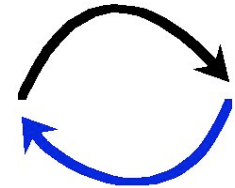


BCS Bristol Spring School



The Future of Agile

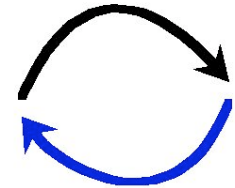
Allan Kelly

Software Strategy

<http://www.softwarestrategy.co.uk>

(c) Allan Kelly, 2009

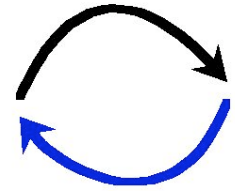
Allan Kelly, BSc, MBA



- Help companies achieve *Operational Excellence in Software Development with Agile* methods
 - Consulting
 - Training: Agile, Lean, Scrum, etc.
 - Agile Coaching
- Over 10 years as developer
- Author:
 - *Changing Software Development: Learning to be Agile*, Wiley 2008.



What is Agile?

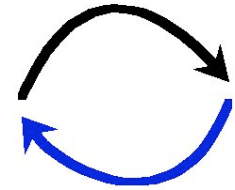


Three tests of Agile

1. Is the team delivering business value to the customer (market) ?
2. Is the team learning, changing and improving?
3. When the consultants, trainers and coaches who introduced Agile leave, are you still Agile?



Nokia Agile test



1. Are you doing Iterative Development?

- Iterations timeboxed to less than 4 weeks
- Features tested and working at the end of each iteration
- Iteration starts before specification is complete

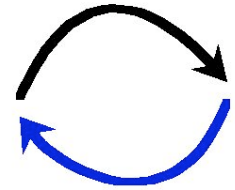
2. Are you are doing Scrum?

- You know who the product owner is
- There is a product backlog
 - Prioritized by business value
 - With estimates created by the team
- Team generates burndown charts and knows velocity
- No project managers (or anyone else) disrupting the work of the team

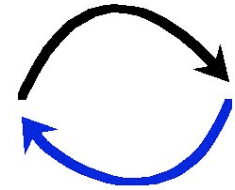


Operational Excellence

- *Agile* and *Lean* are a means to achieving Operation Excellence
- Agile is a version of Lean
- Lean is a journey



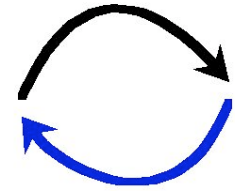
The Future, A summary



- The Future is Lean
 - Kanban: the New Kid on the Block
- Return of the Product Owner
 - Product Managers
 - Business Analysts
- The Change question
 - How do you get Agile?
 - How do you stay Agile?
 - What next?
- People - the meta-issue



