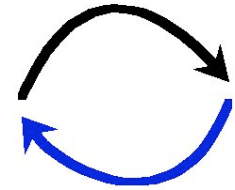


ACCU Oxford 2009



The Future of Agile

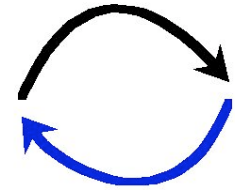
Allan Kelly

Software Strategy

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

(c) Allan Kelly, 2009

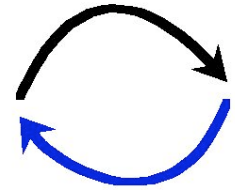
Allan Kelly, BSc, MBA



- Consulting in
 - Agile software development
 - Business of software development
 - Product requirements
- Training in
 - Agile, Lean, Scrum, etc.
 - Product development
- Agile Coaching
- Over 10 years as developer
 - Decided the problem is elsewhere
- Author:
 - *Changing Software Development: Learning to be Agile*, Wiley 2008.



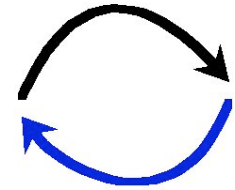
What is Agile?



There is no strict definition



Audience participation time

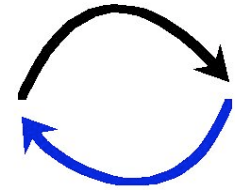


Where do you think Agile
is going?

