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:

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

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

Jim Highsmith, 2002
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

- It's the business need, *stupid*
Traditional approach

Business Analysis / System Analysis

Royce, 1970, “Managing the Development of Large Software Systems”
Traditional approach:
- Slice through work
- Everything in iteration
- End-to-end
- Deliver functionality

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

- 6+ months

Traditional approach:

- Decide requirement
- Analysis / Design
- Code & Unit Test
- Merge & Release

Iteration 1 (2 weeks)

Iteration 2 (2 weeks)
● Slice through work
● Everything in iteration
● End-to-End
● Deliver business functionality

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

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

More philosophical: value, idea based

Applicability

Organizational Learning

Lean thinking

Agile

XP

Scrum

Kanban
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

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

Agile, Agile methods & Agile toolkit

The Agile Toolkit

Scrum, XP, DSDM, Kanban, ...
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

- **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

Business Analyst

Product Manager
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

- **Corporate IT Department**
  - Banks, Pharmaceutical, Oil, …
  - Product Owner is Business Analyst
  - Business Analysts supports non-technical Product 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?

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

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

We often encounter requirements churn of 30% to 50%

Mary & Tom Poppendieck
Implementing Lean Software Development 2007
Project constraints

- Features
  - Product Owner needs to make these trade-offs
- Resources (People)
  - Fixed in the short run (Brooks Law)
- Cost = Time x People
- 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](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

‘Alignment trap’
11% companies
- IT spending +13% above average
- Sales -14% over 3 years

‘Maintenance zone’
74% companies
- Average IT spending
- Sales -2% over 3 years

‘IT Enabled growth’
7% companies
- IT spending 6% less than average
- Sales growth +35% over 3 years

‘Well-oiled IT’
8% companies
- IT spending 15% below average
- Sales growth +11% over 3 years

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

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

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

Project Manager

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

BA/Product Owner

Development team

Project Manager
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