The Best Cookie She Ever Had (Part 2)

As you may recall in "The Best Cookie She Ever Had (Part 1)", after some cookie talk with my 6-year-old neighbor, I confirmed at the most basic level that **written requirements must greatly differ between a maker audience and a consumer audience**.

To make my future backlog management processing as efficient as possible, I needed to ensure each story's acceptance criteria reflected only the voice of the "requirement requestor", which could be a product owner, consumer, architect, quality assurance engineer, or even a chef.

Based on my experience as a process improvement guy the easiest way for me to improve was to start by writing down the user story creation requirements I wanted to fulfill as a BA then execute these requirements in a standardized way. With standardization in place, my performance could be monitored over time for continuous improvement. Listed below are the requirements I developed to help me improve.

Randy's User Story Information Requirements:

1. Each user story shall include the following sections and information:
    a. A "**Summary**" section with a title for the story that includes the "I need <feature or functionality>" text of the user story statement.
    b. A "**Description**" section (aka "why") that includes the following information:
        i. A "**User Story**" section with a story statement told in the Agile "As a <enterprise or system role title> I need a <feature or functionality>, so I can <business action>.
        ii. A "**Background**" section with a brief explanation of why the feature or functionality is required (no more than a sentence or two). *This will help establish*

context for consumers of the information and will promote buy-in if written correctly.

   c. A "**Who**" section with a definition of who will see/use the feature and functionality whether is people or a system.
   d. A "**What**" section with details that define the specific feature/functionality that is required to be built and tested before the story is considered done and ready for deployment to production and that meets the following INVEST mnemonic standards:
      i. **Independent** – the requirements can be developed and tested independently.
      ii. **Negotiable** – the requirements can be modified throughout refinement.
      iii. **Valuable** – the requirements add quantifiable value to the product.
      iv. **Estimable** – the scope of the requirements is clear; there is a clear end and a clear beginning so the team can estimate complexity or duration.
      v. **Small** – the requirements are written clearly and concisely, and the story can be completed in one Sprint (2-week period) or less.
      vi. **Testable** – the requirements are not general. They are specific, so anyone can test them to confirm the requirement was met.
2. A "**Where**" section to illustrate where the feature/functionality will exist. This will help the team understand where to test and control the scope.
3. A "**Supporting Information**" section (aka "How") that includes, but is not limited to the following information:
   a. List the names and contact information for key support stakeholders like the product owner and solution architect
   b. Description of how the work will be tested
   c. Where to find supporting documentation, both technical and non-technical
4. An attached static .pdf design where available
5. A Story Point value that complies with the following rules:
   a. Points shall reflect the number of days it should take to build and test the story *(Although true to Agile I found pointing to complexity instead of days is slow and consistently causes confusion without significant informational gain. There seems to be a direct correlation between increased complexity and increased development/testing time, estimating with working days might be the best approach for unified clarity.)*
   b. Points shall follow the Fibonacci Sequence
6. Development and testing work instructions shall be defined in sub-task work items that are linked to each applicable user story and estimated in work hours by the assigned developer or quality engineer.
7. User stories can be created for business or technical requirements, but the "WHAT" section shall remain pure to the requirement requestor (think product team OR builders OR testers OR…).

After implementing my new user story information requirements, I saw the following improvements:

1. Standardizing the information in each story to reflect the voice of the requirement requestor made the event easier for the refinement participants
2. When refinement participants clearly understood what was required to fulfill a story's requirements, refinement became more efficient
3. Questions about each story, post-refinement were greatly reduced

The Best Cookie She Ever Had (Part 2)

To illustrate this approach, I created two example cookie user stories that can be reviewed by copying this link https://jira.external-share.com/issue/99425/best_cookie and pasting it into your browser.

The "**Distribute a high-quality chocolate chip cookie product**" User Story is written in a way that is meaningful to a product team and the "**Include 14 chips in each chocolate chip cookie**" user story is written in a way that is meaningful to cookie creators. Notice the spirit of each is completely different and specifically applicable to the requirement requestor, so refinement and final deliverable demonstrations are more easily executed.

If you decide to try some of this yourself, please share a comment on how it did or didn't work for you. Thank you for tuning in!