Groups of 4
10 minutes
3 predictions



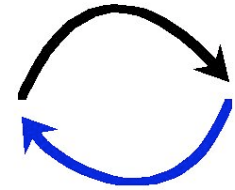
The Future, A summary



- 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



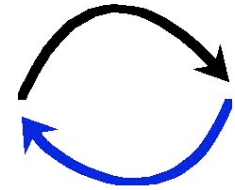
The Future



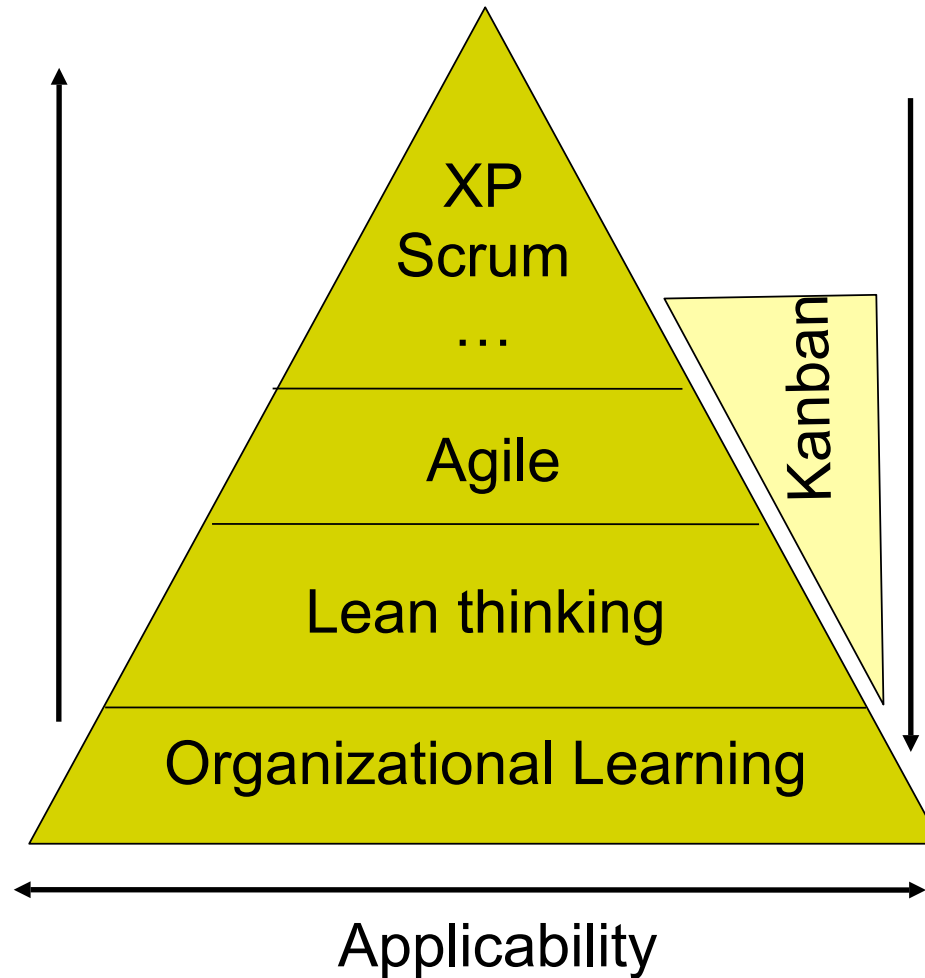
1.The Future is Lean



Agile in context

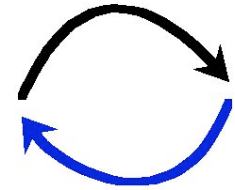


More
prescriptive

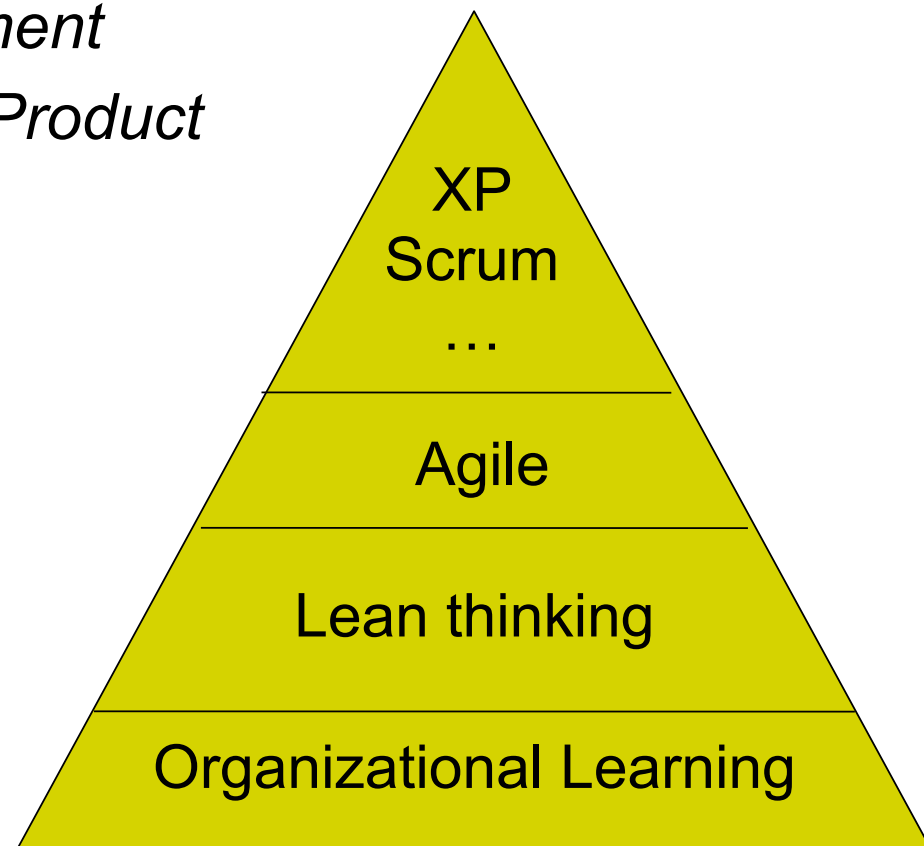


More
philosophical:
value, idea
based

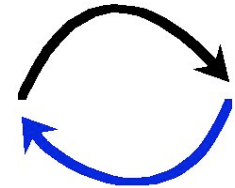
Agile == Lean



- Agile Software Development is
 - *Lean Product Development*
 - *aka Knowledge Based Product Development*

