Scrum Alliance® (/)

Search   https://www.scrumalliance/community/articles/2013/sept...

JOIN » (MEMBERSHIPS/JOIN)   **LOGIN » (/LOGIN)**

Why Scrum? (/why-scrum) | Certifications (/certifications) | Courses & Events (/courses-events) | Community (/community) | Memberships (/memberships)

Careers (/careers)

Community

Member Directory (/community/member-directory)

Expert Blog: Spotlight (/community/spotlight)

Member Articles (/community/articles)

User Groups (/community/user-groups)

  SCRUM USER GROUP® Guiding Principles (/community/user-groups/user-group-guiding-principles)

  SCRUM USER GROUP® Logo Posting Guidelines (/community/user-groups/scrum-user-group-logo-posting-guidelines)

Newsletters (/community/newsletters)

Sponsorship Application (/community/sponsorship-application)

About Us (/About-Us)

Contact Us (/Contact-Us)

FAQs (/FAQs)

Home (/) » Community (/community) » Member Articles (/community/articles) » 2013 (/community/articles/2013) » September (/community/articles/2013/september) » Agile User Stories

Agile User Stories

The Building Blocks for Software Project Development Success

20 September 2013

Venkata Gayatri Krishnamurty Pammi (/community/profile/vpammi/)
Ivy Comptech Limited

Introduction

If you don't get the user needs right, it doesn't matter how well you execute the rest of the project. Requirements are the foundation for all the project work that follows. The purpose of a software development project is to build a product that provides value to a particular set of customers. Thus customer involvement is the most critical contributor in eliciting requirements.

Let us go through a famous survey on this topic. The Standish Group surveyed IT executive managers in 1995 for their opinions about why projects succeed. The three major reasons that a project will succeed are user involvement (15.9%), executive management support (13.9%), and a clear statement of requirements (13%). Opinions about why projects are impaired and ultimately canceled included incomplete requirements (13.1%) and lack of user involvement (12.4%) at the top of the list. In his now classic ROI presentation at XP 2002, Jim Johnson shared research regarding the productivity achieved by those software projects that do deliver something. Among these, he found that 45% of product features are never used, 19% are rarely used, and 36% are routinely used. This tells us that traditional requirement engineering practices are not helping us deliver value to our customer through our traditional practices.

In my experience, traditional requirement engineering practices applied through Waterfall models have numerous limitations, including the following:

1. **Why?** It is difficult to correlate each requirement with the corresponding end business value it maps to. Thus it is becoming difficult for the development team to understand the business reason behind each of the specified requirements. This also leads to ineffective prioritization and thus impacts ROI-based decisions.

2. **Who?** It is difficult to understand the mapping between each requirement and the person who uses it. It creates a huge gap between users and the developers.

3. **What?** It is difficult to keep requirement documents current and synchronized, especially when the project involves multi-site and multi-group involvement.

4. **Users' perspectives or business analyst's interpretations:** Often, we see standard requirement specifications leaning toward the business analyst's interpretations rather than toward clear capturing of user perspectives.

5. **De-focus:** Lengthy requirements descriptions may derail developers trying to understand the exact intent behind the requirement specification.

6. **Changes happen:** It is inevitable that requirements will change. Business needs evolve, new users or markets are identified, business rules and government regulations are revised, and operating systems change over time. When changes happen to overall business goals, it is difficult to reflect those changes across the associated requirement specification documents pertaining to various products.

7. Exact **user acceptance criteria** are often not clearly known to developers when they build the features. This often leads defective products.

8. Even the best requirements documents cannot -- and should not -- replace **human dialogue**. We often see limited interactions with users and more focus toward documents and processes. This leads to defective product feature implementation.

9. **Validation of requirements** does not often take place against business ROI and user satisfaction. This leads to less usage of product features and less profit.

10. **Periodic feedback** is not enabled between users and developers in the Waterfall development model till user acceptance testing. Thus it incurs huge costs to incorporate user feedback as a result of user acceptance testing.

(/Community/User-Groups)

What happens in Agile?

Kent Beck coined the term "user stories" in Extreme Programming in 1999. A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. The essence of user-centric and usage requirements elicitation is to focus on what the user wants to have, not what the user wants the system to do.

Agile projects, especially Scrum ones, use a product backlog, which is a prioritized list of the user stories to be developed in a product or service. Let us learn these basics in detail.

The user story is accompanied by acceptance criteria

A user story typically follows a simple template:

As a <type of user>,
I want <to perform some task>
so that I can <achieve some goal/benefit/value>.

User stories need to be confirmed with the 3C guidelines below, suggested by Ron Jeffries:

1. **Card:** User stories are traditionally written on index cards or sticky notes, in short form. User narratives further explain these cards. Thus the main intention is to describe the user story in short form to allow common understanding of the user

2. **Conversation:** User stories shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

3. **Confirmation:** Acceptance tests confirm that the story was delivered correctly.

