BUILDING PRODUCTS & SERVICES
with Agile Development

Sean Ammirati
@SeanAmmirati
Partner, Birchmere Ventures
Adjunct Professor, Carnegie Mellon University
March 4, 2015
Build

Learn

Measure
“that version of a new product [or service] which allows a team to collect the maximum amount of validated learning about customers with the least effort”

-- Eric Reis
What do you build?

Storyboards
Landing Pages
Demo Videos
Wizard of Oz
Storyboards

A panel or series of panels of rough sketches outlining the sequence of events a consumer will experience before, during and after using our product/service.
It’s a mirror

Encourages customers to be more honest

YOU won’t fall in Love
It’s a Mirror

Telling the customer’s story from their context
- how they will use the product / service
Encourages Customer Honesty

Easier to react honestly to rough sketches than final pixel perfect images
YOU will NOT Fall in Love

Never seen anyone fall in love with their story board

Storyboards
3 Benefits
Storyboards
Sample

http://www.slideshare.net/IntuitInc/conserve-code-storyboard-experiences-with-customers-first?next_slideshow=1
Storyboards

Sample

http://www.slideshare.net/IntuitInc/conserve-code-storyboard-experiences-with-customers-first?next_slideshow=1
Wrong Problem

http://www.slideshare.net/IntuitInc/conserve-code-storyboard-experiences-with-customers-first?next_slideshow=1
Storyboards

Wrong Problem

Wrong Benefit

http://www.slideshare.net/IntuitInc/conserve-code-storyboard-experiences-with-customers-first?next_slideshow=1
Storyboards

Sample

Problem

Solution

Benefit

http://www.slideshare.net/IntuitInc/conserve-code-storyboard-experiences-with-customers-first?next_slideshow=1
Landing Pages
High Performing

User Centric
Clear & Concise
Conversion Focused
Shareable
Measurable
Consistent
Demo Video

https://www.youtube.com/watch?v=7QmCUDHpNzE
Wizard of Oz

1. Submit

2. Email to you / your team

3. “automated” response
Storyboards
Landing Pages
Demo Videos
Wizard of Oz
Five Misunderstandings about MAPS
#1: Viable ≠ Crappy
#2: Not a destination!
<table>
<thead>
<tr>
<th>MAPS #1 &gt; Concept</th>
<th>MAPS #2 &gt; P-M Eval</th>
<th>MAPS #3 &gt; P-M Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAP #1</strong> PPC Campaign</td>
<td><strong>MAP #2</strong> Product Drawings</td>
<td><strong>MAP #3</strong> Prototype</td>
</tr>
<tr>
<td>Landing Page</td>
<td>Detailed Spec</td>
<td>Functional Output</td>
</tr>
<tr>
<td><strong>Features / Benefits</strong></td>
<td><strong>Face-to-face technology</strong></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>demo</td>
<td></td>
</tr>
<tr>
<td>‘More info’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call to Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Market insight</strong></td>
<td><strong>Locate Strategic</strong></td>
<td><strong>Field Pilot</strong></td>
</tr>
<tr>
<td>Find early adopters</td>
<td>partners</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Seed Funding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Paid-beta customers</td>
<td></td>
</tr>
<tr>
<td><strong>Investment</strong></td>
<td><strong>Revenue</strong></td>
<td><strong>Customer Validation</strong></td>
</tr>
<tr>
<td>$</td>
<td>$$</td>
<td>Capital Investment</td>
</tr>
<tr>
<td><strong>Customer Interaction</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Learning Objective</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Objective</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
#3: Validates or invalidates key hypothesis
#4: Doesn’t have to be a product at all
#5: Not always a landing page
AGILE Development

How do you build?
What do we mean by agile?

We value …

*Individuals & Interactions* over process and tools

*Working products* over documentation

*Customer collaboration* over negotiation

*Responding to change* over following a plan
Why agile development?

You’re wrong more then you are right

Key Metric: How fast can you iterate?

Need to Predict Delivery Times
How to do *agile* development?

Scrum

Engineering Practices

This will be unique for each of you based on your team, type of solution being developed and personal preferences.

Could be: XP, Feature Driven Development, Crystal, Kanban or any other process your engineering team is comfortable with.

*(often pull aspects from each)*
Why do we focus on **scrum**?

