

How do you develop a Shared Understanding on an Agile project?

Exploring common approaches and their adaptations from real-world projects

Share this ebook.



Having a shared understanding of customer requirements is central to all aspects of Agile project development. User stories play a big part in cultivating a shared understanding. Thus getting your stories right is important.



story (n)

A story is a concise description of a customer's requirement on a card, used as a unit of work on an Agile project that can be tracked. It serves as a placeholder for a collaborative conversation between the customer and the team. Its completeness is verified by confirmation with the customer's criteria.



So...how do you write the perfect story?



There is no such way.

That being said, let's explore some common ways of thinking about stories and how you can use those to create stories that best suit your team.

INVEST in a
good story

The INVEST guideline for story writing

Independent

Negotiable

Valuable

Estimable

Small

Testable



INVEST in a
good story

Independent

Negotiable

Valuable

Estimable

Small

Testable

Pay by Visa

Pay by MasterCard

Pay by AmEx

Pay by Credit Card

- ✓ Do not overlap your stories in concept.
- ✓ When sequencing the stories, try to find their natural order.



The order of stories should not restrict your customer's ability to re-prioritize or move stories out of scope.

INVEST in a good story


Independent
Negotiable

Valuable

Estimable


Small

Testable



As a purchaser, I want the receipt to display the date of my purchase in ISO 8601 format Comic Sans 12pt font with 9pt leading, so that I can maintain my records.

As a purchaser, I want the receipt to indicate when I completed the purchase, so that I can maintain my records.



- ✓ Stories are negotiable...and negotiated.
- ✓ Remember, your story is the essence of the requirement and not an explicit contract.
- ✓ Sign off stories with working software.



Avoid signing off written stories before they are played, as it creates contractual documents that shift the focus to documentation.

INVEST in a good story


Independent
Negotiable


Valuable

Estimable

Small

Testable

 As a developer/DBA, I want a new table in the Orders DB to capture shipping information, so that ???

As a customer, I want to be able to specify my preferred shipping details, so that I can ship to an address other than my billing address 

- ✓ Your stories need to be valuable to and understandable by your customer.
- ✓ They need to be framed from your customer's perspective.

If it is difficult to write the "so that...." part easily, you might want to consider the story's value and purpose.

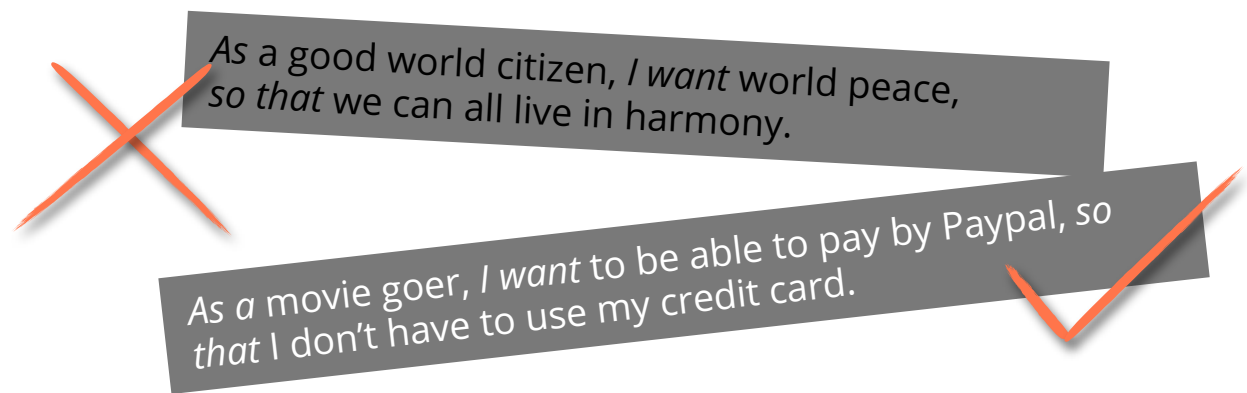


Avoid layer-based development. Instead choose vertical slices of functionality.

