10 TIPS FOR WRITING GOOD USER STORIES

By Roman Pichler, 24th March 2016

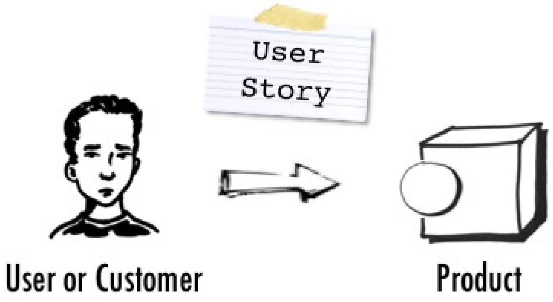
Photo courtesy of Pexels

User stories are probably the most popular agile technique to capture product functionality: Working with user stories is easy. But telling effective stories can be hard. The following ten tips help you create good stories.

1 Users Come First

As its name suggests, a user story describes how a customer or user employs the product; it is told from the user's perspective. What's more, user stories are particularly helpful to capture a specific functionality, such as, searching for a product or making a booking. The following picture illustrates the relationship between the user, the story, and the product functionality (symbolised by the circle).

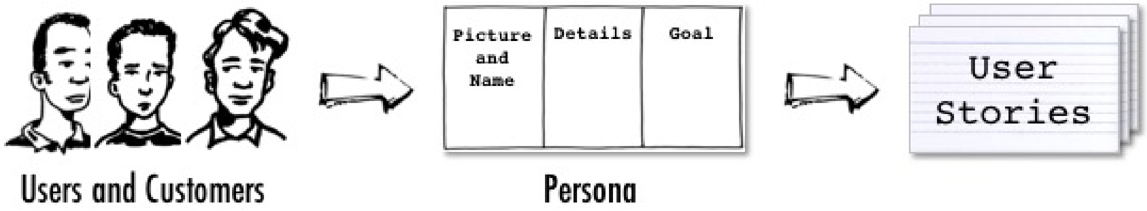
If you don't know who the users and customers are and why they would want to use the product, then you should **not** write any user stories. Carry out the necessary user research first, for example, by observing and interviewing users. Otherwise, you take the risk of writing speculative stories that are based on beliefs and ideas—but not on data and empirical evidence.



2 Use Personas to Discover the Right Stories

A great technique to capture your insights about the users and customers is working with personas. Personas are fictional characters that are based on first-hand knowledge of the target group. They usually consist of a name and a picture; relevant characteristics, behaviours, and attitudes; and a goal. The goal is the benefit the persona wants to achieve, or the problem the character wants to see solved by using the product.

But there is more to it: The persona goals help you discover the *right* stories: Ask yourself what functionality the product should provide to meet the goals of the personas, as I explain in my post From Personas to User Stories (http://www.romanpichler.com/blog/personas-epics-user-stories/). You can download a handy template to describe your personas from romanpichler.com/tools/persona-template (http://www.romanpichler.com/tools/persona-template/).

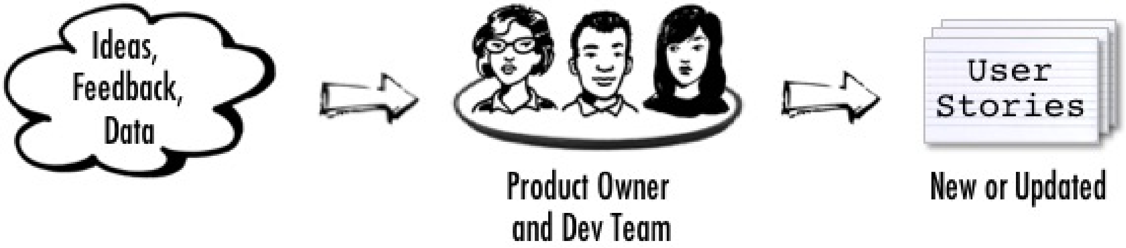


3 Create Stories Collaboratively

User stories are intended as a lightweight technique that allows you to move fast. They are not a specification, but a collaboration tool. Stories should never be handed off to a development team. Instead, they should be embedded in a conversation (http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/): The product owner (http://www.romanpichler.com/blog/roles/one-page-product-owner/) and the team should discuss the stories together. This allows you to capture only the minimum amount of information, reduce overhead, and accelerate delivery.

You can take this approach further and write stories collaboratively as part of your product backlog grooming process (http://www.romanpichler.com/blog/grooming-the-product-backlog/). This leverages the creativity and the knowledge of the team and results in better user stories.

If you can't involve the development team in the user story work, then you should consider using another, more formal technique to capture the product functionality, such as, use cases.



4 Keep your Stories Simple and Concise

Write your stories so that they are easy to understand. Keep them simple and concise. Avoid confusing and ambiguous terms, and use active voice. Focus on what's important, and leave out the rest. The template below puts the user or customer modelled as a persona into the story and makes its benefit explicit. It is based on by Rachel Davies' popular template, but I have replaced *user role* with *persona name* to connect the story with the relevant persona.

As <persona> ,
I want <what?>
so that <why?>.