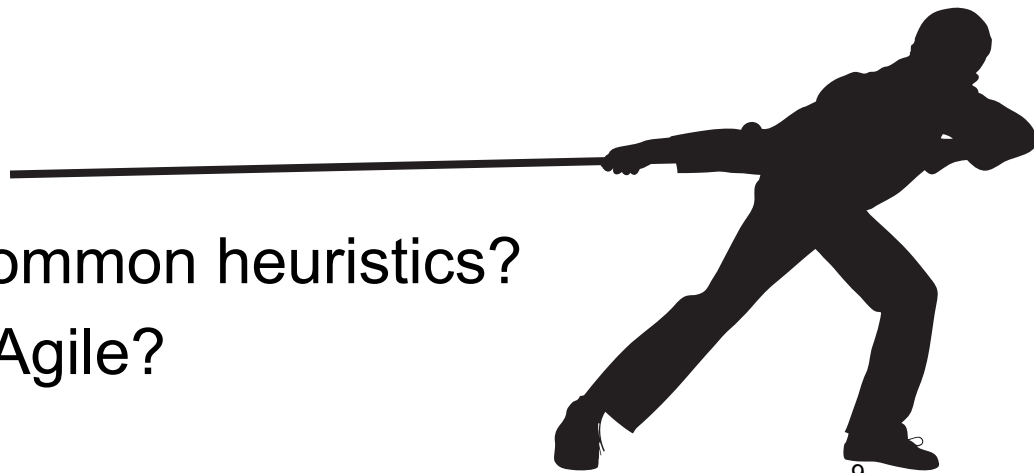


Watch Kanban - new tools!

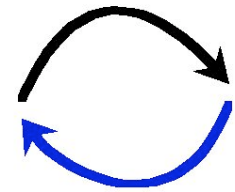


- The Kanban innovation
 - Explicitly limit Work in Progress
 - Break flow down into more stages
- Work is pulled from business
 - MMF - Minimally Marketable Feature
- Is Kanban
 - Post-Agile?
 - A collection of common heuristics?
 - Dangerously unAgile?

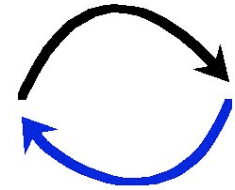
From David
Anderson



Kanban board

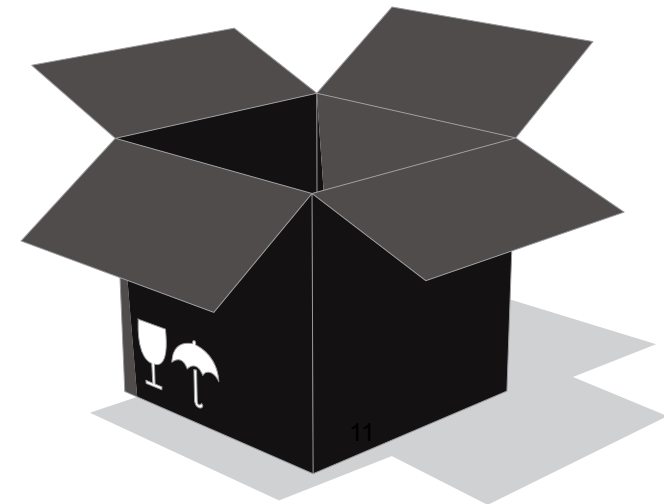


Kanban

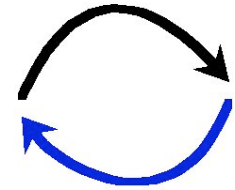


“a subtle difference between kanban and typical agile processes such as Scrum. Scrum focuses on being agile which may (and should) lead to improving. Kanban focuses on improving, which may lead to being agile. However, being agile itself is not important - it just happens to be the best way we (or at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

Karl Scotland,
Early Kanban adopter
December 2008



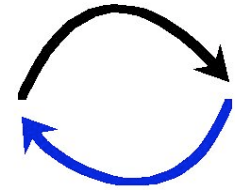
The Future



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



Who is the Product Owner?