Let us take an example.

User story title: Customer withdraws cash.

As a customer,
I want to withdraw cash from an ATM
So that I don't have to wait in line at the bank.

Acceptance Criterion 1:

Given that the account is creditworthy
  And the card is valid
  And the dispenser contains cash,
When the customer requests the cash
Then ensure the account is debited
  And ensure cash is dispensed
  And ensure the card is returned.

Acceptance Criterion 2:

Given that the account is overdrawn
  And the card is valid,
When the customer requests the cash
Then ensure the rejection message is displayed
  And ensure cash is not dispensed.

Product backlogs and epics

User stories are written throughout the Agile project. Usually a story-writing workshop is held near the beginning. Everyone on the team participates, with the goal of creating a product backlog that fully describes the functionality to be added over the course of the project or a three- to six-month release cycle within it.

Some of these Agile user stories will undoubtedly be epics. Epics are large user stories and will later be decomposed into smaller stories that fit more readily into a single iteration. Additionally, new stories can be written and added to the product backlog at any time and by anyone.

While a product backlog can be thought of as a replacement for the requirements document of a traditional project, it is important to remember that the written part of an Agile user story ("As a user, I want . . . ") is incomplete until the discussions about that story occur.

Anyone can write user stories. It's the product owner's responsibility to make sure a product backlog of Agile user stories exists, but that doesn't mean that the product owner is the one who writes them. Over the course of a good Agile project, you should expect to have user story examples written by each team member. Also, note that who writes a user story is far less important than who is involved in the discussion of it.

Agile helps us get the requirement right

1. **Why?** The user story clearly describes the goal or benefit the user is trying to achieve through it. Thus it is easy to correlate each user story with the corresponding end business value it maps to. Using KANO and MoSCoW prioritization techniques, the product owner can optimize the product's ROI.

2. **Who?** The user story starts with the persona who is needed in this functionality. Thus, it connects users who consume the capability with the developers.

3. **What?** In Agile, one product will have one product backlog. The product backlog can be subdivided into product area backlogs for the respective multiple groups. In this format, user stories can be grouped under epics so that we can establish a common understanding about the user cases across the product backlog.

4. **Users' perspectives:** User stories are written to capture user perspectives only. We also capture user acceptance criteria up front for the related user story. This indeed offers great help to the development team.

5. **Focused:** User stories are short forms of the user's needs. We capture all the related information in user narratives. This helps the development team understand the exact intent of the user need.

6. **Changes happen:** User stories recognize the importance of change. User stories absorb change through the means below.
   - If the scope of the user story becomes large, they can be split into smaller user stories
   - By adding "conditions" to acceptance criterion

7. Exact **user acceptance criteria** are clearly known to developers up front. This leads to better connection between users and developers and thus facilitates a collaborative, value-driven path.

8. User stories using the 3C guidelines are driven by **human dialogue**. Please remember that the written part of an Agile user story ("As a user, I want . . . ") is incomplete until the discussions about that story occur.

9. **Validation of user needs** is clearly documented through acceptance criteria, and the same is captured for each user story. This ultimately leads to user satisfaction, as it is communicated early to the development community.

10. **Periodic feedback** is enabled through sprint reviews in the Scrum framework. Thus, users are given working software well up front and their feedback is periodically incorporated into the product iteratively.

Conclusion

Agile projects are successful three times more often than non-Agile projects, according to the 2011 CHAOS report from the Standish Group. The report goes so far as to say, "The Agile process is the universal remedy for software development project failure. Software applications developed through the Agile process have three times the success rate of the traditional Waterfall method and a much lower percentage of time and cost overruns" (page 25). In this process, Agile user stories certainly act as building blocks for software project development success.

References

http://www.mountaingoatsoftware.com/blog (http://www.mountaingoatsoftware.com/blog)

http://agileproductdesign.com (http://agileproductdesign.com/)

Cohn, Mike. *User Stories Applied: For Agile Software Development.* Addison-Wesley Professional. 2004.

Wiegers, Karl E. *Software Requirements.* Microsoft Press. 2003.

Tags:

Article Rating

Current rating: 3.8 (14 ratings)

Comments

Emile Bakker (/community/profile/ebakker/), CSPO, 9/20/2013 12:31:47 PM

I am sorry, but I have to disagree with nice write up completely. In my humble opinion you are trying to unleash the waterfall documentation need on scrum. Yes, stories are used to create clarity about what needs to be developed. But you are trying to get the Product Owner out of the equation during the sprint. I can imagine that you see it as a goal to keep the shortest possible grooming sessions and sprint planning by specifying as much as possible. This kills scrum, communication is vital. Not only because you transfer knowledge but also because it is an handshake of understanding. If a story does not trigger a question, I have failed to write a good story. I sense that you try to reach the opposite.
The 13.1% projects that are canceled based upon incomplete requirements should be translated towards insufficient understanding of what needs to be build. So lack of presents of the product owner.