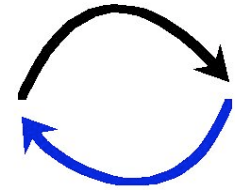
The Future



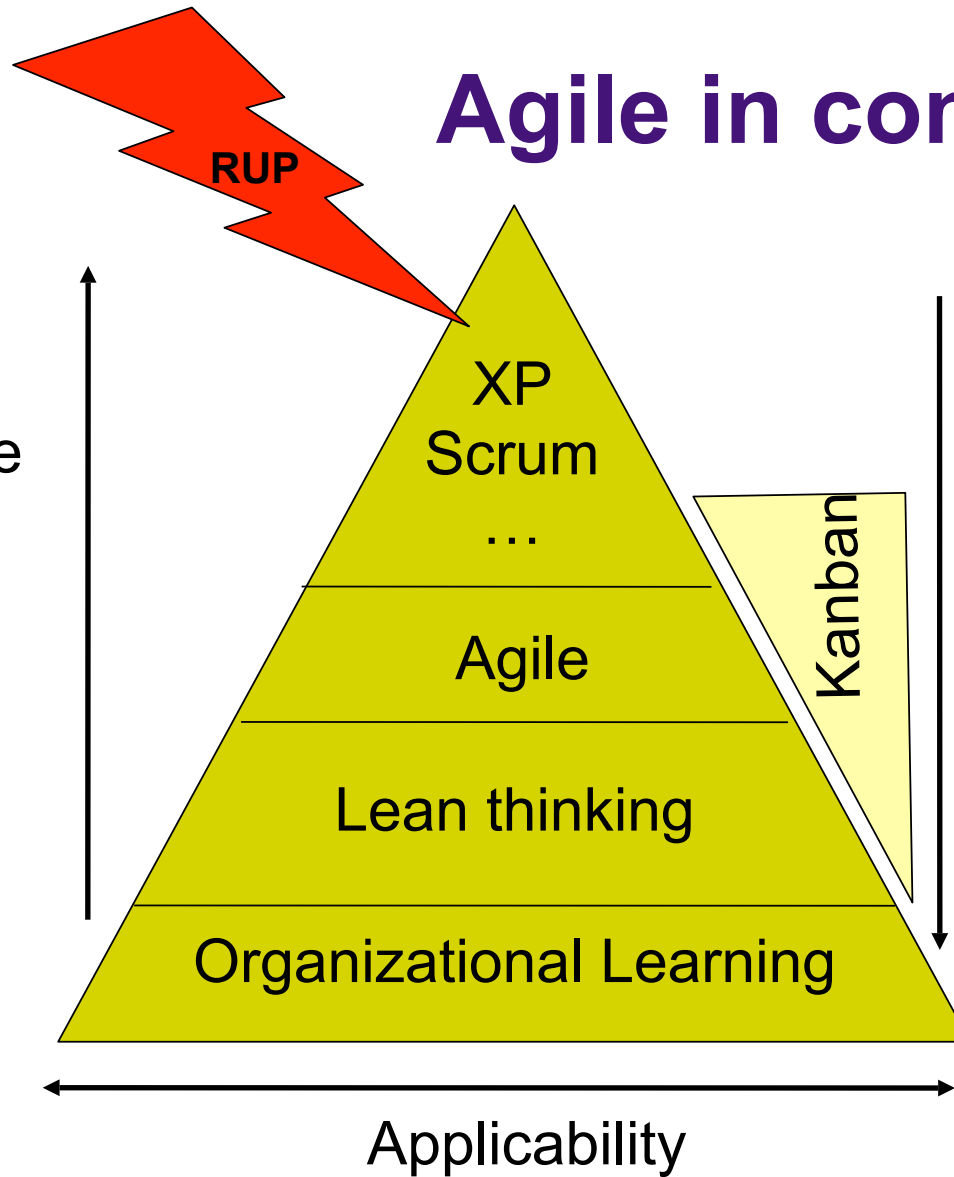
1.The Future is Lean



Agile in context

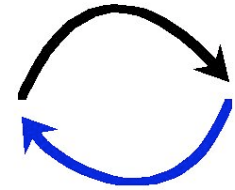


More
prescriptive

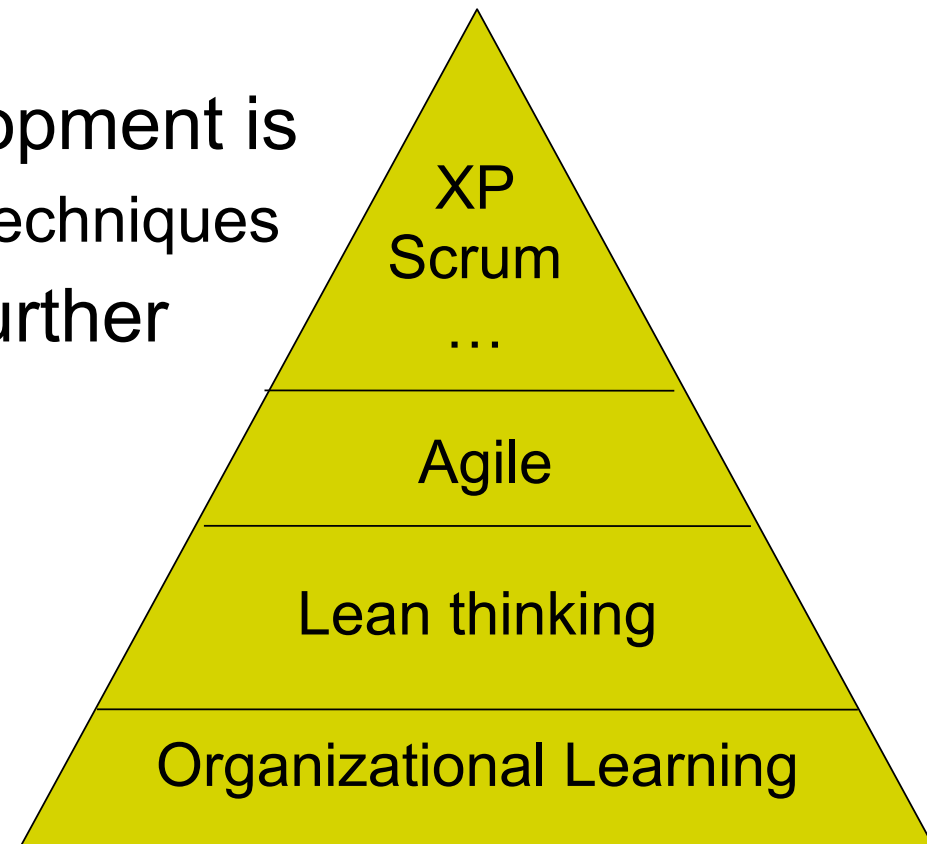


More
philosophical:
value, idea
based

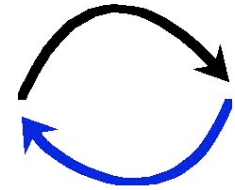
Agile == Lean



- Agile Software Development is Lean Product Development
- Agile Software Development is
 - Toolbox of well known techniques
- Lean Thinking goes further

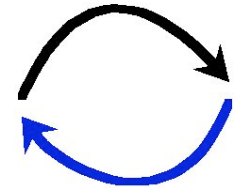


Lean principles



- Eliminate waste
 - 7 Wastes
- Level flow
- Visual management
 - Kanban
- Continual improvement
 - Organizational learning

7 Wastes

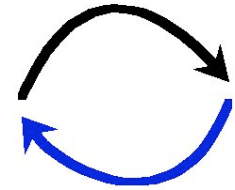


Inventory	Partially done work
Extra processing	Extra processes
Over production	Extra features
Transportation	Task switching
Waiting	Waiting
Motion	Motion
Defects	Defects

