Agile Eastern Europe, Kiev, October 2010

More important than ever: The Business Analysts' role in Agile software development

Allan Kelly

allan@allankelly.net

http://www.allankelly.net

Software Strategy

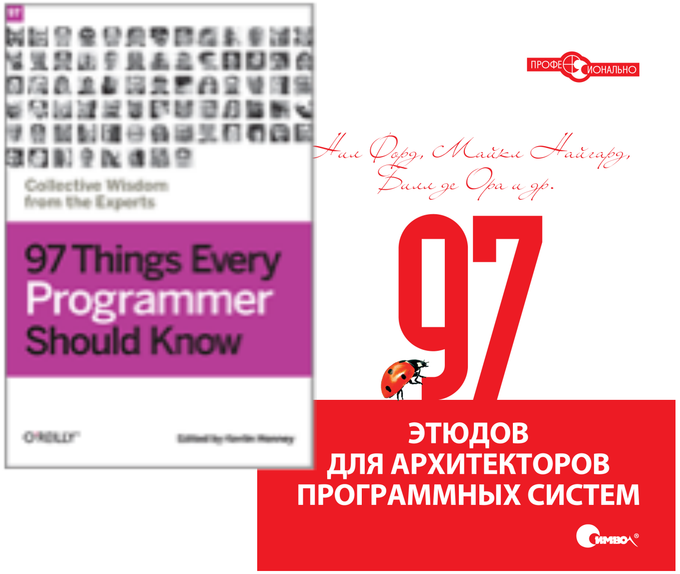
http://www.softwarestrategy.co.uk

Allan Kelly, BSc, MBA

- Training, Consulting & Coaching in Agile software development
- Author:
  - *Changing Software Development: Learning to be Agile*, Wiley 2008.

97 Things Every Programmer Should Know, Henney, 2010

Context Encapsulation in *Pattern Languages of Program Design* volume 5, 2006


Collective Wisdom from the Experts
97 Things Every Programmer Should Know
O'REILLY
Edited by Kevlin Henney
ПРОФЕССИОНАЛЬНО
Нил Форд, Майкл Найгард, Билл де Ора и др.
97
ЭТЮДОВ ДЛЯ АРХИТЕКТОРОВ ПРОГРАММНЫХ СИСТЕМ
Символ

What is Agility?

“Agile processes promise to react flexibly to changing requirements, thus providing the highest business value to the customer at any point in time”

Jutta Eckstein 2004

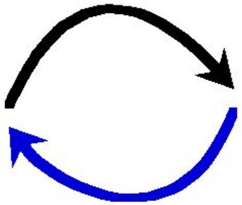
Jim Highsmith, 2002

“Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.”

What is Agility?

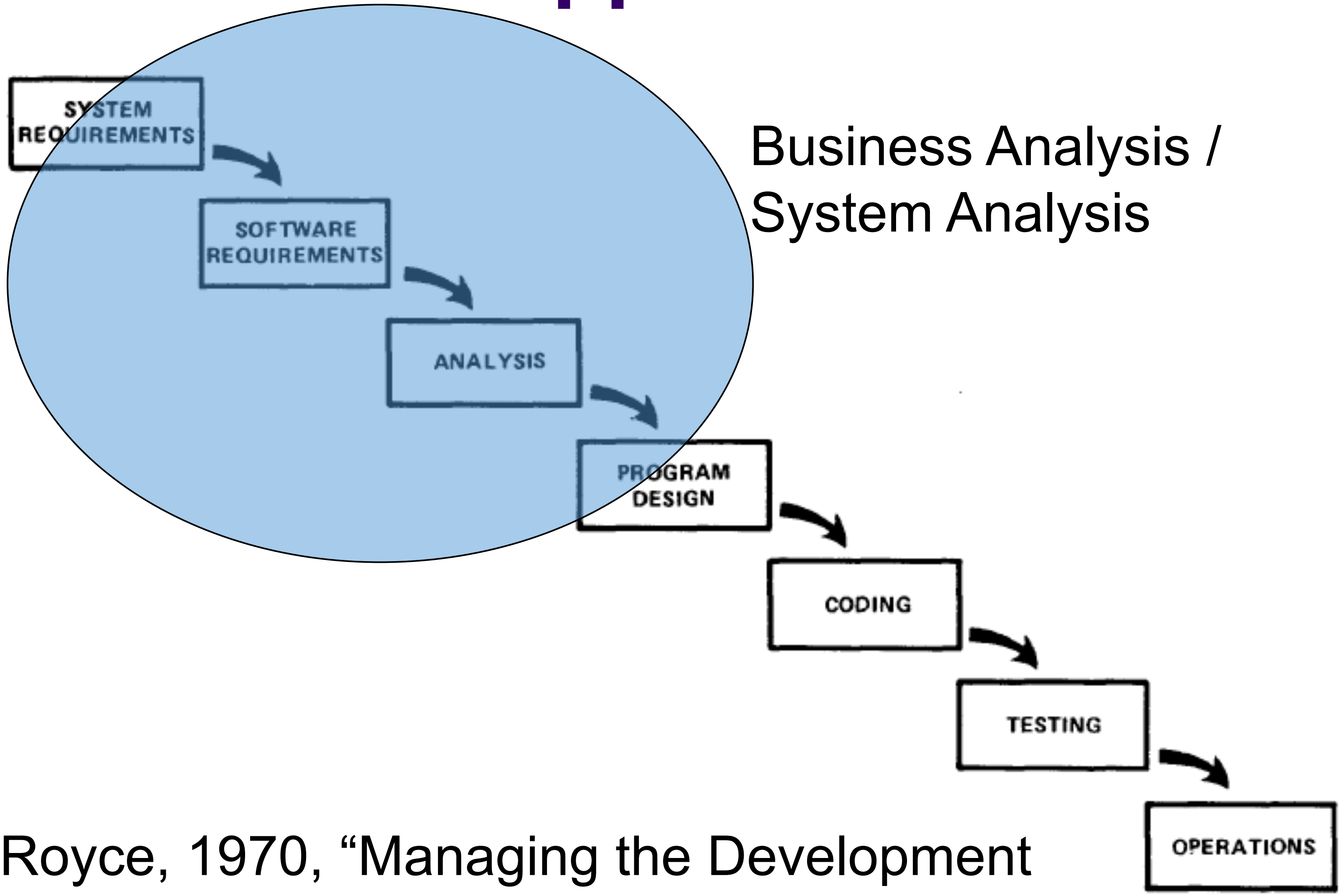
- Agile Today: *Agile as Better*
  - Respond to changing (business) environment
  - Faster, more productive, higher quality
  - Happier customers
- Agile Tomorrow: Agile creates new business models
  - Opportunities for those not confined by traditional IT

