Building Products & Services with Agile Development

Sean Ammirati
@SeanAmmirati
Partner, Birchmere Ventures
Adjunct Professor, Carnegie Mellon University
February 24, 2016
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
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
Storyboards
3 Benefits

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 storyboard
Storyboards

Sample

Wrong Problem

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

Wrong Problem

Wrong Benefit

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

**Problem**

- AI just faxed this—is it what I need?
- Where is the original request?
- In Email?
- In these files?
- This is not what I needed. Now I have to type up another email (for the 23rd time).
- What else does AI owe me? Oh, it takes forever to figure this out.
- All my requests and his responses are scattered between files and emails.

**Solution**

- It's so easy to keep track of what AI sent me because it's attached to the checklist.
- I also need the receipt from AI for this transaction. Let me add a note and re-publish this item to the checklist.
- Having all my requests and responses managed in one place saves me up to 2 hours of typing and hunting through files.

**Benefit**

- Let me review the receipt attached. It's just what I needed, I'll mark it complete.

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
Wizard of Oz

Submit

Email to you / your team

“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>Customer Interaction</th>
<th>Learning Objective</th>
<th>Investment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP #1 PPC Campaign Landing Page</td>
<td>Market insight Find early adopters</td>
<td>$</td>
</tr>
<tr>
<td>MAP #2 Product Drawings Detailed Spec</td>
<td>Locate Strategic partners Seed Funding Paid-beta customers</td>
<td>$$</td>
</tr>
<tr>
<td>MAP #3 Prototype Functional Output</td>
<td>Revenue Customer Validation Capital Investment</td>
<td>$$$</td>
</tr>
</tbody>
</table>

**MAPS #1 > Concept**
- Features / Benefits description
- ‘More info’
- Call to Action

**MAPS #2 > P-M Eval**
- Face-to-face technology demo

**MAPS #3 > P-M Fit**
- Field Pilot

**What do you measure & learn?**

- **MAPS #1 > Concept**
  - Market insight
  - Find early adopters

- **MAPS #2 > P-M Eval**
  - Locate Strategic partners
  - Seed Funding
  - Paid-beta customers

- **MAPS #3 > P-M Fit**
  - Revenue
  - Customer Validation
  - Capital Investment

**Customer Learning Objective Investment**

- **MAPS #1 > Concept**
  - Market insight
  - Find early adopters
  - $ |

- **MAPS #2 > P-M Eval**
  - Locate Strategic partners
  - Seed Funding
  - Paid-beta customers
  - $$ |

- **MAPS #3 > P-M Fit**
  - Revenue
  - Customer Validation
  - Capital Investment
  - $$$
#3: Validates or invalidates key hypothesis
#4: Doesn’t have to be a product at all
#5: Not always a landing page
How do you build?

AGILE Development
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 than you are right

Key Metric: How fast can you iterate?

Need to Predict Delivery Times
How to do agile development?

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=0AoBzAQ22ynH5dEpRkNBvGRCVU1HMmk2enBMYUZGS2c&mode=public](https://docs.google.com/previewtemplate?id=0AoBzAQ22ynH5dEpRkNBvGRCVU1HMmk2enBMYUZGS2c&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) 0.75d - Factor out patterns</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 1d automated test infrastructure development</td>
<td>Brian</td>
<td></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 1.0d configure Quartz 0.5d other odds and ends 1.0d test and rework</td>
<td>Dean</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>85</td>
<td>Tech</td>
<td>2</td>
<td>1d - Rails Learning 0.5d - Testing 0.5d - First Migration</td>
<td>Brian</td>
<td>Database Migration Infrastructure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>Tech</td>
<td>1</td>
<td>1d - investigation</td>
<td></td>
<td>Explore Tomcat redeployment issues</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 0.5d - Programmatic URL creation research 1.0d - Implementation of Atom generation 3d - manual testing</td>
<td>Evan</td>
<td></td>
<td></td>
<td></td>
<td>Sample of manual test</td>
</tr>
<tr>
<td>12</td>
<td>II</td>
<td>5.5</td>
<td>0.5d test image maps in FRs and browsers 2d badge generator API and algorithm design 2.5d development 5d manual test</td>
<td>Brian</td>
<td></td>
<td></td>
<td></td>
<td>Sample of manual test</td>
</tr>
<tr>
<td>75</td>
<td>II</td>
<td>4</td>
<td>2d design &amp; HTML layout 1.5d development 0.5d manual test</td>
<td>Brian</td>
<td></td>
<td></td>
<td></td>
<td>Sample of manual test</td>
</tr>
<tr>
<td>36</td>
<td>R</td>
<td>8</td>
<td>1d remaining tweaks from last round with Jamie 2d first pass at improving scalability 1d first pass at testing scalability 1d finish scalability</td>
<td>Dean</td>
<td>Memes can leverage topics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the user clicks on the badge or "why was I shown this?" link in the feed item, they are taken to a web page with a badge and a representation of the item. On this web page, they can hover over the memes in the badge to see descriptions or click to get the "property edit" dialog.

A subject such as "Hurricane Katrina" or "the surge in Iraq" that lasts from a few days to a few months, and that is too ephemeral to reliably get an appealing human-readable name.
<table>
<thead>
<tr>
<th>Category</th>
<th>Product</th>
<th>Description</th>
<th>Priority</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tech</td>
<td>1</td>
<td>New feature</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Tech</td>
<td>2</td>
<td>Enhance UI</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>Fix bug</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>Add new functionality</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>Improve performance</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>Enhance security</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>7</td>
<td>Optimize deployment</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>Add new feature</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>9</td>
<td>Enhance usability</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>10</td>
<td>Fix bug</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>11</td>
<td>Add new feature</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>12</td>
<td>Enhance security</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>13</td>
<td>Optimize deployment</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Note: Priority levels range from 1 (highest) to 5 (lowest).
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”
Estimates

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></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<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>
<td>4</td>
</tr>
<tr>
<td>Port</td>
<td>Port</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvements to thumbs-up and thumbs-down landing pages</td>
<td>Place 'powered by mSpoke' image on Left side nav bard</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Show rating widget in thumbs-up page</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Move page header above cred space on items membes page</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Change &quot;thanks for letting us know&quot; on thumbs-up</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Make &quot;Powered by Technorati&quot; image smaller</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Move &quot;Powered by Technorati&quot; icon underneath results on thumbs-up page</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Add box around Technorati results on thumbs-up page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improve Thumbs Up / Down Interaction</td>
<td>John Investigate</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>John Fix / Improve</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td></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>
<td>0</td>
</tr>
<tr>
<td></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>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></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>
<td>2</td>
</tr>
<tr>
<td></td>
<td>#2003 Meme display doesn't animate in NetNewsWire</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Better error handling implementation in the client across the board</td>
<td>Brian Centralized UI for error handing</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Brian Develop useful messages for current errors thrown</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Better architecture for error handling between client and server in general</td>
<td>Brian 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>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Brian Integrate Spring's Message Resources</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Strip out malicious HTML / JavaScript @ Our WWW Display (HTML)</td>
<td>All Identify Cases</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Controller Unit Test Infrastructure</td>
<td>Zach Learn Spring</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Zach Research existing implementations</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Zach Build framework</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>500 user load test</td>
<td>Ian / John Test</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>#2032 Need to implement Status Edit link as spec'ed</td>
<td>Sheet1</td>
<td>4</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
Velocity / Burn Down

![Velocity / Burn Down Graph](image-url)
Sprint Review

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