## AgileTour12 - Dublin

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Speaker(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:00-2:15</td>
<td>Welcome</td>
<td></td>
</tr>
<tr>
<td>2:15-3:00</td>
<td><strong>Timebox or Flow? Choices to consider in your Agile/Lean adoption</strong> – Colm O’hEocha, AgileInnovation</td>
<td><strong>From the coalface</strong> – John Vaughan, SoftCo</td>
</tr>
<tr>
<td>3:00-3:45</td>
<td><strong>Agile Requirements Management with User Stories</strong> – Fran O’Hara, InspireQS</td>
<td><strong>ScrumBan and beyond!</strong> - Gail Park and Glen Lockhart, Openet</td>
</tr>
<tr>
<td>3:45-4:15</td>
<td>Coffee &amp; Networking</td>
<td></td>
</tr>
<tr>
<td>4:15-5:00</td>
<td><strong>The Three Revolutions</strong> - Claudio Perrone aka AgileSensei</td>
<td><strong>Experiences in Implementing Agile Development</strong> - John Abbott, Karl Heery, Mairead Mulligan, Aisling Ni Cheallaigh, Aegon Insurance</td>
</tr>
<tr>
<td>5:00-5:30</td>
<td>Panel Session</td>
<td></td>
</tr>
</tbody>
</table>

**Sponsored by AgileInnovation and InspireQS**
AgileTour12

Timebox or Flow?
Choices to consider in your Agile/Lean adoption

Colm O’hEocha – AgileInnovation

Slow Down to Speed Up
This material is Copyright Protected under the following Creative Commons License:

Attribution-ShareAlike 3.0

Please use this material freely and derive further work from it. BUT, attribution must be given to the original author, and these same rights must govern all derived works.

Please use the following attribution:
“Colm O’hEocha, AgileInnovation Ltd. 2010: www.agileinnovation.eu”
**Agile/Lean Delivery Models**

- **Timeboxed**
  - Scrum, XP, FDD,
  - Work Planned to fit the Timebox
  - Duration Fixed, Scope Variable
  - Commitment -> Estimation & Planning
  - Macro Level – Pull; Micro Level - Push

- **Flow Based**
  - Kanban,...?
  - Work Planned based on Capacity
  - Scope Fixed, Duration Variable
  - Commitment -> Waste Elimination
  - Macro & Micro Level Pull
Timebox – A Container for Knowledge Work

• Container Constrains the Team, the Work and the Practices
• Container protects the Team from Churn & Interference
• The Container is Transparent

• **ScrumMaster** – Maintain the Container – prevent Leaks
• **Product Owner** – Decide what Work gets into the Container
• **Developers** – Execute the work in the Container
Flow – A Way to Maximise Value Delivery & Minimise Waste

• Pipeline/Value Stream Constrains the ‘Work in Process’ (WIP)
• WIP Limits & CoS protect the Team from Churn & Interference
• The Pipeline is Transparent

• **Explicit Policies** – Maintain the Pipeline – address bottlenecks
• **Product Owner** – Decides the next most important piece of work
• **Makers** – Focus on Finishing over Starting
## Scrum Board

<table>
<thead>
<tr>
<th>Sprint Backlog</th>
<th>To Do</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td>t7, t8</td>
<td>t5, t6</td>
<td>t4</td>
</tr>
<tr>
<td>S3</td>
<td>t10, t11</td>
<td>t9</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>t12, t13, t14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here is an example Kanban board. Notice the stages of the process, represented as columns. Notice the WIP limits in parentheses beside the column names.

Let's say testing of ticket 1 is complete. We move it downstream to indicate that it is ready to deploy.

Now we have 2 items in Test, with a WIP limit of 3, so we may pull a ticket from Development Done into Test, leaving 3 in Development, which has a WIP limit of 4, so we may pull a ticket into Development from Analysis Done, and so forth.

Because our WIP limit in Development covers both In Progress and Done, when we complete development on a ticket, we may move it to the Development Done column to indicate that it is completed, without breaking a WIP limit.

WIP limits stop work being pushed downstream when there is insufficient capacity to receive the work. If development work on ticket 5 is completed, it cannot be "pushed" into Test, because Test is already at its WIP limit.