Agile



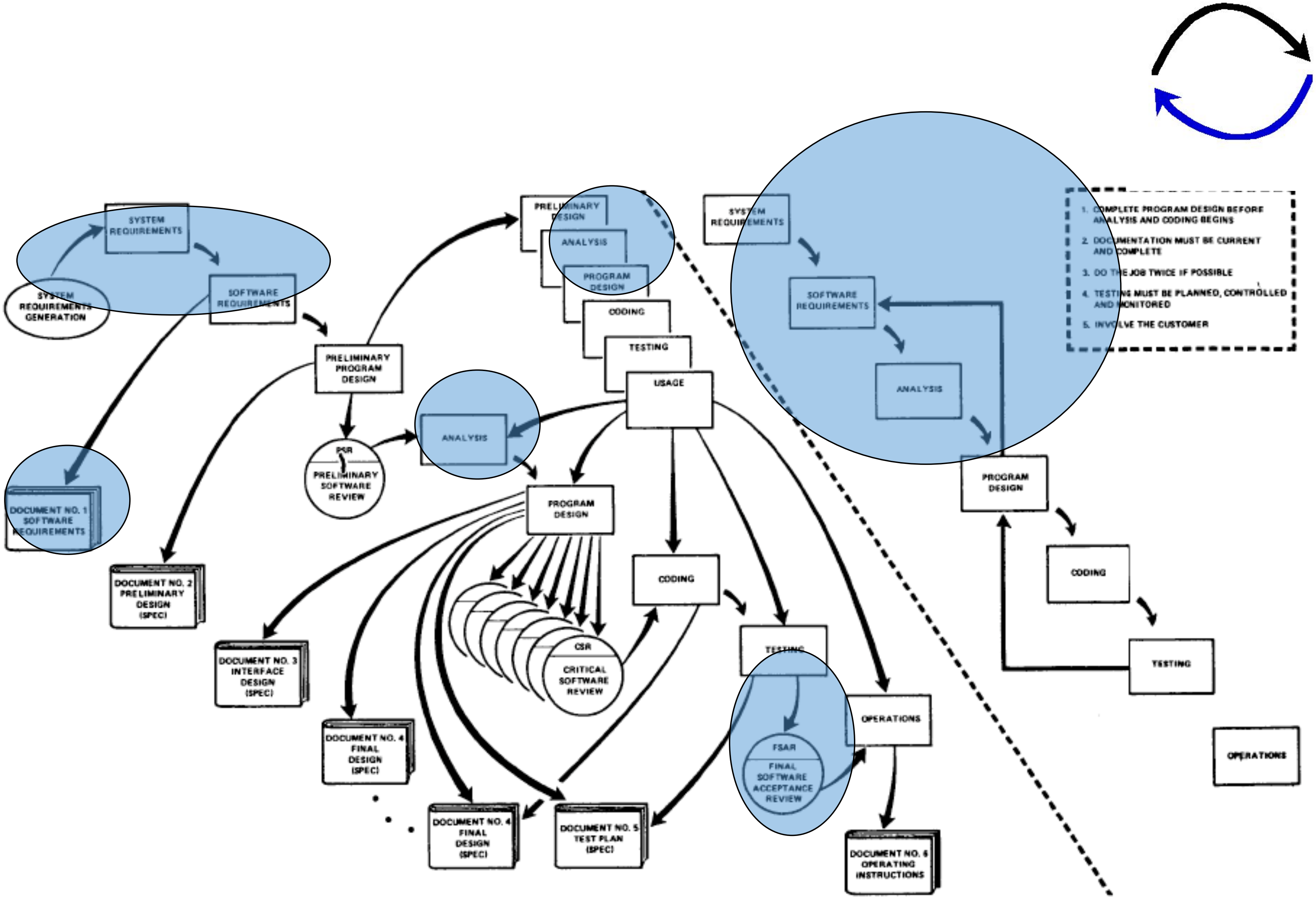
- Its the business need, *stupid*

Traditional approach



Business Analysis /
System Analysis

Royce, 1970, “Managing the Development of Large Software Systems”



SYSTEM REQUIREMENTS GENERATION
SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
DOCUMENT NO. 1 SOFTWARE REQUIREMENTS
PRELIMINARY PROGRAM DESIGN
DOCUMENT NO. 2 PRELIMINARY DESIGN (SPEC)
PSR PRELIMINARY SOFTWARE REVIEW
ANALYSIS
PRELIMINARY DESIGN
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
USAGE
PROGRAM DESIGN
DOCUMENT NO. 3 INTERFACE DESIGN (SPEC)
DOCUMENT NO. 4 FINAL DESIGN (SPEC)
DOCUMENT NO. 4 FINAL DESIGN (SPEC)
CSR CRITICAL SOFTWARE REVIEW
CODING
TESTING
DOCUMENT NO. 5 TEST PLAN (SPEC)
FSAR FINAL SOFTWARE ACCEPTANCE REVIEW
OPERATIONS
DOCUMENT NO. 6 OPERATING INSTRUCTIONS
SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
OPERATIONS

1. COMPLETE PROGRAM DESIGN BEFORE ANALYSIS AND CODING BEGINS
2. DOCUMENTATION MUST BE CURRENT AND COMPLETE
3. DO THE JOB TWICE IF POSSIBLE
4. TESTING MUST BE PLANNED, CONTROLLED AND MONITORED
5. INVOLVE THE CUSTOMER

