

# 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

## 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.

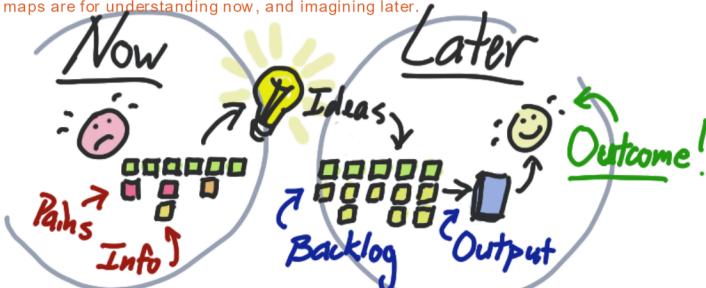
Identify the target outcomes of your slice in a sticky note or card to the left of the slice.

## 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.



©2013 Comakers LLC, www.comakewith.us, youshould@comakewith.us for more info

## Goal-Level

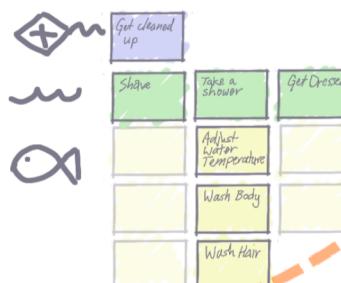
The actions that users take in order to reach their larger goals have a goal level themselves that's tied to user behavior.

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.



## Activities

Activities organize tasks done by similar people at similar times to reach a goal. For your email software activities might include:

- Going through my inbox
- Configuring my email client
- Organizing messages into folders

## Narrative Flow

The left to right axis in a story map is organized in the order you'd tell the story about your user to someone else.

Of course any specific user might choose to do different things in a different order. Use conversation to explain the details and variations.

If you're looking for the precision of a workflow model, flow chart, or UML model, then a story map isn't your best choice.

A story map will take lots of conversation to use effectively. But then that's the purpose of stories.

## Details, Details...

Break down high goal level tasks into:

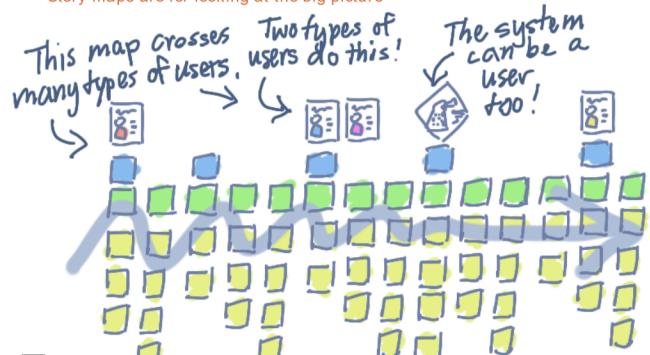
- Sub-tasks
- Alternative tasks
- Exceptions
- Details

Down in the details of the map, it's OK to include details about what UI might look like or what the system might do in the background.

## 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



This map crosses many types of users, users do this! Two types of users! The system can be a user too!

©2013 Comakers LLC, www.comakewith.us, youshould@comakewith.us for more info

# Story Map Process



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



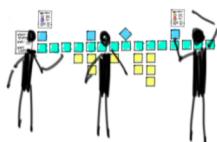
## 1 Frame EO 00

Before mapping, create a short product or feature brief to frame and constrain what you map. Think of this as the big story.<sup>[x]</sup>

What names the product, feature to add to a product, or problem you'd like to solve.<sup>[x]</sup>

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.<sup>[x]</sup>

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.<sup>[x]</sup>



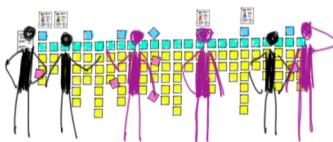
## 2 Map the Big Picture EO 00

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.<sup>[x]</sup>

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.<sup>[x]</sup>

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.<sup>[x]</sup>



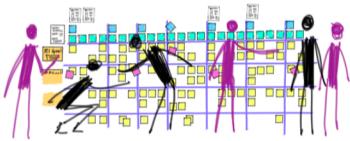
## 3 Explore EO 00

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.<sup>[x]</sup>

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.<sup>[x]</sup>

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

Involve others. Tell your product's story to others that understand users and use. They'll find holes in your story and help build it up. Tell your product's story to software developers. They'll point out risky or expensive areas, and add in great technology solutions.<sup>[x]</sup>



## 4 Slice Out Viable Releases EO 00

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.<sup>[x]</sup>

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.<sup>[x]</sup>

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.<sup>[x]</sup>

## 5 Slice Out a Development Strategy EO 00

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.<sup>[x]</sup>

• 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.<sup>[x]</sup>

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.<sup>[x]</sup>

This development strategy will help you release the best product possible in the time constraints you have.<sup>[x]</sup>

• 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.<sup>[x]</sup>

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.<sup>[x]</sup>

• 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.<sup>[x]</sup>

• 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.<sup>[x]</sup>

