User Stories Done Right: Requirements

Jeff Sutherland, Ph.D.
Co-Creator of Scrum
http://jeffsutherland.com/scrum
For those of you doing Scrum

- You know who the product owner is
- There is a product backlog prioritized by business value
- The product backlog has estimates created by the team
- The team generates burndown charts and knows their velocity
- There are no project managers (or anyone else) disrupting the work of the team
Scrum of One
Scrum starts with Product Owner

- One person;
- Sets development schedule by prioritizing backlog;
- Can be influenced by committees, management, customers, sales people, but is the only person that prioritizes;
- Responsible for ensuring that the most important business value is developed first;
- This mechanism ensures that only one set of requirements drives development; and
- Eliminates confusion of multiple bosses, different opinions, and interference.
Product Backlog

- List of functionality, technology, issues
- Issues are placeholders that are later defined as work
- Emergent, prioritized, estimated
- More detail on higher priority backlog
- One list for multiple teams
- Product Owner responsible for priority
- Anyone can contribute
- Maintained and posted visibly
- Derived from Business Plan or Vision Statement, which sometimes have to be created with customer
Product Backlog

- Attributes:
  - Cost (ITD’s), Tests, Value ($, H/M/L, x/1000)
- Relationships:
  - Sprint, Feature, Specification, Task
- Average Size: Varies across time but seems to average:
  - 15 ITD’s for top 20%
  - 20 ITD’s for next 20%
  - 30+ ITD’s for last 60%
- Average size of individual item – half of Sprint length in days
- 80% of the value comes from 20% of the functionality
- 60% of the functionality delivered in successful projects is rarely or never used.
Product Owner Responsibility: 
Manage the Cone of Uncertainty

Any communication that does not recognize and communicate where a task is in the cone of uncertainty creates confusion and illusion.
## Product Owner Sprint Responsibilities

- Prepare the Product Backlog for Sprint Planning
- **Sprint Planning**
  - Present the Prioritized Product Backlog
  - Clarify business value
  - Work with team to load Product Backlog into Sprint
  - Help team with commitment to deliver
- **Sprint - support the team during a Sprint**
- **Sprint Review**
  - Assess whether Product Backlog is DONE
  - Work with team to define next Sprint goal
  - Reprioritize the Product Backlog

© Jeff Sutherland 1993-2007
Product Owner – Prepare for Sprint Planning

- Clarify and prioritize market requirements by business value
- Break high level requirements into chunks that are estimable and testable
- Have development team estimate chunks using planning poker and determine how much of the Product Backlog can go into the next Sprint
- Determine the right amount of estimated Product Backlog for the next Sprint Planning meeting
- Update the Product Roadmap
1. Template
“As a <type of user>, I want to <feature> so that <benefit>.”

2. Tests

3. Conversation

User story slides are courtesy of Mike Cohn, 2005

© Jeff Sutherland 1993-2007
INVEST in your backlog

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

Thanks to Bill Wake for the acronym. See www.xp123.com.
Independent

Avoid introducing dependencies
- Leads to difficulty prioritizing and planning

- The first of these stories will take 3 days to develop
  - It doesn’t matter which is first
  - The others will take 1 day
Making stories independent

Combine the stories
- A customer can pay with a credit card.

Split across a different dimension
- A customer can pay with one type of credit card.
- A customer can pay with two other types of credit cards.

Write two estimates and move on
- 3 days if first; 1 otherwise
Negotiable

- Stories are not
  - Written contracts
  - Requirements the software must fulfill
- Do not need to include all details
- Too many details give the impressions of
  - false precision or completeness
  - that there’s no need to talk further
- Need some flexibility so that we can adjust how much of the story gets implemented
  - If the card is contract then it needs to be estimated like a contract
A company can pay for a job posting with a credit card.

Note: Accept Visa, MasterCard, and American Express. Consider Discover. On purchases over $100, ask for card ID number from back of card. The system can tell what type of card it is from the first two digits of the card number. The system can store a card number for future use. Collect the expiration month and date of the card.
How about this one?

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

Note: Will we accept Discover cards?
Note for UI: Don’t have a field for card type (it can be derived from first two digits on the card).
Valuable

Stories must be valuable to either:

- A user can search for a job by title and salary range.