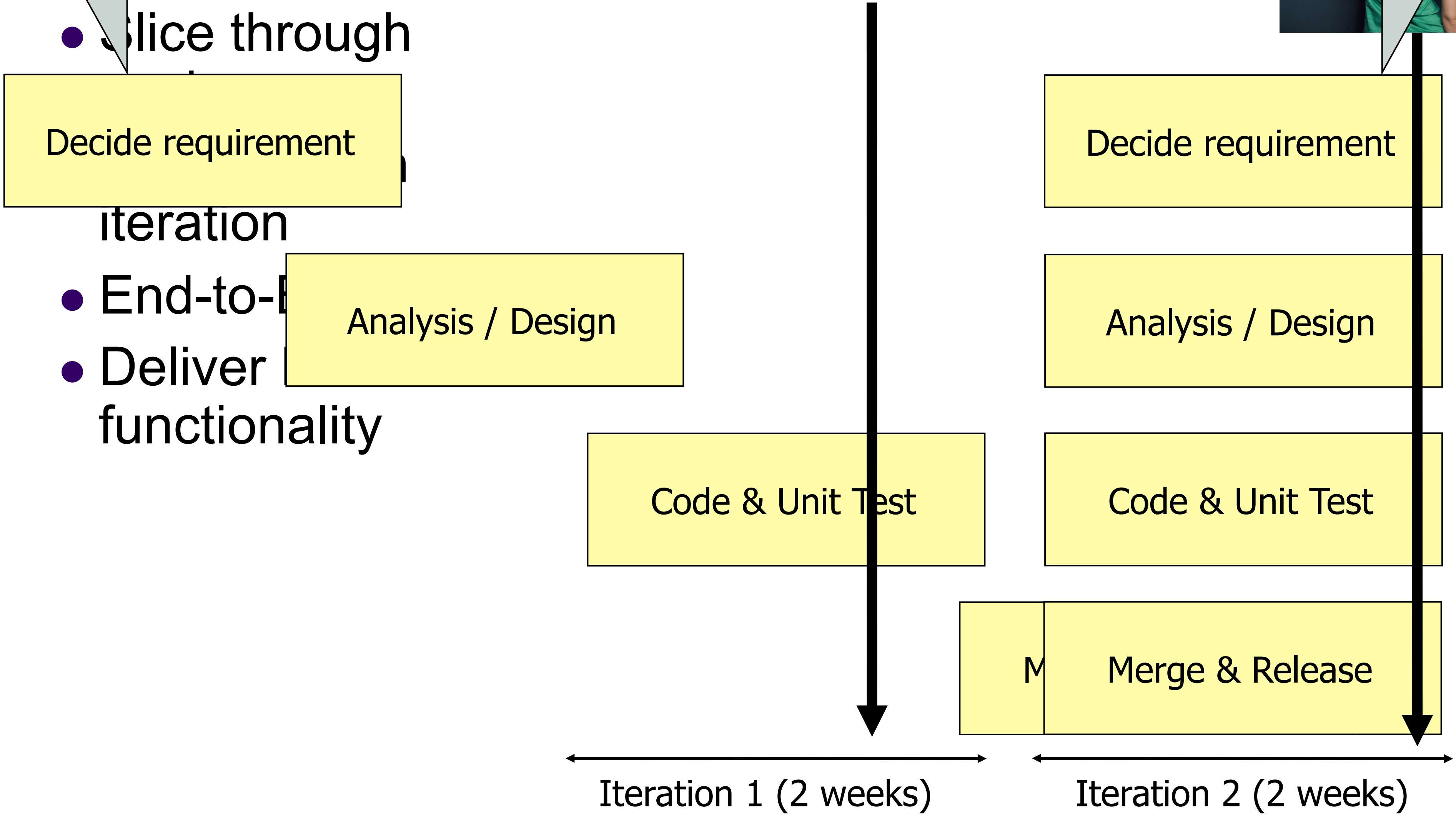
Traditional approach

Agile approach

BA/Product Owner works ahead of team - scouting out requirements

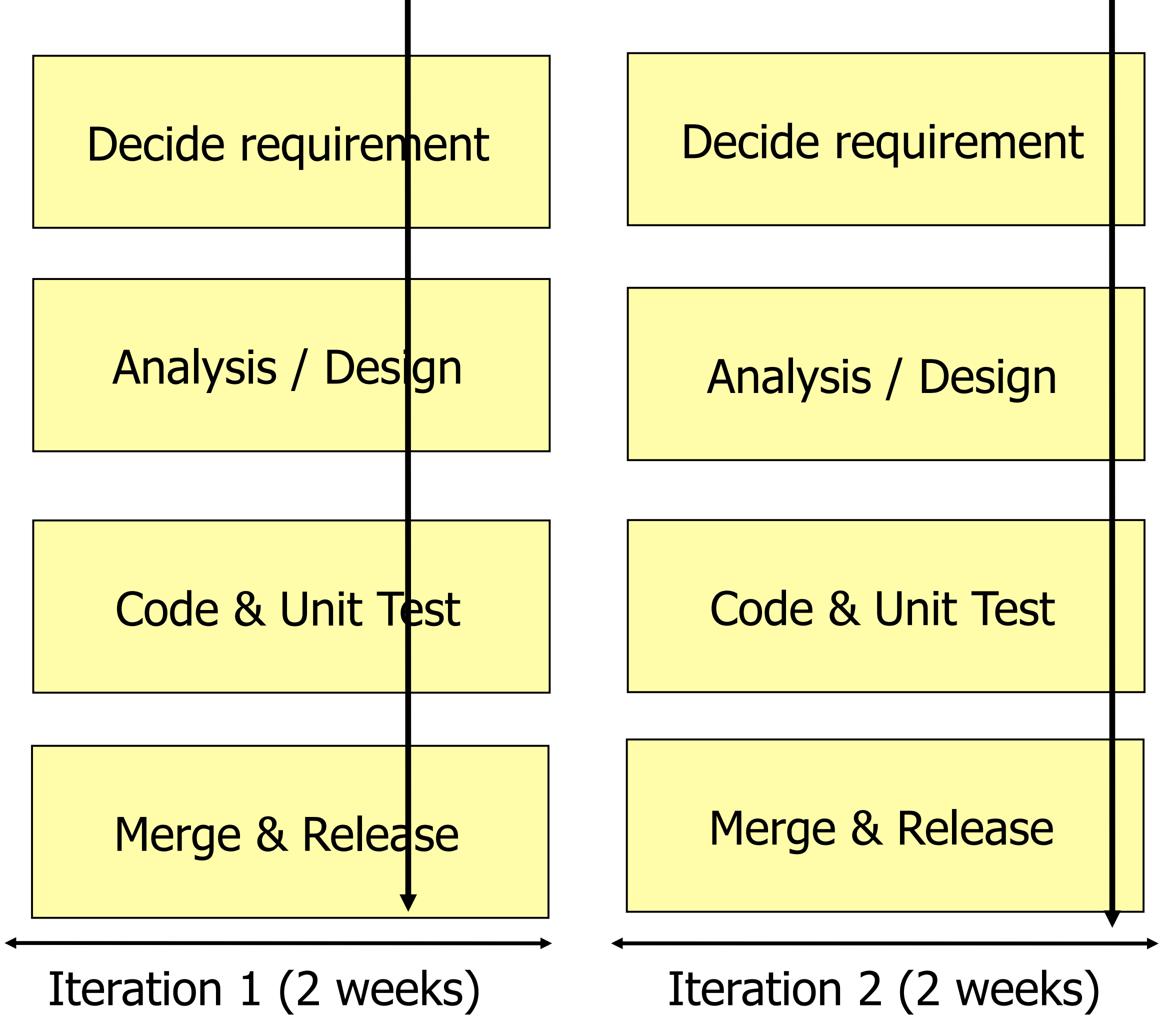
6+ months

- Slice through
- iteration
- End-to-E
- Deliver
  functionality



BA/Product Owner works ahead of team - scouting out requirements

- Slice through work
- Everything in iteration
- End-to-End
- Deliver business functionality



Decide requirement

Analysis / Design

Code & Unit Test

Merge & Release

Iteration 1 (2 weeks)

