Building Better Products Using

User Story Mapping

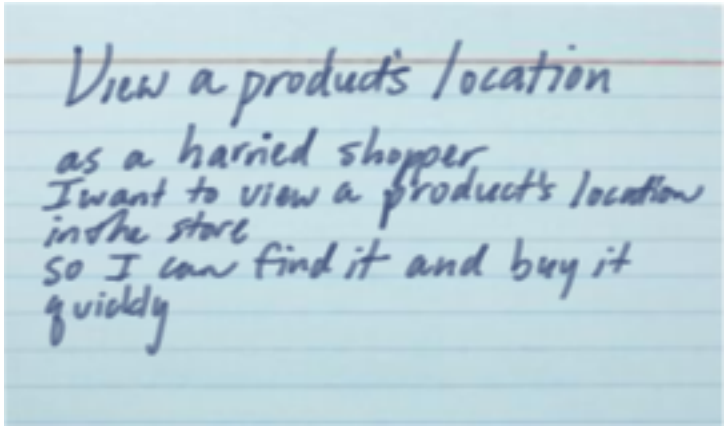
Jeff Patton
jpatton@acm.org
www.agileproductdesign.com

User Stories are multi-purpose

Stories are a:

- User's need
- Product description
- Planning item
- Token for a conversation
- Mechanism for deferring conversation

** Kent Beck coined the term user stories in Extreme Programming Explained 1st Edition, 1999*



Stories gain detail over time

Start with **a title**

Add a **concise description** often using this useful template:

As a [type of user]

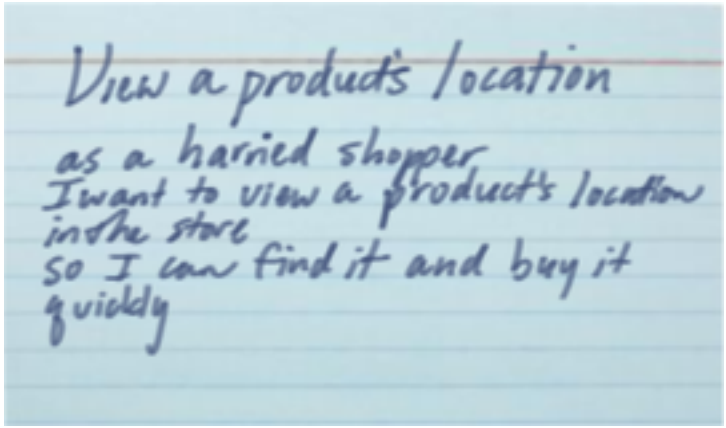
I want to [perform some task]

so that I can [reach some goal]

Remember — that's just a thinking template. No need to write all your stories this way.

Add other relevant **notes**, **specifications**, or **sketches**

Before building software write **acceptance criteria** *(how do we know when we're done?)*



Agile customers or product owner prioritize stories into a backlog

A collection of stories for a software product is referred to as the **product backlog**

