# Tools and Techniques Agile Methodologies

# 

# 

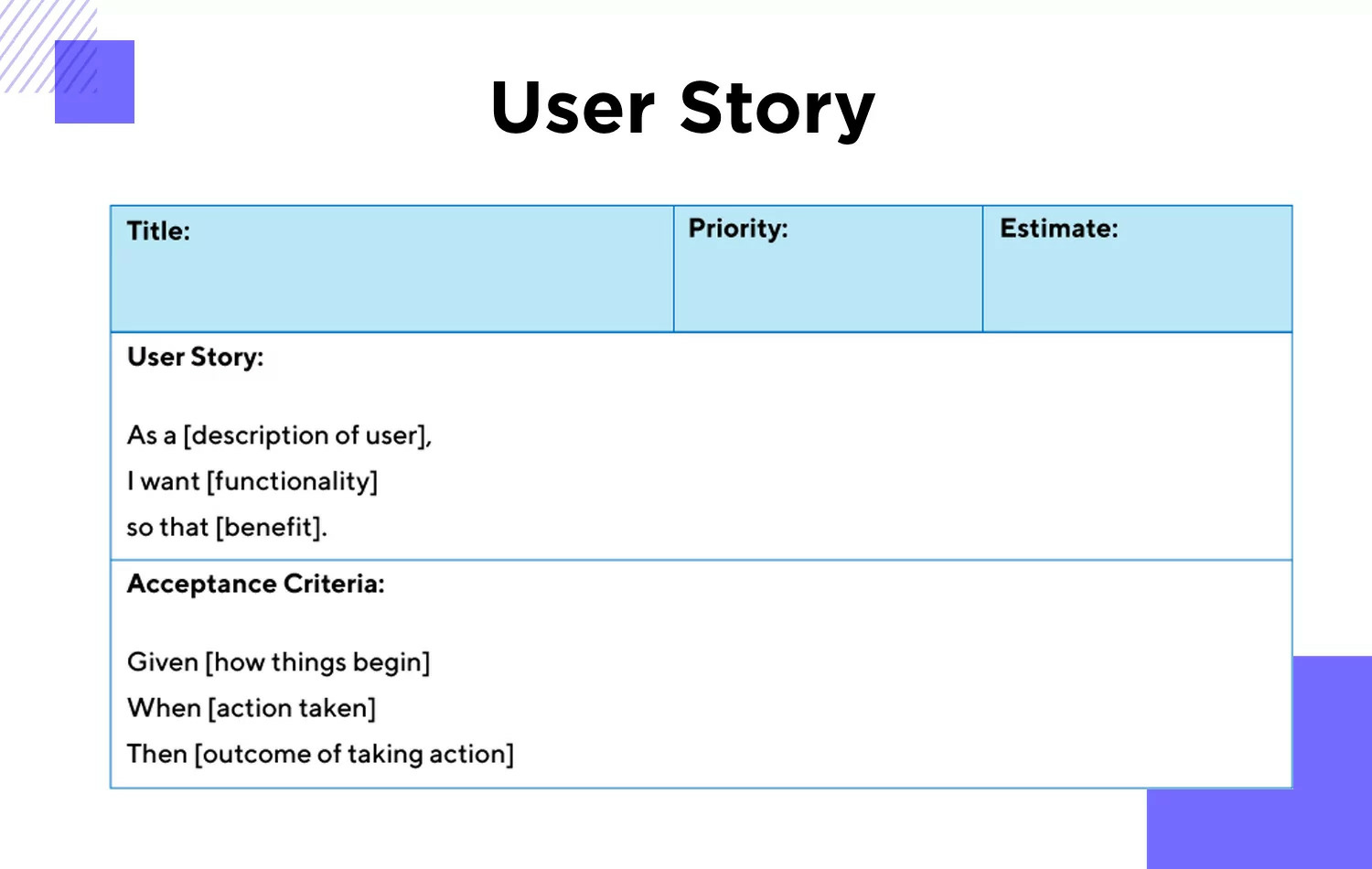
# Mapping User Stories in Agile

Summary: User-story maps help Agile teams define what to build and maintain visibility for how it all fits together. They enable user-centered conversations, collaboration, and feature prioritization to align and guide iterative product development.

In user story mapping, teams create a dynamic outline of a representative user’s interactions with the product, evaluate which steps have the most benefit for the user, and prioritize what should be built next. For agile organizations, it provides an alternative to building a flat list of backlog items or working from lengthy requirements documents.

In traditional product-development processes, teams often rely on wasteful and lengthy business requirements documents and functional design specifications to move from a vision for a digital product to outlining what it should include and how it should work. Instead of having an ongoing conversation about users, problems, ideas, and solutions, teams expect distributed documentation to suffice.