Exercise: Identify wastes for your team

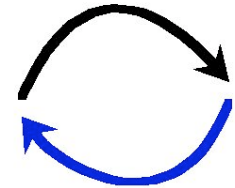
From Lean Software Development, Poppendieck & Poppendieck, 2003

Japanese terms



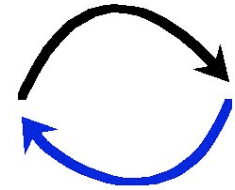
Andon	Signal of a problem
Gemba	Place of truth / Place of work
Genchi / Gembutsu	Go and see for yourself
Jidoka	Automation (with a human touch)
Kaizen	Continuous improvement
Kaikaku	Radical improvement / Kaizen blitz
Kanban	Signal for action / Visual signboard
Muda	Waste
Pika pika	Spic & span / Clean & tidy
Poka-yoke	Mistake proofing

A method from under the floorboards



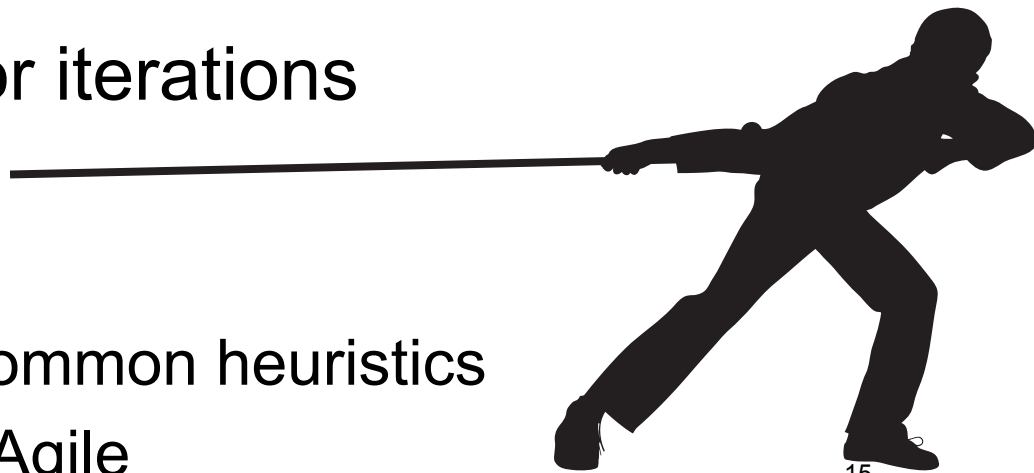
- Agile has been a bottom-up change initiative
 - Developers just started doing it
 - Little or no management instruction
 - Managers seen as a problem
- Top-down change initiatives have poor record
 - ISO-9000, CMM(I), BPR, ...
- Largely ignored business need (requirements)
 - Focus on development effectiveness

Kanban

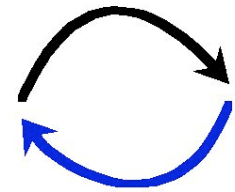


- The Kanban innovation
 - Explicitly limit Work in Progress
 - Break flow down into more stages
- Work is pulled from business
 - MMF - Minimally Marketable Feature
- No estimation or iterations
- Kanban is
 - Post-Agile
 - A collection of common heuristics
 - Dangerously unAgile

From David
Anderson



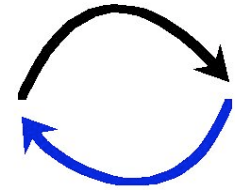
Kanban board



Kanban

Limited
queue

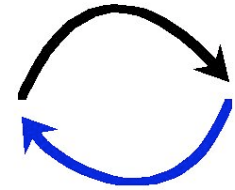
Limited
queue



Agreed	New	Ready (Prioritised) 5	Work in progress 4	Blocked	Done

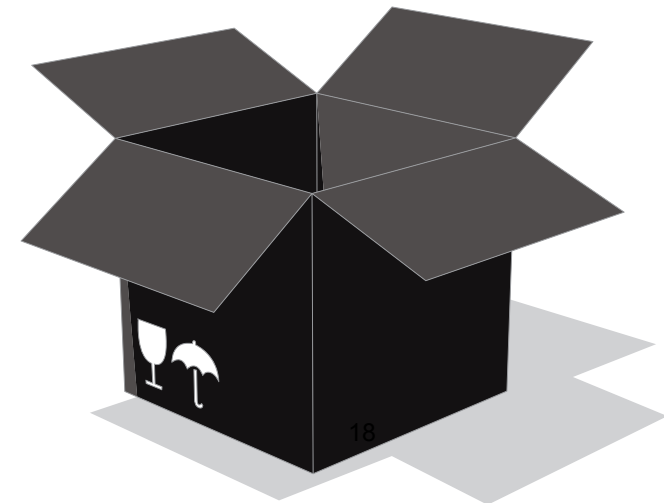
Count block
causes

Kanban

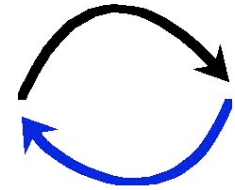


“a subtle difference between kanban and typical agile processes such as Scrum. Scrum focuses on being agile which may (and should) lead to improving. Kanban focuses on improving, which may lead to being agile. However, being agile itself is not important - it just happens to be the best way we (or at least I) know at the moment.”

