

User Stories for Requirements Elicitation

by Nick Naumovich, nick@naumovich.com

Plano, TX h. 972-398-8501 / c. 214-650-8501

This article was originally written on Mach 20, 2007 to help a team transition into Agile and solicit assistance and involvement from the stakeholders in a positive way.

Since our team is new to an Agile development methodology a primer on how it works is in order. In an Agile environment a User Story is a written tool used in the requirements gathering process to describe the specification of a software feature. Typically, a user story is brief as it consists of only one or two sentences.

It is used extensively in the Agile iterative software design process. User stories are central to requirements creation in the Agile sub-disciplines of Extreme Programming (XP) and SCRUM, but user stories are useful as a tool to elicit involvement of the business customer in many different design methodologies.

When we are using these Agile methodologies we will see that they are based on the values of simplicity, communication, feedback, and courage. It works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are and to tune the practices to their unique situation. Central to this discipline is the concept of user stories. By understanding them and communicating these concepts to our business users we have a better chance of delivering product that these business users will like and be both on time and within budget.

What are User Stories?

User Stories and their associated acceptance tests specify software requirements. They are written as things that the system under development needs to do for the business customer. It is a software system requirement formulated as one or two sentences in the everyday language of the business user. Ideally, the user stories are written by the customer and are their main method of influence on our development.

A user story is an informal statement of the requirement instead of a large requirements document. The story communicates to the design team WHAT is needed and does not specifically address anything about HOW to implement what is needed because the HOW part is strictly in the domain of the IT development team. The real intention of a user story is to provide the team with the ability respond quickly to user wants and needs. It creates less overhead in the face of rapidly changing real-world requirements or discovery of new requirements based upon the work in progress. It is not

specifically a description of a feature in a program, but the underlying real world problem that the software component is designed to solve for the business user.

A User Story contains whatever we think is necessary to jog the memories and inspirations of the development team who will later explore the story with greater depth. Throughout the process, a series of conversations will take place between the customer (see below) and the development team. These conversations captured as additional documentation that will be attached to the card along with any discovered acceptance test criteria. We combine a focus on verbal communication with automated tests to communicate requirements. The result is much lower need for written requirements within the team.

In this document I will use InterGalacticMining as our company for illustration purposes. By the way, InterGalacticMining was a central character in one of my favorite science fiction books.

If we have a business need for a specific requirements document for InterGalacticMining, the customer should request the document in the same way as a request for a feature: with a story card. The team will estimate the cost of the document, and the customer may schedule it in any iteration they wish.

Getting Customer Involvement in the Process

One of the few requirements of XP is to have the customer available to help the development team and to be a part of it as well. All phases of an XP project require communication with the customer, preferably face to face and on site.

By making the process of gathering user stories as painless as possible, we will give our business users a tool to gather user stories, and therefore requirements, internally to make sure we implement features that derive the most benefit for our design and its subsequent implementation. The business user and development team can then prioritize the stories and implement a solution within the time allotted. This level of business user involvement gives our customer a sense of confidence in the project and a strong sense of buy-in by handing them small, easily digestible stories on index cards, not a 100 page requirements document. We are asking them what they want, not telling them to sign off on what we have proscribed based upon our understanding of the business process. They, not the IT design team, are the subject area experts, and their input is vital to a successful implementation.

Creation of User Stories

User stories in XP have three components: Cards (their physical medium), Conversation (the discussion surrounding them), and Confirmation (tests that verify them).

User Stories are written down on a note card with a name and a description. If later we find the user story is lacking in some way (too large, complicated, imprecise), it will be rewritten until it is

satisfactory to all parties. However, it is stressed that user stories are not to be definite once they have been written down. Here is one simple story format that will help us to start to communicate within the framework of a user story. It is best to be mindful to not treat a story as a mini-requirement spec where we focus on the written word; rather we use stories as tools for driving a conversation at a later time.