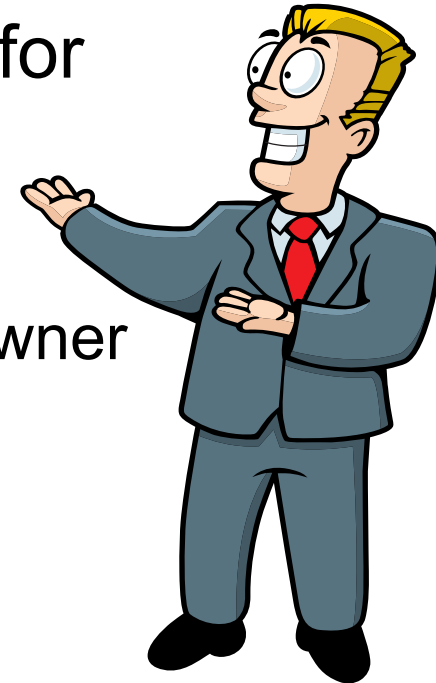
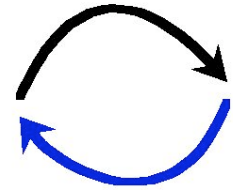


- Business Analyst
 - Corporate IT
 - External Service providers
 - Internal focus - needs inside business
- Product Manager
 - Independent Software Vendors
 - Companies which sell software to a market
 - External focus - needs in the market

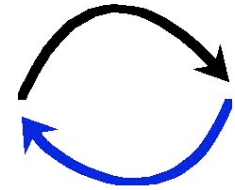


Agile (as we know it) deficient in requirements

- 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



The Alignment Trap



IT Highly aligned

Doing the right thing

'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

Less aligned

IT Less Effective

Doing things right

IT More Effective

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

Better requirements are needed

Better business understanding needed

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.

