VISUAL REQUIREMENTS MANAGEMENT WITH KANBAN

Mahesh Singh
Co-founder/ Sr. VP – Product, Digite, Inc.
Agenda

- Quick Introduction/ Context
- How We Were.. (“Traditional” Requirements Management, Release Scoping/ Planning)
- How We Are Today (Kanban to the rescue)
- Key Process Changes
- Benefits Achieved
Quick Introduction

- **My background**
  - Project Management → Yourdon ADT → Business/Systems Analyst → Project Manager → Business Manager → Entrepreneur
  - Nearly 25 years in the industry, 20 of them in the Silicon Valley
  - Last 12 years in two startups

- **My company – Digite, Inc.**
  - “Lean/Agile ALM” Tools company
  - 95+ customers/ >300,000 users
  - ALM – Digite Enterprise (Soon to be SwiftALM) - 2003
  - Kanban/Scrumban – SwiftKanban - 2011
  - Integration bus – SwiftSync - 2012
## How we used to work...

<table>
<thead>
<tr>
<th>Month -1</th>
<th>Month 1</th>
<th>Month 2</th>
<th>Month 3</th>
<th>Month 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Scoping</td>
<td>Development (features + critical/high priority defects)</td>
<td>Defect Fixing</td>
<td>Test Cycle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IR2</td>
<td>IR3</td>
</tr>
<tr>
<td>Test Case Development</td>
<td>Dev Testing</td>
<td>Scope Testing</td>
<td>Regression Testing</td>
<td>Defect Convergence</td>
</tr>
</tbody>
</table>

© Digite, Inc.
# How we work today...

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 2</th>
<th>Week 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spec Story 7-8</td>
<td>Spec Story 9-10</td>
<td>Spec Story 11-12</td>
<td>Spec Story 13-14</td>
</tr>
<tr>
<td><strong>Design Story 5-6</strong></td>
<td>Design Story 7-8</td>
<td><strong>Design Story 9-10</strong></td>
<td><strong>Design Story 11-12</strong></td>
</tr>
<tr>
<td>Develop Story 3-4</td>
<td>Develop Story 5-6</td>
<td>Develop Story 7-8</td>
<td>Develop Story 9-10</td>
</tr>
<tr>
<td>Validate Story 1-2</td>
<td>Validate Story 3-4</td>
<td>Validate Story 5-6</td>
<td>Validate Story 7-8</td>
</tr>
<tr>
<td><strong>Release 1-2-3-4</strong></td>
<td></td>
<td></td>
<td><strong>Release 5-6-7-8</strong></td>
</tr>
</tbody>
</table>
How we were...

Managing with traditional Requirements Management
Requirements Management: An Old Problem

- Requirements Management is a big chunk of ALM problems for all the usual reasons
  - 60% – 80% of project failures can be attributed directly to poor requirements gathering, analysis, and management (Meta Group)
  - 25% – 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon)
  - 68% of IT Projects Fail: primary cause identified as poor requirements according to a study by IAG
Requirements Management?

- Documenting
- Analyzing
- Tracing
- Prioritizing
- Agreeing on requirements
- Controlling change
- Communicating to relevant stakeholders

Our Requirements Problem:  
Upstream Planning, Visibility and Prioritization

- “What do we build in Release X.Y?”
  - Market Demand? - Customer / Competition/ Technology

- Extensive Documentation….
  - MRD/ PRD/ Product Roadmaps
  - Multiple versions (Traceability issues)
  - Detailed line items in our own Requirements Management tool

- … But Poor Visibility!!
  - “Requirements were buried 6 feet under!”
  - No one, especially management, wanted to read long documents or log into a system to review requirements

© Digite, Inc.
Our Requirements Problem: Upstream Planning, Visibility and Prioritization

- Extensive Back and Forth on Freezing Release Scope
  - Word/Excel documents over Email
  - Keeping track of various documents/emails/discussions
  - Long exhausting meetings

- Delayed Start to many Releases
  - Work started only after last release done – everyone, including Product Management, busy with last release
  - Release Scope freezing took too long
Challenge for PM (and consequently Dev!)

- How to be ready for the next release/sprint. What to build (where do we start?)
- How to prioritize (keeping in mind all stakeholders)
- How to make sure we were not writing specs for features that wouldn’t get built
- How to make sure we are not building the wrong product
- How to keep all stakeholders abreast of what is coming
A Very Costly Process! A LOT OF EFFORT before Use Cases or Epics/ User Stories made it to the Backlog!

When we spoke to our customers and other Bay Area companies, we realized we were not alone!

Getting from Vision/ Roadmap to User Stories in a Backlog is a lot of work – and is usually not very organized!
How we are today...