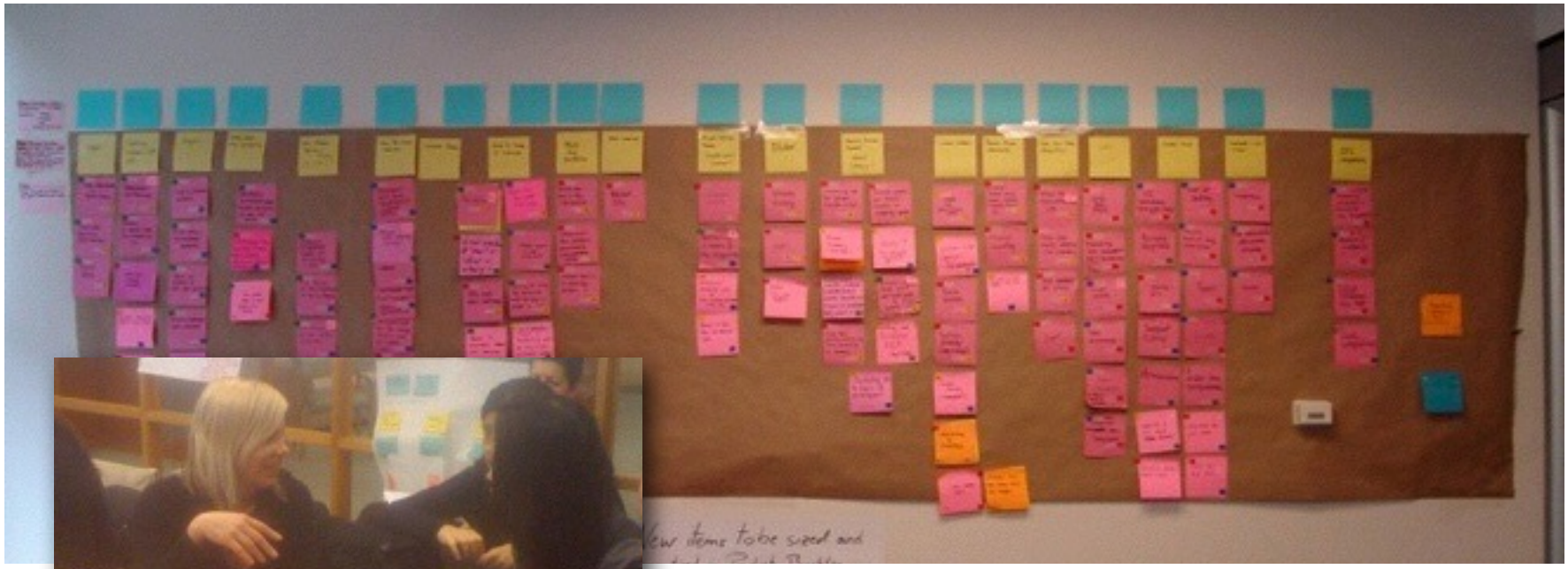
The backlog is prioritized such that the most valuable items are highest

User Story Mapping is an an approach to **Organizing** *and* **Prioritizing** user stories

Unlike typical user story backlogs, Story Maps:

- make **visible** the workflow or value chain
- show the **relationships** of larger stories to their child stories
- help confirm the **completeness** of your backlog
- provide a useful **context** for prioritization
- Plan releases in complete and **valuable slices** of functionality.

User Story Mapping is an an approach to **Organizing** *and* **Prioritizing** user stories



Story Maps support the primary intent of user stories, **rich discussion**

The foundational building block of a story map is the **user task**

People achieve goals through interaction

problem or goal

How I'd like to feel, or what I'd like to achieve

goal evaluation

Is my goal met or problem resolved?

Take some action

action evaluation

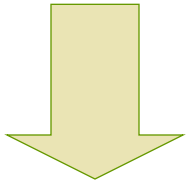
Did that action deliver the results I expected?