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

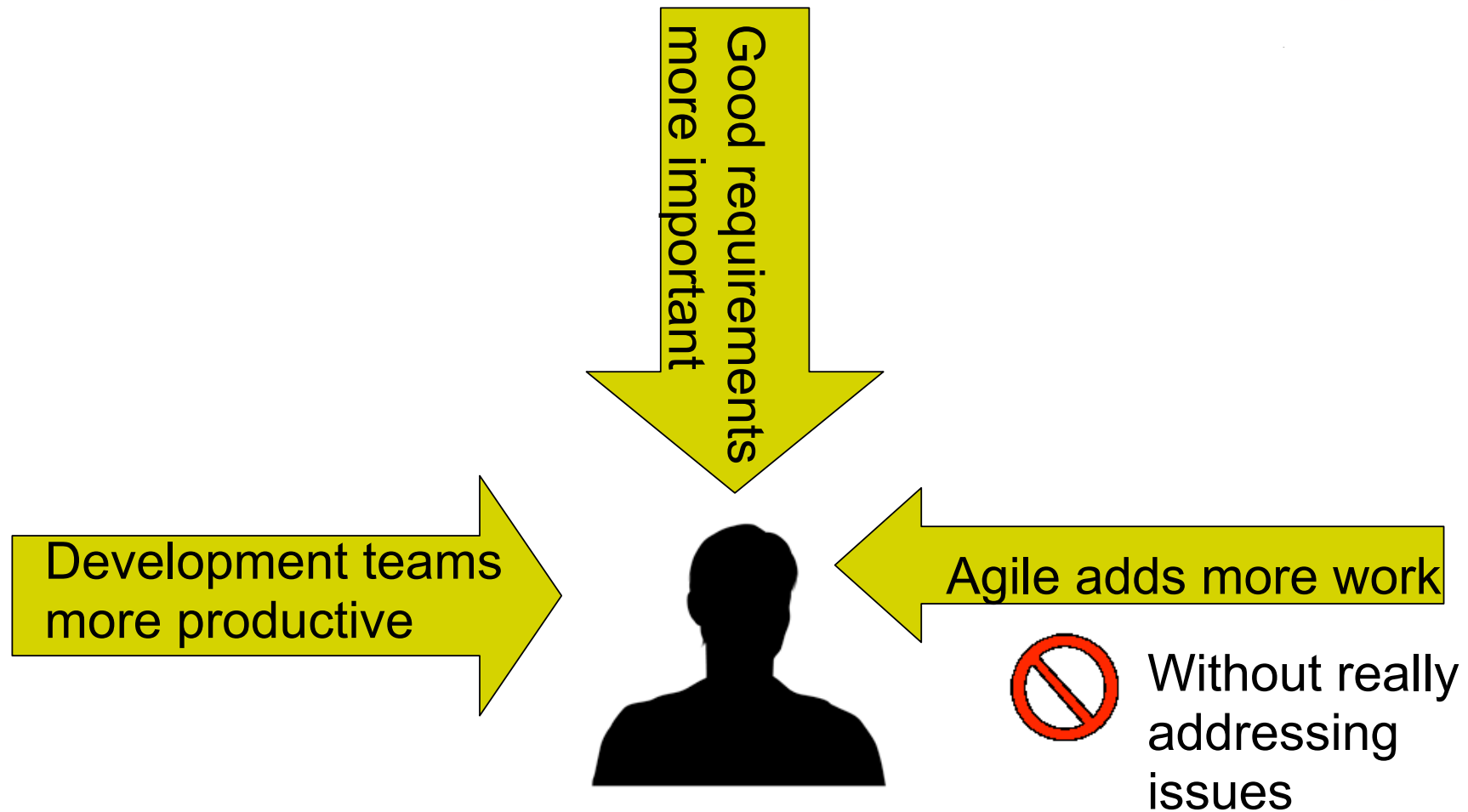
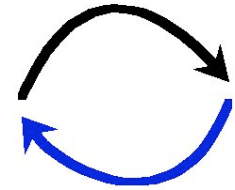
We're often encountered requirements churn of 30% to 50%

Mary & Tom Poppendieck
Implementing Lean Software Development 2007

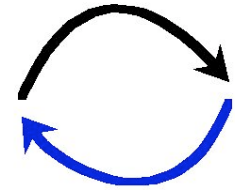
Tom Gilb
Competitive Engineering, 2005

Pressure on Product Owners

Product Manager / Business Analyst



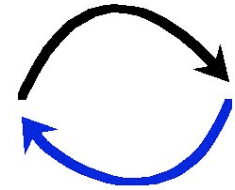
Getting better at Requirements



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



The Future

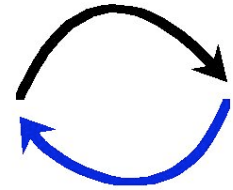


1. The Future is Lean
2. Return of the Product Owner
3. **The Change Question**

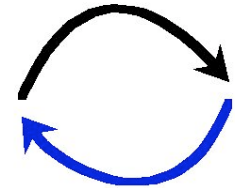


Lets assume

- You want to get better
 - Adopt Agile
- You want to get better at what you do
 - Do Agile better

