Lean IT and Agile 101

October 2013

John Okoro – Point B Agile Service Offering Lead – jokoro@pointb.com

Pat Edmonds – Agile Service Offering Team – pedmonds@pointb.com
Objective

• Introduction to Agile (including definition, concepts, components, and benefits)

• Understand how Agile and Lean principles align, and how they can be complementary in the organization

• Understand how Agile fits into a Lean IT approach

• Learn from case studies of how other organizations have used Agile to create business value

• Gain a strong foundation for how your organization can leverage Agile
What is Agile?

From Wikipedia:

- Agile software development is a group of **software development methods** based on **iterative and incremental development**, where requirements and solutions evolve through **collaboration** between **self-organizing, cross-functional teams**.

- It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.

**Examples of Agile methodologies**: Scrum, XP, AUP, Kanban, etc.
A Brief History

History of Agile

Waterfall Model
(Winston W. Royce)

Concept of
"Adaptive Software Development"
(Edmonds, E. A.)

Rapid App. Development
(James Martin)

Scrum
(Ken Schwaber, Jeff Sutherland)

Adaptive Software Development (ASD)
(Jim Highsmith, Sam Bayer)

FDD
(Jeff De Luca)

DSDM
(DSDM Consortium)

1970 1974
1980
1990
1991
1995
1996
2000
2003

Agile Manifesto
(Marry & Tom Poppendieck)

Crystal Clear
(Alistair Cockburn)

XP
(Kent Beck, Ward Cunningham and Ron Jeffries)

http://bogus212.egloos.com/294217
Agile & Lean

The New New Product Development Game
Takeuchi & Nonaka 1986

Japan’s Manufacturing Industry

Toyota’s Production System
Kanban
Taichi Ohno 1978 Japanese
1988 English

Lean Software Development
Mary & Tom Poppendieck 2003
2009 - Scrumban - Corey Ladas

Kanban
David J. Anderson 2010

Lean Startup
Eric Ries 2011

Scrum
Sutherland & Schwaber

Extreme Programming
Kent Beck 1999

2001 (Mike Beedle & Schwaber)

Design Patterns
Gang of Four, 1994

Agile

The Four Steps to the Epiphany
Steven G. Blank 2005

Startup

Principles of process optimization and Agile

Key Agile Concepts

- Individuals and interactions over processes and tools
- Customer collaboration over contract negotiation
- Respond to change over following a plan
- Work [business outcomes] over comprehensive documentation

Core Lean Principles

- Respect for People
- Value - Customer-centricity
- Flow
- Responsiveness & Flexibility
- Pull
- Pursuit of Perfection
Some Common Lean & Agile Practices

**Agile Practices**
- Daily Standups
- Backlog (WIP or Work in waiting)
- Retrospectives

**Some Lean Corollaries**
- Daily Huddles
- Work Breakdown Structure
- Lessons Learned
- Continuous Improvement Plan
Case Study – Entertainment Client

• Large Entertainment Client
• $1 billion program, included physical properties and Web Site re-launch
• Outcome: Successful re-launch for this key program
• Scrumban was used within shared services groups that had frequent requests from VPs and executive team – “fly ins”
• Scrumban = Scrum + Kanban
• 500+ Person Agile Engagement

Source - http://leansoftwareengineering.com
Case Study – Financial Services Client

• Captive Financial Services Client (Automotive Manufacturer)
• Use Lean principles outside of the IT space
• Generally uses waterfall in the IT space
• Working to align internal projects with vendors that use Agile
• This is an Agile hybrid case – common in industry
  • In some cases by design, in some cases de-facto
## Additional Case Studies

<table>
<thead>
<tr>
<th>Problem</th>
<th>Agile Practices Used</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy Development</td>
<td>• Weekly Scrum Standups&lt;br&gt;• Co-location&lt;br&gt;• Timeboxes&lt;br&gt;• Backlogs</td>
<td>Greater accountability and engagement achieved&lt;br&gt;More solutions developed to deliver high quality and rich recommendations</td>
</tr>
<tr>
<td>Content Development</td>
<td>• Scrum&lt;br&gt;• Ideal Time&lt;br&gt;• Kanban&lt;br&gt;• Retrospectives</td>
<td>Ability to prioritize work in conjunction with production support issues. Ability to incorporate a pull system to bring additional work items into a sprint</td>
</tr>
<tr>
<td>Home Renovation</td>
<td>• Backlog&lt;br&gt;• Epics&lt;br&gt;• Timeboxes&lt;br&gt;• Prioritization</td>
<td>Were able to breakdown the work into consumable chunks and lessen the inconvenience of rooms being out of commission</td>
</tr>
</tbody>
</table>
Agile Software Development Using Scrum

October, 2013

John Okoro – Point B Agile Service Offering Lead – jokoro@pointb.com
Agile & Lean

The New New Product Development Game
Takeuchi & Nonaka 1986

Japan’s Manufacturing Industry

Toyota’s Production System
Kanban
Taichi Ohno 1978 Japanese
1988 English

Lean Software Development
Mary & Tom Poppendieck 2003

2009 - Scrumban - Corey Ladas

Kanban
David J. Anderson 2010

Lean Startup
Eric Ries 2011

Scrum
Sutherland & Schwaber

Design Patterns
Gang of Four, 1994

Extreme Programming
Kent Beck, 1999

Agile

2001 (Mike Beedle & Schwaber)

Lean

Startup

Contents

• Agile Planning
• Why Scrum?
• Scrum – in 30 Slides or Less : )
• Wrap Up
Many companies only implement Agile up to release level planning.