Use the template when it is helpful, but don’t feel obliged to always apply it. Experiment with different ways to write your stories to understand what works best for you and your team.

---

5 Start with Epics

An epic is a big, sketchy, coarse-grained story. It is typically broken into several user stories over time—leveraging the user feedback on early prototypes and product increments. You can think of it as a headline and a placeholder for more detailed stories.

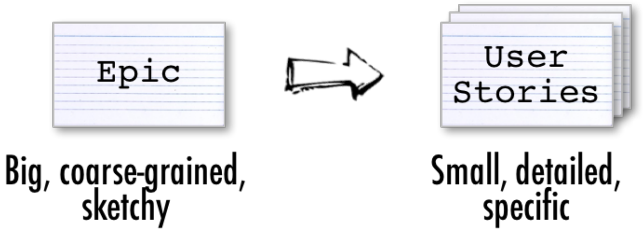
Starting with epics allows you to sketch the product functionality without committing to the details. This is particularly helpful for describing new products and features: It allows you to capture the rough scope, and it buys you time to learn more about how to best address the needs of the users.

It also reduces the time and effort required to integrate new insights. If you have many detailed stories in the product backlog, then it’s often tricky and time-consuming to relate feedback to the appropriate items and it carries the risk of introducing inconsistencies.

---

6 Refine the Stories until They are Ready

Break your epics into smaller, detailed stories (http://www.romanpichler.com/blog/product-canvas/epics-and-ready-stories/) until they are ready (http://www.romanpichler.com/blog/product-backlog/the-definition-of-ready/): clear, feasible, and testable.
All development team members should have a shared understanding of the story's meaning; the story should not be too big and comfortably fit into a sprint; and there has to be an effective way to determine if the story is done.



7 Add Acceptance Criteria

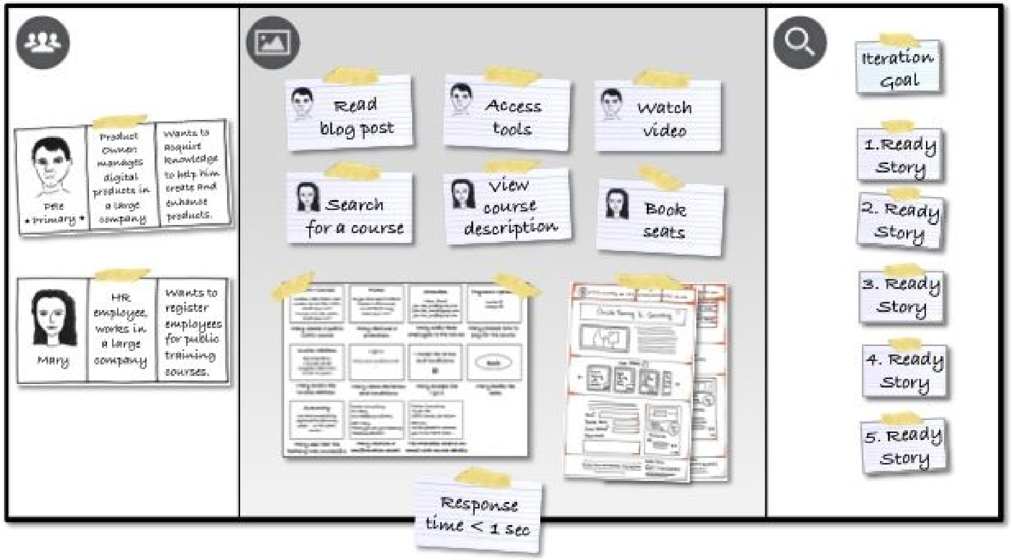
As you break epics into smaller stories, remember to add acceptance criteria. Acceptance criteria complement the narrative: They allow you to describe the conditions that have to be fulfilled so that the story is done. The criteria enrich the story, they make it testable, and they ensures that the story can be demoed or released to the users and other stakeholders. As a rule of thumb, I like to use three to five acceptance criteria for detailed stories.

8 Use Paper Cards

User stories emerged in Extreme Programming (XP), and the early XP literature talks about story *cards* rather than user *stories*. There is a simple reason: User stories were captured on paper cards. This approach provides three benefits: First, paper cards are cheap and easy to use. Second, they facilitate collaboration: Every one can take a card and jot down an idea. Third, cards can be easily grouped on the table or wall to check for consistency and completeness and to visualise dependencies. Even if your stories are stored electronically, it is worthwhile to use paper cards when you write new stories.

9 Keep your Stories Visible and Accessible

Stories want to communicate information. Therefore don't hide them on a network drive, the corporate intranet jungle, or a licensed tool. Make them visible, for instance, by putting them up on the wall. This fosters collaboration, creates transparency, and makes it obvious when you add too many stories too quickly, as you will start running out of wall space. A handy tool to discover, visualise, and manage your stories is my Product Canvas (http://www.romanpichler.com/tools/product-canvas/) shown below.