The suggested syntax of a user story:

As a ... (role or actor) (**Who**)
I want ... (what capability or feature do they need) (**What**)
so that ... (why is it of business value or benefit) (**Why**)

When I originally wrote this article I had written the syntax in both 1st and 3rd person. I had sent a draft to Mike Cohn, *User Stories Applied, For Agile Software Development*, Addison-Wesley, 2004, questioning the use of 3rd person. His comment was to definitely use 1st person when crafting a story. He said for me to write it as if I was in the shoes of the user wanting the story. From a psychological prospective the use of 1st person makes the story more understandable. In that case, be sure to identify the role you are assuming within the story:

"As a *type of user* I want *capability or feature* so that *business value or benefit*"

The "Who" and "What" are essential to the story, but the "Why" only helps with clarity and sets up the acceptance test. Stories help you ask the right questions about the context and reason for the request from the prospective of the person requesting the feature.

Here are some examples of User Stories for the InterGalacticMining.com website:

- As a new visitor to InterGalacticMining.com I see what the latest new content has been added so I can see most current news instead of having to read through everything including items I've already encountered.
- As a returning visitor to InterGalacticMining.com I want to know what new content has been added since my last visit so that I don't miss some important to me.
- As a Sales Associate at InterGalacticMining I want to publish a new case study relevant to my existing clients so that I can promote a new process I am offering.
- As a Sales Associate at InterGalacticMining I want to change the access to a piece of materials newly published case study relevant to my existing clients so that I can promote a new process I am offering.
- As a visitor returning to InterGalacticMining.com I want to visit InterGalacticMining.com site and have it remember who I am so that I don't have to remember my password.
- As a visitor to InterGalacticMining.com I want to click a button to have InterGalacticMining quickly e-mail me my login credentials so that I don't have to remember it or have to reregister it if I cannot.

- As a registered visitor to InterGalacticMining.com I want to access information on products and services that interest me so that I can learn more about what InterGalacticMining has to offer.
- As a registered visitor to InterGalacticMining.com I want to find content that I am entitled to that is not available to the general public so that I can use this special material including research, white papers and market information.
- As a visitor to InterGalacticMining.com I want to be able to search multiple InterGalacticMining sites so that I don't have to go to each of the separate corporate sites to find content I am looking for.
- As a registered visitor to InterGalacticMining.com I want to mark content and files so that I can retrace my tracks so that I can find some information I may have seen and subsequently misplaced.
- As a visitor to InterGalacticMining.com I want a navigation marker telling me exactly where I am in the navigation tree so that I can backtrack immediately to a page where I chose this pathway so that I can quickly pick another branch to traverse.
- As a visitor to InterGalacticMining.com I want to receive InterGalacticMining product, services and support content specifically suited to my industry so that my visit to the site becomes a more personalized experience.
- As a visitor to InterGalacticMining.com I want access to context specific help to use any feature on InterGalacticMining.com that I am currently viewing so that I can understand what the site has available for me to do next.
- As a visitor to InterGalacticMining.com I want to be able to participate in demos and tutorials so that I can understand how to use RSS feeds, play pod casts and any other feature beyond simple web browsing.
- As a Content Manager at InterGalacticMining I want to load specific content and target the so that I can promote a new process I am offering.

From the above example and from a brainstorming session we have identified the following types of users of InterGalacticMining.com:

- InterGalacticMining.com Unregistered Visitor
- InterGalacticMining.com Registered Visitor
- InterGalacticMining Sales Executive
- InterGalacticMining Profile Manager
- Financial Analyst
- Job Seeker
- CXO level Executive
- VP/Sr. VP level Executive

- Business Analyst
- Business Partner
- Academic

Also, we can generate a list of features and capabilities that InterGalacticMining.com should have giving us something to estimate and test.