Technical debt are not user stories.

INVEST in a good story

Independent
Negotiable
Valuable
Estimable
Small
Testable




- ✓ Your stories should have boundaries so you know when you are “done” and what is required to be “done”.
- ✓ Your stories should be digestible by the team so they can size them.
- ✓ Keep your stories understandable and of consistent granularity.
- ✓ “Spike” stories that your team has difficulty understanding.




Avoid “catch-all” stories with uncertain estimates.
Don’t get bogged with precision and detail.

INVEST in a good story

Independent
Negotiable
Valuable
Estimable
Small
Testable

 As a movie goer, I want to be able to find and purchase movie tickets online, so that I have something to do tonight.

As a movie goer, I want to be able to find a movie by title, so that I can quickly locate the details of a movie I am interested in. 

- ✓ Keep your stories small enough to be measured and tracked.
- ✓ Keep your story descriptions short and concise.



Stories should be measured in days *not* weeks.

INVEST in a
good story

Independent
Negotiable
Valuable
Estimable
Small
Testable

Improve readability

All acronyms replaced with complete terminology

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

New screens appear within 2 seconds 95% of time

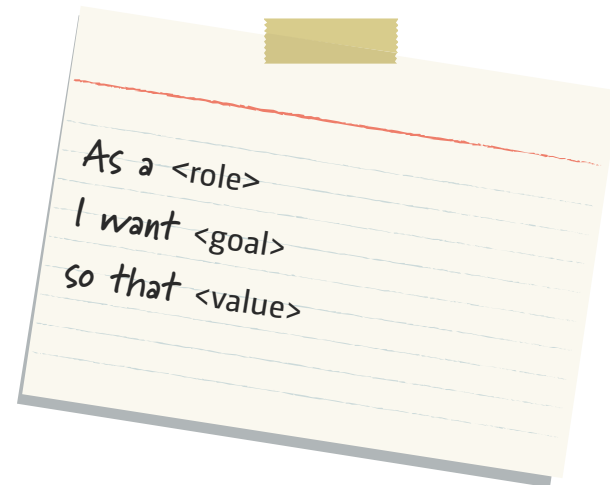
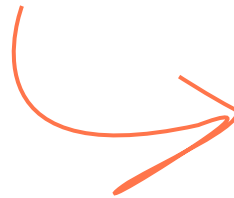
- ✓ To know when your story is done, it needs to be testable.
- ✓ Define acceptance criteria that are clear and precise so you know when you are done and have delivered value.



First define your tests and then develop.

Try a format that works for you

User focused



- ✓ It is written in the first person.
- ✓ Keeps the value in the "so that".

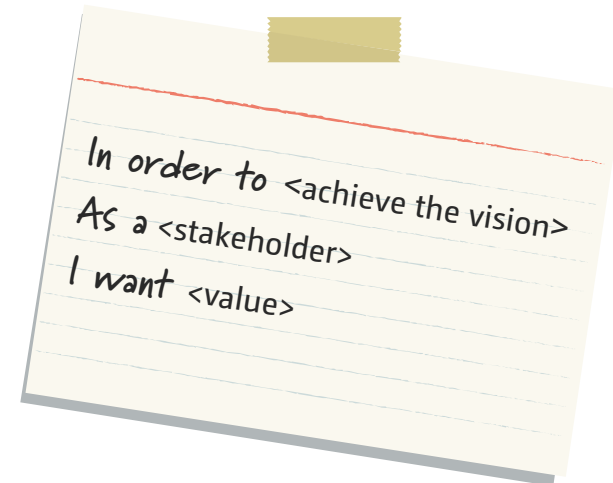
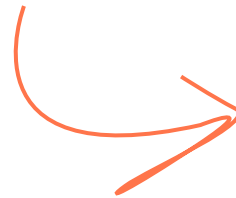


Do not skip the "so that" part, that is very important.

As value is not always clear, this format is hard for some teams.