Pete
• Primary •
Product Owner: manages digital products in a large company
Wants to acquire knowledge to help him create and enhance products.
Mary
HR employee, works in a large company
Wants to register employees for public training courses.
Read blog post
Access tools
Watch video
Search for a course
View course description
Book seats
Response time < 1 sec
Iteration Goal
1. Ready Story
2. Ready Story
3. Ready Story
4. Ready Story
5. Ready Story

(http://www.romanpichler.com/tools/product-canvas/)

---

10 Don’t Solely Rely on User Stories

Creating a great user experience (UX) requires more than user stories (http://www.romanpichler.com/blog/agile-product-innovation/user-stories-enough-for-a-great-user-experience/). User stories are helpful to capture product functionality, but they are not well suited to describe the user journeys (http://www.romanpichler.com/blog/agile-product-management-tools/agile-scenarios-and-storyboards/)and the visual design (http://www.romanpichler.com/blog/agile-product-innovation/agile-user-interface-design/). Therefore complement user stories with other techniques, such as, story maps, workflow diagrams, storyboards, sketches, and mockups.

Additionally, user stories are not good capturing technical requirements. If you need to communicate what an architectural element like a component or service should do, then write *technical stories* or—which is my preference—use a modeling language like UML.

Finally, writing user stories is worthwhile when you develop software that's likely to be reused. But if you want to quickly create a throwaway prototype or mockup to validate an idea, then writing stories may not be necessary. Remember: User stories are not about documenting requirements; they want to enable you to move fast and develop software as quickly as possible—not to impose any overhead.

Learn More

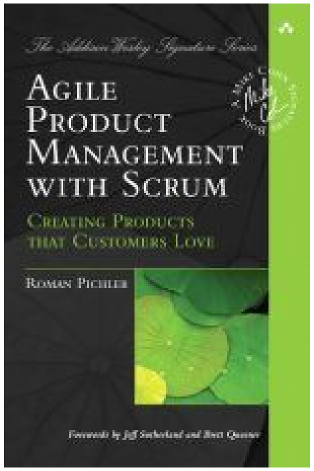
Learn more about user stories with the following:

TRAINING

Certified Scrum Product Owner Course

(https://www.romanpichler.com/training-courses/certified-scrum-product-owner-course/)

BOOK


The Addison-Wesley Signature Series
Agile Product Management with Scrum
Creating Products that Customers Love
Roman Pichler
Forewords by Jeff Sutherland and Brett Queener

(https://www.romanpichler.com/romans-books/agile-product-management-with-scrum/)

BLOG POST

5 Common User Story Mistakes

🕒 6 MIN READ

(https://www.romanpichler.com/blog/5-common-user-story-mistakes/)

Source: https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/ (https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/)

---

73 comments on “10 Tips for Writing Good User Stories”

jane

DECEMBER 28TH, 2018

Hi Roman,
I have a question for you , we doing a homework about user story,
my question is user stroies are written for whole system or part of the system ?

Reply

Roman Pichler

JANUARY 3RD, 2019

Hi Jane,

Thanks for sharing your question. My article “User Story Modelling (https://www.romanpichler.com/blog/user-story-modelling/)” should help you come up with the right answer.

Good luck with your homework,
Roman

Reply

Maciej Lewandowski

DECEMBER 13TH, 2018

Hello. Do you agree that not the form of the user story is most important but the expressed value?

Reply

Roman Pichler

DECEMBER 14TH, 2018

Absolutely Maciej. As I say in the article: “Use the [user story] template when it is helpful, but don’t feel obliged to always apply it. Experiment with different ways to write your stories to understand what works best for you and your team.” There is no one right way to tell a story.

Reply

Tony Bresse

OCTOBER 13TH, 2018

Hi Roman
In your final statement you wrote
“Remember: User stories are not about documenting requirements”

As a certified scrum Master and Product Owner, I always considered User Stories as the new structured way of documenting functional requirements.

And the User Story backlog is the replacement of the traditional BRD Business Requirements Document that listed the functional requirements.

If user stories are not requirements, then how are the requirements represented in the Agile framework? And what do they look like?

Reply

Roman Pichler

OCTOBER 15TH, 2018

Hi Tony,

Thank you for sharing your question. A traditional functional requirement is replaced by one or more user stories plus a conversation between the product owner and development team. In order to effectively apply user stories, product owner and dev team must discuss product functionality together on a regular basis, for example, as part of the product backlog grooming or refinement work (https://www.romanpichler.com/blog/grooming-the-product-backlog/). If this doesn't work for you, then I recommend experimenting with use cases, see my article "User Stories or Use Cases? (https://www.romanpichler.com/blog/user-stories-or-use-cases/)".

Hope this helps!

Reply

Ondrej

JUNE 27TH, 2018

Hi Roman,

I need to implement a fraud detection solution into my digital product. This is solely just for the business to evaluate if the user is a human being or a bot.

So is it bad to write story of "As business I want to have a fraud detection implemented so I can lower the risks in the business process"? Is user really just the one who is using the frontend of the app (customer)? Isn't it the same as stakeholder instead that means any affected party? I mean.. Users of some features (story) are sometimes someone else then a user of the whole application, right? Other example – I need to upgrade to a newest technology so it's easier for me to develop faster and more precise. User story of "As developer I want to have XX technology implemented so I can develop faster and more precise" is not good then?