the world

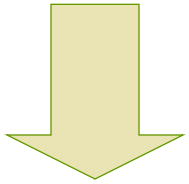
Information and tools

Think of three levels: goal, task, and tool

goal

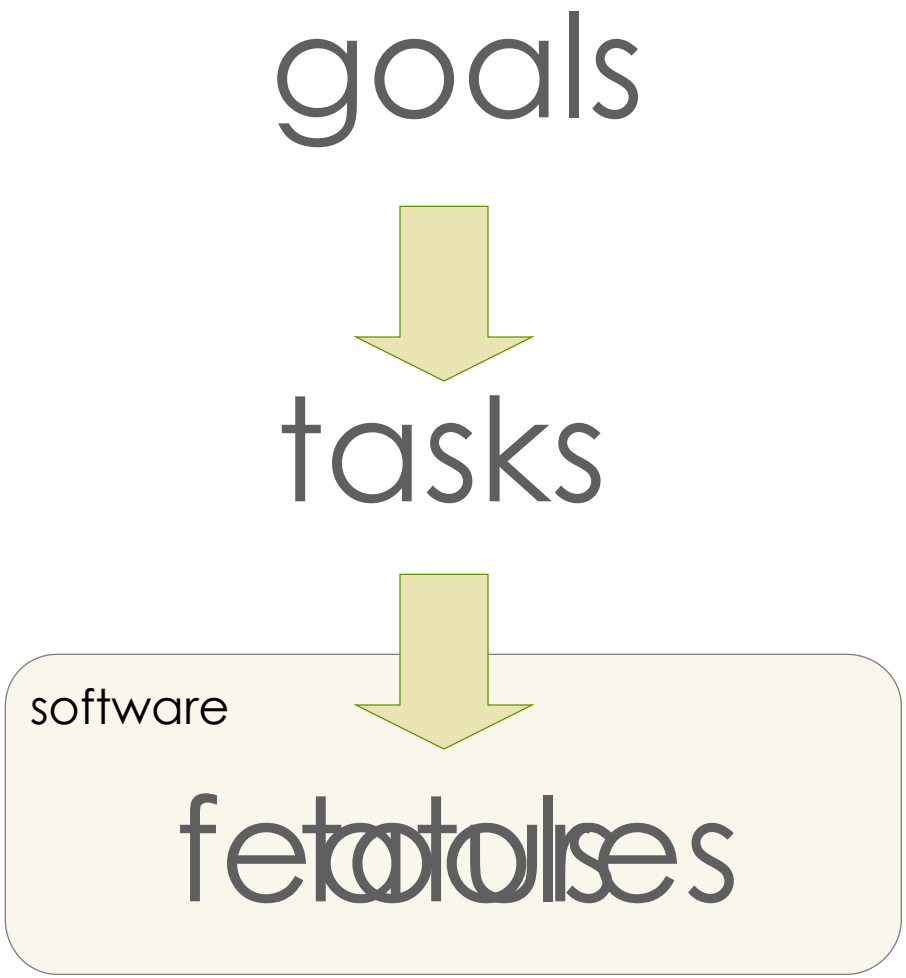


task



tool

Software contains features that support a variety of tasks and a variety of goals



User tasks are decompose to smaller tasks and organize into activities

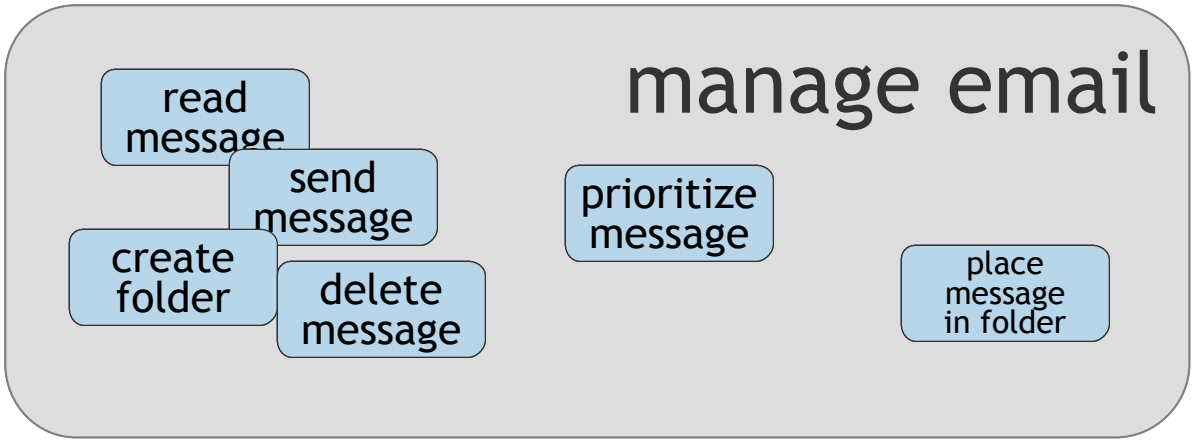
Tasks require **intentional action** on behalf of a tool's user

Tasks have an **objective** that can be completed

Tasks **decompose** into smaller tasks

Tasks often **cluster** together into activities of related tasks

"Read an email message" is a task, "Managing email" is an activity.



Be sensitive to your user task's "altitude"

Too abstract

Activity or "Kite level"

Longer term goals often with no precise ending. I'll perform several functional tasks in the context of [illegible] activity

Think about user experience at this level

Functional or "Sea level"