Try a format that works for you

Feature Injection



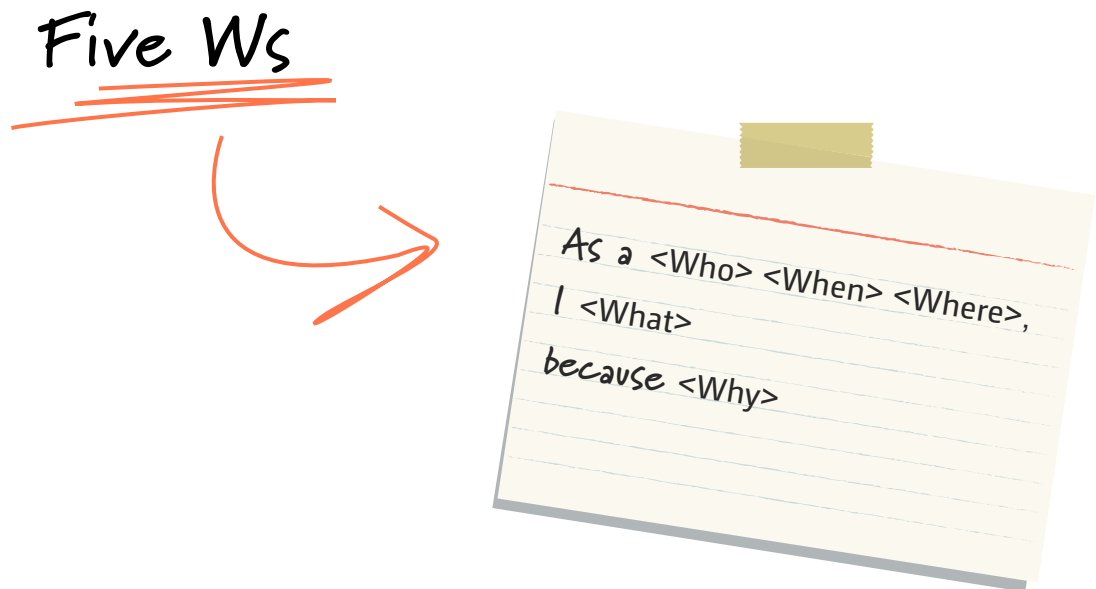
- ✓ Helps you see the big picture by putting the vision first.
- ✓ Focuses on the user and the why.
- ✓ Shows that features are necessary to realize the vision, not the other way.



Don't focus on the wording. What's important is to find a way of keeping the business value in the conversation.

When talking about the capabilities, focus on the user.

Try a format that works for you



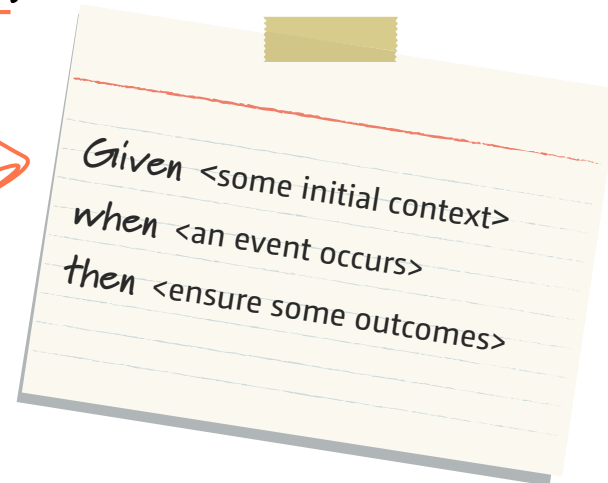
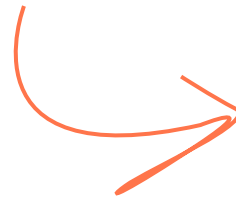
- ✓ The first person is in the “who”.
- ✓ “When” and “where” help with context and scenario setting.
- ✓ The value is in the “why”.



The “what” can drive to solutions too soon.

Try a format that works for you

Given, when, then



- ✓ Probably the most common format that is used to break down a story into acceptance criteria.
- ✓ Can be used as scenarios to break down a larger story with specifics.
- ✓ Good for handling uncertain and unclear scenarios.
- ✓ Can help with BDD (Business Driven Design) and automation.



