Finding and Writing Effective User Stories

Mike Hall
Three Beacons, LLC
mike@threebeacons.com
214.783.3936
User Stories – origin

- User Stories come from Extreme Programming in 1999
- Originally used to help create time estimates for release plan
- Discovered that they can also be used as an “informal requirements spec”
  - Less detail than a formal requirements spec
  - Must query the customer or proxy for more details
User Stories – who?

- User Stories - as originally defined - are written by the customer

- Often this is impractical*

- In Scrum, the responsibility for developing User Stories lies with the Product Owner
  - Customer interface
  - Business-side interface
  - Sales & Marketing interface
  - Etc.
User Stories – who?

- User Stories are ideally written by the customer, from the **point of the user** of the system or another **system**

- User Stories are often written, reviewed, and improved collaboratively
  - Product Owner
  - Business Analysts
  - Customer proxy
  - QA
  - Developers
  - Etc.

**Advice:**
Have the Scrum development team assist in the writing/review of the user stories.
User Stories – what?

• They are a brief informal description of an end-to-end action that the system must perform to meet the user’s goal

• Much smaller than a requirement spec

• They are written on index cards (or within tool) - 3x5 or 4x6

• **Represent requirements** instead of documenting them

• A reminder that a conversation needs to be held to clarify the details of the requirement

**Purposefully high-level**

[0.5] As a loan rep I want to see all pending loans in my pipeline, so that I have a visual reminder of follow-up calls.
User Story - format

• Name – always start with verb!

• Concise problem statement
  • “As <role>,
  • I want <function>,
  • so that <business value>”

• Acceptance test case(s)
  • On back side
  • Always required!

1. View Pending Loans
   As a loan rep, I want to see all pending loans, so that I have a visual reminder of follow-up calls.

2. When I select the Pending Loans menu item, I expect a new window with pending loans.
   When I select the Pending Loans menu item, I expect to see only pending loans for me.
   When I close the window, I expect to be taken back to the Main window.
Acceptance Tests

- Simple format:
  - *When I* <action>, *I expect* <result>

- Review the stories and acceptance tests
  - Augment for completeness, boundary, stress, etc.
  - New requirements

- “Test First” thinking
### Story Name: Accept Incoming Call

**Description:**

As a NL6 user, I want to be able to accept an incoming call if I am connected to my phone via Bluetooth, so that I can use the NL6 as my speakerphone.

**Acceptance Criteria:**

While connected to the NL6 with my phone using BT, when an incoming call notification appears, I expect an option to be presented to me to accept the incoming call.

While connected to the NL6 with my phone using BT, when I select the option to accept an incoming call, I expect the NL6 to accept and answer the incoming call, and indicate that my call is active and in-progress. See "Call In-Progress" user story for more details.

**Questions/Notes:**

While connected to the NL6 with my phone using BT, when I select the option to accept an incoming call, I expect the NL6 to establish audio input and output between the two devices.

### Others:

- Launch My Phone
- Pair phone
- Unpair phone
- Ignore incoming call
- Adjust volume up/down
- Etc.
User Stories – why?

• Lightweight
• Easy to use, easy to maintain
• Allows for fast start of project development
• Encourages collaboration and participation
• Nice fit for iterative development
• Build up tacit knowledge and awareness
• Rallying point for organizing development into bite-size chunks
• Well-known Agile technique
  – Used in all Agile Methods (Scrum, Lean, XP, DSDM, FDD, etc.)
User Stories – how to find? (1 of 2)

• “Trawl” for user stories
  - Discuss project vision
  - Workshops
  - Start writing down the user story names
  - Additional user stories will emerge as details are discussed

• Determine user roles
  - Attributes that characterize a population of users
  - Monster.com user roles
    - Job poster
    - Job seeker
    - New grad
    - Geographic searcher
    - Resume screener
    - Administrator, Etc.

• Focus on goals for each user role
User Interviews

- Default approach
- Interview users with different roles

Observation

- Watch users
- Focus groups

Questionnaires

- When you need answers from a large number of users
- Does not lend itself to follow-up or clarification
Exercise: Follow the Script!

- Work as **individuals**.
- Take the DWS project vision handout and some index cards.
- Develop 1 user story for this system. Include at least 2 acceptance test cases on the back.
- I’m the customer, so engage me!