I'd reasonably expect to complete this in a single sitting

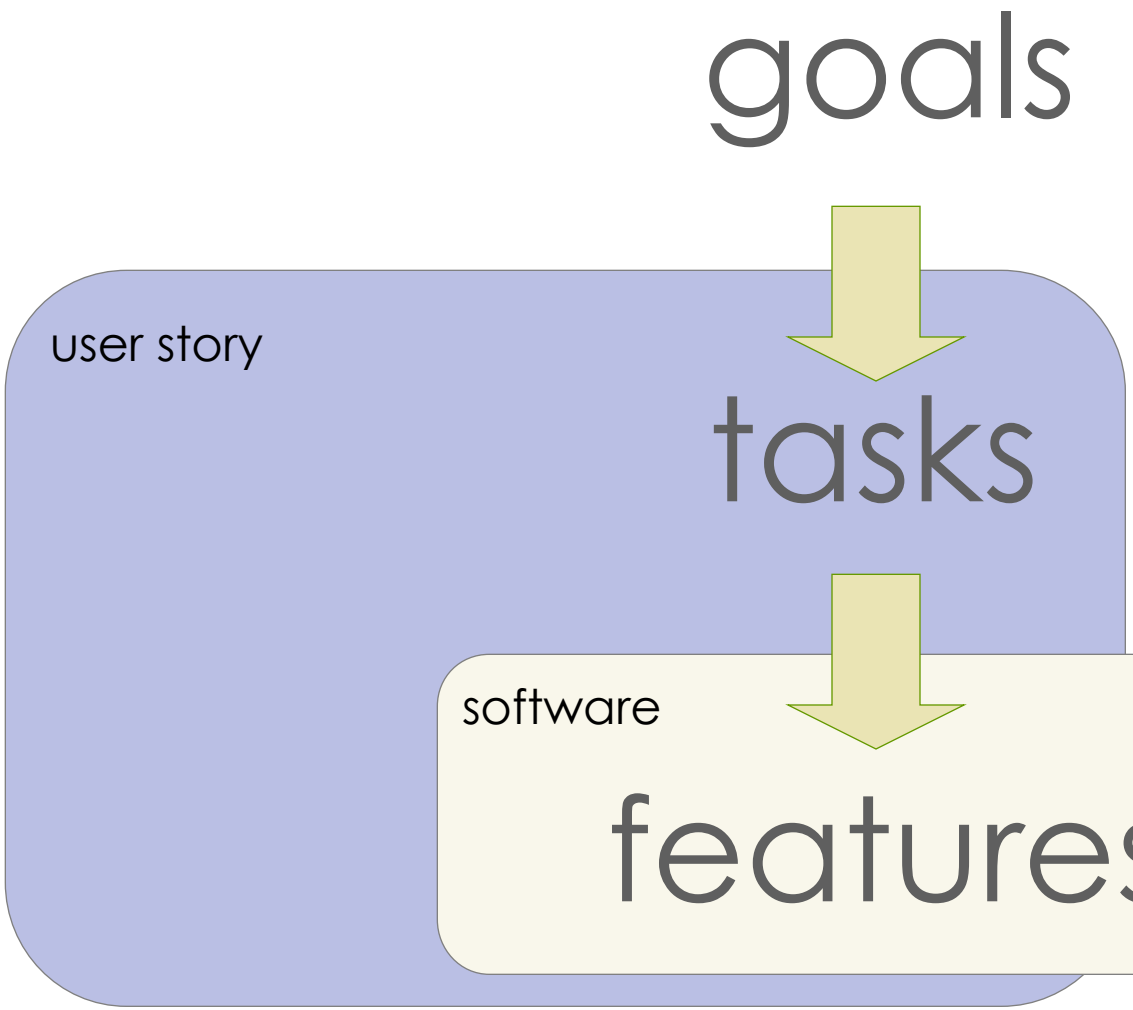
Sub-Functional or "Fish level"

Small tasks that by themselves don't mean much. I'll do several of these before I reach a functional level goal

Too detailed

* from Cockburn's Writing Effective Use Cases

In practice user stories may be written to describe user tasks or the tools that support them



More task-centric:

As a weekend gardener
I want **to** **dig a hole**
so that I can plant a tree

More tool-centric:
(or feature-centric)

As a weekend gardener
I want **a** **shovel**
so that I can [dig a hole to] plant a tree

What are the goals & tasks?

Business goals or pain points?

Types of users using this system?

User's goals or pains?

How will users of the system reach their goals?

Start with an understanding or user experience

A user scenario is a simple way to think through user experience concretely

A user scenario tells a story about a main character with a problem or goal

- Describes how that character reaches their goal
- contains important facts
- describes external context
- describes goals and concerns of our character

include interesting plot points that help us envision important aspects of the system

A scenario can gloss over uninteresting details

Imagine the user experience as a scenario

Separate your team of four into two pairs

One person imagine out loud the experience of someone using the kiosk. Think about:

- Typical use, including typical problems
- Interesting plot points
- Goals and pains of your user

The other ask questions to better understand

After 5 minutes of discussion write out the scenario

Pair **one** think about **Carol**:

casual browser

"I want to find something good from a band I heard on the radio."