I think that it does not really matter who is the user but the fact that group of people will really use the object of that story. Am I wrong? Purpose of the user story is to tell a user's story so the developer or whoever for who the task is intended understands a view of that user. I find it not beneficially to try to write every story from the view of the customer as it is not describing the reality then.

I would really appreciate your opinion here as I respect you a lot but in the same time I believe this rule "user = customer / ending user" is contra-beneficiary.

Reply

Roman Pichler

JUNE 27TH, 2018

Thanks for sharing your question Ondrej. User stories are just a simple technique to help you describe a product from the perspective of a user, customer, administrator, or another role. It's entirely up to you, of course, if and how you use the technique.

Having said that, I find it beneficial to carefully consider who will benefit from a product and describe the individuals using personas. This avoids a solution-centric mindset where we worry more about how to build the product rather than for who and why.

When it comes to capturing development work, I would simply add an item to the product backlog that states "Investigation of new tools and technologies to increase development productivity". Don't feel obliged to describe all product backlog items as (user) stories.

Hope this helps!

Reply

Costas

NOVEMBER 23RD, 2017

Thank you Roman for all the helpful information on your site!

What would you do if, due to certain system components being unavailable, the DEV team requests stories that describe slices of user functionality to be broken down further at system component level?
For example, AS a buyer, I WANT to see the total price for the selected product, SO THAT I can decide if I want to buy it. Due to commercial reasons, and for the next few sprints, the DEV team has no code access to the back-end system that will process the logic to derive the price. What we did in this case is to leave the story as-is with the prefix [Front-end]. Without a real calculation in place the user will see a dummy price and the story will pass. We created an identical story with the prefix [Back-end] which remained in the backlog until the calculation logic could be built.

What are your thoughts? How would you handle this?

Thank you

Reply

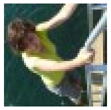
Roman Pichler

NOVEMBER 24TH, 2017

Hi Costas,

Thanks for your feedback and sharing your question. Sounds like you found a solution that works for you, which is great. An alternative way to handle the issue is to delay the implementation of the story until you have gained access to the back-end system. Personally, I would prefer this option to avoid splitting the story by architecture and effectively ending up with a partially implemented one. This might also help highlight the real issue: the inability to access the back-end system. Hope this helps!

Reply



Lora

OCTOBER 20TH, 2017

Great post. Added it to the Awesome List of Agile Software Development https://github.com/lorabv/awesome-agile-software-development/blob/master/User-Stories.md (https://github.com/lorabv/awesome-agile-software-development/blob/master/User-Stories.md)

Reply

Roman Pichler

OCTOBER 23RD, 2017

Thanks Lora!

Reply

Alan

SEPTEMBER 1ST, 2017

Hi Roman,
We have engineers who document the technical solution they're going to implement in the story description. It seems wrong, but where else would they document it?

Reply

Roman Pichler

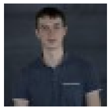
SEPTEMBER 1ST, 2017

Hi Alan,

I recommend keeping user stories free from implementation details. The stories should capture how users employ product functionality, not how the solution is built. I've found it helpful to capture technical decisions in form of UML diagrams like component, activity, and sequence diagrams. Additionally, your dev team may want to use an overall architecture model that documents key decisions affecting the entire solution. Scott Ambler has done a lot of good work on agile modelling techniques, and I suggest that you take a look at his site http://agilemodeling.com (http://agilemodeling.com).

Hope this helps!

Reply



Anton Belonovich

JULY 6TH, 2017

Hi Roman. Thanks for your articles and blog!

But I have a question. You wrote “Stories should never be handed off to a development team”. But here http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/ (http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/) mentioned that “[story card] often handed to the programmers”.

We are now doing exactly this: passing user stories to devs, then devs discuss them and create issues.

Can you please explain why do you think stories should not be passed to devs?

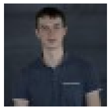
Reply

Roman Pichler

JULY 6TH, 2017

Hi Anton, Thanks for your feedback and question. Well done for checking the source. It’s perfectly fine in my mind to hand the user stories to the development team if an effective conversation about the story did take place beforehand and the conversation involved the product owner and development team. AsRon Jeffries put it: “The requirement itself is communicated from customer to programmers through conversation: an exchange of thoughts, opinions, and feelings. (http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/)” Please bear in mind that Ron describes how user stories should be used in Extreme Programming where planning differs compared to Scrum Kanban, and the roles product owner role, self-organising development team, and ScrumMaster don’t exist (as defined in Scrum). Does this help?

Reply



Anton Belonovich

JULY 9TH, 2017

Thanks for your answer Roman. Yes, there is a discussion between product owner and dev team before passing user stories to devs.

Reply

Sergey

JUNE 30TH, 2017

Hi, Roman, thank you for the excellent post. I have a question related to the number “10 Don’t Solely Rely on User Stories” I have the following architecture: Portal for schools to post jobs, a separate Portal for candidates to apply for a job, a CRM system as a backend system to manage/mediate a process between Schools and Candidates. First question, should we have user stories across all three types of users? What would be the best artefact to cross reference these user stories? Would you recommend process maps, customer journey or something else? I would appreciate you input.