Firsthand observed it transform & improve my last software company - mSpoke

Being used at some of the largest technology companies in the world today (Google, Yahoo!, Adobe, etc ...)

Provides a great framework for entire team to understand what is going on.

*Disclaimer: Still hard to build innovative products & services and not a silver bullet*
Why do we focus on Scrum?
Key Themes from Scrum
The process of scrum?

Tips / Tricks for scrum?

Important for everyone to buy in and get familiar with vocabulary.

Tools for implementing scrum (not necessary):

- Google Doc Template - [https://docs.google.com/previewtemplate?id=0AoBzAQ22ynH5dEpRRkNBVGRCVU1HMmk2enBMYUZZZGc&mode=public](https://docs.google.com/previewtemplate?id=0AoBzAQ22ynH5dEpRRkNBVGRCVU1HMmk2enBMYUZZZGc&mode=public)
Product Backlog

Prioritized list or queue of requirements

Rough Estimates of level of effort to complete (not all estimates need to be equally thorough / higher priority can be more thorough)

Ultimately Product Owner sets the priority

Any one (customer, employee, board member, advisor) can add to product backlog

Should be shared with the full-team
## Product Wish List

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Total - Sprint 5</th>
<th>Effort Estimates</th>
<th>Planner</th>
<th>Difficult or Uncertain Issues</th>
<th>Abstract Requirement</th>
<th>Summary of Spec</th>
<th>Notes / Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>Tech</td>
<td>1.25</td>
<td>0.5d - Identify common patterns (Brian and Evan)</td>
<td>Evan</td>
<td>Improve front-end server architecture</td>
<td>Improve Javascript unit testing</td>
<td>Factor out common resource patterns</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Tech</td>
<td>2</td>
<td>1d research JUnit automated testing features</td>
<td>Brian</td>
<td>Organize the software for automatic operation on multiple nodes: e.g., automatic input feed processing, personalized feed processing. Integration testing and debugging of the software running as a whole.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>Tech</td>
<td>3</td>
<td>0.5d set up persistent Quartz environment</td>
<td>Dean</td>
<td>Database Migration Infrastructure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>Tech</td>
<td>2</td>
<td>1d - Rails Learning</td>
<td>Brian</td>
<td>Explore Tomcat redeployment issues</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>Tech</td>
<td>1</td>
<td>1d - Investigation</td>
<td>Evan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>IC</td>
<td>5</td>
<td>0.5d - Atom generation library research</td>
<td>Evan</td>
<td>The personalized feed URL should deliver feed content when pulled.</td>
<td>Generate an atom feed on each request containing all Personalized feed items that were created in the last 72 hours. (This line item includes basic styling.)</td>
<td>Sample of manual test</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>II</td>
<td>5.5</td>
<td>0.5d test image maps in FRs and browsers</td>
<td>Brian</td>
<td>When an item is delivered in a personalized feed, it includes a visual representation of how relevant the system believes it is and possibly why.</td>
<td>Generate a badge (per TH's spec) that the feed item references as an image (e.g., IMG tag) with image-map rollover (alt) text for each meme.</td>
<td>Does it make sense for this just to be a controller in our current spring Mvc server? Or should we think about making this more easily distributable across multiple machines separate from the other spring Mvc controllers?</td>
<td>Sample of manual test</td>
</tr>
<tr>
<td>75</td>
<td>II</td>
<td>4</td>
<td>2d design &amp; HTML layout</td>
<td>Brian</td>
<td>The user is able to provide &quot;thorough&quot; feedback on the preferences that drove the selection of a particular content item.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>R</td>
<td>8</td>
<td>1d remaining tweaks from last round with Jamie</td>
<td>Dean</td>
<td>Memes can leverage topics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: The table is a partial view and may contain additional rows and columns not shown here.*
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Product</th>
<th>Wish List</th>
<th>Analysis Required</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tech</td>
<td>Feature A</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>3</td>
<td>Open</td>
</tr>
<tr>
<td>Tech</td>
<td>Feature B</td>
<td>1</td>
<td>2</td>
<td>Low</td>
<td>3</td>
<td>Close</td>
</tr>
<tr>
<td>C</td>
<td>Feature C</td>
<td>1</td>
<td>2</td>
<td>High</td>
<td>3</td>
<td>Close</td>
</tr>
<tr>
<td>A</td>
<td>Feature D</td>
<td>1</td>
<td>2</td>
<td>Medium</td>
<td>3</td>
<td>Open</td>
</tr>
</tbody>
</table>

