WRITING GOOD USER STORIES

EDGAR C. WORTS, III
AGENDA

1. What, Why, and Who
2. Characteristics of a User Story
3. Product Backlog Refinement
   1. Epics vs Features vs User Stories
   2. Acceptance Criteria
   3. Agile Estimation
4. Sprint Planning: Task Breakdown
5. Bad and Good Examples
6. Definition of Ready
7. Open Space
WHAT, WHY, WHO
A user story represents a small piece of business value that a team can deliver in an iteration.
Because all stakeholders should be able to understand what a system must do to deliver the desired value to their customers and because we want to promote conversation and clarification.
**USER STORIES: WHO**

**Product Owner (P.O.)**
- Gathers Feedback
- Makes Business Decisions
- Voice of the Customer
- Owns Value
- Owns the Product Backlog

**Scrum Team**
- Anyone can write user stories.
  Who writes a user story is far less important than who is involved in the discussions of it.
CHARACTERISTICS OF A GOOD USER STORY
USER STORIES: INVEST

• If Product Owners and their teams work together to INVEST in good user stories the learning curve of working together will be much shorter.

• Characteristics of a Good Story

  - **Independent**: It should be self-contained.
  - **Negotiable**: Not carved in stone and should leave space for discussion.
  - **Valuable**: It must deliver value to the identified persona.
  - **Estimable**: It is possible to estimate its size.
  - **Small**: It must fit within an iteration.
  - **Testable**: It must provide the necessary information to be tested.
USER STORIES: VALUE

Like this!  Working Product Increment

1. 2. 3. 4. 5.
PRODUCT BACKLOG REFINEMENT:

EPICS VS FEATURES VS USER STORIES
THE SIMPLE WAY

AGILE - SCRUM

I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.

AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."

OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.
USER STORIES: VSTS PRODUCT BACKLOG

- Epic
  - Large initiatives delivering new products, solutions, or services to customer
  - Comprised of a large collection of features

- Feature
  - Capabilities that the product owner is interested in
  - Provides values to users
  - Realized by some number of user stories

- User Story
  - Represents a user’s need
  - Planning item
  - Causes a conversation to occur

Same Epic or Feature may appear in multiple Sprints or even Releases
PRODUCT BACKLOG REFINEMENT:
ACCEPTANCE CRITERIA
ACCEPTANCE CRITERIA

• Like User Stories it is written in **simple language**
• Define the **conditions of success/satisfaction**
• Provide clear user story **boundaries**
• **Remove ambiguity** by forcing the team to think through how a feature or piece of functionality will work from the user’s perspective
• Establish the basis for **acceptance testing**
  • **Steps to test the user story** (given-when-then scenarios)
PRODUCT BACKLOG REFINEMENT:

ESTIMATION
• Planning Poker (Story Points)

• T-Shirt Sizing
ESTIMATION

- Complexity
- Risk
- Implementation
- Deployment
- Interdependencies

The diagram illustrates the complexity and risk levels associated with various transportation modes, from skateboard (0) to Space Shuttle (100), representing different stages of development and deployment.
ESTIMATION: VELOCITY & RELEASE PLAN

Velocity

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Completed</th>
<th>Incomplete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 2</td>
<td>55</td>
<td>12</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>55</td>
<td>10</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>58</td>
<td>2</td>
</tr>
</tbody>
</table>

56 Story Points

Release Burndown

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Story Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 1</td>
<td>299</td>
</tr>
<tr>
<td>Sprint 2</td>
<td>265</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>210</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>152</td>
</tr>
<tr>
<td>Sprint 5</td>
<td>96</td>
</tr>
<tr>
<td>Sprint 6</td>
<td>40</td>
</tr>
</tbody>
</table>

299 Story Points
Product Backlog
SPRINT PLANNING TASK BREAKDOWN
TASK BREAKDOWN – MAPPING THE VALUE STREAM

- Accept Credit Card Payment
  - State: Committed
  - Effort: 3

- Research Legal Requirements
  - Unassigned

- Get Approval to Deploy in QA
  - Unassigned

- Create Test Cases
  - Unassigned

- User Interface
  - Unassigned

- Apply Database Changes
  - Unassigned

- Review Test Cases
  - Unassigned

- Update User Documentation
  - Unassigned

- Create Database Model
  - Unassigned

- Code Gateway Integration
  - Unassigned

- Test
  - Unassigned

- Update Deployment Process
  - Unassigned

Concurrent Work
BAD AND GOOD EXAMPLES
Slice it!

As a Manny’s food service customer, I need to save my list so that I can reorder from the list to create more accurate food orders.

As a Manny’s food service customer, I need to copy my list so that I can use it as a starting point for creating another list.

As a Manny’s food service customer, I need to print my list so that I can check a received shipment against the printed list.

As a Manny’s food service customer, I need to email my list so that I can have someone who doesn’t use the system review my list.
• As a Manny’s food service customer, I want to see different food item types displayed in different colors—RGB = #FF0000 for meats, #A52AFA for grains, and #808000 for vegetables and fruits—so that I can quickly identify my food items by food type.
• As a customer ordering food, I want to locate previous food order lists so that I can see all the lists that I have.

As a customer ordering food, I want to see my saved food order lists so that I can reuse the list for future orders, making ordering faster and more accurate.
USER STORIES: 5 COMMON MISTAKES

Criteria Crisis
None, restate the narrative, and hide new stories.
“Confirmation”

Story Mania
Like any technique, user story writing has its strengths and limitations.
WHEN IS MY USER STORY READY FOR DEVELOPMENT?
DEFINITION OF READY

• Enables a team to specify certain pre-conditions that must be fulfilled before a story is allowed into an iteration
  
  • Example:
    • Acceptance criteria must be clear and testable.
    • All tasks assigned to the story.
    • Team is staffed appropriately to complete it.
    • User stories need to be INVEST.
    • Performance criteria exist
    • Ensure the result is demonstrable.
    • Mockups