Karl Scotland,
Early Kanban adopter
December 2008



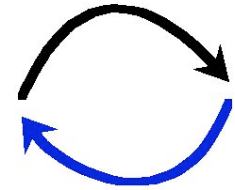
Experience with Kanban



- Easier to introduce
- More responsive than Scrum/XP
- Models current process then you change
- Keep management by routine to start with
 - iterations
 - planning
 - work breakdown
 - estimates
- Greater need to emphasis learning & change
 - No safety net of



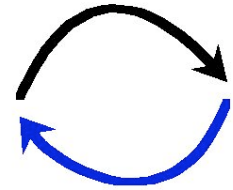
The Future



1. The Future is Lean
2. **Return of the Product Owner**



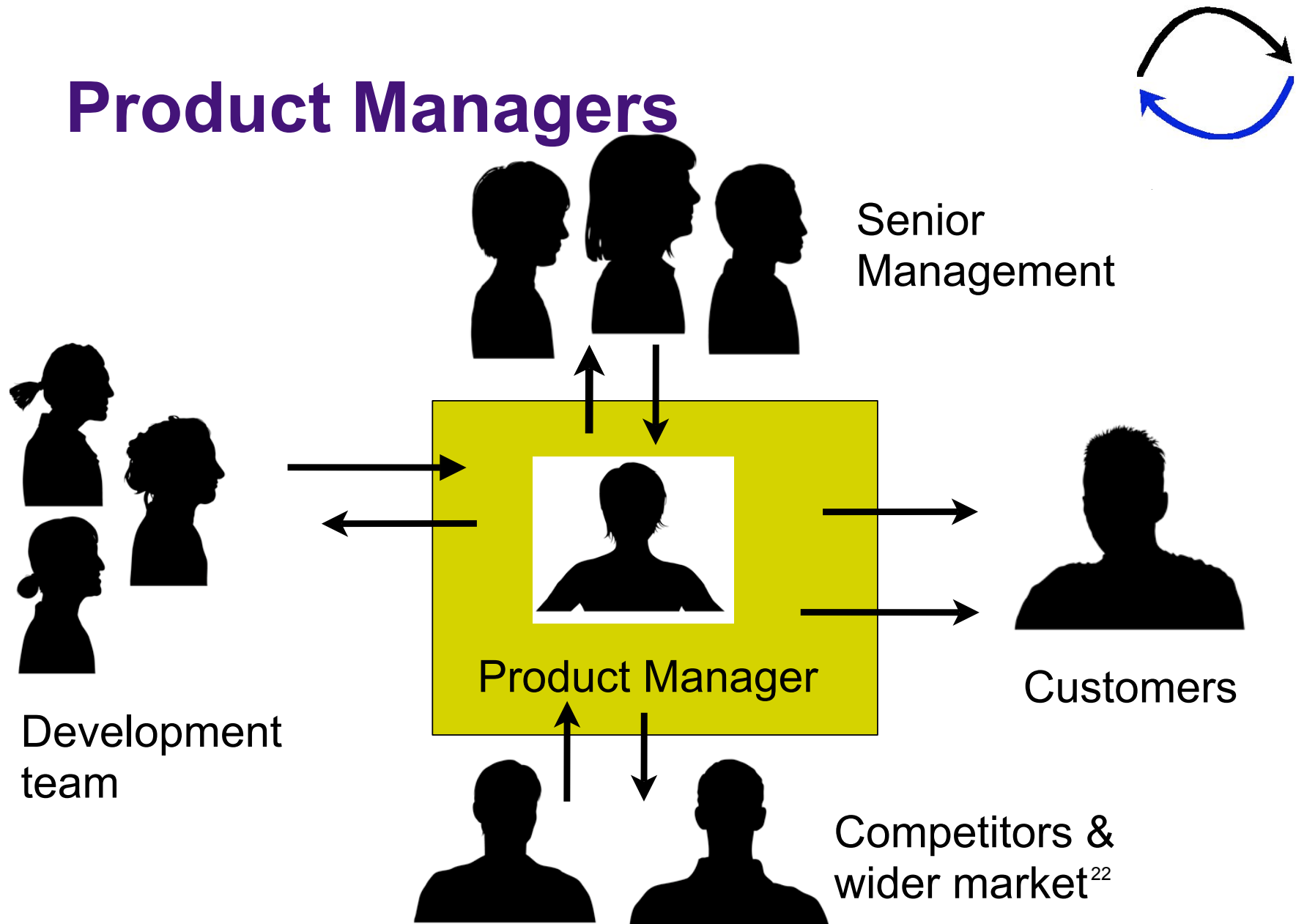
Who is the Product Owner?



- Business Analyst
 - Corporate IT
 - External Service providers
 - Internal focus - needs inside business
- Product Manager
 - Independent Software Vendors
 - Companies which sell software to a market
 - External focus - needs in the market