- Throughout the project, the development team will produce documentation suitable for an ISO 9001 audit.
- The development team will produce the software in accordance with CMM level 3.
- All configuration information is read from a central location.
Stories valued by developers

- Should be rewritten to show the benefit

- All connections to the database are through a connection pool.

- Up to 50 users should be able to use the application with a five-user database license.

- All error handling and logging is done through a set of common classes.

- All errors are presented to the user and logged in a consistent manner.
Embed the user in the backlog item

- Many projects mistakenly assume there’s only one user:
  - “The user”
- Write all stories from one user’s perspective
- Assume all users have the same goals
- Leads to missing features

“As a <type of user>, I want to <feature> so that <benefit>.”
Travel Site—Who’s the user?

- **Mary**: Frequent flier who never knows where she’ll be.
- **Laura**: Wants to schedule her family’s annual vacation.
- **Jim**: Frequent flier who flies every week but always to the same place.
- **Howard**: Mary’s assistant; books her reservations.
- **Dominic**: Hotel chain Vice President; wants to monitor reservations.
User roles

- Broaden the scope from looking at one user
- Allows users to vary by
  - What they use the software for
  - How they use the software
  - Background
  - Familiarity with the software / computers
- Used extensively in usage-centered design
- Definition
  - A user role is a collection of defining attributes that characterize a population of users and their intended interactions with the system.

Common attributes

- **Mary**: Frequent flier who never knows where she’ll be
- **Jim**: Frequent flier who flies every week but always to the same place
- **Laura**: Wants to schedule her family’s annual vacation
- **Howard**: Mary’s assistant; books her reservations
- **Frequent Flier Scheduler**: Infrequent Vacation Planner
- **Dominic**: Hotel chain Vice President; wants to monitor reservations
- **Repeat Traveler**: Insider
Estimable

- Because stories are used in planning
- A story may not be estimatable if:

  - Developers lack domain knowledge
    - New users are given a diabetic screening.
  - Developers lack technical knowledge
    - A user can select to see all text on the site in a larger font.
  - The story is too big
    - A user can find a job.
Small

- Large stories (epics) are
  - hard to estimate
  - hard to plan
    - They don’t fit well into single sprints
- Compound story
  - An epic that comprises multiple shorter stories
- Complex story
  - A story that is inherently large and cannot easily be disaggregated into constituent stories
Compound stories

- Often hide a great number of assumptions

A user can post her resume.

- A resume includes separate sections for education, prior jobs, salary history, publications, etc.
- Users can mark resumes as inactive
- Users can have multiple resumes
- Users can edit resumes
- Users can delete resumes
Splitting a compound story

Split along operational boundaries (CRUD)

- A user can create resumes, which include education, prior jobs, salary history, publications, presentations, community service, and an objective.
- A user can edit a resume.
- A user can delete a resume.
- A user can have multiple resumes.
- A user can activate and inactivate resumes.
Splitting a compound story, cont.

A user can add and edit educational information on a resume.
A user can add and edit prior jobs on a resume.
A user can add and edit salary history on a resume.
A user can delete a resume.
A user can have multiple resumes.
A user can activate and inactivate resumes.
Testable

- Tests demonstrate that a story meets the product owner’s expectations
- Strive for 90+% automation

A user must find the software easy to use.

A novice user is able to complete common workflows without training.

A user must never have to wait long for a screen to appear.

New screens appear within 2 seconds in 95% of all cases.
Stocking the backlog

1) Write user stories that will be the initial product backlog for your online dating site.

Tip: Try this template
“As a <role>, I want to <action> so that <benefit>.”
Product Owner Gets Product Backlog Ready

- Vision
- Roadmap
- Product Backlog
- Prioritize
- Estimate
- Size for Sprints
- Agile Specification
- Product Backlog “ready”
Product Owner Presents Highest Priority Features
ScrumMaster and Product Owner at Work
Team Breaks Features into Smaller “Stories”
Delivery Team Gives Story Estimates
Team Moves Stories Across Sprints
Team Executes
What makes Scrum work

Self-organization

Japanese view: Self-organization is the process of creating explicit knowledge from tacit knowledge!


Questions?
Bibliography