To fully realize benefits, Agile should be used all the way up to strategic planning.
Why Scrum?

• First, some truths about software development:
  • Building software rarely means building repeatable widgets
  • It’s difficult to plan a project that’s more than a few weeks, since you don’t know when you’ll encounter something unexpected, or when business priorities will change

• For a **product manager**, Scrum permits:
  • Frequent changes in scope
  • Flexible changes to a project’s end-date

• For a **development team**, Scrum permits:
  • A guarantee that scope won’t change for a predictable period of time
  • A team to self-organize as opposed to being told what to do

• For an **organization**, Scrum permits:
  • Quality software that more reliably meets the needs of the organization
  • Software projects that can be delivered on time
A Typical Project

Design → Develop → Test → Release
Giving Credit Where It’s Due…

• The bulk of the following slides have been borrowed in full or in part from Mike Cohn (with Mike’s permission, of course…)

• Mike Cohn:
  http://www.mountaingoatsoftware.com/
What is Scrum

• Scrum is one of the many flavors of Agile
  • Scrum is the flavor of Agile used by many Point B clients
• Scrum was started by Jeff Sutherland in 1993
  • He borrowed the term Scrum from a 1986 study by Takeuchi & Nokata Published in the HBR
  • The HBR study compares high performing cross functional teams to the Scrum formation used in Rugby
• Scrum is the leading Agile development framework used by Agile development teams around the world
• There are several other flavors of Agile including XP, AUP, Kanban
The Agile Manifesto – a Statement of Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Over</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuals and Interactions</td>
<td>over Process and Tools</td>
<td>Working Software over Comprehensive Documentation</td>
</tr>
<tr>
<td>Customer Collaboration</td>
<td>over Contract Negotiation</td>
<td>Responding to Change over Following a Plan</td>
</tr>
</tbody>
</table>

Source: www.agilemanifesto.org
Sprints

- Scrum projects make progress in a series of “sprints”
  - Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint
Putting it All Together

Image available at www.mountaingoatsoftware.com/scrum
No Changes During a Sprint

• Plan sprint durations around how long you can commit to keeping change out of the sprint
Scrum Framework

Roles
- Product owner
- ScrumMaster
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
Scrum Framework

Roles
- Product owner
- ScrumMaster
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
Product Owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results
The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences
The Team

• Typically 5-9 people
• Cross-functional:
  • Programmers, testers, user experience designers, etc.
• Members should be full-time on the project
  • May be exceptions (e.g., database administrator)
• Teams are self-organizing
  • Ideally, no titles but rarely a possibility
• Membership should change only between sprints
Scrum Framework

Roles
- Product owner
- ScrumMaster
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
Everyone Answers 3 Questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the ScrumMaster
  - They are commitments in front of peers
Scrum Framework

Roles
- Product owner
- ScrumMaster
- Team

Ceremonies
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts
- Product backlog
- Sprint backlog
- Burndown charts
Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

This is the product backlog
## A Sprint Backlog

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write the foo class</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Add error logging</td>
<td></td>
<td>8</td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Tasks</td>
<td>Mon</td>
<td>Tues</td>
<td>Wed</td>
<td>Thur</td>
<td>Fri</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Hours Graph](image)
A Sprint Burndown Chart
Remember this?

Design → Develop → Test → Release
How about this instead?
Wrap Up

• Why Scrum?
  • **Focus** – Sprints allow teams to focus on an achievable goal
  • **Ownership** – Self-organization foments a feeling of ownership of the product
  • **Transparency** – everyone can see the requirements and the progress, which allows for course corrections to happen before it’s too late
  • **Continuous Improvement** – It’s ok to re-do what wasn’t done right the first time. Retrospectives cause teams to constantly look for areas of improvement.
  • **It’s All About the Team** – over time, team members develop respect for one another and are more willing to pitch-in to reach a common goal
Resources

• Where to learn more:
  • Scrum Alliance: http://www.scrumalliance.org/
  • Mike Cohn’s Site: http://www.mountaingoatsoftware.com/topics/scrum
  • Point B Agile Thought Leadership & Case Studies: http://www.pointb.com
  • Wikipedia (Agile, Scrum)
  • YouTube (lot’s of great short videos)
  • Books:
    • Agile Software Development with Scrum (Schwaber/Beedle)
    • Agile Estimating and Planning (Cohn)
    • Agile Project Management: Creating Innovative Products (2nd Edition - Highsmith)
    • Lean IT: Enabling and Sustaining Your Lean Transformation (Bell)
Appendix
Resources

• PointB.com Web Site – https://pointb.com
• Agile Service Offering Team: John Okoro, Alex Stockdale, Pat Edmonds (Lean), Scott Ikeda
The Business Environment

- Many organizations have embraced Agile for their software development practices
- A growing percentage of these organizations are starting to use it outside of the IT department
- Application of Agile practices in non-IT environments include:
  - Daily stand-ups
  - Backlog
  - Retrospectives
- With a slight tweak of the Agile Manifesto from software to **business outcomes**, the intent behind the manifesto becomes relevant to most implementation-type projects
Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

http://agilemanifesto.org/principles.html
What is Lean?