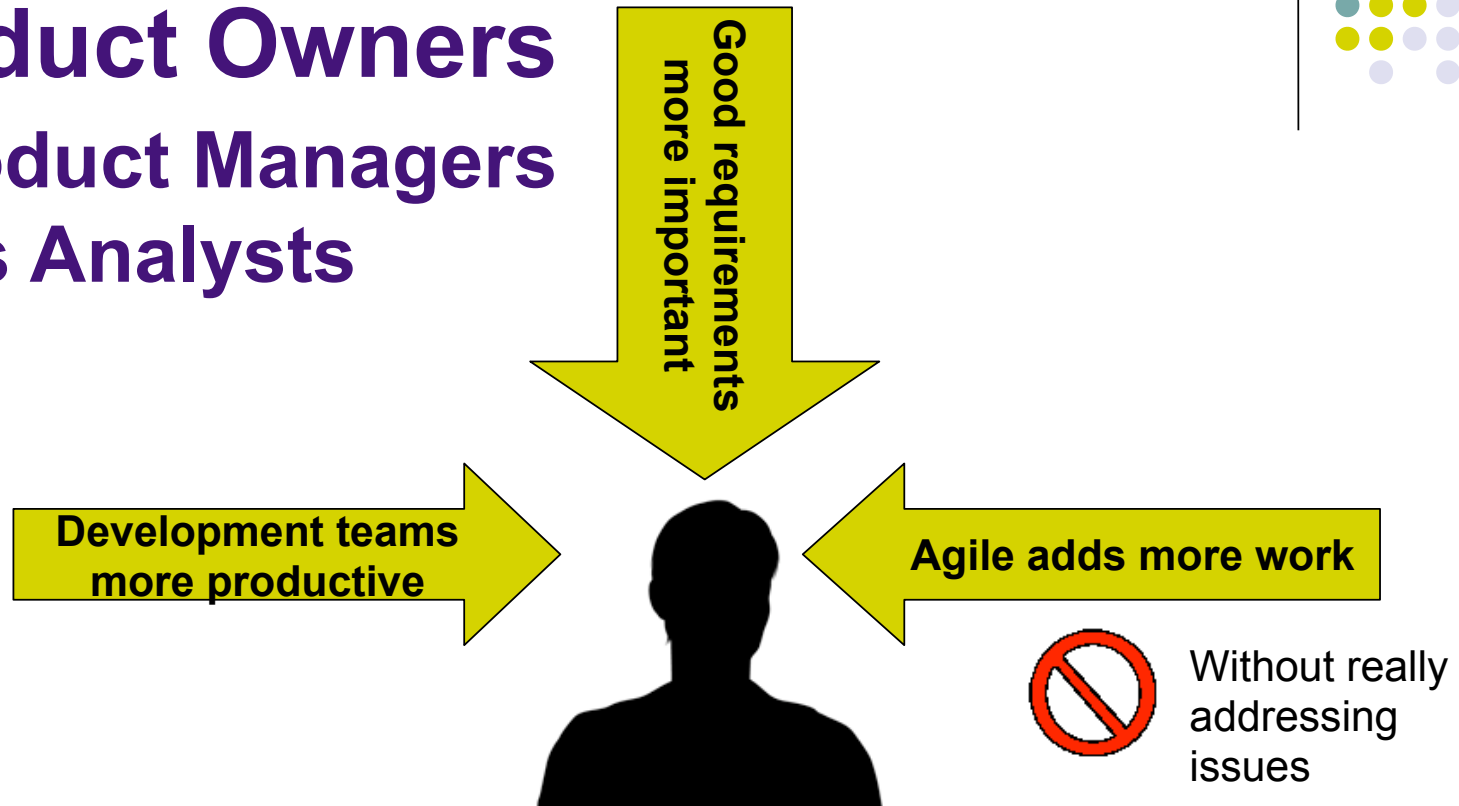


Product Managers



Pressure on Product Owners

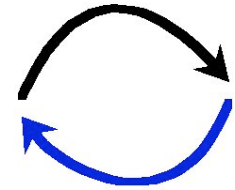
- Product Managers
- Bus Analysts



Need to:

- Pay more attention
- More staff: 1 product mgr to 3-7 developers

The Alignment Trap



Challenge 1:

- Get Agile
- From Maintenance to Well-oiled

Challenge 2:

- From Well-oiled
- To Growth

IT Highly aligned

Doing the right thing

Less aligned

'Alignment trap'
11% companies
+13% IT spending
-14% 3 year sales growth

'IT Enabled growth'
7% companies
-6% IT spending
+35% 3 year sales growth

'Maintenance zone'
74% companies
Avg IT spending
-2% 3 year sales growth

'Well-oiled IT'
8% companies
-15% IT spending
+11% 3 year sales growth

IT Less Effective

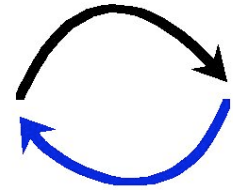
Doing things right

IT More Effective

Source: Shpilberg, Berez, Puryear, Shah: MIT Sloan Review, Fall 2007

Agile (as we know it) deficient in requirements


- XP Onsite customer too simplistic for most
- Scrum Product Owner better
 - but Scrum doesn't say how Product Owner knows
- User stories good... but
 - Life's more complicated
 - Many teams ignore scenarios, actors, personas
- But IT DOESN'T MATTER (so far)
 - Developer centric view helped avoid the Alignment Trap



Requirements

Mary & Tom Poppendieck
Implementing Lean Software
Development 2007

Only about 20% of
features & functions
in typical custom software
are used

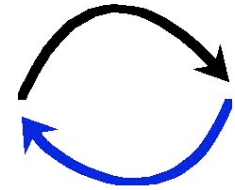


We're often
encountered
requirements churn of
30% to 50%

From the point of view of understanding “competitiveness”, “levels of achievement” and “associated risk,” the **performance requirements** are by far the most interesting requirements. Yet, traditionally, too much attention has been given to specification of functional requirements and resource requirements.