Avoid stating the obvious and creating too many scenarios.

How do we adapt these patterns to suit our needs?

Let's see how some ThoughtWorkers did it.



“Shu Ha Ri”



Jeffrey, **Principal Consultant, BA**

I once started a debate on how to write a User Story. The most popular style by far is “As a <role>, I want <functionality> in order to <achieve a certain value>.” I am not a big fan of this style because I find most teams either write poor value statements or don’t write them at all. There is power in understanding the value and agile teams are missing this when the user stories are short changed. I find it’s easier to write a user story when we start with value: *In order to <achieve some value>, as a <role> I want <something>*. A part of Feature Injection, this puts more importance on the value portion of the story. This means we talk about the good stuff more and it even helps reduce scope creep.

The problem I had with the debate was not the answers though. The problem was when other survey respondents said, “I just do whatever I think will work best.” Argh!

Losing Mastery

Our best chance for success comes from knowing how to use our tools. My problem with picking “any old style” is its result: failing to spend time mastering any style. We are constantly jumping to the next big thing.

Unfortunately, the deck is stacked against us because the rate of change is accelerating. Technology and our professions are running at an ever-faster pace. Fast enough that we’re losing even the chance to focus.

No masters. Skills, entire professions, especially in tech, now run a 100-year life cycle in a decade or less. No one gains the wisdom of years. - John McWade, [The Vanishing Master](#)

The nature of learning and mastery is the need to take time and practice. You cannot get good at something without making the effort to practice over time, deliberately trying to get better. For example, I used to interview a lot of candidates who often had a title like “Senior Analyst” or “DBA III.” But these candidates had not learned enough to perform the basics of their job, they just stayed in the position long enough to be promoted. Without practice, the years spent in the role had not translated into growth or expertise, just time. Time spent practicing, however, can lead to big changes. Watch the beginning of the [TED Talk by Benjamin Zander](#). He starts with visual and auditory demonstration of how practice leads to huge impacts over the course of a few years.

(continued)

"Shu Ha Ri"



Jeffrey, **Principal Consultant, BA**

(continued)

Gaining Mastery

I was recently coaching a very process bound client. Multiple times a week I would get asked, "What's the best practice for this?" I try to tell them, "It depends," but they don't hear this message very well. They are still beginners and they recognize this. They want one path to follow because the freedom of Agile is troubling. This isn't a bad thing. In fact, I think their self-awareness of wanting a "best way" is probably a good thing. They are deep in a learning phase and it makes sense they study and practice one thing until they get it right.

Supporting me, is the Shu Ha Ri model. Here's how Gojko Adzic described it in Bridging the Communication Gap:

Shu-ha-ri is a learning model associated with Aikido. It roughly translates to "obey-detach-leave." At the first level (Shu - "obey"), a student learns by closely following one model. At the second level (Ha - "detach"), the student learns that there are multiple models and solutions. At the third level (Ri - "leave"), the student goes beyond following models.

Forced Constraints

- As I read about people who are really good at their craft, I read about how current masters are rediscovering the joy of going back to Shu - "obey." It's striking to me how often I read an article about how people who are excelling are discovering their creativity shines brighter when they find themselves stuck within constraints. They achieve more within the restrictions than they were able to outside of them!

My Conclusion

I am afraid we are reaching outside of the constraints stories give us before we have really mastered them. My fear is we are taking the easy way out. Reaching for the next thing or what's easiest. Missing the journey leading us to be masters of our craft. Maybe it's just in my head, but I know too many Business Analysts and Product owners who focus on writing quickly to get it done, not thinking about what's right.

However, regardless of which format you use, we must be mindful throughout the writing journey to assure we produce and deliver value, and with reason.

"None of the above"



Melissa, BA

I've written stories with both the traditional Agile (ala Mike Cohn) and Feature Injection formats and find both unsatisfactory. Something just doesn't sit well when I'm trying to fit the essence of a story into a prefab form. If I have to follow a format, I spend more time trying to conform to the format than distilling the valuable work that needs to get done.