Goal: : have fun finding something interesting

Pair **two** think about **Isaac**:

Impatient buyer

"I wan to find a specific hard to find foreign film."

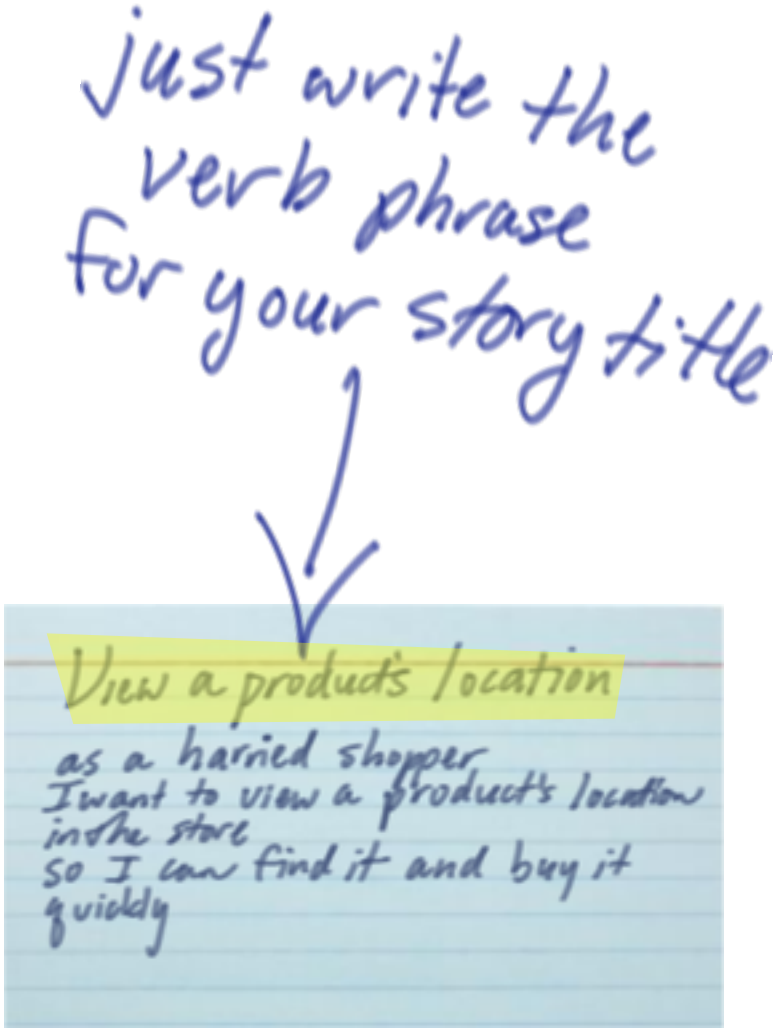
Goal: : Find it and get out quickly without wasting my time

Extract task-centric user stories from your user scenarios

Come back together as a team

Reviewing each others scenarios, extract task-centric stories using yellow stickies

Write only the tasks **verb-phrase** for your title



Organize user stories into a map that communicates experience

By arranging activity and task-centric story cards spatially, we can tell bigger stories

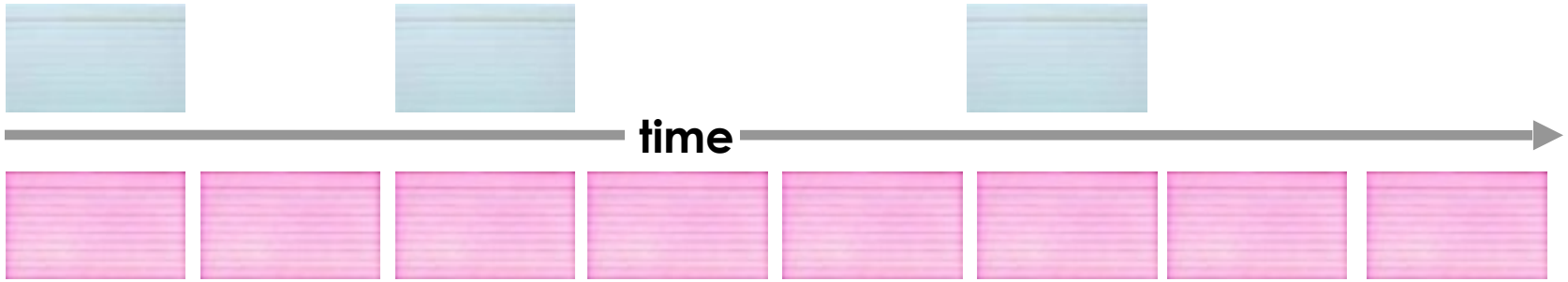
Tell a big story of the product by starting with the major user activities the kiosk will be used for