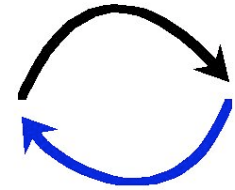


Change from below



- 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

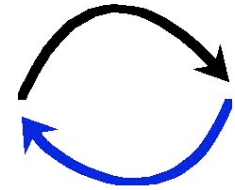
Managers



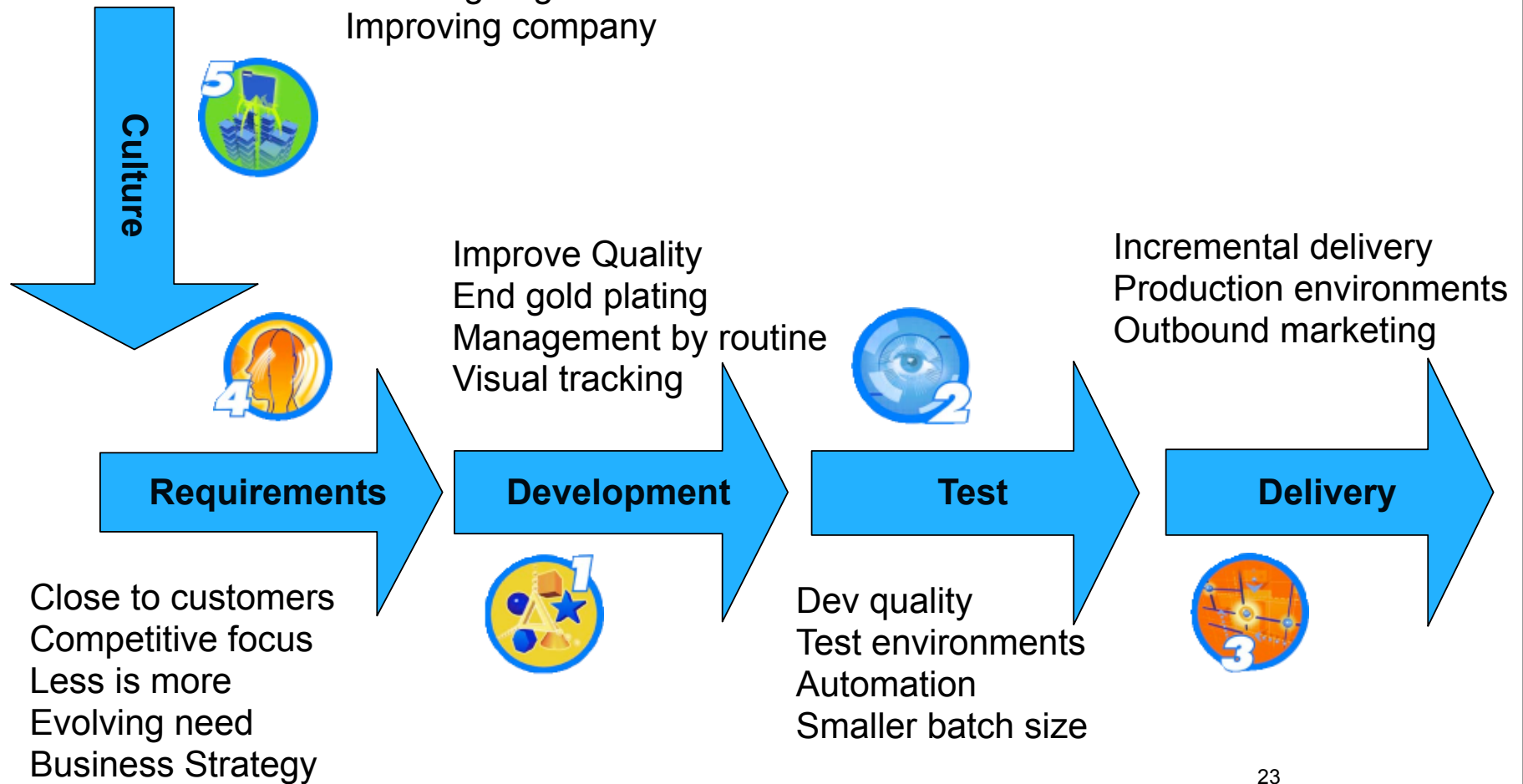
- Traditional Top-down change not the Agile way
- Pincer movement
 - Leadership over authority
 - Seed learning
 - Kindle the fire of Agile
 - Support bottom-up change
- Be ready to change yourself
 - Have bias for action



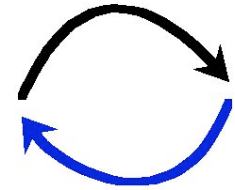
Follow the Bottleneck



Learning Organization
Improving company



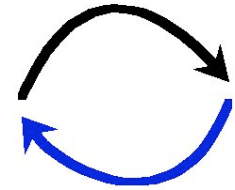
Recipe for change



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!