Decide requirement

Analysis / Design

Code & Unit Test

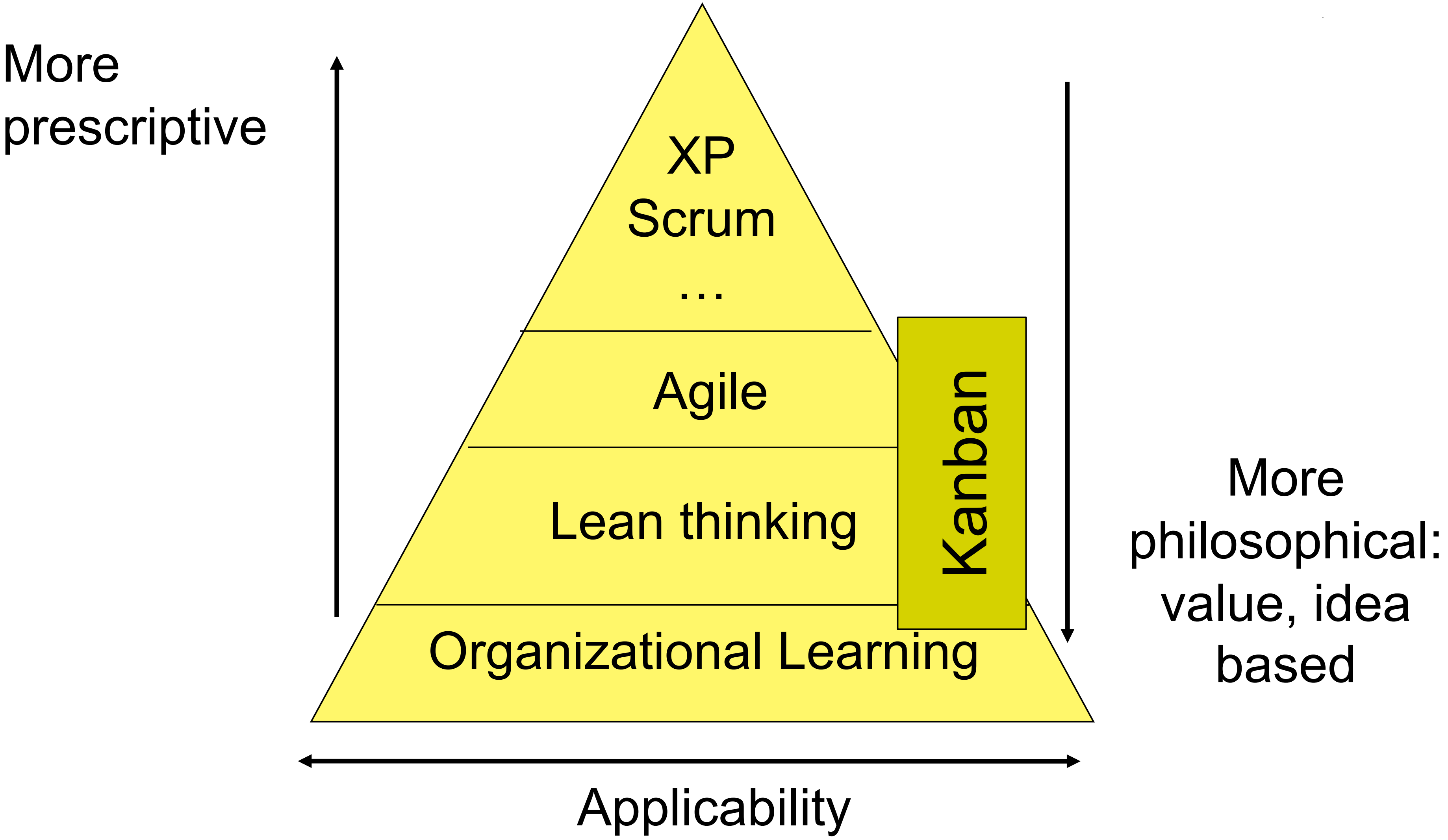
Merge & Release

Iteration 2 (2 weeks)

Lesson 1: Continual requirements

- Discovery
- Understanding
- Refinement
- Valuation
- Prioritization
- Explanation

Agile in context



More prescriptive

XP
Scrum
…

Agile

Lean thinking

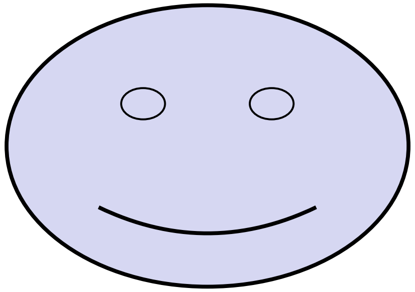
Organizational Learning

Kanban

More philosophical: value, idea based

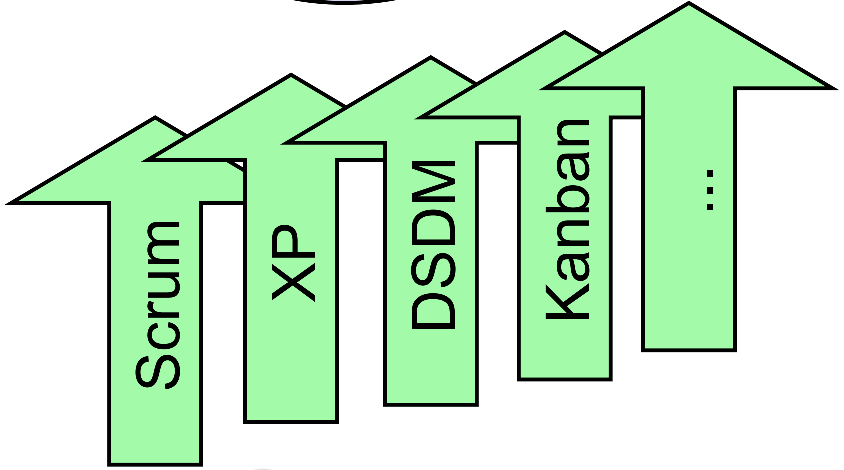
Applicability

Agile, Agile methods & Agile toolkit



The State of *Agile* (our objective)

- Quick on our feet
- Deliver quickly
- Respond to change rapidly
- Seize opportunities



