

# Story Map Concepts

Use simple maps to visualize the stories you tell about your software



## Users

A map tells a story about a type of person doing something to reach a goal. Make sure to include them in your map along with a little information about them.

Try using lightweight persona sketches to describe your users.

## User Tasks

User's tasks are short verb phrases that are the basic building block of a map. If I ask you what you did earlier today when using email, you'll likely respond with tasks like:

- Read an email message
- Respond to a message
- Mark a message as spam

## Goal-Level

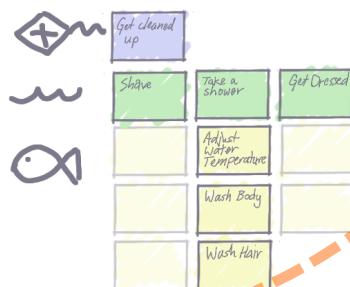
The actions that users take in order to reach their larger goals have a goal level themselves that's tied to user behavior.

**Summary:** lots of tasks done in support of a bigger goal.

**Functional:** I'd expect to complete this task before taking a break.

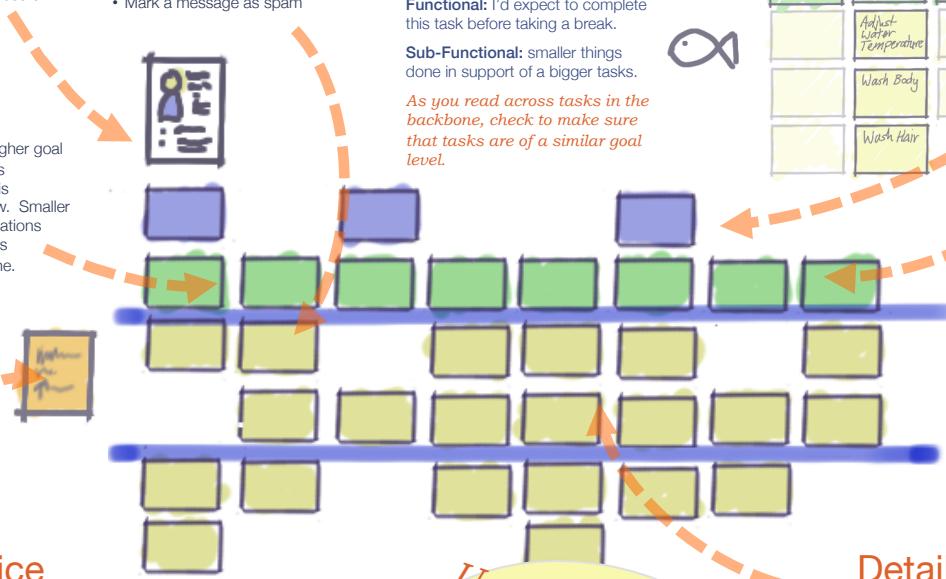
**Sub-Functional:** smaller things done in support of a bigger tasks.

*As you read across tasks in the backbone, check to make sure that tasks are of a similar goal level.*



## Backbone

Activities and tasks at a higher goal level give the story map its structure. The backbone is arranged in a narrative flow. Smaller sub-tasks, details and variations hang down to form the ribs connected to the backbone.



## Release Slice

Use a tape line to identify slices of tasks that users might use your software for to reach their goals. The smallest number of tasks that allow your specific target users to reach their goal compose a viable product release.

*Use release slices to identify small experiments, minimal viable product releases, or a "walking skeleton" version of your product.*

Identify the target outcomes of your slice in a sticky note or card to the left of the slice.

*User Tasks make great story titles!*

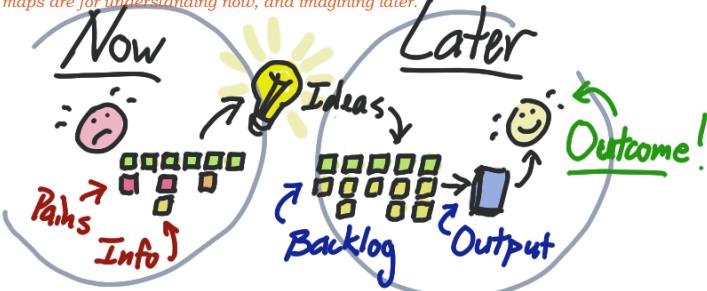
*Write short verb phrases on cards or stickies. Use them later as your story titles. If you use the story template to write descriptions, the task fits nicely right after "I want to," the activity fits nicely right after "so that..."*

## Map Now & Later

Use a map to describe the world as it is today. Inject pains, joys and rewards, details and observations, and solution ideas.

Evolve the map using design and discovery to describe the behavior you expect users to have in the future.