<table>
<thead>
<tr>
<th>FRONT</th>
<th>BACK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong>&lt;br&gt;“As &lt;role&gt;, I want &lt;function&gt;, so that &lt;business value&gt;”</td>
<td><strong>When I</strong> &lt;action&gt;, I expect &lt;result&gt;&lt;br&gt;<strong>When I</strong> &lt;action&gt;, I expect &lt;result&gt;</td>
</tr>
</tbody>
</table>
User Stories
- Workshop
User Story Workshop

- User Stories can be developed by one person

- Ideally, they are developed in a “workshop”
  - Collaboratively
  - Great for teamwork
  - Different perspectives – cause discussions
  - Thoroughness, completeness

- Workshops can be called by anyone
  - Customer
  - Customer proxy
  - Product owner
  - Business analyst
  - Development team members
  - Etc.
Exercise: User Story Workshop

- Huddle up into your sub-teams – hold a User Story Workshop.
- Take the DWS project vision handout and new index cards.
- Develop all the user stories for this system collaboratively using consensus.

Reflect – how was this compared to individually?

When I do <action>,
I expect <result>

When I do <action>,
I expect <result>
User Stories
Quality
User Story - format

• May also contain other relevant info
  - Notes
  - Screen mocks (to help describe requirements)
  - Non-functional test cases (performance, usability, etc.)
  - Story point estimate from Eng
  - Priority designation
  - Etc.

• Avoid details of specific technology, data base layout, and algorithms

• Keep stories focused on user needs and benefits ("what") as opposed to specifying solutions ("how")
User Story Attributes

The acronym "INVEST" can remind you that good stories are:

- I - Independent
- N - Negotiable
- V - Valuable
- E - Estimable
- S – Small, Sized Appropriately
- T - Testable

[from XP Explored by Bill Wake]
User Stories - **Independent**

- **Stand-alone**
  - Starts with some stimulus
  - Ends with some result
  - Vertical development vs. horizontal

- **Ideally, no dependencies with other stories**
  - Dependencies can make planning more difficult
  - Dependencies can make prioritization more difficult
  - Dependencies can make estimating more difficult
  - To avoid: combine user stories, or split them differently

- **Order Independent** - can be scheduled and implemented in any order
User Stories - *Negotiable*

- A user story is not a contract
- It is a “reminder for a conversation”
- Purposefully high-level
  - Captures the essence, not the details
  - Clarified through conversations
  - Adjusted as needed
User Stories - **Valuable**

- Must be valuable to the customer / end user
  - Best if development team and PO understand the value also
  - No discernable value? Don’t do it.
  - Avoid user stories that are only valued by developers

- Should deliver some business value
  - Exceptions should at least be preparatory in nature
User Stories - **Estimable**

- Estimate does not have to be exact
  - But good enough in relation to other user stories
  - But good enough to help sprint planning
  - But good enough to help in prioritization
  - But good enough to help release planning

- Planning Poker
  - Story point estimates
  - Relative in size to all other stories
  - Allows team to determine their *velocity*
  - Eases release planning
User Stories – **Small / Sized Appropriately**

- Each story should take less than your iteration length (2 weeks, 4 weeks) to implement
  - Lines up well with Scrum sprint
  - Larger stories are more difficult to estimate and plan
- Ideally, no bigger than 2 – 3 person weeks of effort
- Each story can be broken down into a reasonable number of tasks
- Longer than iteration length?
  - Break it down into more granular stories
- Epic?
  - Break it down into separate stories as it bubbles up in priority
  - “A job seeker can find a job.” – monster.com epic
Example - Small / Sized Appropriately
Example - Small / Sized Appropriately

Estimate is too big for a sprint

Right-sized pieces*

Demark point

Sprint n + 1

Sprint n
User Stories - Testable

• 3 C’s of the user story:
  - Card
  - Conversation
  - **Confirmation** (via testing)

• “Testability” is a characteristic of a good requirement
  - Helps the customer assess business value
  - Helps the dev team know what is required
  - Magic question: “How will I know I’ve done that?” Answer: acceptance criteria

• Acceptance tests cases written prior to story implementation
  - Helps the team be more productive
  - Builds quality in up front
  - Allows for automation efforts
User Stories – User Extra Check

On behalf of the user, ask yourself these additional questions:

• How does the user know when this user story is done?

• Does the system need to provide interim status to the user?

• Are there any user actions that would halt this story?

• Are there any user actions that would take this story in a different direction?

• Is there any part of the user story that might be confusing to the user?

• Does the story work well for novice users as well as experienced users?
On behalf of the development team, ask yourself these additional questions:

• Does the story use vocabulary that is consistent with the project?
  - e.g. “account” vs. “medical workflow session”

• Put yourself in the developer’s position. Are any of the words confusing or easily misconstrued?

• Are the nouns used self-documenting? Would they benefit from qualifiers?
  - e.g. “accept button” vs. “Facebook Posting button”

• Are the verbs used the right ones? Are they explicit enough?
  - e.g. “changed” vs. “modified and saved for future use”
Exercise: Good or Bad User Story?