*Note: Table continues with similar columns and entries.*
<table>
<thead>
<tr>
<th>Category</th>
<th>Priority</th>
<th>Duration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tech</td>
<td>High</td>
<td>1 week</td>
<td>Implement new feature for improved user experience</td>
</tr>
<tr>
<td>Bug</td>
<td>Medium</td>
<td>2 days</td>
<td>Fix critical bug causing crashes in the application</td>
</tr>
<tr>
<td>Feature</td>
<td>Low</td>
<td>3 months</td>
<td>Add new feature for enhanced productivity</td>
</tr>
</tbody>
</table>

Notes:
- Ensure all new features are tested thoroughly before release.
- The team is working on improving the user interface for the next major release.
- Continuous integration and quality assurance is a priority for maintaining software reliability.
Everyone on team should easily be able to see the backlog

I’ve found one “administrator” helpful logistically

If not using a full scrum tool, you can do this easily in a shared spreadsheet (eg Google Docs)
Each sprint:

- Lasts a defined number of days (time box)
- Has a specific set of requirements from backlog allocated to it (defined during “sprint planning meeting”)
- Has specific goals for the team to achieve (set up front) - “sprint goal”
All estimates are **forward looking**

How much will it take to complete this feature / requirement?

Increasing an estimate based on learned complexity is accepted by the team

Sprint backlog estimates should be updated regularly
The average decrease in estimates for the total effort / time remaining is a sprint’s velocity.

Overtime velocity becomes very helpful for planning purposes.

The chart showing daily total of time remaining is called a burn down chart or sprint’s signature.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><strong>Who</strong></td>
<td><strong>Description</strong></td>
<td><strong>Original</strong></td>
<td><strong>6/20/07</strong></td>
<td><strong>6/21/07</strong></td>
<td><strong>6/22/07</strong></td>
<td><strong>6/25/07</strong></td>
<td><strong>6/26/07</strong></td>
</tr>
<tr>
<td>4</td>
<td>Jim</td>
<td>Create a QA environment in data center</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Jim</td>
<td>Place 'powered by mSpoke' image on Left side nav bard</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Jim</td>
<td>Show rating widget in thumbs-up page</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Jim</td>
<td>Move page header above cred space on items memes page</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Test</td>
<td>Change &quot;thanks for letting us know&quot; on thumbs-up</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Test</td>
<td>Improve Thumbs Up / Down Interaction</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>Fix / improve</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>John</td>
<td>#1896 should prioritize initial processing of new input feeds over re-processing of old ones</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>John</td>
<td>#1908 - need user administration page for deleting a user and/or personalized feed</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>Evan</td>
<td>#1977 Meme's Icon and Label below mouse pointer should resize accordingly on drag</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Evan</td>
<td>#2003 Meme display doesn't animate in NetNewsWire</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Evan</td>
<td>Better error handling implementation in the client across the board</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>Brian</td>
<td>Better architecture for error handling between client and server in general</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>Brian</td>
<td>Better passing of error context from server to client</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>Brian</td>
<td>Integrate Spring's Message Resources</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>Brian</td>
<td>Strip out malicious HTML / JavaScript @ Our WWW Display (HTML)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>Zach</td>
<td>Identify Cases</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>Zach</td>
<td>Controller Unit Test Infrastructure</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>Zach</td>
<td>500 user load test</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>Ian / John</td>
<td>#2032 Need to implement Status Edit link as spec'd</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Velocity / Burn Down

Graph showing the trend of hours remaining over time, indicating a decrease in remaining hours as time progresses.
Velocity / Burn Down

Graph showing the relationship between hours remaining and time, with a downward trend indicating the burn down process.
Sprint Review

At the end of the sprint, the team demonstrates what they have built

Compared against the sprint’s goals

Retrospective to look for improvements at the end of the sprint
Thank you

@SeanAmmirati

Subscribe to Newsletter: seanammirati.com/email

sean@birchmerelabs.com
Creative Commons

Slide 18: http://www.flickr.com/photos/kylemay/526384959

Slide 23: http://www.flickr.com/photos/mattimattila/8687764367

Slide 19: http://www.flickr.com/photos/isolino/3238550310

Slide 20: http://www.flickr.com/photos/superfamous/7550687770/sizes/o/