However, these documents usually fail; no one has the time or attention to read them, and even those who do read them end to end will likely come away with vastly different interpretations of what to build. Rather than propelling productivity, these heavy documents stifle creativity, communication, collaboration, and innovation from the start. As an alternative, user-story maps work much better as lightweight representations of the digital product that an Agile team intends to build.



User-Story Mapping Defined

Definition: User-story mapping (also known as user-story maps, story maps, and story mapping) is a lean UX-mapping method, often practiced by Agile teams, that uses sticky notes and sketches to outline the interactions that the team expects users to go through to complete their goals in a digital product.

Jeff Patton popularized the method, which replaces the lengthy, technical requirement gathering and siloed updating processes found in waterfall development. Story maps are intended to spark collaboration and conversation among Agile team members, while providing them with the bigger picture of how the digital product flows and fits together. This latter quality of story maps is important in the Agile environment, because losing sight of the product as a whole is a common challenge, likely to arise when teams work from a discrete list of user stories in a backlog.

User story mapping employs the concept of user stories — which communicate requirements from the perspective of user value — to validate and build shared understanding of the steps to create a product users love. Teams write user stories in a format that captures business value and can be completed within a development iteration (usually called a sprint). The user story format — As a [type of user], I want to [action] so that [benefit]. — can be helpful in thinking about product interactions from a user’s perspective.

A user-story map depicts 3 types of actions at different granularity: activities (the most general actions), steps, and details (the most specific actions). User activities and steps display horizontally across the top of the map, and the details stack vertically underneath their respective steps in priority order. To define each level of a story map, we’ll use a feature for depositing checks through a bank’s mobile application as an example:

Activities represent the high-level tasks that users aim to complete in the digital product — for example, Check account balance or Deposit a check. Depending on the type of application or website you're creating, you may only have a few high-level activities. These can display in sequential order or in parallel if multiple paths for various user types exist. Exploratory research about top user tasks should inform this level of the map.

Steps sit directly underneath activities and also display in sequential order. They represent the specific subtasks that users will go through in the product to complete the activity above. For example, the activity, Deposit a check can be deconstructed into the steps of Enter mobile deposit details, Sign check, Photograph check, Submit deposit, and Confirm deposit.

Details are the third level of the story map and describe the lowest-granularity interactions that the team anticipates users will experience to complete the step above. For example, Enter username or email and Enter password appear as two separate details underneath the Log in step.

Story maps contain 3 levels; activities, steps, and details.

User-story maps contain 3 main levels of actions (activities, steps, and details) that users will take in a digital product to complete their goals.

Why It’s Called User-Story Mapping

If you’re new to the concept of user stories, they are informal, natural language descriptions of features, UI elements, or tasks, written from the perspective of the user. They’re intended to get the team talking to each other about solutions in the context of end users and the benefit they’ll receive. These conversations help everyone arrive at shared understanding much faster than reading a requirements document. User stories can be written at a high level to describe a full product or feature and what it enables users to do or at low level, to outline an interface element and its value. For example:

High-level user story: “As a checking account holder, I want to deposit a check from my mobile device, so that I don’t have to waste time going to the bank.”

Low-level user story: “As a checking account holder, I want to save my credentials, so that I don’t have to input my username and password each time I log in.

Agile teams commonly rely on small, high-value user stories to plan and estimate what to work on each sprint. In the user-story map, activities, steps, and details are captured as short, succinct verb phrases representing user actions. These serve as the basis for the first half of the user-story format, describing what the user needs or wants to do. The story can then be elaborated upon to include the key benefit to complete the second half of the narrative. Thus, the mapping method is called user-story mapping because it can be used to evolve the verb phrases captured on the map into fully fleshed-out user stories that can be discussed further, eventually paired with acceptance criteria, and added to the product backlog for prioritization and estimation.

When and How to Create a User Story Map

Story maps can be used at any point in the product-development process to drive discussion and align the team. You can create a story map to plot the experience for a new product, after initial discovery work, or for an existing product, after usability testing. In either case, the story map begins to illustrate solutions to the problems uncovered in the research. Once created, teams will maintain and refer back to their story map over time; they add to it, modify it to reflect the actual state of the product, and use it to guide what to work on and release in subsequent sprints.

Story maps are best constructed by small teams with representatives from product, UX, development, and QA working collaboratively to discuss and shape the product plan. It’s important to establish the following context before beginning:

User goals and needs: Outline what users are trying to do, why the product or feature(s) you are story-mapping are important, and what real problems you’re solving.

Scope of the story map: State whether your story map will portray a current or a future iteration of the product and whether you will map an entire product, just one feature, or a section of the experience. Use caution in mapping massive products; breaking story maps up into manageable scopes and segments is often better than tackling an entire large-scale product in one story map. Being transparent about the scope of the story map upfront helps the team stay on topic and on task.