The rigidity of formats can compromise the ability to communicate value and purpose; they make the story writing process feel unnatural, if not forced. So, I find myself moving toward "free form" story writing.

This all said -- and here I go contradicting myself -- *if* I find the structure of traditional formats helpful in distilling the value of a story, I'll use them, as in the following example:

As a *scrum master* ***I want*** *to create a card by dragging the magic card onto a cell in the grid view* ***so that*** *I can intuitively specify where I want the newly created card to appear on the grid view.*

I simply don't want to have to use them. Teams should be able to work the way want to. Whatever form works best for the story, the writer and the team should be the form that is use.

One of our customers recently asked a similar question about how to achieve consistency across user stories. My response: "I would encourage BAs to write stories that capture value in a manner that is best suited to the team and the project. Forcing a structure or format could inhibit teams from effectively capturing value or create more needless work (i.e. waste) by making it more difficult to. As long as the story captures value and the team understands why you're playing it, the format doesn't really matter."

(continued)

"None of the above"



Melissa, BA

(continued)

There has been a lot of debate around the best way to write user stories. Regardless of your stance, it's important to recognize the exercise traditional story formats provide: focusing our attention on value and our efforts on answering the question, "Why?" Why are we investing in this work? Why will it provide value to the customer? (You might have other "why" questions to answer, but these are a few to keep in the forefront of your mind.)

My point, in short:

- Traditional formats can be great for training purposes (and could be beneficial for multi-lingual distributed teams). But once you learn the rules, you should have the right to break them. Focus on fluidity and getting real.
- Stories—regardless of what format they're written in—should spark conversations that result in a shared understanding among the team and help continuously deliver value to the customer.

“So that...so what?”



JK, PM, BA, Agile coach

One of my biggest frustrations on a project is when we have stories that capture a solution rather than a goal. Often this leads to long conversations with the business to drive out exactly what it is that should be built and hinders the creativity of the development team in finding the best solution for the problem.

A few of us got talking while doing an inception that the ‘so that’ statement was often either repeating the requirement, was too general or was left off completely when capturing stories. This meant that we were missing the true goal of the business, leading to problems with scope and misunderstandings of what a story truly meant once we got into delivery. Let us examine some headlines:

- *As a viewer I want to change channels on the TV with my remote so that the I can use the remote to change channels.* Such headlines repeat the requirement as the business value and would be difficult to prioritize because we don't understand the benefit to the user.
- *As a viewer I want to change channels on the TV with my remote so that I can watch different channels.* Such vague headlines cause debate over alternative solutions that may or may not solve the problem. Is using the remote really necessary to watch different channels?

- In really bad cases you end up with no business value at all which are both difficult to identify alternative solutions for and to prioritize - *As a viewer I want to change channels on the TV with my remote control*
- Of course there are always the times when a headline captures the true value of what the business is trying to achieve - *As a viewer I want to change channels on the TV with my remote control so that I can watch different channels without needing to be in reach of the television.* This story can be easily prioritized against other stories and alternative solutions identified, but the headline is too long winded to easily grasp what we are aiming to achieve.

(continued)

"So that...so what?"



JK, PM, BA, Agile coach

(continued)

So, how do we ensure that the true goal of the business is the most important thing captured in a headline?

This isn't a silver bullet, but a potential solution that I've now used successfully for almost a year is:

As a (role) I want (business value) by (method/requirement)

This forces the writer to capture the value and be specific with what the story is aiming to achieve. At first I found it quite challenging, but soon realized that the difficulty wasn't because of the format, but that it was forcing me to have headlines which clearly stated the goal the business was trying to achieve. I wasn't allowed to wimp out! The feedback that I have received from the developers on the team has been very positive and it has helped to ensure a common understanding with the business.

Our remote control example now reads - *As a viewer I want to change channels without being in reach of the tv by using my remote control.* This states the same benefit to the viewer and solution as the 'good' headline above, but in a much clearer and natural statement. We can prioritize it easily and the team has the ability to think of alternative solutions. It also makes it very difficult to run into the 'bad' headlines from above.