- Huddle up into your sub-teams.

- Take the handout of 8 user stories.

- Brainstorm and rate each one as “good” or “bad” – and why.
Story 1: Display Video for Badge Reader Button Push

As the video administrator,

I want a pop-up to play video for 5 seconds when someone presses the call button on the badge reader,

so that I can visually verify the identity of the person and decide whether or not to unlock the door.

Story 2: Download & Play Selected Video File

We need to be able to access the server and download a video file selected from the list of files. It is expected that the server runs a database, preferably SQL so that this search is easy. And make sure it is one of those “free” databases. If we could do it using a web-based client it would be awesome. Also, be sure and add in wireless access in your spare time. Irregardless, make sure the UI has some snazz to it.
**Exercise: Good or Bad User Story?**

**Story 3: Configure Screen Areas for Specific Cameras**

*As* the video administrator,

*I want* to be able to designate certain areas of my PC screen to each of the video devices

*so that* I can easily monitor multiple cameras at once.

Example:

Note: use of highlighting to indicate video activity would be helpful. Note: may want to consider alignment with phone button colors (?)

**Story 4: View Medicaid report**

*As* a fraud analyst, *I want* to see a report of the duplicate Medicaid filings for every month.
Story 5: Accept Job Posting Payment

A company can pay for a job posting with a credit card.

Story 6: Identify cost trends

As an HMO administrator, I want to access data from the Operations data mart (SoCal region only), so that I can identify cost trends.
Story 7: System Responsiveness

As a user, I want to never have to wait very long for any screen to appear, so that I can work quickly.

Story 8: Create patient churn report

As a business analyst, I want to run reports from the Accounting data mart, so that I can aggregate the reasons for our patient churn.

Note: certain reasons should not be included in the analysis, such as death, moved to foreign country, and reached age of 65.
Lifecycle
User Stories – in relation to project lifecycle

Requirements clarification using User Stories

Initial User Story Development

User Stories

Development Sprints

Vision

Official Project Kickoff

Product Releases

Project Complete

© Three Beacons LLC, 2013
Within a Sprint

<table>
<thead>
<tr>
<th>Items</th>
<th>To Do</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Story</td>
<td>Task</td>
<td>Task</td>
<td></td>
</tr>
<tr>
<td>2. Story</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td>3. Story</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td>4. Story</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
</tr>
</tbody>
</table>

Sprint Goal
Genesis Project
Exercise: Genesis Project

• Huddle up into your sub-teams – hold a User Story Workshop.

• Take the Genesis project vision handout and new index cards.

• Create the user stories for this system.

• Hint: determine story names first, then split the stories up to write them. Discuss internally for consensus.

• Be prepared to present!

Exercise:

When I do <action>, I expect <result>

When I do <action>, I expect <result>

As who, I want what so that why.
Final Thoughts
Takeaways – Effective User Stories

- The most popular Agile technique
- Use the syntax, easy to learn
- I-N-V-E-S-T
- Lightweight technique for managing (ever-changing?) requirements
- If you are currently doing “traditional requirements”, please be advised – your competition is very likely using User Stories!
## Traditional Reqs vs. Agile Reqs

<table>
<thead>
<tr>
<th>Traditional</th>
<th>User Stories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow project start</td>
<td>Fast project start</td>
</tr>
<tr>
<td>Document-focused</td>
<td>Collaboration-focused</td>
</tr>
<tr>
<td>All details written down</td>
<td>Essence written down</td>
</tr>
<tr>
<td>Details exist in written words</td>
<td>Details exist in the user story + conversations</td>
</tr>
<tr>
<td>Written words - easy to misinterpret</td>
<td>Conversations – wavelength sync</td>
</tr>
<tr>
<td>Delivery: document</td>
<td>Delivery: stories + understanding</td>
</tr>
<tr>
<td>Cumbersome to update</td>
<td>Easy to update</td>
</tr>
<tr>
<td>Slower project delivery to market</td>
<td>Faster project delivery to market</td>
</tr>
</tbody>
</table>
References
Sign UP

- Sign up for PDF at front
- Sign up for semi-annual newsletter
Agile Methods training / coaching

All courses can be delivered onsite at your location!

Michael Hall, mike@threebeacons.com
214.783.3936

www.threebeacons.com

User Stories
Scrum Team Training
Agile Immersion
Extreme Programming
Distributed Agile
Seminars, Workshops
Process Diagnosis
Project “Kick-off”
Sprint Planning/Review/Retrospective

© Three Beacons LLC, 2013
As who, I want what so that why.

Mike Hall
Three Beacons, LLC
mike@threebeacons.com
214.783.3936