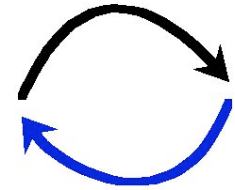
Tom Gilb
Competitive Engineering, 2005

Net Lessons



1. Don't get stuck in Alignment Trap
2. Alignment (Requirements) can lead you astray
3. Focus on effectiveness first
 - Agile as we know it (XP, Scrum, ...)
4. Then focus on alignment
 - Real benefit
 - Expect Lean/Agile to overflow development
 - Really getting Lean/Agile need requirements

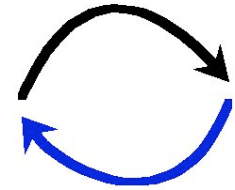
Getting better at Requirements



1. Product Managers or Business Analysts?
 - Decide which is right for you
2. Stop using to Project Managers to understand business need
 - Different role, different skills
3. Staff Product Owner role sufficiently
 - 1 Product Manager per 3 to 7 developers
4. BAs need to learn from Product Management



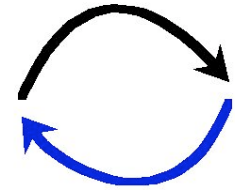
The Future



1. The Future is Lean
2. Return of the Product Owner
3. **The Change Question**



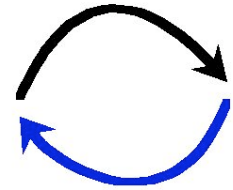
4 types of company



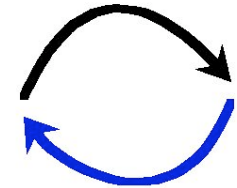
1. In the Maintenance zone / tar pit
 - Don't care / Make enough money elsewhere / Believe IT is always SNAFU
2. Are changing to Agile
 - All good things are now Agile
3. Happy to be Well Oiled
 - Top 8% of companies; pretty good place to be
4. Moving on Up - Growth
 - Good - want to be Great!

Lets assume

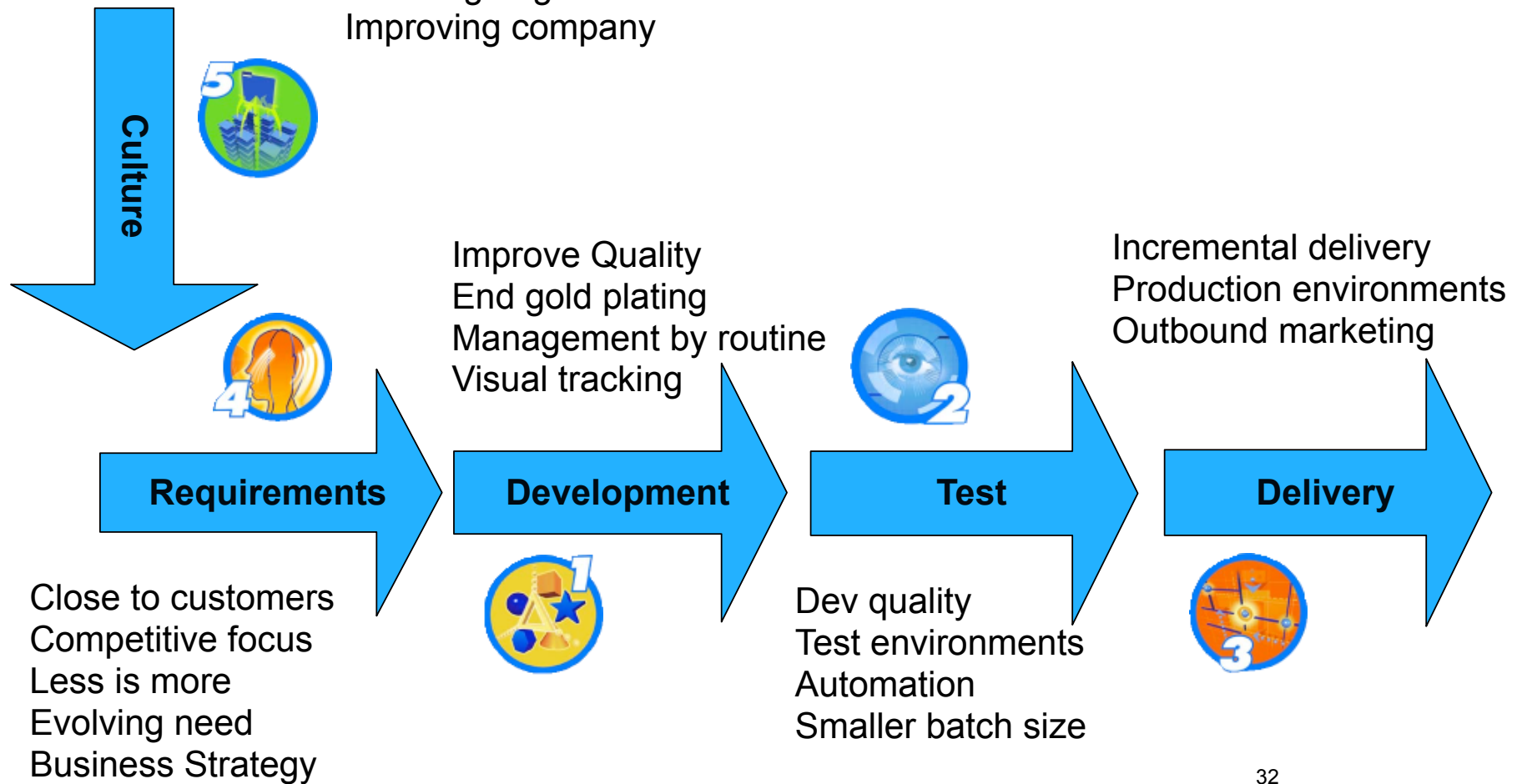
- You want to get better
 - Type 2 - Get Agile
- Or
- Type 4 - Move on Up
- You want to change