A ticket completes Testing, a slot comes free, and ticket 5 may now be pulled into Test.

The Input queue may be replenished to its WIP limit with new work from the customer.

Kanban Mechanics in a Nutshell

As work is completed it moves to Done. But because Test is proving a bottleneck, preceding operations quickly 'back up'. Eventually 'Development' cannot take on any further work - it has reached its WIP limit. This is like a signal (kanban) from Test to Development to stop producing more work than can be tested.

Similarly, Analysis quickly reaches its WIP limit and is disallowed from taking on new work.
## Some Things to Consider...

<table>
<thead>
<tr>
<th>TimeBoxed</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Somewhat-Predictable Demand</td>
<td>Interrupt Driven Demand</td>
</tr>
<tr>
<td>• Over Short Planning Horizon</td>
<td>• Day to Day, Hour to Hour</td>
</tr>
<tr>
<td>• Sub-Iteration Features</td>
<td>• Highly Variable/Unpredictable Task</td>
</tr>
<tr>
<td></td>
<td>Durations</td>
</tr>
<tr>
<td>Focus Time</td>
<td>Less Focus Time</td>
</tr>
<tr>
<td>• Reduce Churn &amp; Distraction</td>
<td>• Churn/Distraction unavoidable</td>
</tr>
<tr>
<td>Shared, Collaborative Work</td>
<td>High Specialisation</td>
</tr>
<tr>
<td>• X-Functional Teams (7±2)</td>
<td>• Individuals (∞)</td>
</tr>
<tr>
<td>• Generalists</td>
<td>• Hand-Offs &amp; Dependancies</td>
</tr>
<tr>
<td>• Parallel Work</td>
<td>• Parallel &amp; Sequential Work</td>
</tr>
<tr>
<td>Scope Can Vary</td>
<td>Scope Often Fixed</td>
</tr>
<tr>
<td>Predictability Paramount</td>
<td>Responsiveness Paramount</td>
</tr>
</tbody>
</table>

**E.g. Software Development**

**E.g IT Operations**
Timeboxed (Scrum) and Flow (Kanban)

**Scrum**
- Timeboxing (Sprints & Delivery)
- Cadence
- Prioritized backlogs
- Daily standup meetings
- Demo after each iteration
- Inspect & Adapt the Process

**Kanban**
- Make the Work Visible
- Limit ‘Work In Process’ (WIP)
- Manage Flow (blockages, bottlenecks)
- Stabilise (Explicit Policies)
- Nurture Feedback Loops
- Drive Collaboration to Improve

**Scrum & Kanban**

**Scrumban**

**Kanbum**

Copyright © 2012 AgileInnovation
On Scrum & Kanban

• Scrum focuses on being agile which may (and should) lead to improving.

• Kanban focuses on improving, which may lead to being agile.

- Karl Scotland
# Scrum & Kanban Summary - Differences

<table>
<thead>
<tr>
<th><strong>Iteration Based (Scrum)</strong></th>
<th><strong>Flow Based (Kanban)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on maximising work done in an iteration</td>
<td>Focus on minimising the time to get a piece of work from start to finish</td>
</tr>
<tr>
<td>Velocity the default metric</td>
<td>Lead Time the default metric</td>
</tr>
<tr>
<td>Burndown chart prescribed</td>
<td>No particular type of diagram is prescribed (CFD common)</td>
</tr>
<tr>
<td>WIP limited indirectly (per sprint)</td>
<td>WIP limited directly (per workflow state)</td>
</tr>
<tr>
<td>Sprints Immutable</td>
<td>Capacity Driven Input Cadence</td>
</tr>
<tr>
<td>Scope extends to work of the Team</td>
<td>Scope extends across a Value Stream</td>
</tr>
<tr>
<td>Board Reset between Sprints</td>
<td>Board state Persistent</td>
</tr>
<tr>
<td>Design for any type of Work</td>
<td>Design for the Work Expected: WIT, CoS, SLA</td>
</tr>
<tr>
<td>Periodic Retrospectives</td>
<td>Stop the Line – Fix as it Happens</td>
</tr>
</tbody>
</table>
Some Questions to ask yourself...