Reply

Roman Pichler

JUNE 30TH, 2017

Thanks for your feedback and question, Sergey. I recommend that you write user stories for the school employees (presumably the head teachers) and the job seekers. Capture your software architecture decisions in a architecture model using, for instance, UML. User stories describe functionality from an end user’s perspective and should be free from architecture and technology concerns. To put it differently, user stories describe the what−what a product should do−and not the how−how the product is built. Does this help?

Reply

Sergey

JUNE 30TH, 2017

Thank you, Roman.
What about CRM users? Do we need to have user stories for them? They are basically managing processes needed to support both, schools and candidates. They are using out of the shelf solution which we customise. I think process maps (with swimlane) is the medium to link different user groups user stories together in the end to end flow. What do you think?

Reply

Roman Pichler

JULY 3RD, 2017

Hi Sergey, Sounds like you should write stories for the CRM users, assuming that they directly interact with the system. You may find it helpful, though, to take a step back, determine the users and customers of the product, and create personas (http://www.romanpichler.com/blog/10-tips-agile-personas/) before you write more user stories. Good luck!

Reply

Diogo

MAY 12TH, 2017

Hi. Very good tutorial on writing user stories.
I have a question though. As a web development intern I was given the task to create the user stories of a tool that will possibly be implemented in the company, for our own use. I say possibly, because there already exists a tool that apparently does what we need, but seems to be harder to integrate and customize for our needs. Being part of the development team I am basically writing my own experiences on using the existing tool, so what should be my approach in writing the user stories? Something like: As a business analyst, I would like to (what) so that I will be able to (why). As simple as that? As the project progresses and as part of the customization, I assume that existing functionalities will be suppressed and not be implemented and new one possibly created to fulfill our needs. Are there any tricks and tips I should know or pitfalls I should avoid when creating these user stories? I also assume that even being a developer, I should take the hat of the company business analysts or managers, as to think as they would when using the tool to be developed. Am I right?
Thank you for your post!

Reply

Roman Pichler

MAY 15TH, 2017

Hi Diogo,

Thanks for your feedback and for sharing your question. No matter what product you develop, always starts with the users and the value the product should create for them. In your case, I would ask what benefit the tool should offer or which problem it should address, and who the users are.

Once you can confidently answer these questions, create personas and get together with the people who should build the tool and capture the key pieces of functionality as epics. My article From Personas to User Stories (http://www.romanpichler.com/blog/personas-epics-user-stories/) explains how you can do this.

Good luck!

Reply

Muhammad

MAY 3RD, 2017

How would you write a user story from a user prospective if the story is to redesign the page cosmetically. The story has been instigated by the content designer. I was thinking of below but when you say think from user prospective then it would be wrong.

As a content designer
I want to improve the layout of the xxxxxx homepage
So that I can enrich the customer experience with quality visual designs

Reply

Roman Pichler

MAY 3RD, 2017

Hi Muhammad,

Thanks for your questions. User stories should always be written from the perspective of the user. To do so, think about how user interface design changes would benefit the users. Will it to simplify a user journey? Will it make the page more intuitive and easier to use? Will it help them find what they want quicker? Asking these questions is not only helpful to write effective users stories. It makes sense from a business perspective: why should you company invest money in improving the page design? How will it benefit the users and ultimately your business?

Hope this helps.

Reply

Rantanen, Veli-Matti

MARCH 3RD, 2017

As a electrical commissioning engineer, the critical things are the engine transient load test and engine load sharing, from the FAT, the engine transient load test was very good result, can we get the parameter of engine used for FAT test? How can we get the best engine load sharing?

Reply

Roman Pichler

MARCH 3RD, 2017

Hi Veli-Matti, Thanks for your comment. I am not sure I fully understand your question. Are you referring to a factory acceptance test? I'm afraid I am not a test expert.

Reply

James Sessions

FEBRUARY 11TH, 2017

Hi Roman,
Can you tell me if the following can be categorised as a User Story or would it be correct to categorise it as acceptance criteria? Thank you for your help. Kind regards James Sessions.
"As a Recruiting Manager I want to be able to input advert details into a recruitment form, so that a new post can be advertised"

Reply

John Ferguson Smart

MARCH 2ND, 2017

Hi James,

So: “As a Recruiting Manager I want to be able to input advert details into a recruitment form, so that a new post can be advertised”

I would argue that, no, you don’t want to input advert details (after all, typing is a pain), you want to post a new job advert. And why do you want to do that? I presume, to recruit more suitable candidates. So I would write this story like this:

In order to attract more suitable clients
As a Recruiting Manager
I want to make the relevant details of a job accessible to potential and suitable candidates

Reply

Hawwa
FEBRUARY 6TH, 2017

Hi Roman, I’m currently part of a project with a group at University, and writing up the user stories.
For the ‘so that..’ section of user stories, I was wondering what exactly its capturing.