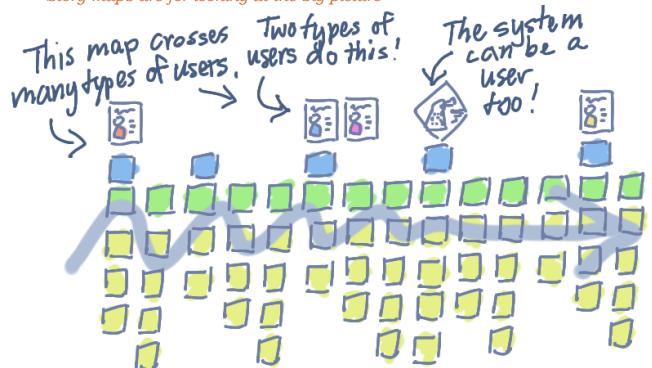
*Story maps are for understanding now, and imagining later.*



## Map the Whole System

Map a whole process as it crosses through a number of types of users

*Story maps are for looking at the big picture*



# Story Map Process

The story map evolves with your understanding of your users and your product solution



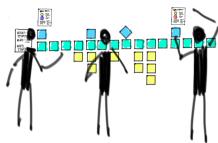
## 1 Frame

Before mapping, create a short product or **feature brief** to frame and constrain what you map. Think of this as the big story.

**What** names the product, feature to add to a product, or problem you'd like to solve.

**Who** names the different types of users who will use the product, and the "chooser" or customers who will buy it. For each user and chooser state the benefit they get from using the product.

**Why** describes the benefit your organization gets by building the product or feature. Say what users do and how their use leads to increased revenue or reduced costs.



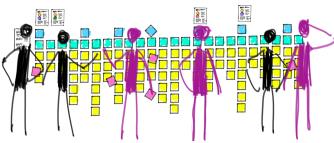
## 2 Map the Big Picture

**Focus on getting the whole story.** Think "mile-wide, inch deep". The activities and high-level user tasks that tell the whole story form the backbone of your story map.

**Start with the user type most critical to your product's success.** Imagine a typical day in your user's life with your new product. Map the steps they take as user tasks left to right.

**Identify user activities** – groups of tasks that work together to support a common goal. Activities often emerge after you see more of the story.

**Add in additional users.** As you follow the typical use of your product, you may find other types of users enter your story. Continue modeling their story left to right.

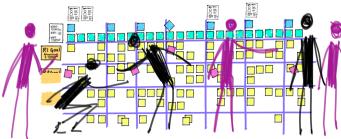


## 3 Explore

**Fill the body of your story map** by breaking down larger user tasks into smaller subtasks and user interface details. During this phase you'll add cards, split one card into two, rewrite cards, and reorganize them.

**Use this phase to think "blue sky"** about all the great possibilities. Use this time to think of everything that could go wrong. Don't worry if your ideas are "in or out of scope." You'll deliberately move things out of scope later.

- Play "wouldn't it be cool if..." to help think of great product ideas.
- Look for variations: What else might users of the system have done?
- Look for exceptions: What could go wrong, and what would the user have to do to recover?
- Consider other users: What might other types of users do to reach their goals?
- Add in other product details like: Description of proposed UI; Business rules; Data elements



## 4 Slice Out Viable Releases

**Slice your map into holistic product releases** that span the users and use of the product. These slices form an incremental product release roadmap where each release is a minimal viable product release.

**For each release name the target outcomes and Impact.** Outcomes and impact says how this release contributes to the overall goal in the "big why" that motivates building the product, and how users will behave in a way that helps us reach the goal.

**For each release, identify product success metrics.** Answer the question: "what would we measure to determine if this product was successful?" Ideally you'll find specific changes in user's behavior as they use the product the way your story map imagines.

## 5 Slice Out a Development Strategy

**Slice the first release of your map** into three or more delivery phases that allow you and your team to learn fast and avoid risk. Think of the opening, mid, and end-game phases of a chess game.

This development strategy will help you release the best product possible in the time constraints you have.

• **Opening Game** builds a "functional walking skeleton" – the simplest possible functional version of the product. As you finish "Opening game" vet the product with users and other stakeholders. Begin validating performance and scalability.

• **Mid Game complete all major functionality** and makes existing functionality richer and more complete. Continue user testing and leverage feedback to adjust the product. Continue testing performance and scalability.

• **End Game refines the product** in preparation for release. Continuously assess release readiness based on your release level product goals. Count on unforeseen work to emerge during this last stretch of development.

**Plan the work necessary to refine stories.**

Workshop stories with developers and testers to work through details and agree on acceptance criteria.

**Plan development and testing.**

**Build and verify parts of working software.**