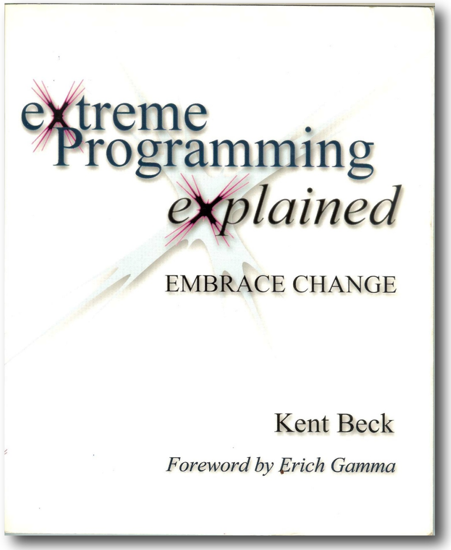
Agile Methods

Promise to create the state of Agile

The Agile Toolkit

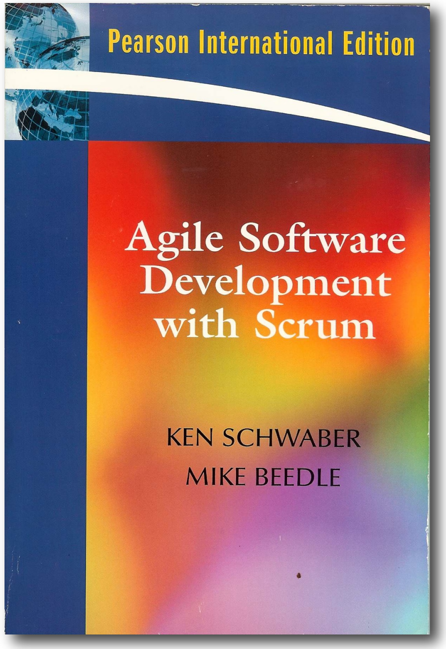
- Test Driven Development, Refactoring
- Iterations, Time boxing
- User Stories, Feature injection
- Retrospectives, ...

1999-2004: Agile = XP

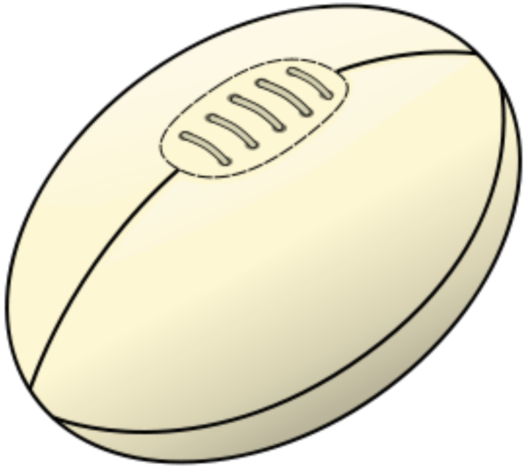


- Extreme Programming
  - First Agile method to gain popularity
  - Developer centric practices and literature
- Business need from *onsite Customer*
  - Customer on C3 was a **Business Analyst**
- “Customer” view simplistic
  - Short sighted
  - Assume customer knows
  - No discussion on how the customer knows

2005-today: Agile = Scrum


Pearson International Edition
Agile Software Development with Scrum
KEN SCHWABER
MIKE BEEDLE

- Scrum
  - A project management method without a project manager
- *Product Owner* specifies need
- Scrum silent on how the Product Owner knows



Who is the Product Owner?

Subject Matter /Domain Expert

Product Manager

Business Analyst

3 Types of Software Makers

Software Product Company

Microsoft, Oracle, Adobe, TargetProcess
Product Owner is Product Manager

Software Service Company

Accenture, Luxoft, Tata, EDS (HP), …
(Proxy) Product Owner is Business Analyst

ESP - External Service Provider

Corporate IT Department

Banks, Pharmaceutical, Oil, …

•Product Owner is Business Analyst

Or

•Business Analysts supports non-technical Produce Owner

Lesson 2: Know your Product Owner

Business Analyst is either

- Is the Product Owner

Or

- Supporting Product Owner

Business Analysts Role

- Ensure Agile teams meet business need
- Help explain business need to development team
- Act as “Product Owner” when
  - No other Product Owner
  - Product Owner does not have time or skills
- Advise development team on business details

How to help a team

- Work closely with the team
  - Sit with the team
  - Be part of the team
- Stay involved
  - Do not leave after initial stages
- Requirements gathering is ongoing process
  - Not only at the start
- Dialogue over document

How to help a team

- Be ONE step ahead
  - Ready to answer questions
  - But not too far ahead
- Goals and objectives
  - Replace *Big Requirements Documents*
  - Under continual review
- Delivered functionality changes and evolves
  - In direction of the goal and objective

Less (software) is more (value)

Potentially 80% of software development work is waste

- Better understanding can reduce demand by 80%

If 30+% of requirements change then

- Why bother doing work on them in the first place?

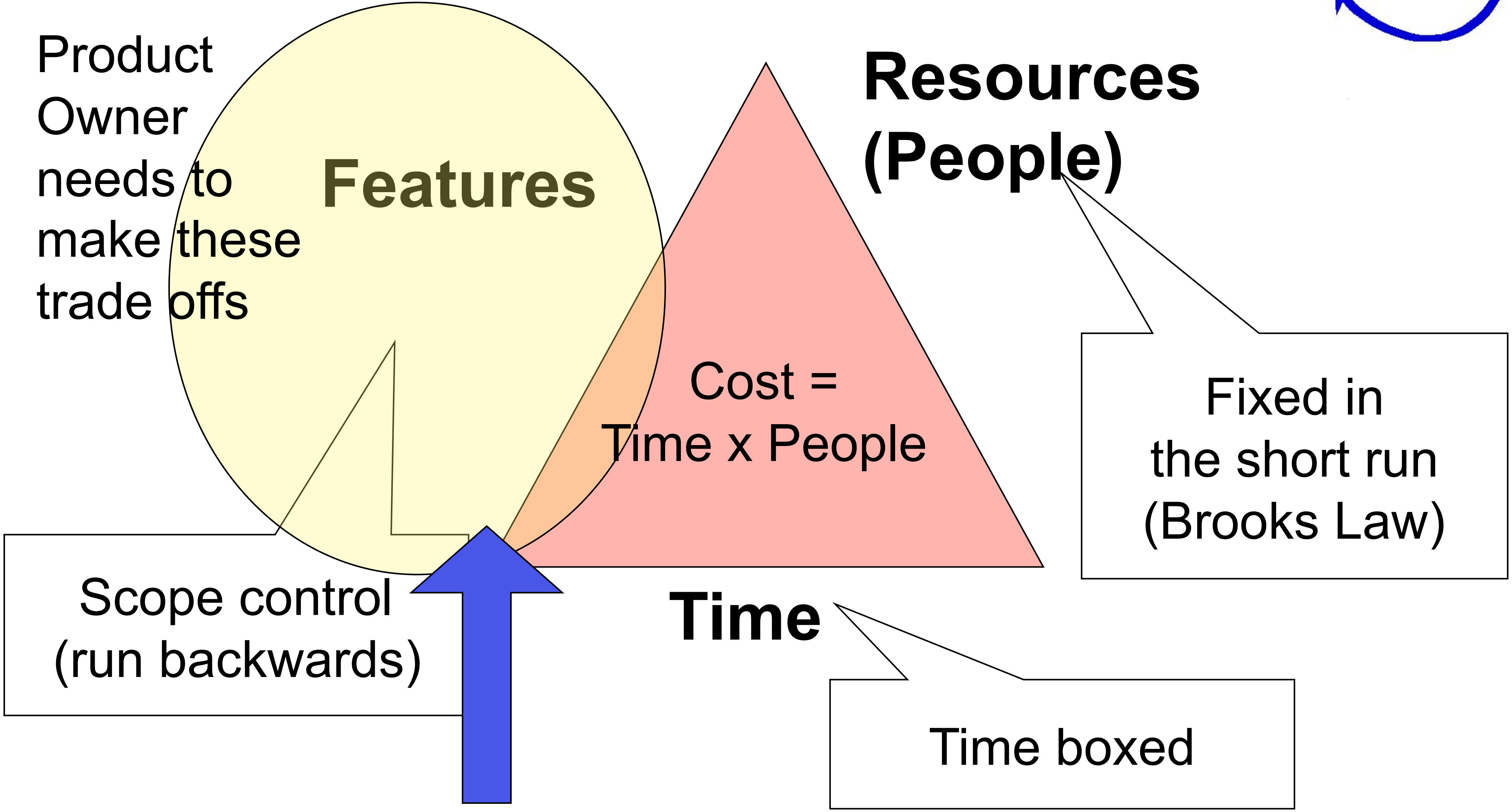
Only about 20% of features & functions in typical custom software are used