Outcomes: Talk about what your users will be able to do as a result of launching the product or feature outlined in the map. This information will help the team maintain focus on the outcomes instead of getting too bogged down by specific solutions and tools. Focusing on outcomes also sets realistic beginning and end points for the map.

To create a story map, in-person teams use sticky notes, white boards, and open wall space; remote teams can take advantage of video conferencing and collaborative spreadsheets, presentation slides, or web-based programs. Everyone should work together on the map; no one person or role should dominate the others.

Assign differently colored sticky notes (whether real or virtual) to each row of activities, steps, and details to keep the story map visually organized. It’s also important to frame your activities, steps, and details by what the user is doing at that particular point in the product, not what the product is technically doing for the user. For example, if you were creating a digital product using artificial intelligence and machine learning, a step in the story might display as Share preferences, not Train the AI.

Remote teams can use collaborative tools and video conferencing to build a story map.

Teams working remotely can rely on videoconferencing and collaborative, web-based tools such as CardBoard, spreadsheets, or even presentation slides to create a story map together.

User-Story Mapping vs. Customer-Journey Mapping

When we introduce user-story mapping during our UX Conference course Lean UX & Agile, practitioners often ask how it differs from customer-journey mapping. The key differences are that each map type is visualized from a different perspective and they’re used for a different purpose. A customer-journey map takes the user’s perspective, showing the steps she goes through to complete a goal, coupled with her thoughts, emotions, channels, and devices used along the way.

A user-story map takes the perspective of the product. It aims to guide the planning and implementation of features and functionality to solve users’ problems. Put simply, a user-story map connects what we uncover in customer-journey mapping to what we’re going to intentionally do about it in the product we create, beyond listing out general ideas and opportunities.

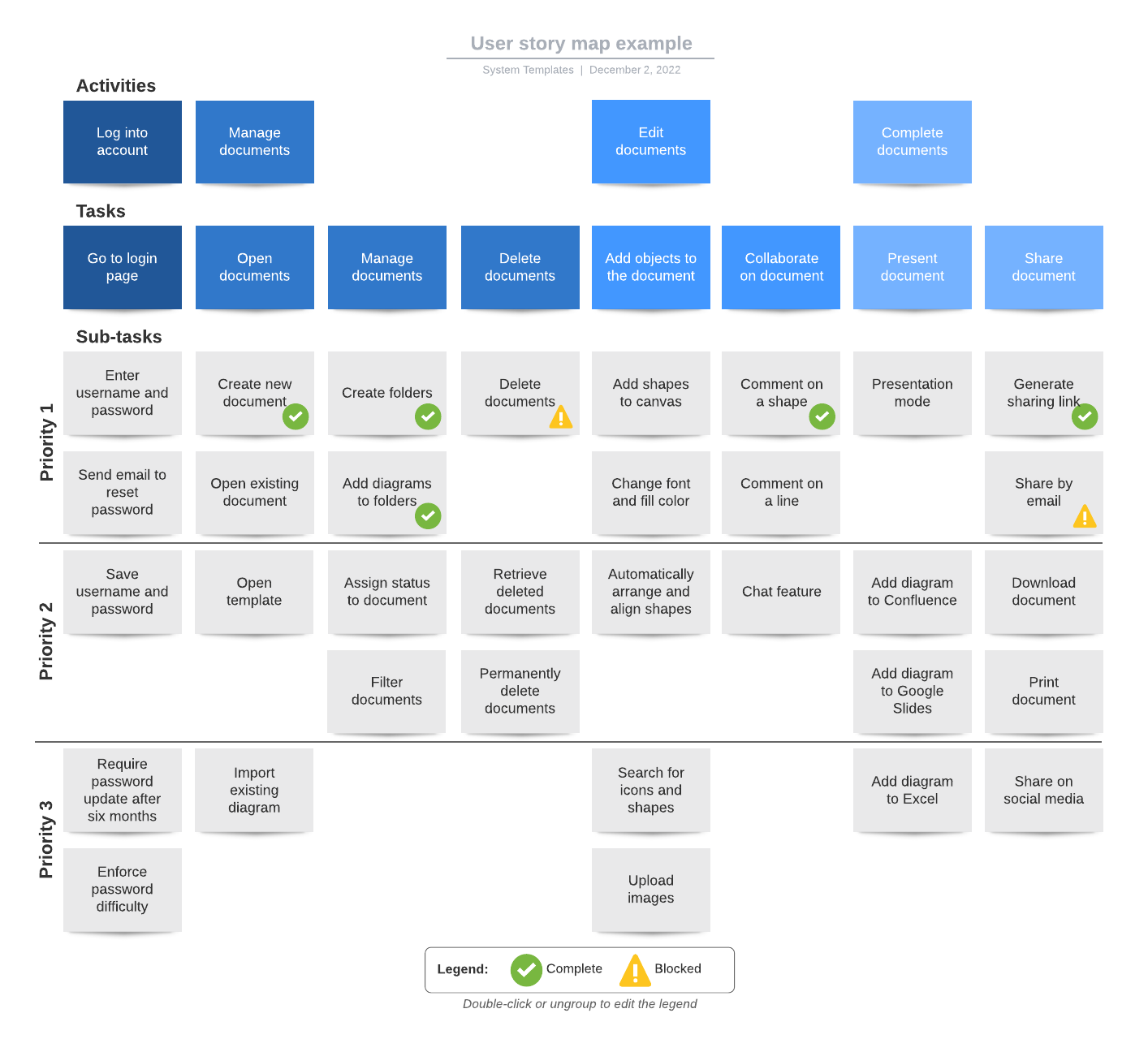
A customer-journey map can easily evolve into a user-story map by adding the activities, steps, and details. Similarly, a user-story map can morph into a customer-journey map if the users’ context, thoughts, and feelings are added. These two map types can work well when combined, but are also effective when used independently, as the research methods used to inform and create each of them are often the same.