For example, some user stories directly relate to some sort of action such as:
‘As a Student,
I want to be able to enrol to courses,
so that I can access all content of that course’

However, I find that some are not as 'clear-cut' in the so that..section. e.g. We have to include a feature where a Lecturer can include a map as part of a group meetup invite.
'As a Lecturer,
I can send a map in a group meetup invite
so that…' <- how can this directly relate to some sort of action?

Could you provide any input as to how user stories can tackle the problem of capturing a functionality such as this?

Reply

Roman Pichler

FEBRUARY 6TH, 2017

Thanks for your question, Hawwa. I find it helpful to derive user stories from personas and to link the "so that" part of a user story to the persona's need–the main problem the persona wants to solve or the primary benefit the character wants to realise–and I'd recommend that you try this approach.

I also find that stating a reason why it is necessary to provide the appropriate functionality works well for epics. As epics get broken them down into smaller, fine-grained stories, stating a reason can feel artificial: Users typically don't want to access individual pieces of functionality, such as, set a password, but rather achieve a bigger goal like "personalise the app". You therefore shouldn't feel obliged to add a reason to all of your stories, particularly your smaller ones.

Does this help?

Reply

Hawwa

FEBRUARY 6TH, 2017

Thanks for the reply, your advice was quite helpful.

How do you think UI requirements from the client are best captured? It's tricky trying to pin down exact functionalities without it seeming too specific.

Reply

Roman Pichler

FEBRUARY 6TH, 2017

Please take a look at my post Agile User Interface Design (http://www.romanpichler.com/blog/agile-user-interface-design/).

Reply

Jyothis Narayanan

OCTOBER 28TH, 2016

Hi Roman,
Thank You so much for such informative posts.
One question I have is about the data validation on text fields.
Do we need to cover them as part of acceptance criteria. If so, then is there any standard format that can be followed for specifying data validation details.
Thanks
Jyothis N

Reply

Roman Pichler

OCTOBER 31ST, 2016

Hi Jyothis,

My preference is to create separate validation stories, particularly if the validation rules are not straightforward. If in doubt, decide together with the development team what top do.

All the best!

Reply

Vit

OCTOBER 20TH, 2016

What about some example user stories?

Reply

Roman Pichler

OCTOBER 20TH, 2016

Hi Vit, You can find user story examples in my other posts on user stories (http://www.romanpichler.com/blog/category/user-stories/). I would hope, though, that my tips allow you to start telling stories about how users interact with your product. Remember: A user story is simply a story, no more and no less. And we all can tell stories.

Reply

vicque l amos

OCTOBER 19TH, 2016

I'm new to agile methodology working on a new product using agile. I'm having a difficult time understanding what to do with technical stories. Initially, these were included as part of the functional story but the story point was 34 so split into a technical and functional story. However, the agile coach said there should be no technical story so now have to combined technical with the functional. Totally confused. My thought was to leave the functional and technical split and assign to an epic. your thoughts.

Reply

Roman Pichler

OCTOBER 20TH, 2016

Hi Vicque, Thanks for sharing your question. When splitting user stories, try to break them into smaller functional units, as I explain in my posts Epics and Ready Stories (http://www.romanpichler.com/blog/epics-and-ready-stories/) and Refining User Stories (http://www.romanpichler.com/blog/refining-user-stories/). Don't make any assumptions about how the user stories will be implemented, for instance, which layers, components, or services will be affected. The development team should make those decisions, typically in the sprint planning meeting. Hope this helps!

Reply

gapsel

SEPTEMBER 9TH, 2016

I'm learning writing user stories. This help much. Thank you.

Reply

Lloyd Lawrence

SEPTEMBER 8TH, 2016

Hi Roman,

Thanks for the very informative article. I work in the IT support field (general support, not specific to application support), and have been investigating the possibility of using User Stories for defining issues or experiences that users go through in their IT working environment. Do you think User Stories are adaptable to a scenario such as this, eg:
Joe Bloggs is an Engineer. Joe needs to be able to open, fill and sign PDF files in order to approve work orders. Joe often experiences freezing or crashing of the PDF file while trying to fill one out.
Thanks.

Reply

Roman Pichler

SEPTEMBER 9TH, 2016

Hi Lloyd,

Thanks for your question. You can certainly use user stories to describe the requirements stated in your sample scenario. If you want to describe the entire workflow or interaction, however, I recommend using a scenario, storyboard, or story map, see my post Agile Scenarios and Storyboards (http://www.romanpichler.com/blog/agile-scenarios-and-storyboards/). Hope this helps!

Reply

Raviavi

AUGUST 16TH, 2016

Dear Roman,

Could you please give atleast one example of an Epic ?
Do they follow the same format as the smaller User Stories, or are they in a different format ?
Is it for example ok to have an Epic as follows : –
As a [Manager] I want to [view monthly reports] so that I can [manage my branch well]
Or is it traditionally written in a different way ?