We often encounter requirements churn of 30% to 50%

Solution: **Just In Time Requirements** Identify, implement, deliver in quick succession

Mary & Tom Poppendieck
Implementing Lean Software Development 2007

Project constraints



Product Owner needs to make these trade offs

Features

Resources (People)

Cost = Time x People

Fixed in the short run (Brooks Law)

Time

Time boxed

Scope control (run backwards)

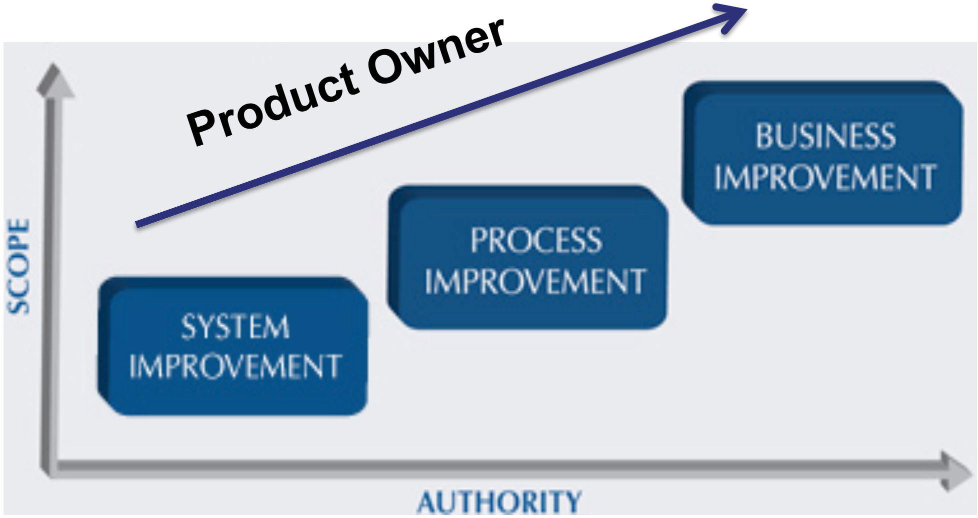
Agile projects negotiate over requirements rather than resources or time

Lesson 3: Negotiate over What

- Time is fixed
  - Fixed iterations
  - Fixed delivery dates
- Resources are fixed in short term
  - Can only increase slowly with time
- Therefore….
  - Negotiate over what will be delivered
- Therefore…
  - Requirements understand more important

Business Analysts, move on up

- BA Product Owner needs
  - Move up the BA Maturity Model
  - From “Requirements gather” / “Order taker”
  - To “Internal Consultant”
- Improving business effectiveness



Source and more details on BAMM:
http://www.assistkd.com/bamm/bamm.html

Lesson 4: Think BIG!

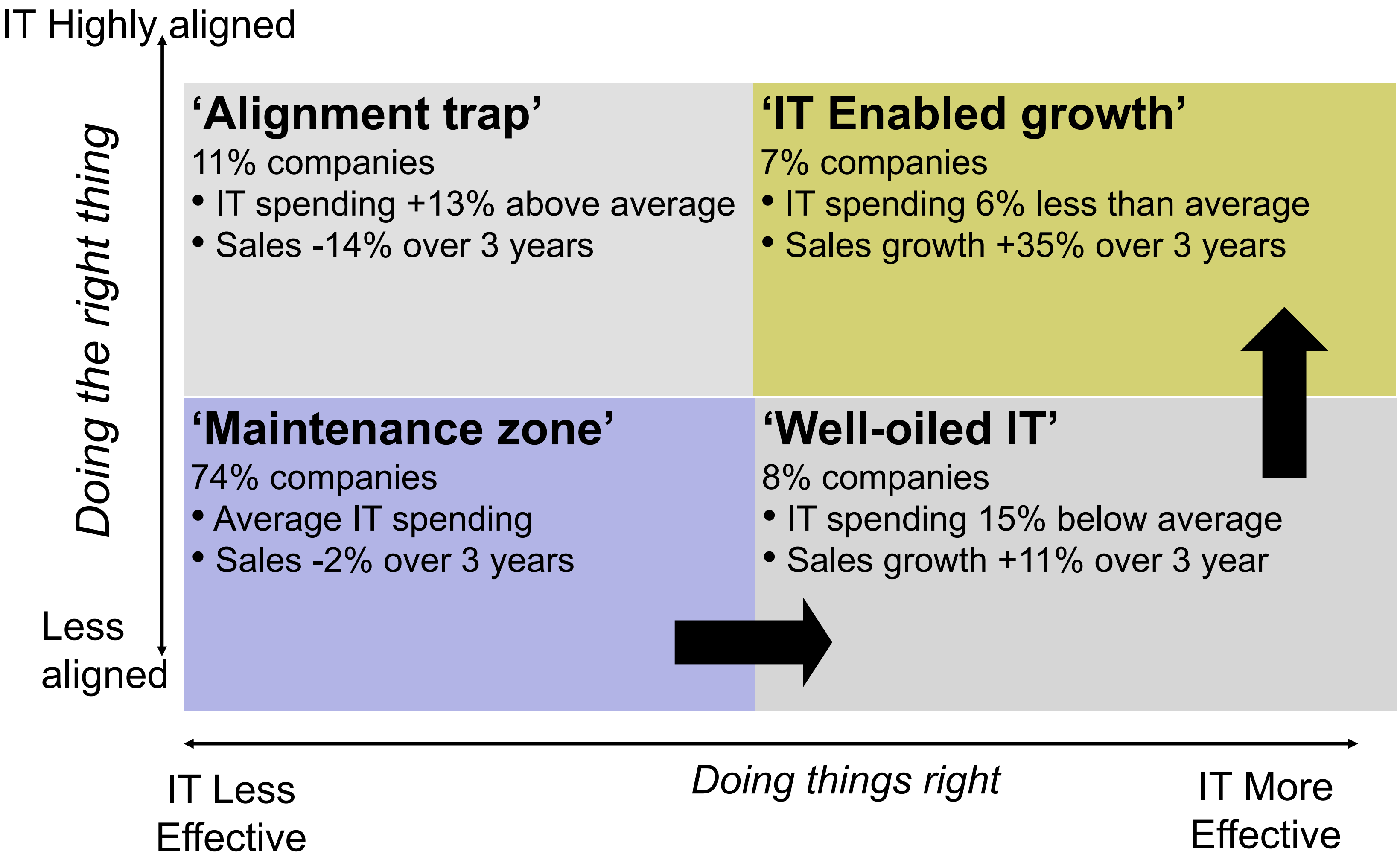
- Business Analysts think about
  - The business system
  - the business objective
  - the need
- Be more than an *order taker*