Kanban to the Rescue!
Our Kanban Journey

- **SwiftKanban – Brief History**
  - Decision to build SwiftKanban – April 2010
  - Beta – Nov 2010
  - GA – July 2011
- **Advice/input from several thought leaders** – David Anderson, Jim Benson, Al Shalloway, Masa K Maeda, Yuval Yeret..
- **Kanban training** – Q4, 2010
- **We started using it as soon as we launched Beta**
Benefits we sought from Kanban

- Small(er) batch size
- Continuous (more frequent) delivery
- Visibility to all stakeholders (including external customers)
- Overall, greater throughput and quality

We were EXCITED! We were raring to go!
### First version of our Kanban – As Is (2010-11)

#### Swift-Kanban Development

<table>
<thead>
<tr>
<th></th>
<th>Ready (0/5)</th>
<th>Develop (0/5)</th>
<th>Validate (0/5)</th>
<th>Approve (0/5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In-Progress (0/3)</td>
<td>Done (0/2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Defects

#### Issue

© Digite, Inc.
Version 2 of our Kanban (2011-2012)  
Greater Attention to Planning/Scoping
Version 2 of our Kanban (2011-2012)
Execution separated from Planning/Spec-intoing
Version 2 of our Kanban (2011-2012)
Execution separated from Planning/Spec-ing
Version 3 of our Kanban (2012-Present)
Roadmapping in a separate board

<table>
<thead>
<tr>
<th>General Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics/Analytics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Idea Queue</th>
<th>Table Stakes</th>
<th>Catchup</th>
<th>Differentiators/Cool</th>
<th>Hygiene</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backlog</td>
<td>Backlog</td>
<td>Backlog</td>
<td>Backlog</td>
</tr>
<tr>
<td>Idea 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Digite, Inc.
Version 3 of our Kanban (2012-Present)
Execution separated from Planning/ Spec-ing
“The shorter the Project, the more planning it needs.” – old (Project Management) Jungle saying...!
Key (Product) Management Process Changes

- Separate “Weekly Standup Meeting” (tho’ we continue to participate in Engg Daily Standup, mostly as observers) on the Planning Board
- “Monthly Prioritization Meeting” with leadership team on Roadmap Board
- Access to leadership to all boards. CEO/ Sales/ Support look at the board to see when they can expect a feature to be released
How Work Really Gets Done in Dev/ QA

**User Story Planning**
- Backlog consists of New features and Enhancements
- Backlog Prioritization by Rank
- Estimation pulled in PM, Dev, UI team
- Splitting Large User Stories into smaller Deployable Stories
- Elaborating the stories (Specification writing)
- Available in ‘Ready For Development’

**Other Cards**
- Issues, Internal And Customer Defects are put directly into ‘Ready For Development’ lane with highest priority
- Engineering Tasks are treated like user Stories
How Work Really Gets Done in Dev/ QA

**Work Execution**

- Each Team Member Pulls next available card from ‘Ready For Development’ lane.
- Each Card flows through Development Value stream:
  - Validated Cards are put into ‘Ready for Deployment’ lane.
- Release when 20 Cards in ‘Ready For Deployment’
  - Generally, release cadence: 2 wks

**Work Tracking**

- Daily standup Calls
- Generally, 1 card at a Time for all team members.
- Monitoring and Controlling WIP violations
- Focus on Impediments (Block)
Going further – “Tribal Mashup…”?

<table>
<thead>
<tr>
<th>Backlog (By Source)</th>
<th>Prioritized</th>
<th>Scheduled</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing/ Product Mgt (5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering (2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Representation from various sources
Cross-Board Decomposition and Drilldown
How We Have Benefited

- **Areas of improvement**
  - Right level of Features visibility throughout the cycle – from concept to deployment
  - Right level of User Story breakdown/ sizing for predictable delivery
  - Early commitment by Engg teams based on Estimation
  - Second level Prioritization by sales/ mgmt/ customers based on ‘cost of development”
  - Work on “truly relevant” requirements by PM team
  - Greater representation for all types of requirements (strategic → tactical → technical debt related)
  - Greater representation for all functions - non-sales/marketing including Support/ Engg!
Cumulative Flow Diagram (Minus backlog/ archive)

Average Cycle Time of User Stories

300% Reduction in Cycle Time thru a combination of factors!
Business Benefits

- Improved collaboration between functional teams and faster development
- Improved sales/ customer expectation management
- Faster feedback loops
- Better overall product/ market fit
- Faster time to market
About us

- Contact Information
  - mahesh@digite.com
  - @maheshsingh
- Learn more at
  - www.swift-kanban.com
  - www.digite.com
- Connect with us
  - Twitter - @swiftkanban/@digite
  - Facebook – www.facebook.com/digite

About Kanban

- Blogs and Articles
  - The Principles of Kanban Method
  - Kanban Applied to Software Development
  - Personal Kanban
  - Scrumban
  - Kanban vs. Scrum
  - 10 example Kanban boards
  - Explaining Cumulative Flow Diagrams
  - Your Family, Agile, and You
- Kanban Communities
  - Kanban Dev Group
  - Lean Agile Group
  - Lean Development
  - Kanban-Ops
  - IT Kanban

© Digite, Inc.