Reply

Roman Pichler

AUGUST 16TH, 2016

Thanks for asking the question, Raviavi. Epics are simply big, coarse-grained user stories. They are typically so large that they cannot be implemented in one sprint and therefore have to be eventually broken down into smaller stories. You can write epics using the format stated in tip number four (if that works for you) but you don't have to. There is no one right way to formulate a story. What matters is that the development team understands the epic or user story. Your example sounds like a nice epic. I would recommend, though, that you make the goal more specific: State how viewing reports helps managers do a good job. You can find more information on epics together with sample epics in my post Epics and Ready Stories (http://www.romanpichler.com/blog/epics-and-ready-stories/). Hope this helps!

Reply

Raviavi Read more http://www.romanpichler.com/blog/10-tips-writing-good-user-stories/

AUGUST 18TH, 2016

Thanks very much Roman.

Reply

Randy

AUGUST 6TH, 2016

we take a business need (user story/feature) and through conversations we design the solution and create stories for that work. All this prior to starting the sprint for that bus need? Is that correct?

Reply

Roman Pichler

AUGUST 8TH, 2016

Thanks for your question, Randy. A business need or goal is distinct from a user story or feature description. A business need might be to acquire new customers, generate revenue, or increase conversion. A user story captures how someone interacts with the product and uses a price of functionality. I recommend that you determine the business need and the product’s value proposition before you create any user stories. Hope this helps!

Reply

Randy r

AUGUST 6TH, 2016

With regards to team understanding of stories. I’m being told that the team gets a business need or story. They size it and then during the sprint is the time the design a solution for implementing the story. Seems the conversation needs to happen sooner. How can you size it without some idea of WHAT the solution will be. Is that correct?

Thanks

Reply

Roman Pichler

AUGUST 8TH, 2016

Hi Randy,

User stories are commonly sized (or estimated) for three reasons:

1. Help prioritise the product backlog (by performing a cost-benefit analysis).
2. Track the projects progress, for instance, by using a release burndown chart.
3. Ensure that the sprint goal is realistic and help the development team pull the right amount of work into the sprint (assuming that the team uses velocity-based iteration planning).

I recommend that you are clear on why you size stories to choose when you should so it.

The last point in time for the user stories conversation to happen is the sprint planning. But I would recommend including the team or some team members in the product backlog management work (aka grooming or refinement), as I discuss in my post Grooming the Product Backlog (http://www.romanpichler.com/blog/grooming-the-product-backlog/).

Hope this helps!

Reply

Harsha

DECEMBER 9TH, 2015

Hi Roman,

I was working in Operations for 10 years and now have joined as a Business Analyst in a bank for an Agile project. The Product Owner/SME has already written Epics and created a few product backlogs. He now wants me to write new backlogs relevant to the Epic and also write User Stories for each of the backlog items.He is knowledgeable but always short of time. How do I approach him with a set of questions for which I need his answers? I get totally blank and have no idea how I should proceed with writing new product backlog items and also writing User stories for each of these items. The Epic is at a very high level and I just have become clueless on how to proceed further. Can you pls guide me ?

Reply

Roman Pichler

DECEMBER 10TH, 2015

Hi Harsha, I recommend that you understand who the customers and users are and what problem they would like to see addressed before you write epics and user stories. A great technique to capture your customer and user insights are personas, from which you can derive epics, as I describe in the following post:

http://www.romanpichler.com/blog/personas-epics-user-stories/ (http://www.romanpichler.com/blog/personas-epics-user-stories/)

In addition to understanding the market, you should also address the collaboration between you and the product owner. It sounds to me as if you were at risk of becoming a poxy product owner, which I tend to find an ineffective solution, see my posts:

http://www.romanpichler.com/blog/avoiding-common-product-owner-mistake/
(http://www.romanpichler.com/blog/avoiding-common-product-owner-mistake/)
http://www.romanpichler.com/blog/business-analysts-in-scrum/
(http://www.romanpichler.com/blog/business-analysts-in-scrum/)

Hope this helps!

Reply

Deepu Kumar Gouli
SEPTEMBER 22ND, 2015

Hi Roman,

Thank you for such a detailed tips for writing user stories.
I have query, I'm currently working in auto financial domain for the first time in my experience and finding it hard to create user stories for technical topics like Master Data Management (Data Standardization, Data Historization.. etc). How to tackle this kind of situation. Can you please give some input to this.

I see no simple which are maintained in past in these topics as well, The domain has data flow between different systems and creating business rules to standardize hundred of data field.

It will be a great help if you give some input to me.

Thank you in advance.

Best Regards,
Deepu Kumar Gouli

Reply

Roman Pichler

SEPTEMBER 23RD, 2015

Hi Deepu, Thanks for your comment. I don’t recommend writing user stories for technical topics like the ones you mentioned. User stories are great to describe end user functionality but not how a product or system works. If you want to capture architecture and design requirements, then I suggest that you use a modelling language like UML, which is better suited than natural language IMO. Hope this helps!

Reply

Bruce

JUNE 5TH, 2015

Great Post! Thanks for the advice and the templates. I’ll let you know how my project goes.

Reply

Roman Pichler

JUNE 5TH, 2015

Thanks for your feedback Bruce!