But....

- There is a time and a place for everything
- ....

Requirements come second when changing to Agile

The Alignment Trap



IT Highly aligned

Doing the right thing

Less aligned

‘Alignment trap’

11% companies

- IT spending +13% above average
- Sales -14% over 3 years

‘IT Enabled growth’

7% companies

- IT spending 6% less than average
- Sales growth +35% over 3 years

‘Maintenance zone’

74% companies

- Average IT spending
- Sales -2% over 3 years

‘Well-oiled IT’

8% companies

- IT spending 15% below average
- Sales growth +11% over 3 year

IT Less Effective

Doing things right

IT More Effective

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

Lesson 5

Build effective delivery machine

Then

Move outwards and upwards

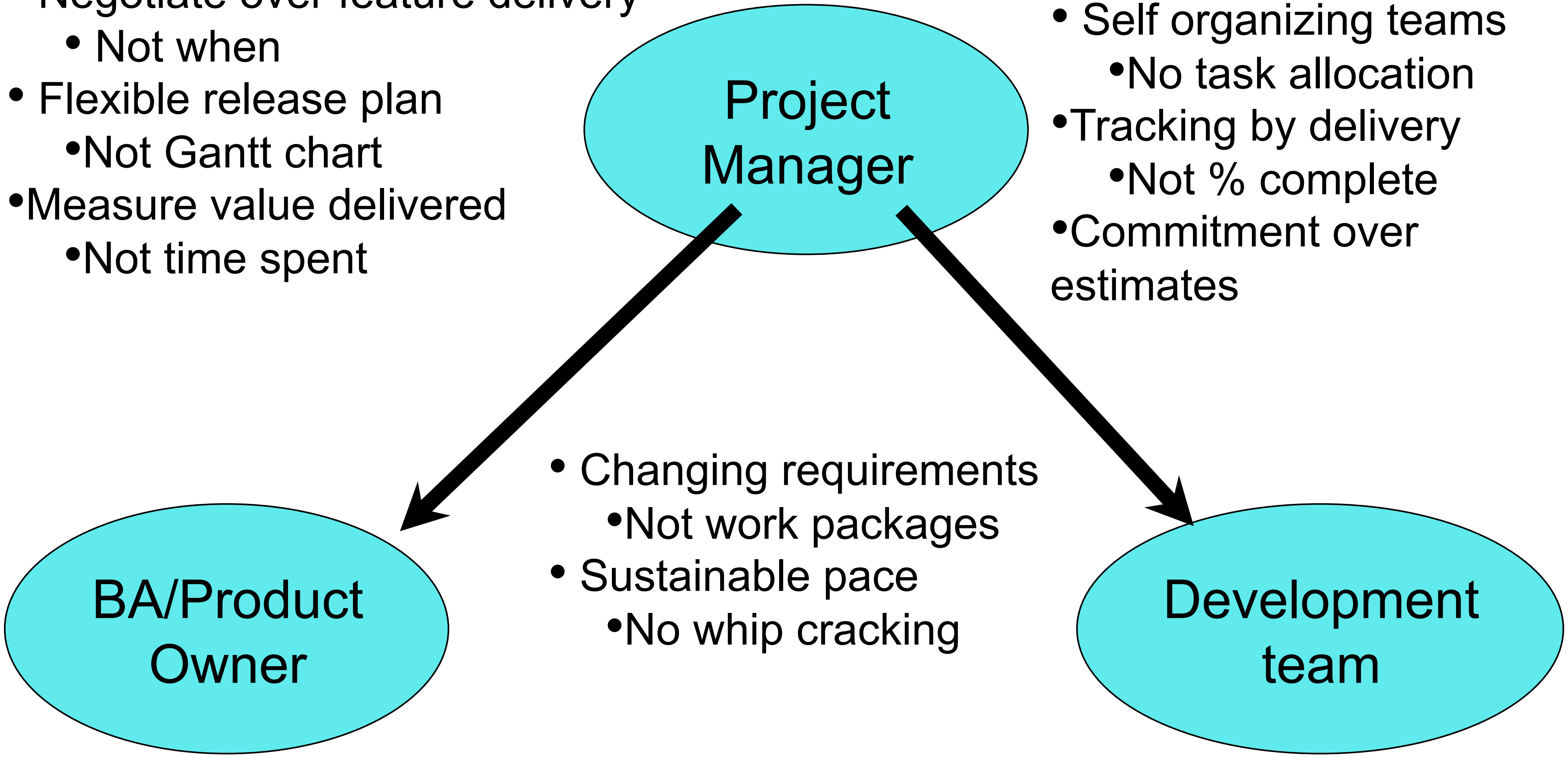
When adopting Agile

1. ***Do it right:*** Focus on the development team
2. Do not emphasis requirements or BA role
3. Get developers more effective

Then

4. ***Do the right thing***: Focus on the what
5. Long term benefits in BA role

More work for Product Owners
Less work for Project Managers



- Negotiate over feature delivery
  - Not when
- Flexible release plan
  - Not Gantt chart
- Measure value delivered
  - Not time spent

Project Manager

- Self organizing teams
  - No task allocation
- Tracking by delivery
  - Not % complete
- Commitment over estimates

- Changing requirements
  - Not work packages
- Sustainable pace
  - No whip cracking