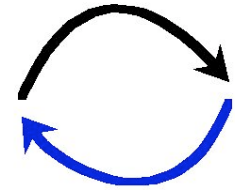
The Future



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

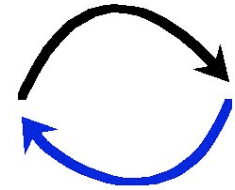


Agility demands change

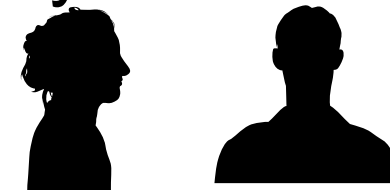
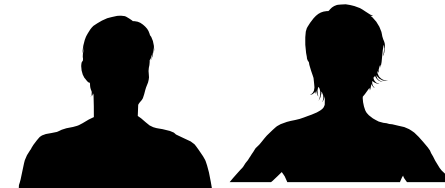


- Change to Agile
 - Keep changing to improve
- People have to change

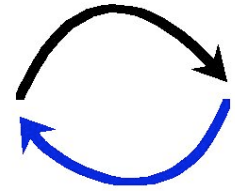
It's the People, stupid



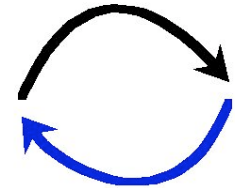
- Software development is all about people
 - Not processes
 - Not methods
 - Not tools
- To get better software development you want better people
 - How you going to do that?
- Only the Organizational Learning promises better people



Closing



Software Craftsmanship



On the one
hand...



And on the
other...

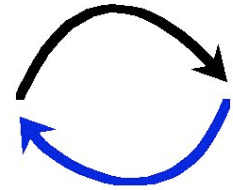


- Better coding is needed
- Technical practices neglected as Scrum overtaken XP
- Need for improvement all around
- Coding is small part of the issue
- Emphasis on technical practices may set back movement
- Alienated Product Owners & Managers

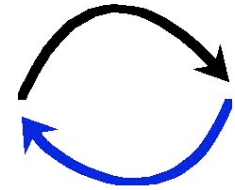
The Downturn

- 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



Audience participation time

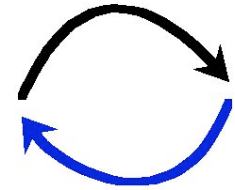


Now you've heard me...
Where do you think Agile
is going now?

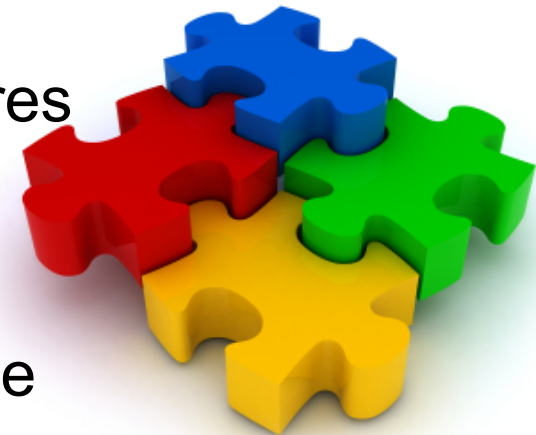
Groups of 4
10 minutes
3 predictions



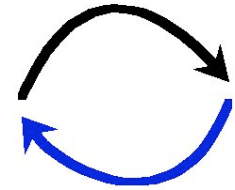
Future predictions



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



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