A philosophy, mindset and set of tools focused on delivering value to customers through the elimination of waste in the business process.
Scrum in 100 Words

• Scrum is an agile process that allows us to focus on delivering the **highest business value** in the **shortest time**.
• It allows us to rapidly and repeatedly inspect **actual working software** (every two weeks to one month).
• The **business sets the priorities**. Teams self-organize to determine the best way to **deliver the highest priority features**.
• Every **two weeks to a month** anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.
  • For creative teams this may mean delivering demonstrable experiences (working deliverables)
Sprint planning meeting

Sprint prioritization
- Analyze and evaluate product backlog
- Select sprint goal

Sprint planning
- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours (or points)
Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours or points)
  - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

- Code the middle tier (8 hours)
- Code the user interface (4 hours)
- Write test fixtures (4 hours)
- Code the foo class (6 hours)
- Update performance tests (4 hours)
The Daily Scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings
Everyone Answers 3 Questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are *not* status for the ScrumMaster
  - They are commitments in front of peers
The Sprint Review

• Team presents what it accomplished during the sprint
• Typically takes the form of a demo of new features or underlying architecture
• Informal
  • 2-hour prep time rule
  • No slides
• Whole team participates
• Invite the world
Sprint Retrospective

• Periodically take a look at what is and is not working
• Typically 15–30 minutes
• Done after every sprint
• Whole team participates
  • ScrumMaster
  • Product owner
  • Team
  • Possibly customers and others
Start / Stop / Continue

- Whole team gathers and discusses what they’d like to:

- Start doing
- Stop doing
- Continue doing

This is just one of many ways to do a sprint retrospective.
User Stories

- **What is it?** A short, simple description of a feature told from the perspective of the actor who desires the new capability.

- **Template:** As an `<actor>` I want to `<goal>` so that `<reason or intent>`.

- **Example:** As a Customer, I want to have my product gift-wrapped so that when I buy a gift for someone, it appears personalized and special.

- **Notes:**
  - Focus is always on providing business value
  - Usually considered the start of a conversation and are not intended to contain all requirements for a feature
  - Assume that IAs/designers/developers/testers are experienced enough to make good design decisions on the details
  - Epics are used for large features and should contain many corresponding user stories
### A Sample Product Backlog

<table>
<thead>
<tr>
<th>Backlog item</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow a guest to make a reservation</td>
<td>3</td>
</tr>
<tr>
<td>As a guest, I want to cancel a reservation.</td>
<td>5</td>
</tr>
<tr>
<td>As a guest, I want to change the dates of a reservation.</td>
<td>3</td>
</tr>
<tr>
<td>As a hotel employee, I can run RevPAR reports (revenue-per-available-room)</td>
<td>8</td>
</tr>
<tr>
<td>Improve exception handling</td>
<td>8</td>
</tr>
<tr>
<td>...</td>
<td>30</td>
</tr>
<tr>
<td>...</td>
<td>50</td>
</tr>
</tbody>
</table>
The Sprint Goal

- A short statement of what the work will be focused on during the sprint

**Database Application**
- Make the application run on SQL Server in addition to Oracle.

**Life Sciences**
- Support features necessary for population genetics studies.

**Financial services**
- Support more technical indicators than company ABC with real-time, streaming data.
Managing the Sprint Backlog

- Individuals sign up for work of their own choosing
  - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known
Estimating Work

• Estimating the time it takes to build software or design a UI is really difficult to nail

• … But it’s not so tough to get it pretty close to right

• Estimating to achieve perfection can waste a ton of time

• So instead of spending all your team’s time estimating, let them take their best guesses and see how things work out

• Patterns will emerge…

• Treat estimates as best guesses rather than contracts
Challenges Adopting Scrum

• Org structure, support and sponsorship
  • Scrum adoption requires executive sponsorship as well as complete buy-in from Creative/dev/test/product mgmt teams

• Physical environment
  • Scrum values communication, putting a team in the same room is best, but often can’t be achieved

• Geographical location
  • In today’s world of distributed teams, time zones and distance can make it difficult to communicate

• External customers
  • Implementing Scrum when the Product Owner is outside of your company can be difficult – distance, contracts and trust can get in the way.

• Scope
  • Letting go of the notion of fixed scope throughout the course of the project