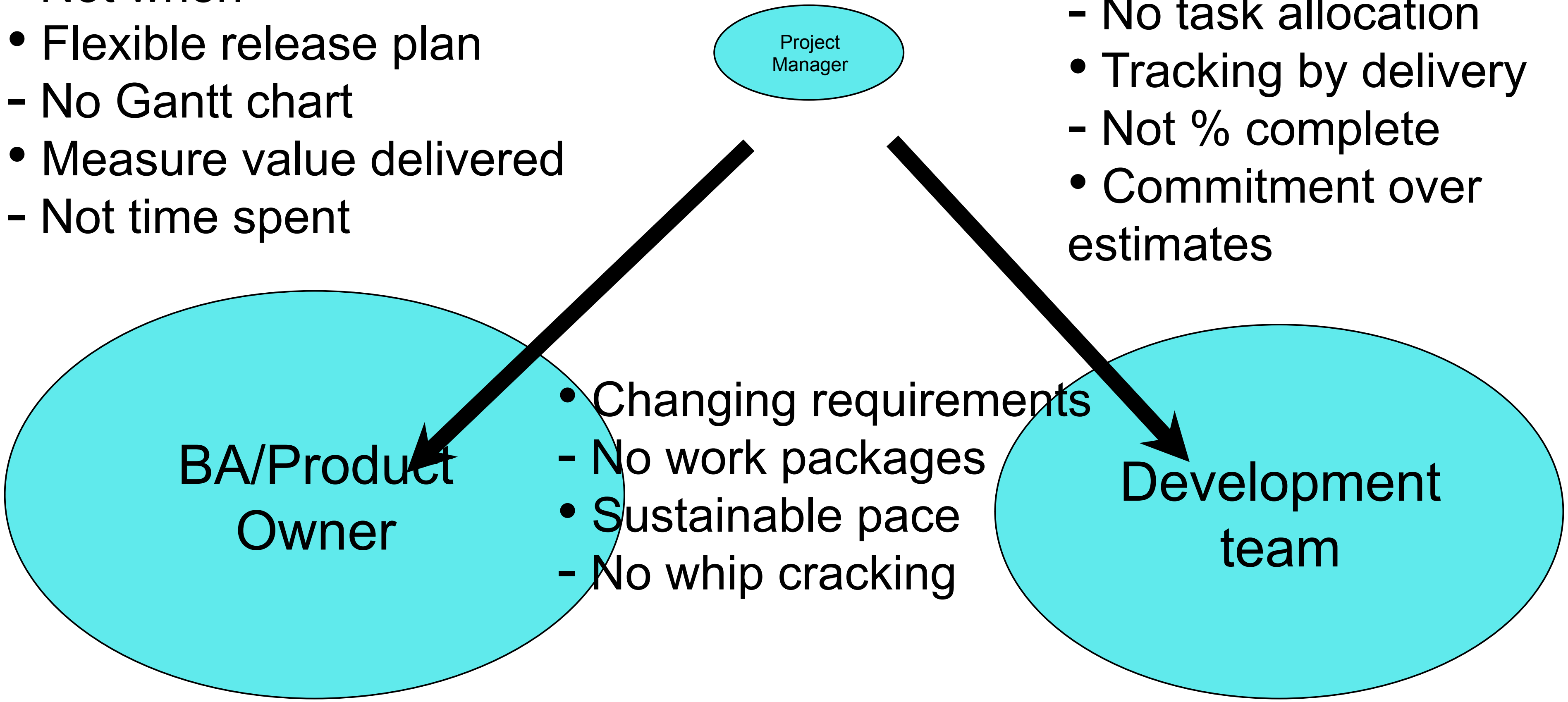
BA/Product Owner

Development team

More work for Product Owners Less work for Project Managers

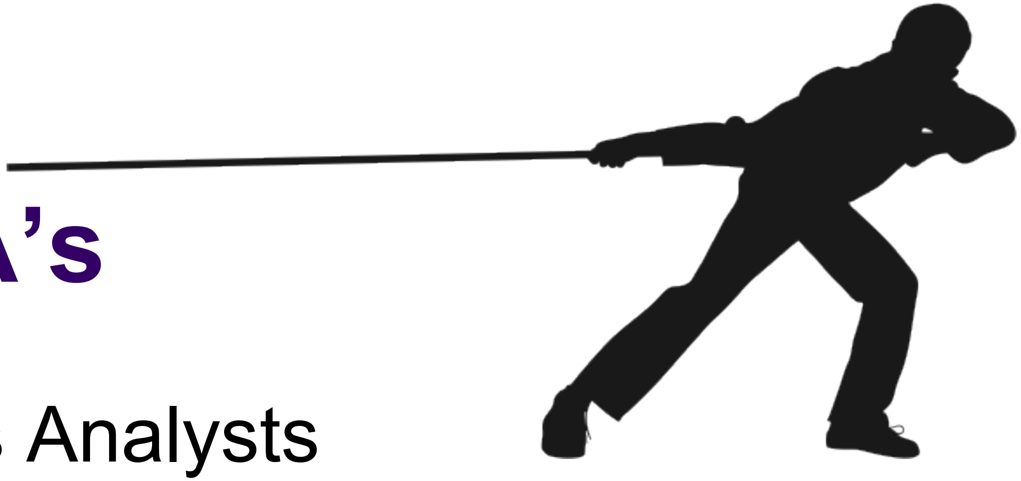
- Negotiate over feature delivery
  - Not when
- Flexible release plan
  - No Gantt chart
- Measure value delivered
  - Not time spent

- Self organizing teams
  - No task allocation
- Tracking by delivery
  - Not % complete
- Commitment over estimates



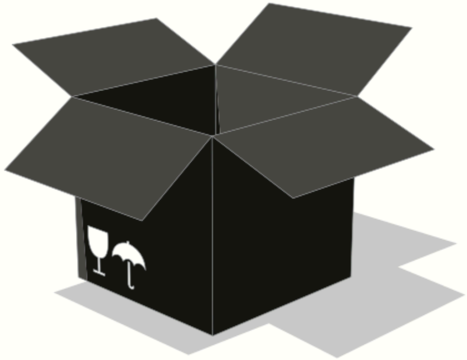
- Changing requirements
  - No work packages
- Sustainable pace
  - No whip cracking

More work for BA’s



- More work for Business Analysts
  - More/better analysis can reduce work load in time
  - More responsible for value delivered
  - More conversations with Developers
  - Writing/Creating acceptance tests
  - Slack for *Just in time requirements* (Queuing theory)
- Move from ***requirements push*** to ***needs pull***
- Therefore... 1 BA for every 3 to 7 developers
  - Stable product: 1 BA -> 7 developers
  - Rapid change: 1 BA -> 3 developers

Take aways



1. Being Agile means delivering business needs
2. Product Owner is often a BA
   - If not then BA supports Product Owner & Dev team
   - Agile process does not remove need for requirements
3. BA take a back seat in early transition
   - Step forward as team becomes effective
   - Key in reducing work to be done
4. Product Owner role is larger than BA role
   - Need greater staffing
   - Shift from *Requirements Push* to *Need Pull*

Thank you

allan@allankelly.net

http://www.allankelly.net

http://blog.allankelly.net

Twitter: allankellynet

Software Strategy Ltd

http://www.softwarestrategy.co.uk

Training & Consulting in Agile Software Development