- Arrange activities left to right in the order you'd explain them to someone when asked the question: "What do people do with this system?"

By arranging task-centric story cards spatially, we can tell bigger stories

Add task-centric stories in under each activity in workflow order left to right.

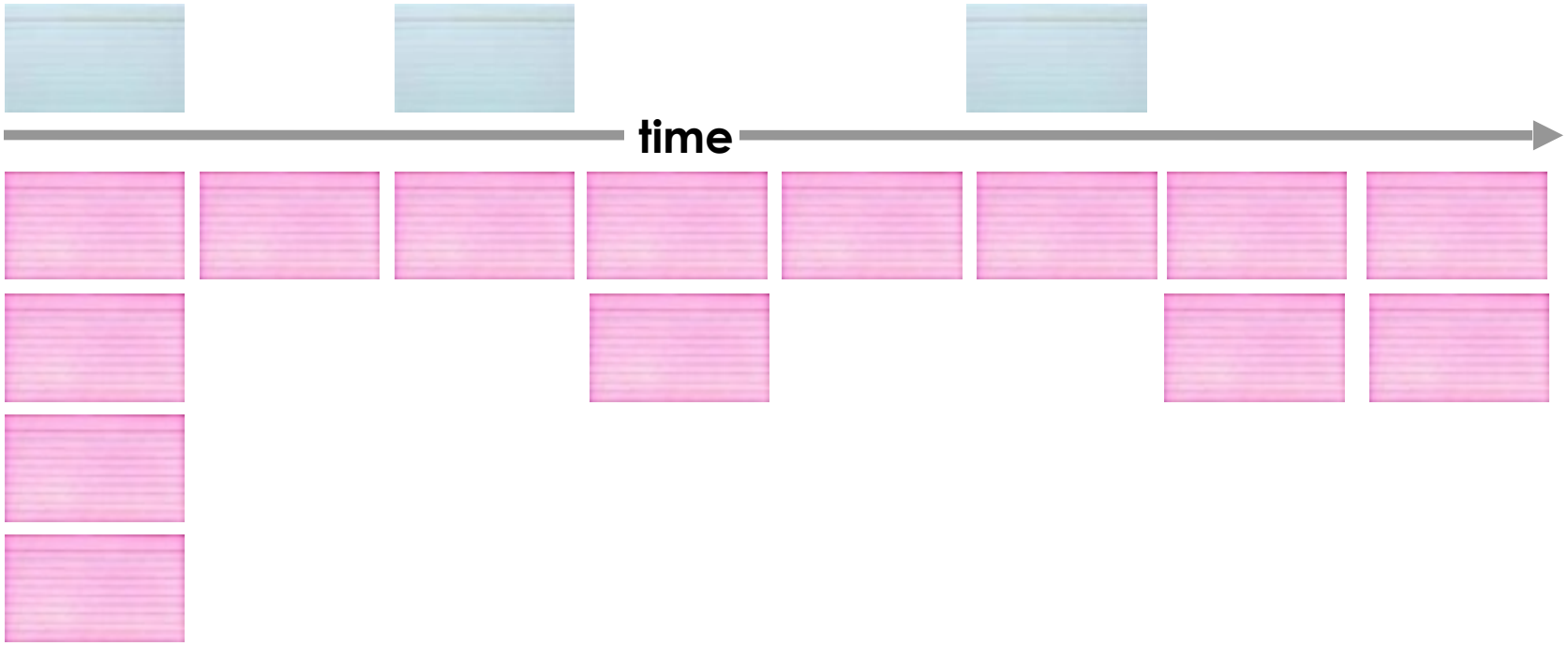
- If you were to explain to someone what a person typically does in this activity, arrange tasks in the order you'd tell the story. Don't get too uptight about the order.