Additionally, the business value is attached to the strongest words of the statement ('I want') and the method of delivering the business value is attached to the weaker word ('By'). In the traditional I want/so that format it is the other way around. This makes a big difference in the conversations with the business when an alternative solution has been identified.

Now you've got the perfect* story.



For your team.

How do you share it?

Involve the Team



Huimin, **BA**

*"Tell me, I forget.
Show me, I remember.
Involve me, I understand."*

—Chinese Proverb

As do most Agile teams, our teams have group time for product owners or business analysts to share user stories. We consciously practice the following three tactics to ensure quality interactions:

1. Why?

We put a lot of effort in establishing team-wide shared understanding about "why we are doing this?" You can use narratives to do this: telling a story of your user story. For instance during the kick-off for a new release, we immersed our team in one of customer's life. We showed the customer's photo, reproduced a scene from her daily work, and quoted what she said about her pain points. Doing this can activate the team's creativity to solve problems for customers.

2. Where are we?

In Agile teams, breaking work into digestible stories creates the need for a "map" to track and locate them in the overall user journey. When kicking off a specific story, never hesitate to spend time syncing audience with context. You can employ many different tools such as user story mapping or an activity map to track and share your "current location." If you have access to a whiteboard, draw it out. Imagine yourself as a zoomable canvas that carries the map of your team's work. Choose the the right zooming level and use your own style to bring everyone to the same point.

3. What are the Acceptance Criteria?

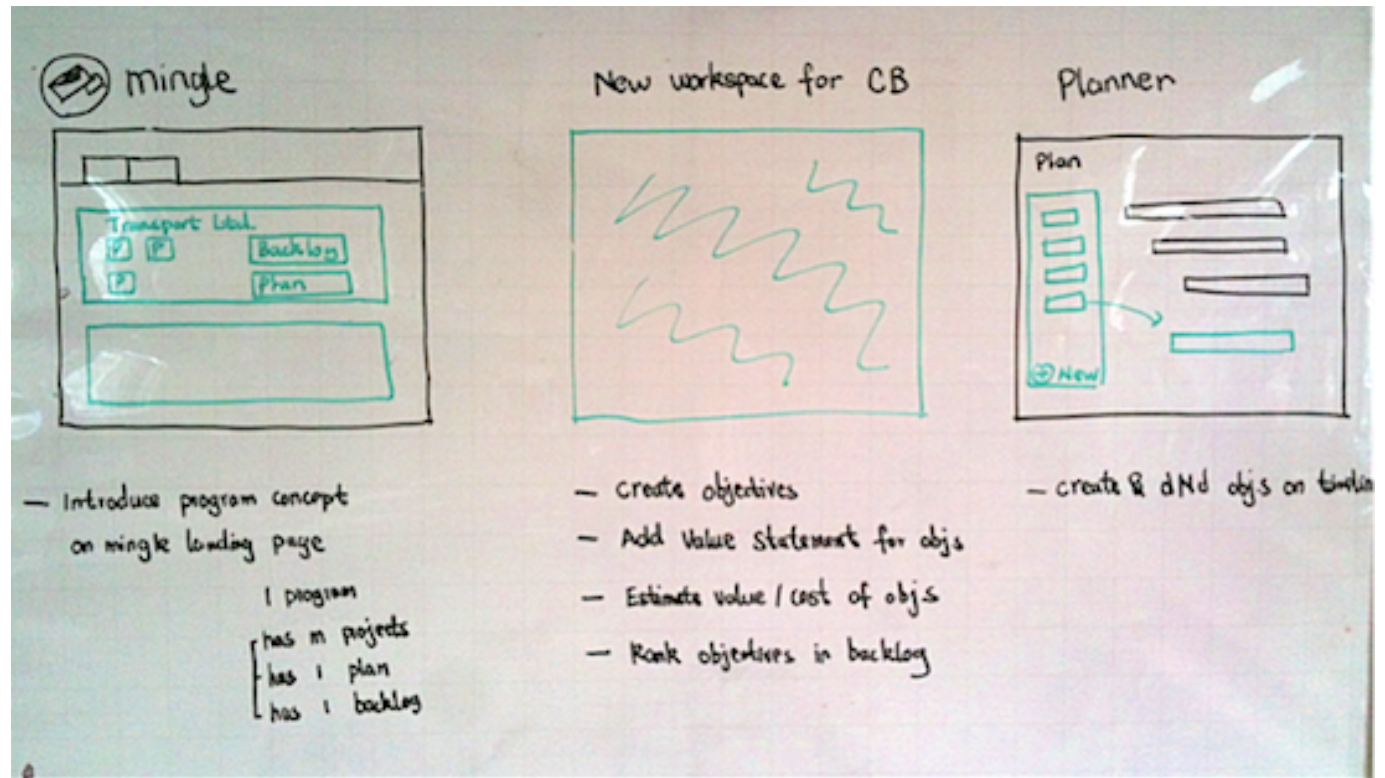
To define "what" the Acceptance Criteria are, I like drawing them out on the whiteboard and letting the team summarize what they see. Here's what I do.

