Avoiding Scrummerfall

What the Agilista forgot to mention...
Ryan Cromwell

- Coder
- Agile Coach

@cromwellryan
http://cromwellhaus.com
Agile Language
Fail
Product Backlog Item (PBI)

- User Story
- Feature
- Requirement
- Idea
- Use Case
What is Scrummerfall?
Project Scrummerfall
LIPSTICK ON A PIG

It's still a pig.

Sprint Scrummerfall

Week 1
Define

Week 2
Design

Week 3
Code

Week 4
Test
How can we fix it?
Don’t stuff the turkey

Complex problem solving requires focus, knowledge, creativity, and space
Fancy Forecasting

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Top 3 Avg</th>
<th>Best Case</th>
<th>Middle 3 Avg</th>
<th>Likely Case</th>
<th>Bottom 3 Avg</th>
<th>Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>23</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>
Can we do all of this well in the next few weeks?

Have we done this much before?
“Working software is the primary measure of progress”
Agile Manifesto
Validate Assumptions

- Product Backlog
- Assumptions
Done is when an item is:

1. Completed & Tested
2. Verified by PM
3. Incredible! (when applicable)

http://www.planbox.com/blog/agile-project-management/done-means-done.html
Continually Refine your Product Backlog
Progressive Enlightenment

Current Sprint

Next Sprint

Next, Next Sprint
HOW TO SPLIT A USER STORY

1. PREPARE THE INPUT STORY
   - Does the big story satisfy INVEST? (independent, negotiable, valuable, estimatable, small, testable)?
   - Combine it with another story if possible, or split it for a good, split, longer story.
   - Is the story size %e to %d of your velocity?
   - You’re done. Continue. You need to split it.

2. APPLY THE SPLITTING PATTERNS
   - Can you split the story so you do the beginning and end of the workflow first and enhance with stories from the middle of the workflow?
   - Can you split the story to do that simple core first and enhance it with later stories?
   - Can you split the story to do a subset of the rules first and enhance with additional rules later?
   - Can you split the story to process one kind of data first and enhance with the other kind later?
   - Can you split the story to handle data from one interface first and enhance with the others later?
   - Does the story have a simple core that provides most of the value and learning?
   - Does the story have a complex core?
   - When you apply the obvious split, is it the easiest story you do first the most difficult?
   - Can you find a small piece you understand well enough to start?
   - Can you define the 1-3 key questions most holding you back?
   - Write that story first, build it, and start again at the top of this process.
   - Write a spike with these questions, do the minimum to answer them, and start against the top of the process.

3. EVALUATE THE SPLIT
   - Are the new stories roughly equal in size?
   - Do each of the stories satisfy INVEST?
   - Is each story about 1% to 10% of your velocity?
   - Try another pattern on the original goal or the larger post-split story.
   - Are there stories you could deconstruct?
   - Try another pattern. You probably have waste in each of your stories.
   - You’re done, though you could try another pattern to see if it works better.

* INVEST - Stories should be:
  - Independent
  - Negotiable
  - Valuable
  - Estimatable
  - Small
  - Testable

Vis: http://www.richardlawrence.info/splitting-user-stories/ for more info on the story splitting patterns
Copyright © 2011-2013 Agile For All. All rights reserved.

Last updated 3/4/2013

Have a Sprint Goal

Providing the vision for a Sprint
Example Sprint Goals

- Increase social networking presence
- Reduce cart abandonment by 10%
- Deliver minimal account self servicing
- Make the application run in AWS
- Increase find accuracy by 20%
Progressive Enlightenment

Big ideas often become goals

Current Sprint

Next Sprint

Next, Next Sprint
“Reduce cart abandonment by 10%”

- Allow checkout without confirming inventory
- Send discount email after 18 hours of inactivity
- Quick checkout from any page
- Integrate Bill-me-later™
- Remove adds from cart page (speed)
- Auto complete fields for return customers
Avoid Narcissistic Backlogs

Prefer business value over technical requirements
Signs of unsustainable pace

“Clean authentication code”

“Add logging to message bus”

“Refactoring Sprint”
Less the business cares...

the more nervous you should be.
Avoid Stabilization Sprints
Tasks Isolate, Tests Unite

Measure Sprint progress in passing tests
Burndown Tests

Current Sprint

Scenario 1
Given ...
When ...
Then ...

Scenario 2
Given ...
When ...
Then ...

Load Test
Smoke Test
### SpecDriver Test Execution Report


#### Summary

<table>
<thead>
<tr>
<th>Features</th>
<th>Success rate</th>
<th>Scenarios</th>
<th>Success</th>
<th>Failed</th>
<th>Pending</th>
<th>Ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 features</td>
<td>100%</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Feature Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Success rate</th>
<th>10 scenarios (10 passed)</th>
<th>139 steps (139 passed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Search</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Feature Execution Details

**Feature: Google Search**

**Scenario:** Basic installation

- **Given** WordPress application is uninstalled
- **When** I install WordPress application
- **And** install Membership Login plugin and activate
- **Then** no errors appeared
80% of each story means nothing is done
Swarming on Solutions

http://scrum.org
What might this look like?

<table>
<thead>
<tr>
<th>PBI</th>
<th>Todo</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
<tr>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
<tr>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
<tr>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
<tr>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
<td>![icon]</td>
</tr>
</tbody>
</table>
What this does **not** look like?

<table>
<thead>
<tr>
<th>PBI</th>
<th>Todo</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>![PBI]</td>
<td>![Todo]</td>
<td>![In Progress]</td>
<td>![Done]</td>
</tr>
<tr>
<td>![PBI]</td>
<td>![Todo]</td>
<td>![In Progress]</td>
<td>![Done]</td>
</tr>
<tr>
<td>![PBI]</td>
<td>![Todo]</td>
<td>![In Progress]</td>
<td>![Done]</td>
</tr>
<tr>
<td>![PBI]</td>
<td>![Todo]</td>
<td>![In Progress]</td>
<td>![Done]</td>
</tr>
<tr>
<td>![PBI]</td>
<td>![Todo]</td>
<td>![In Progress]</td>
<td>![Done]</td>
</tr>
</tbody>
</table>
Unified Demo

A team that builds together ships together
Integrate Often Always

- Demo integrated, Done software
- No slides!
- No tomfoolery!
10 Feature Toggles

Progressive roll-out, canary deployments, etc.
using System;

namespace ExampleWebFormsApplication
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Feature2Button.Visible = new Feature2Toggle().FeatureEnabled;
        }
    }
}
Questions?

Stories?

Ideas?

Thoughts?
Thank You!

ryan.cromwell@appliedis.com
http://appliedis.com

@cromwellryan
http://cromwellhaus.com