Customer journey maps are from the perspective of the user.

Customer-journey maps are from the perspective of a person, depicting the journey she goes through to complete a task, along with her thoughts, feelings, and channels used.

User-story maps are from the perspective of the product.

User-story maps are from the perspective of the product and communicate users’ granular interactions therein. They’re used for alignment and tactical planning for developing and releasing features and product iterations that aim to solve the problems uncovered in the journey map.



User-Story Maps in Agile

User-story maps support the success of Agile product-development teams for several reasons:

Improved collaboration and team alignment. User story maps ease conversations and collaboration around what’s to be built, why, and for whom. They help everyone arrive at shared understanding and direction much more efficiently than creating and reading a 500-page document.

Facilitating the creation and expansion of the product backlog. The second-level steps in your story map translate to epics in an Agile product backlog. An epic is a large user story that must be broken down into multiple, smaller user-stories and tasks. The third-level details in the story map are the inputs for those smaller user stories and tasks, albeit with a bit more detail and acceptance criteria added before they hit the product backlog.

Steps evolve into epics and details become user stories in an agile backlog.

Steps in the story map translate to epics, or groups of user stories in an Agile product backlog. Details are also further broken down into user stories and tasks with acceptance criteria and are added to the backlog.

Minimum-viable-product slicing and well-informed prioritization. Story maps help teams see what their minimum-viable-product release could or should include, along with how and when to release end-to-end product increments with specific goals and outcomes in mind. Teams will often draw release lines directly on the story map, moving the details included in each release above the corresponding lines and leaving deferred details for a subsequent release below. Those deferred details, along with any new activities, steps, and details added to the map, become candidates for future sprints and releases based on what the team learns and prioritizes over time.

MVP slicing in a story map based on a goal.

An example of how to slice a story map to represent a first release: Everything above the line would be included in a prototype to learn if users understand the process (goal) and can successfully deposit a check (outcome).

Support identification of risky assumptions. Creativity can sometimes get the best of us: when we’re working with sticky notes and stories, we may add risky elements to the map — items that are not backed up by user data, aren’t technically feasible, or take our project’s budget or timeline off course. A story map helps us see where these risks exist. We can deprioritize these risky stickies in the story map and replace them with other low-risk ideas with the same value proposition. In this way, we can learn from the leaner alternative first before investing further in sophisticated or time-consuming design and development.

Risky assumption in a user story map.

Story maps expose risky product areas that might offput users or take excess development time. Replace risky stickies with leaner alternatives that support the same value proposition. Experiment and learn how users react before investing additional time and effort to develop the full feature.

Story maps encourage productive, user-centered discussions about product creation, improve visibility for the backlog, and allow teams to see the bigger picture. If done properly, user-story maps reveal logical and releasable slices of product increments that meet users’ needs, while uncovering impacts and areas of risk ahead of development. This allows teams to learn early and often what works and what does not. Savvy teams use this knowledge to drive decisions about where to focus their time to maximize usability, value, and feasibility in subsequent iterations.

And finally, because Agile is all about embracing and reacting to change over following a concrete plan, story maps better facilitate efficient adaptation; it’s far easier to swap out sticky notes than it is to revise hefty requirements documents. Learn even more about how to create a user-story map, the benefits of collaborative user-story mapping, and how to use these maps in your Agile product-development practice in our Lean UX & Agile course at the UX Conference.