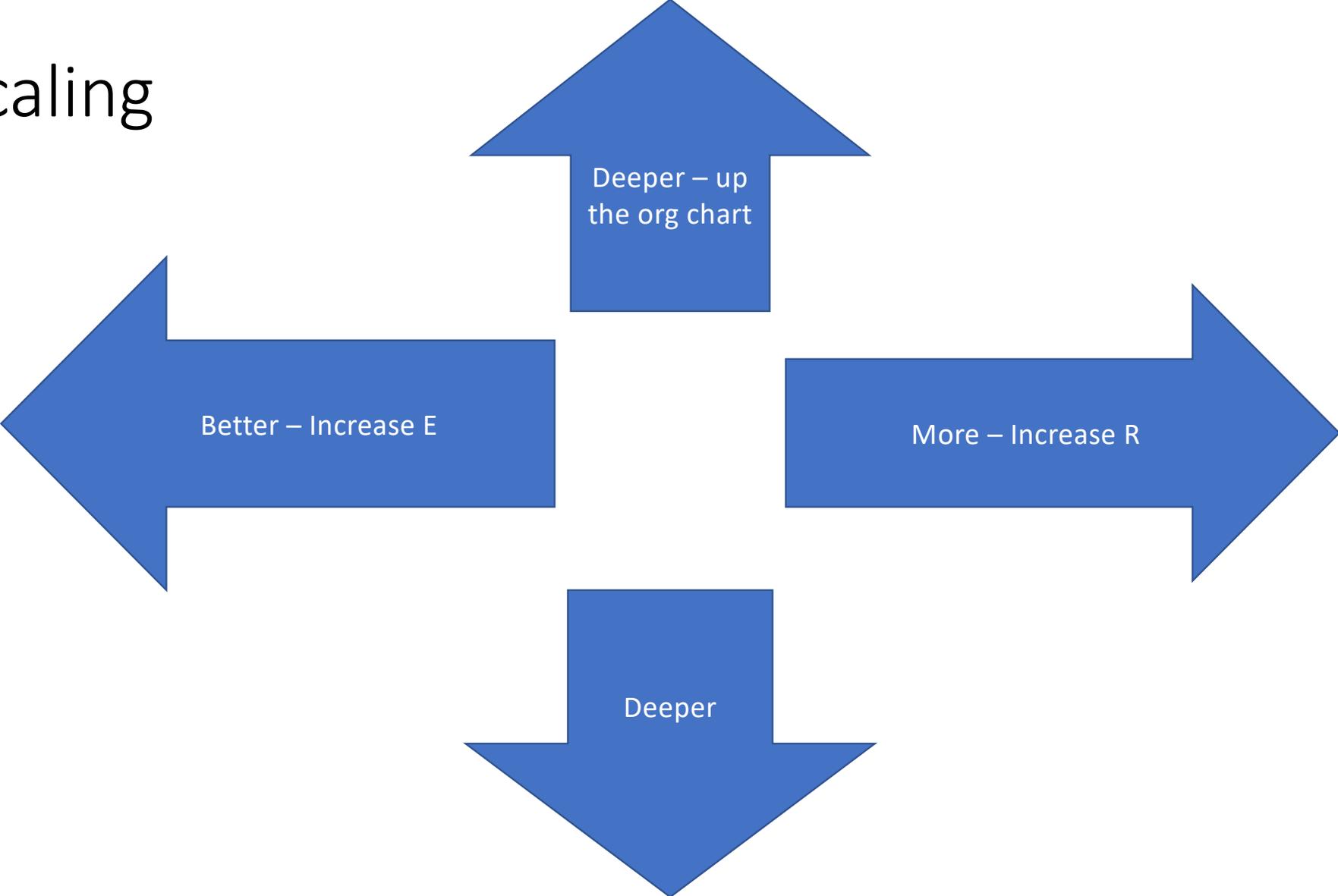
Mode 1 failure

- don't read the book
- don't go on the course
- don't get help (experience coach/guide)
- don't agree as team

Mode 2 failure

- don't have the experience at all
- don't learn from own experiences
- don't reflect on what happened

Scaling



Galbraith

Risk fear

- Can this work? – request for proof
- Fear of failure
- Fear that it is not the best
- And the risk of not doing?
- Profit is the return for risk: take chances