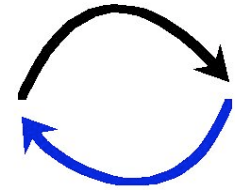
Follow the Bottleneck



Learning Organization
Improving company



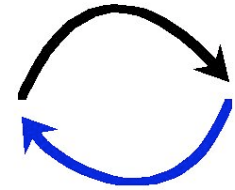
Managers



- Traditional Top-down change not the Agile way
- Pincer movement
 - Leadership over authority
 - Seed learning
 - Kindle the fire of Agile
 - Support bottom-up change
- Be ready to change yourself
 - Have bias for action



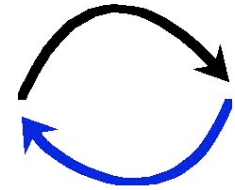
Recipe for change



1. Take a learning approach
2. *Do things Right* comes first then *Do The Right Things*
3. Seed *Bottom-up* change and support with *Top-Down*
4. Go Broad: evangelise Agile
5. Go Deep: one software development team
 - Go Development
 - Go Test
 - Go Deployment
6. Enhance Product Owner
7. Deepen & broaden: pull in more of the company
8. Work down the Agile Triangle
9. Repeat - faster!



Consultants



- Remember:

3. When the consultants, trainers and coaches who introduced Agile leave, are you still Agile?

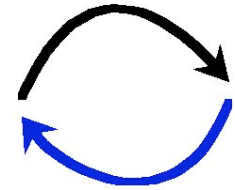
- You don't need consultants - *you know the answers*
- You use consultants to:
 - Go faster
 - Seed learning
 - De-risk
 - Overcome opposition
 - Show commitment



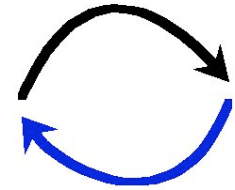
The Downturn

- Will accelerate Agile adoption
- Plays to Agile strengths
 - Reduce waste
 - Reduce inventory
 - Bring revenue forward
- Downturn reduces risk of change
 - Status quo is not an option
- Positions companies for upturn

} Improve cashflow



The Future

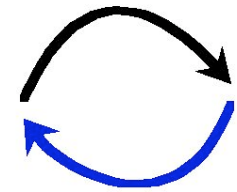
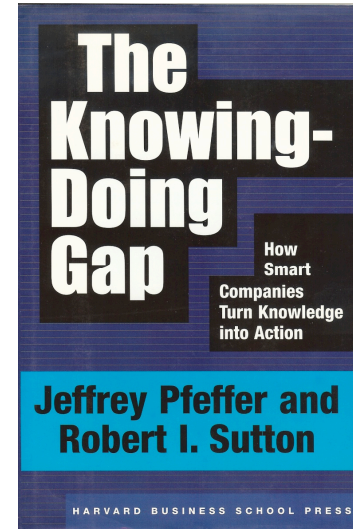


1. The Future is Lean
2. Return of the Product Owner
3. The Change Question
4. **People - the meta-issue**

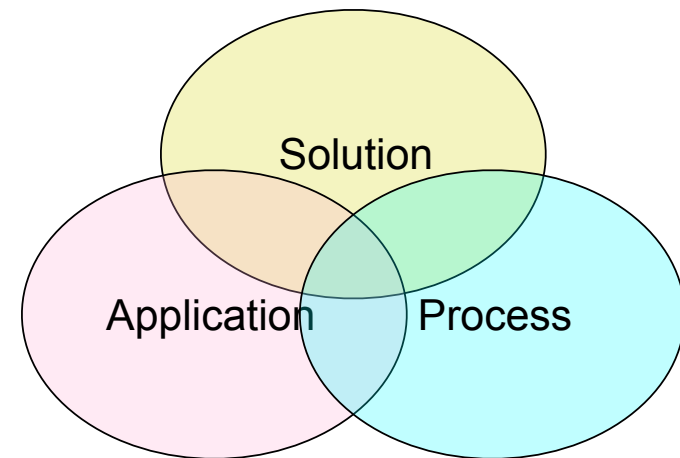


Knowing Doing Gap

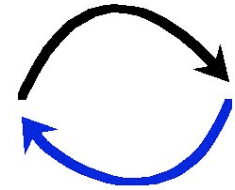
- Agility demands *change*
 - Change to Agile - and keep changing
- We don't act on what we know
 - e.g. TDD
- Why not?
 - Talk over action
 - Memory substitutes for action
 - Fear prevents action
 - Measurement obstructs good judgement
 - Internal competition - friends become enemies
- Agile Training a good start - not enough
 - Organization need to learn & change
 - A learning organization