By arranging task-centric story cards spatially, we can tell bigger stories

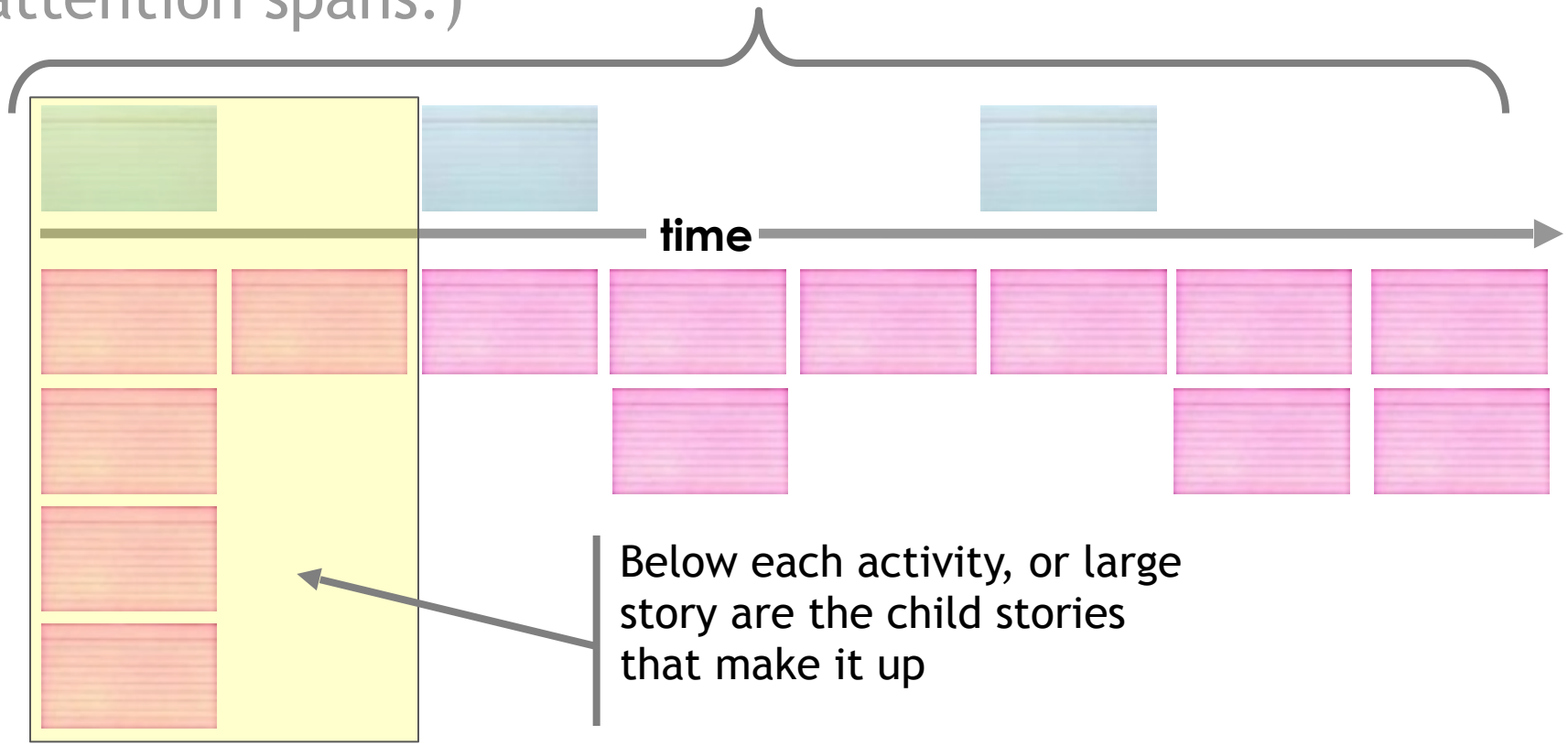
Overlap user tasks vertically if a user may do one of several tasks at approximately the same time

- If in telling the story I say the systems' user typically "does this or this or this, and then does that," "or's" signal a stacking vertically, "and then's" signal stepping horizontally.