(continued)

Involve the Team



Huimin, BA



- Draw the existing scenario/page as the “Given” part.
- Using a different color point out the event trigger, the “When” part.
- Continue to draw out the expected “Then” part.
- Pause, let the team ask questions or make comments
- Using a different color capture the questions and comments.

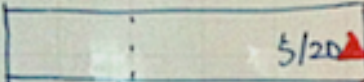
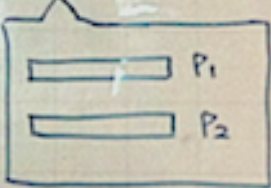
(continued)

Involve the Team



Huimin, BA

13621 Tell pm that some objective work has not started

Today	P ₁	P ₂	Alert?
	no work completed	no work completed	✓
	no work completed	on track	✓
	no work completed	late	✓
	on track	on track	—
	on track	late	✓
	late	late	✓

- Answer the questions on the list and add items which are important but weren't brought up by the team.
- Use this as a checklist along with the "given-when-then" as acceptance criteria for the story.

(continued)

Involve the Team



Huimin, BA

(continued)

FAQs

Why don't I just use screenshots or hi-fi prototypes?

They contain too many detailed page elements. Sketching omits noise, keeping the conversation focused and limiting analysis to include just-enough detail. Whiteboard sketching is generally better than a static picture, because you can change it as the conversation proceeds.

What do people do when I am drawing?

Don't worry about the one or two minutes when people are waiting for you. Revealing the story progressively helps your audience become more actively involved.

What if I have ten scenarios for my story?

I only draw out the happy path in the "given-when-then" format to establish the basic flow of the interaction. For the negative scenarios, I use a table or matrix, depending on situation. A bit of variety can prevent boredom.

What's after these 3 things?

Sharing user stories is not a one-time thing. We need remain flexible to accommodate changes. Don't freeze the business requirement after the group communication. Keep the conversation going as needed; keep improving as we are committed to in the true agile spirit.

References

Cohn, Mike (INVEST)
(2004) "User Stories Applied", Addison Wesley

Jeffries, Ron (2001) Card, Conversation, Confirmation
<http://xprogramming.com/articles/expcardconversationconfirmation/>

Feature Injection
[http://www.infoq.com/articles/pulling-power,](http://www.infoq.com/articles/pulling-power)
<http://www.infoq.com/articles/feature-injection-success>

North, Dan (Given, when, then)
<http://dannorth.net/introducing-bdd/>

Wake, B (2003) INVEST acronym
<http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>



Be in this ebook.

Tell us your story.

We'd love to hear it.
Email us your take on writing
and sharing stories and if it is
interesting we'll include it in
this ebook

Email Us



Agile Project Management

Make decisions, not documentation

The best Agile requirements are the ones the team builds as they work. Mingle generates actionable project records from natural team collaboration.

Share this ebook.