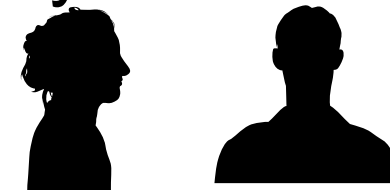
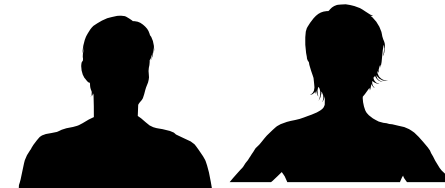
Pfeffer &
Sutton 2001



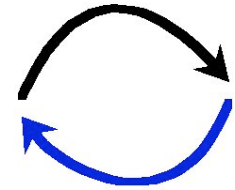
It's the People, stupid



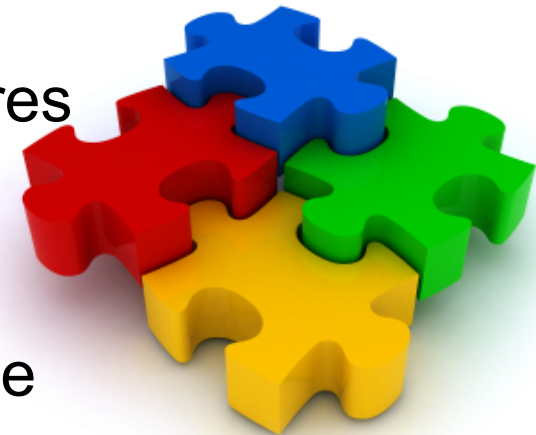
- Software development is all about people
 - Not processes
 - Not methods
 - Not tools
- To get better software development you want better people
 - How you going to do that?
- Only the Organizational Learning promises better people



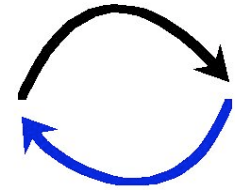
Future predictions



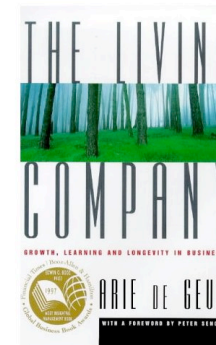
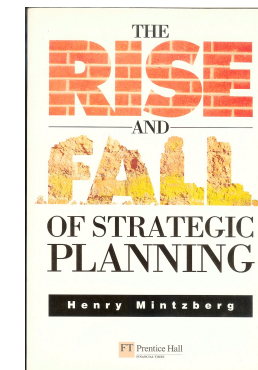
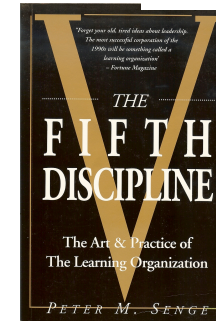
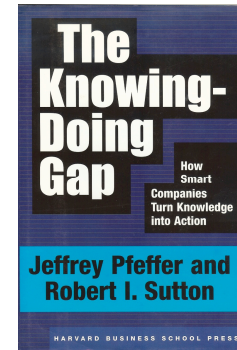
1. Downturn will accelerate Agile
2. More companies will try to adopt
 - Many will fail - expect more Agile failures
 - Most will stop at Well-Oiled
3. Expect to see more Kanban
4. More emphasis on requirements in Agile
5. Best companies move beyond Agile (as we know it)
6. Agile will break out of software development
7. Greater people focus
 - Organizational Learning is the true aim and true measure of success



Further reading Learning and the company

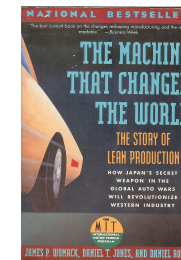
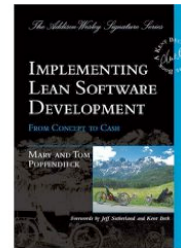
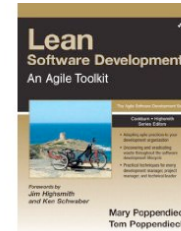


- Knowing Doing Gap
 - Pfeffer & Sutton, 2000
- Fifth Discipline
 - Senge, 1990
- Rise and Fall of Strategic Planning
 - Mintzberg, 1994
- The Living Company
 - de Geus, 1997
- Slack
 - DeMarco, 2001



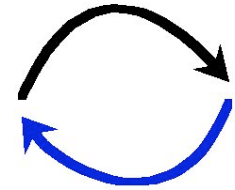
Further reading

- Lean Software Development
 - Poppendieck & Poppendieck, 2003
- Implementing Lean Software Development
 - Poppendieck & Poppendieck, 2007
- Machine that Changed the World
 - Womack, Jones & Roos, 1991
- Product Development for the Lean Enterprise
 - Kennedy, 2003

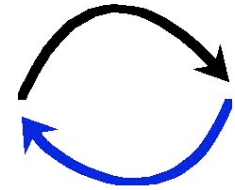


*Changing Software Development:
Learning to become Agile*

Kelly, 2008



Questions?



Allan Kelly

0773 310 7131

allan@allankelly.net

www.allankelly.net

blog.allankelly.net

Software Strategy Ltd.
www.softwarestrategy.co.uk

Training & Consulting for
Operational Excellence in
Software Development

(c) Allan Kelly, 2009