Reply

Anuj Sharma

APRIL 29TH, 2015

If I as a BA getting user story from marketing, for a product meant for general public, is it a good way to write user story ” As a Marketing executive, I want customer to have ability to xxxx so that I or he/she can xxxxxx” . It will help in clearly understanding the owner of user story and objective.

Reply

Roman Pichler

APRIL 30TH, 2015

Hi Anuj, Thanks for your comment but I disagree with your recommendation. As the name suggests, user stories are primarily about the users and the customers. They should therefore feature in your stories and not internal stakeholders. I recommend that you formulate your stories “As , I want … so that …”.

Reply

Matt Adams
MARCH 12TH, 2015

Thanks for the tips Roman. My company is currently in a transition period to Agile. In terms of ADAPT, we have the awareness, desire and the ability is coming fast – can you recommend a good book which focuses purely on writing user stories?

Thanks,
Matt

Reply

Roman Pichler
MARCH 13TH, 2015

Hi Matt, I recommend that you look at the following books:

Mike Cohn. *User Stories Applied*
Gojko Adzic and David Evans. *Fifty Quick Ideas To Improve Your User Stories*
Jeff Patton. *User Story Mapping*

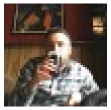
Reply

Matt Adams

MARCH 14TH, 2015

That’s great thanks Roman.

Reply



Thomas Hallett

APRIL 25TH, 2014

Useful tips, thanks.

Reply

Lohic Beneyzet

JANUARY 18TH, 2014

Thank you Roman for your posts. I’ve read this one and the linked one. Very useful! Especially the collaboration tip in writing user stories. I was struggling with how to deal with my role of Deputy Product Owner and now I think I’ve found the clue. Keep up the good job.

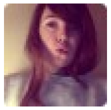
Reply

Roman Pichler

JANUARY 20TH, 2014

Thanks for the feedback. Glad you’ve found my user story tips helpful!

Reply



AnAn

DECEMBER 17TH, 2013

Thank you Roman! Useful post! We’ve recently summarised our “recipe” for writing a user/feature story the right way – looks like there’s a lot of similarities https://netguru.co/blog/posts/doing-features-and-user-stories-the-right-way (https://netguru.co/blog/posts/doing-features-and-user-stories-the-right-way)

Reply

Prashant Khare

FEBRUARY 15TH, 2013

Thank you Roman! Very helpful thoughts…

Reply

Prashant Khare

FEBRUARY 14TH, 2013

Thank you for very fast response! It is very helpful.

Just one query though; when you say estimate the item – do you mean the relative estimation (Poker / T-shirt) ot the time estimation as done for the tasks / sub-tasks for the user stories?
Thanks once again!

Reply

Roman Pichler

FEBRUARY 14TH, 2013

Hi Prashant, I was referring to the former. I typically estimate user stories in a product backlog/canvas workshop: http://www.romanpichler.com/blog/the-product-backlog-grooming-steps/ (http://www.romanpichler.com/blog/the-product-backlog-grooming-steps/) The task break-down happens in the sprint planning meeting.

Reply

Prashant Khare

FEBRUARY 14TH, 2013

Should the technical parts at the start of the project like setting up the environments, continuous integration, etc. be included as stories and also be estimated?

Reply

Roman Pichler

FEBRUARY 14TH, 2013

Hi Prashant, Great question. I don’t capture work results that are necessary to create a successful product but are not meaningful to a customer or user as user stories. I simply add a card that says, for instance, “Install and configure CruiseControl so that software can be continuously integrated and tested”. The example is written similar to an epic. It uses a goal/benefit clause but has no user role or persona. I would suggest, however, to estimate the item, as the team will have to spend time and effort to get it done.

Does this help?

Reply

RD

SEPTEMBER 30TH, 2015

Would you not write stories for environment setup etc and aim to complete those in sprint zero?

Reply

Roman Pichler

SEPTEMBER 30TH, 2015

No, I would not. If the work you mention is required, then I would describe the deliverables as simple statements, for instance, set up the development environment or choose a test automation tool, and add them to the product backlog. Customers and users don't care about the environment setup. They want a product that does a great job for them □

Reply

Agile Scout

DECEMBER 20TH, 2010

Thanks a lot for this. We featured this on our article today □

Reply

Robert Wilson

OCTOBER 18TH, 2010

thanks for the post

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

Subscribe to Roman's Blog

Your email

SUBSCRIBE

RSS Feed (https://www.romanpichler.com/feed/rss/)

Tags related to this post: definition of ready (https://www.romanpichler.com/blog/tag/definition-of-ready/) product definition (https://www.romanpichler.com/blog/tag/product-definition/) product manager (https://www.romanpichler.com/blog/tag/product-manager/) product owner (https://www.romanpichler.com/blog/tag/product-owner/) requirements (https://www.romanpichler.com/blog/tag/requirements/) teamwork (https://www.romanpichler.com/blog/tag/teamwork/)

(https://www.facebook.com/romanpichleragile) (https://twitter.com/romanpichler) (https://www.linkedin.com/in/romanpichler) (https://vimeo.com/user18622296) (https://www.romanpichler.com/feed/)