Lean and Kanban Agile

Michael Cox
Vice President and Senior Consultant
NetObjectives, Inc.
michael.cox@netobjectives.com

“What is all this Agile stuff about, anyway?”
Lean and Kanban

How do they complement each other?
How do you use them?
Why does it work?
Lean

- Value from the customers perspective
- Identify and eliminate waste – non value added activities
- Flow of work at customer demand
- Continuous improvement

Kanban

A management discipline.
A constant exercise of matching demand with supply, to deliver the right thing at the right time.

See also: Visibility, Prioritization, WIP limits, Pull
Agile

Agile is a method that features rapid delivery of functional product iterations

Relies on immediate customer feedback

Allows for evolving understanding of system
Agile is about Business Iterations not Development Cycles
Agility

Predictability of Business Value Realization
“Where did this stuff come from?”
How Did We Get Into This Spot?

- Tremendous rise in the standard of living the past 100 years in all developed countries
- Rise was largely driven by productivity improvements
  - Agricultural up 3 to 5% a year since 1900
    - 50% of workforce in 1900, < 2% today, more production
  - Production up by 3% a year since Depression
    - 35% of workforce in 1940, < 15% today, 100x output rise

Basis has been the Invention and Widespread Adoption of Mass Production Techniques
How Did We Get Into This Spot?

- Managing via hierarchy, command and control
- Scientific management – the one best way
- Economies of scale
- Batch production

Mass Production Techniques & Mindset

Thought Basis for Current Management Practices

Lean Principles have generated Lean Practices
How Did We Get Into This Spot?

- Mass production management techniques in systems and software development have largely failed
  - Documentation = Understanding
  - The right tasks, correct pressure - force it to happen
  - “If they would freeze requirements, we would be fine”
  - “Heroes” called in when program is in real trouble

- A dissatisfied customer community has imposed more controls and rigidity

- Contractors countered with rigid contracts and change orders to batter the customer with cost and schedule

- Product owners were not involved until too late
we are always working with uncertainty
Lean and Kanban help us deal with uncertainty

The result is agility
Lean suggests limit 
TIME between steps

Kanban suggests 
limit # of items 
being worked on in 
each step

size of queue
Understanding Lean

1. Value from the Customer’s Perspective
2. Value stream
3. Flow
4. Pull
5. Perfection

- Define the value
- See the value stream
- Flow and where value comes from
- JIT
- Cycle time
- Reduce waste
“You cannot **build** the right thing if you have not **discovered** it first!”

The product owner must own the **definition of value**!
Usage of Features and Functions in Typical System

- **Always**: 7%
- **Often**: 13%
- **Sometimes**: 16%
- **Rarely**: 19%
- **Never Used**: 45%

Source: Standish Group Study of 2000 projects at 1000 companies
Discover incremental Business Value

Realize it

software product development

Discover how to build & implement it
visualize the entire value stream
The value stream

- Continuous flow of valuable work and features into deployment
- Includes *everybody* from the customer to operations and support engineers, and *not just development*

visualize the entire value stream
Focus on TIME

Optimize the Whole!

Large batches create delay and waste while Small batches create incremental value
### Three ways to do three projects

1. **Do them all at once, switching between them when delayed waiting for answers.**
2. **Do one at a time – may not be politically feasible.**
3. **Do them guided by Minimal Marketable Features.**

**Legend:**
- **Task-Switching and Schedules**
- **Lean Thinking**

_Ean Thinking_ by Michael Kennedy. Oaklea Press. 2003
DELAY IS finding redoing reworking waiting
hand-offs bottlenecks information delay untested code unread requirements transaction related coordination related
Cycle Time is Key!
Requirements ...

Decay and Lose Value over time
Requirements
are not fully understood even
after a formal sign-off
Requirements
change often
during long development cycles
Requirements
piled on
poorly prioritized
long delivery cycles
The work enters a queue.

When someone needs new work, they pull from the queue.

The work goes through a number of stages. When the work is done in a stage, it flows down to the next stage.

Until it is done.
Principles of Lean Software Development

- Optimize the Whole
- Eliminate Waste
- Build Quality In
- Deliver Fast

- Defer Commitment
- Create Knowledge
- Empower People

kanban improves quality and lowers cost by eliminating delays by managing WIP
Workflows can be seen and managed.

You can divide the work into small value adding increments.