<table>
<thead>
<tr>
<th></th>
<th>Scrum</th>
<th>Kanban</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Demand Predictable?</td>
<td>Yes - Short Planning Horizon</td>
<td>No – Interrupt/Event Driven</td>
</tr>
<tr>
<td>Is it a Team Based Work</td>
<td>Yes – Small Cross-Functional Teams</td>
<td>No – Specialists, Hand-Offs, X-Departments</td>
</tr>
<tr>
<td>Environment?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>What is the Organisations</td>
<td>Collaboration/Cultivation</td>
<td>Control/Competence</td>
</tr>
<tr>
<td>Culture?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>What is the Organisational</td>
<td>Low – We need ceremonies, roles &amp;</td>
<td>High – We Continuously Refine our own processes</td>
</tr>
<tr>
<td>Agile/Lean Maturity?</td>
<td>cadences</td>
<td></td>
</tr>
<tr>
<td>What is the appetite for</td>
<td>High – we want/need to shake things up</td>
<td>Low – we want low-risk, gradual change</td>
</tr>
<tr>
<td>change?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed price/scope vs. The</td>
<td>Estimation &amp; Commitment Mandatory</td>
<td>Earliest delivery of Value is what’s important</td>
</tr>
<tr>
<td>Most Important Stuff ASAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the Work always be</td>
<td>Yes – and Sprints encourage this</td>
<td>Not always – we can accommodate big features</td>
</tr>
<tr>
<td>Right-Sized (sub-sprint)?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What it feels like...

• Scrum
  – Teams may struggle to break up work, with transparency & joint responsibility
  – Management may struggle to allow self-organisation and maintain sprint discipline

• Kanban
  – Teams may struggle with a lack of prescribed structure
  – Management may struggle with a lack of commitment and milestones

Copyright © 2012 AgileInnovation
Scrum
Coordinate Work – Sprint
Drive Change – Commitment

Kanban
Coordinate Work – Pull
Drive Change – WIP Limits

Scrum
Kanban
(e.g. WIP Limits, Explicit Policies)

Kanban
Scrum
(e.g. Retros, Planning Poker, User Stories)

Kanban (Product/Project Management)
Scrum
Scrum
Scrum

Kanban (Program/Portfolio Management)
Kanban
Scrum
Kanban
Kanban
Kanban
Kanban
Copyright ©2012 AgileInnovation Limited
Scrum: “Nothing gets into a Sprint that can’t get out”

Kanban: “Stop Starting and Start Finishing”

Q&A
Colm O’hEocha,
AgileInnovation Ltd.
Colm@agileinnovation.eu

Colm is an IT industry veteran, with over 20 years design and development experience. His interest in Lean Thinking began when automating Just-In-Time production lines in the 1980s. Implementing early agile software methods in 2001, Colm has extensive hands-on experiences across teams, organisations and technologies. In varied roles of developer, architect and Director of R&D, he has seen Lean and Agile from a variety of perspectives. More recently Colm has partnered with LERO, the Irish software engineering research institute, investigating how agile and lean methods influence innovation in software development.

AgileInnovation offers agile training advisory and coaching services. Pre and post adoption assessments, coaching services to help teams get the most from agile, developing a agile business case, and targeted workshops on specific agile practices. Customers include Intel, HMH, Aegon, Ericsson, Omnipay, SoftCo, Openet and Accenture.
Some Customers...
Agile Assessments, Training and Coaching

Training Includes:
- Executive Briefing on Agile and Lean (½ day)
- Introduction to Agile Methods (½ day)
- Executive Briefing on Lean/Agile (½ day)
- Succeeding with Agile (2 days)
- ScrumMaster Training (2 days)
- Product Owner Training (1 day)
- Getting Lean through Kanban (2 days)

Consulting Includes:
- Project and Organisational Assessments
- Developing an Agile Adoption Strategy
- Getting the best from Agile and Waterfall
- Agile in Complex Contexts (firmware, B2B, batch, etc)

Coaching Includes:
- Structuring and facilitating
  - Daily StandUps
  - Sprint Planning
  - Sprint Reviews
- Developing User Stories
- Estimation
- Release Planning
AgileInnovation: Embracing the Inevitability of Change