- Display a list of new content published on the site since their last visit.
- The ability for authorized InterGalacticMining personnel to publish specific content for visitors with particular profiles.
- Capability to e-mail the content displayed in a window to an e-mail address provided by the site visitor.
- Provide a way to retrieve a lost user names and passwords via e-mail.
- Display a navigation marker showing the current location in the navigation tree with the ability to backtrack.
- The ability to view case study content matching my profile.

Acceptance Tests

As a central part of the planning game, user stories define what we build in our project. User stories are prioritized to indicate which are most important for the system and are broken down into software engineering tasks and estimated by the development team. When we implement a user story a more formal acceptance test will need to be written to ensure that the goals of the story are fulfilled.

Acceptance tests are created from user stories. During an iteration the user stories selected during the iteration planning meeting will be translated into acceptance tests. A story can have one or many acceptance tests, whatever it takes to ensure the functionality works 100%.

Each acceptance test represents some expected result from the system. We must verify the correctness of the acceptance tests and reviewing test scores to decide which failed tests are of highest priority. A user story is not considered complete until it has passed its acceptance tests.

Things to Keep in Mind When Discussing Stories

As we discuss stories, write cards, and split stories, the INVEST acronym helps remind us of characteristics of good stories. When we break out individual tasks in a task plan, applying the SMART acronym can improve the tasks generated from them.

Characteristics of Good Stories – remember “INVEST”:

- I - Independent
- N - Negotiable
- V - Valuable
- E - Estimable
- S - Small
- T - Testable

Independent

Independent stories are the easiest to work with. If the concepts contained in them overlap it makes it hard to implement them in any order.

Negotiable

A story is not an explicit contract for a feature. It represents what will be created by the customer and programmer during development. A good story captures the essential idea and not specific details.

Valuable

A story needs to be valuable. We don't care about value to just anybody; it needs to be valuable to the customer. Developers may have (legitimate) concerns, but these framed in a way that makes the customer perceive them as important.

Estimable

A good story can be estimated. We don't need an exact estimate, but just enough to help the customer rank and schedule the story's implementation. Being estimable is partly a function of being negotiated, as it's hard to estimate a story we don't understand. It is also a function of size: bigger stories are harder to estimate. Finally, it's a function of the team: what's easy to estimate will vary depending on our team's experience. (Sometimes our team may have to split a story into a (time-boxed) "spike" that will give the team enough information to make a decent estimate, and the rest of the story that will actually implement the desired feature.)

Small

Good stories tend to be small. The best stories represent at most a few person-days worth of work. The details can be elaborated through conversations with the customer.

Testable

A good story is testable. If you can say: "I understand what I want well enough that I could write a test for it", then it is testable and probably is a good requirement. By writing the tests early helps us know whether the goal of the story is met. Not knowing how to test something indicates that the story may not be clear enough, or that it doesn't have true value for the customer. Performance and

usability are things that need to be tested. The feedback cycle of proposing, estimating, and implementing stories will help teach the team what it needs to know.

Characteristics of effective task development – remember “SMART”

S - Specific

M - Measurable

A - Achievable

R - Relevant

T - Time-boxed

Specific

A task needs to be specific enough that everyone can understand what's involved in it. This helps keep other tasks from overlapping, and helps people understand whether the tasks add up to the full story.

Measurable

The task should be able to be completed and marked as done. The team needs to agree upon exactly what “Done” means, but it must at least include doing what it was intended to and be accompanied by the tests.

Achievable

The owner of the task should believe that they can achieve the task. XP teams have a rule that anybody can ask for help whenever they need it; this certainly includes ensuring that task owners are up to the job.

Relevant

Every task should be relevant, contributing to the story at hand. Stories are broken into tasks for the benefit of developers, but a customer should still be able to expect that every task can be explained and justified.

Time-Boxed

A task should be time-boxed: limited to a specific duration. This doesn't need to be a formal estimate in hours or days, but there should be an expectation so people know when they should seek help. If a task is harder than expected, the team needs to know it must split the task, change players, or do something to help the task (and story) get done.