It is possible to develop value-adding increment in a continuous flow, from requirement to deployment.
Kanban for Systems and Software

Limit Work in Process (WIP)
Pull value through
Make it visible
Increase throughput
Prioritized Backlog
Quality is built in
Team continuously monitor and improve

design the kanban board
Kanban boards reflect
Priority
WIP
Process
## Business Value Kanban

### Business Discovery

<table>
<thead>
<tr>
<th>Input</th>
<th>Prioritize</th>
<th>Sequence / incremental</th>
<th>Technical Analysis</th>
<th>Staging</th>
<th>Readiness</th>
<th>Specify</th>
<th>Execute</th>
<th>Deploy &amp; Ready to Use</th>
<th>Implement</th>
</tr>
</thead>
</table>

### Business Delivery

<table>
<thead>
<tr>
<th>Entry</th>
<th>Exit</th>
</tr>
</thead>
</table>

Don’t build features that nobody needs
(right now or in some cases, ever)

Don’t write more specs than you can code

Don’t develop more code than you can test

Don’t test more code than you can deploy
Kanban Success

Focus on Quality
Reduce WIP
Balance demand against throughput
Prioritize
Stop Starting and Start Finishing
The work enters a queue.

When someone needs new work, they pull from the queue.

The work goes through a number of stages. When the work is done in a stage, it flows down to the next stage.

Until it is done.
WIP Limit...

Governs the *maximum* number of work items that can be in that state at any instant.

**Below its limit:**

Receive a work item from upstream

**At its limit:**

Wait for one of its own items to be completed and flowed downstream

*In Knowledge Work, complexity grows exponentially with WIP*
Classes of Service

- Expedite
- Specific Delivery Date
- Standard
  - Maintenance or Break-Fix Work
- Standard
  - New or Value-Added Work
- Development Story
- Outside Impact
- Impediment
- Red Flag Issue

service level agreements
Direct the team in the **priority** of processing work items

1. Class of Service
2. SLA
3. Blocked Items
4. FIFO
## Monitoring flow: Kanban for portfolio

<table>
<thead>
<tr>
<th>Status</th>
<th>Backlog</th>
<th>Specify (right size)</th>
<th>Execute</th>
<th>Validate</th>
<th>Done/Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td><img src="image1.png" alt="Images" /></td>
<td><img src="image2.png" alt="Images" /></td>
<td><img src="image3.png" alt="Images" /></td>
<td><img src="image4.png" alt="Images" /></td>
<td><img src="image5.png" alt="Images" /></td>
</tr>
<tr>
<td>Project X</td>
<td><img src="image6.png" alt="Images" /></td>
<td><img src="image7.png" alt="Images" /></td>
<td><img src="image8.png" alt="Images" /></td>
<td><img src="image9.png" alt="Images" /></td>
<td><img src="image10.png" alt="Images" /></td>
</tr>
<tr>
<td>Project Y</td>
<td><img src="image11.png" alt="Images" /></td>
<td><img src="image12.png" alt="Images" /></td>
<td><img src="image13.png" alt="Images" /></td>
<td><img src="image14.png" alt="Images" /></td>
<td><img src="image15.png" alt="Images" /></td>
</tr>
<tr>
<td>Project Z</td>
<td><img src="image16.png" alt="Images" /></td>
<td><img src="image17.png" alt="Images" /></td>
<td><img src="image18.png" alt="Images" /></td>
<td><img src="image19.png" alt="Images" /></td>
<td><img src="image20.png" alt="Images" /></td>
</tr>
<tr>
<td>WIP Limit</td>
<td>14</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Who Talks?
• Only Team members moving stickies across the board!

Do This
• Start from the **right**
• Work by the **highest priority**
• Pay attention to:
  o Oldest
  o Blocked
  o Class of Service
  o SLA in jeopardy
• Ask
  o Do we have a bottleneck (congestion or gaps in the queues)?
  o Do we have a “blocker” not dealt with?
  o Are we keeping to our WIP limits?
  o Are priorities clear?

When done
• Update the board
• Remove done items from the board
Getting started with kanban

- Agree to goals
- Map the value stream
- Define a set of work item types
- Meet with external stakeholders
- Create board for tracking
- Agree to standup
- Agree to operational review
- Educate the team
- Start doing it

David Anderson. XTC, London 2009, October
Kanban

What you will see:
• Queues start backing up immediately following any blockage
• Predictable consequences
• The entire board will slow down as a result of flow issues
• Teams see issues right away and act together to fix them
Lean and Kanban

Lean is the theory
Kanban is the approach
Agile is the result

Questions