A product backlog is not a replacement of a requirements document. A requirements document does an attempt to be complete (and create loads of waste that way) while a backlog is enough to fuel your next few sprints and is constantly changing.

With regards to the examples of acceptance criteria there are also a few remarks.
"Given that the account is creditworthy": should be part of the story.
"And the card is valid": Separate story. What to do with invalid cards requires a story.

"When the customer requests the cash.Then ensure the account is debited" This is a total redundant statement. What does the word withdraw mean?

"And ensure cash is dispensed" Huh?
"And ensure the card is returned Huh?

The risk of writing your stories like this is that you minimize the involvement of the team, if you write stupid things you get stupid functionality, but no stupid questions. These kind of acceptance criteria are an insult for a well educated developer.

My advice.. spend less time on defining what a story should look like and spent more time on involving the PO with the team.

On my blog I have written in the past an item about acceptance criteria: http://agileowners.wordpress.com/2012/11/21/un-acceptance-criteria/

Venkata Gayatri Krishnamurty Pammi (/community/profile/vpammi/), CSP,CSM, 9/21/2013 12:17:59 AM

Thank you Emile for your comments. The intent of this article is three folds

1) What are the challenges we face with Traditional Requirement engineering model and related practices
2) How Agile User stories address them

In my article, I highlighted 3C practice suggested by Ron Jeffries:
Card: User stories are traditionally written on index cards or sticky notes, in short form. User narratives further explain these cards. Thus the main intention is to describe the user story in short form to allow common understanding of the user need among all stakeholders.

B.Conversation: User stories shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

Confirmation: Acceptance tests confirm that the story was delivered correctly.

I also mentioned that "discussions are more important" than textual user story.

In my experience, In Agile, we maintain Product backlog as opposed to traditional Requirement specification document.

I am happy to discuss with you if you want any further clarity.

Emile Bakker (/community/profile/ebakker/), CSPO, 9/21/2013 2:40:27 AM
I get your point. And shopping through your reply I fully agree on this: "In fact, these discussions are more important than whatever text is written."
So if that is the case focus on the involvement of the PO instead on formalizing stories. I have seen so many discussions about what should or should not be in a story, while the result of a sprint determines if the stories and the conversations have been sufficient. And luckily Scrum allows you to determine that per sprint.
But in my experience of many years being a PO, it really comes down to communication.

Kim Poo Gan (/community/profile/kgan18/), CSM, 11/20/2013 9:02:56 PM
Great article.

Bharath Suryakanthan (/community/profile/bsuryakant/), CSM, 3/26/2014 12:12:16 PM
The comments on this post actually complete the post. However, I notice that you are adding Behavior Driven Development to contemporary User Stories.

That said, we ought to keep in mind that Agile is a WE exercise rather than I; the whole team needs to buy-in to the story before it is accepted as a valid story point.

@Emile: When you say 'communication', what is the medium of communication? How do you ensure that what you sought out to be achieved is eventually achieved?

You must Login (/login?returnurl=~/community/articles/2013/september/agile-user-stories) or Signup (http://www.scrumalliance.org/signup) to comment.

Find a course » (/Courses-Events /Courses)

Certified ScrumMaster (/Courses-Events /Course?type=Csm;)

Certified Scrum Product Owner (/Courses-Events /Course?type=Cspo;)

Certified Scrum Developer (/Courses-Events /Course?type=Csd;)

Extended Education (/Courses-Events /Course?type=Extended%20Education;)

Find Education Providers » (/Community /Member-Directory?cst=true& csc=true& rep=true)

Certified Scrum Trainer (/Community/Member-Directory?cst=true)

Certified Scrum Coach (/Community/Member-Directory?csc=true)

Registered Education Provider (/Community /Member-Directory?rep=true)

Global Scrum Gatherings » (/Courses-Events /Global-Gatherings)

Berlin: 2014 (/courses-events/events/global-gatherings /2014/berlin-2014)

Phoenix: 2015 (/courses-events/events/global-gatherings /2015/phoenix-2015)

Prague: 2015 (/courses-events/events/global-gatherings /2015/prague-2015)

Find a User Group »

Select Region

GO »

(/scrum/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2013-State-of-Scrum-Report_062713_final.pdf)

Transforming the world of work

Why Scrum? (/why-scrum)   Certifications (/certifications)

Courses & Events (/courses-events)   Community (/community)

Memberships (/memberships)   Careers (/careers)

Feedback (/Feedback)

Contact Us (http://scrumalliance.zendesk.com/account /dropboxes/12016)

Legal (/legal)   Privacy Policy (/Privacy-Policy)

Sitemap (/sitemap)

Help (http://scrumalliance.zendesk.com/account/dropboxes /12016)