The map shows decomposition and typical flow across the entire system

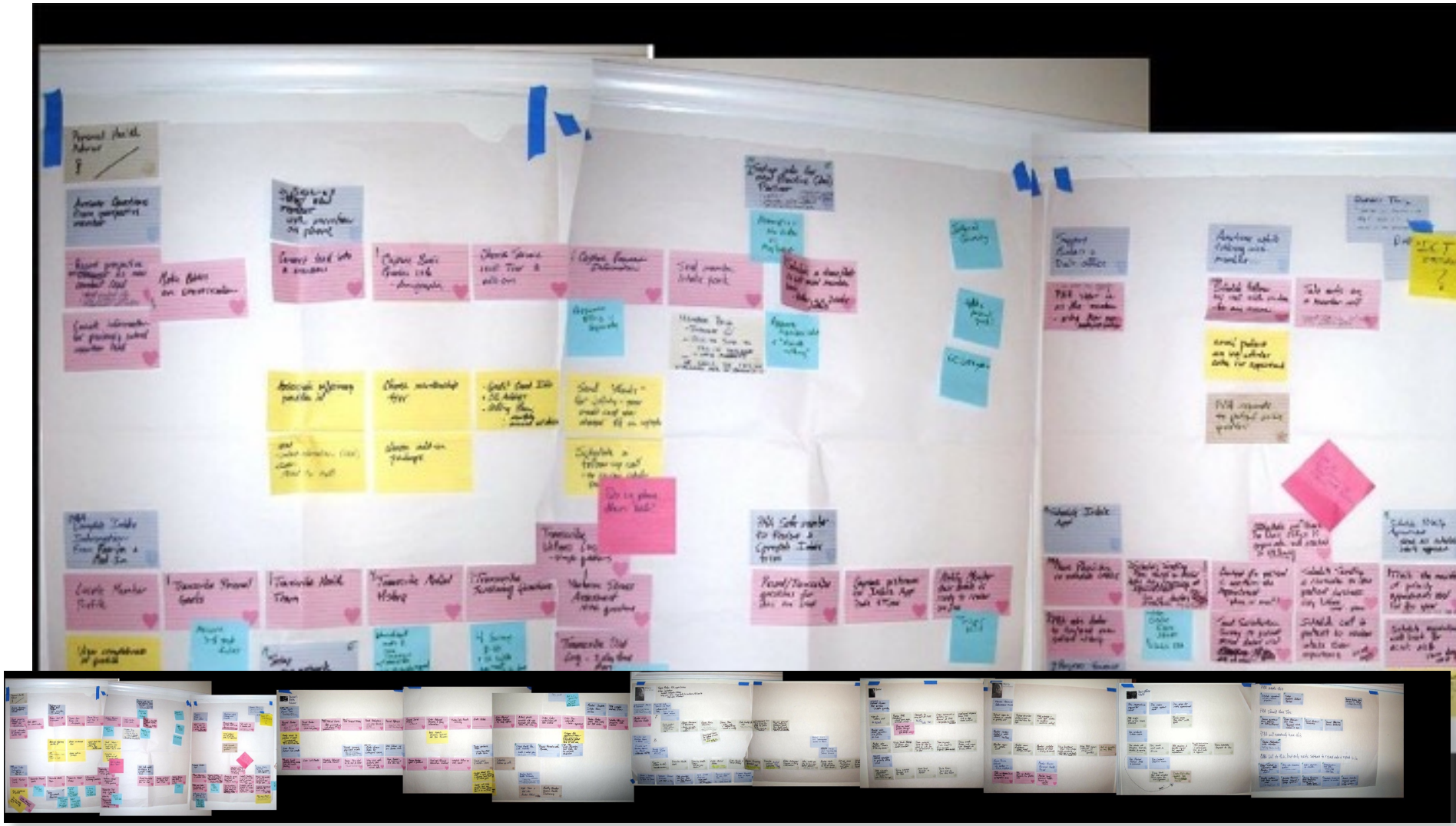
Reading the activities across the top of the system helps us understand end-to-end use of the system. (Talk through just these when talking with people with short attention spans.)



Building a story map helps facilitate discussion – but requires a bit of space

Gary Levitt, owner & designer of Mad Mimi

A story map for a reasonable sized system can fill a room

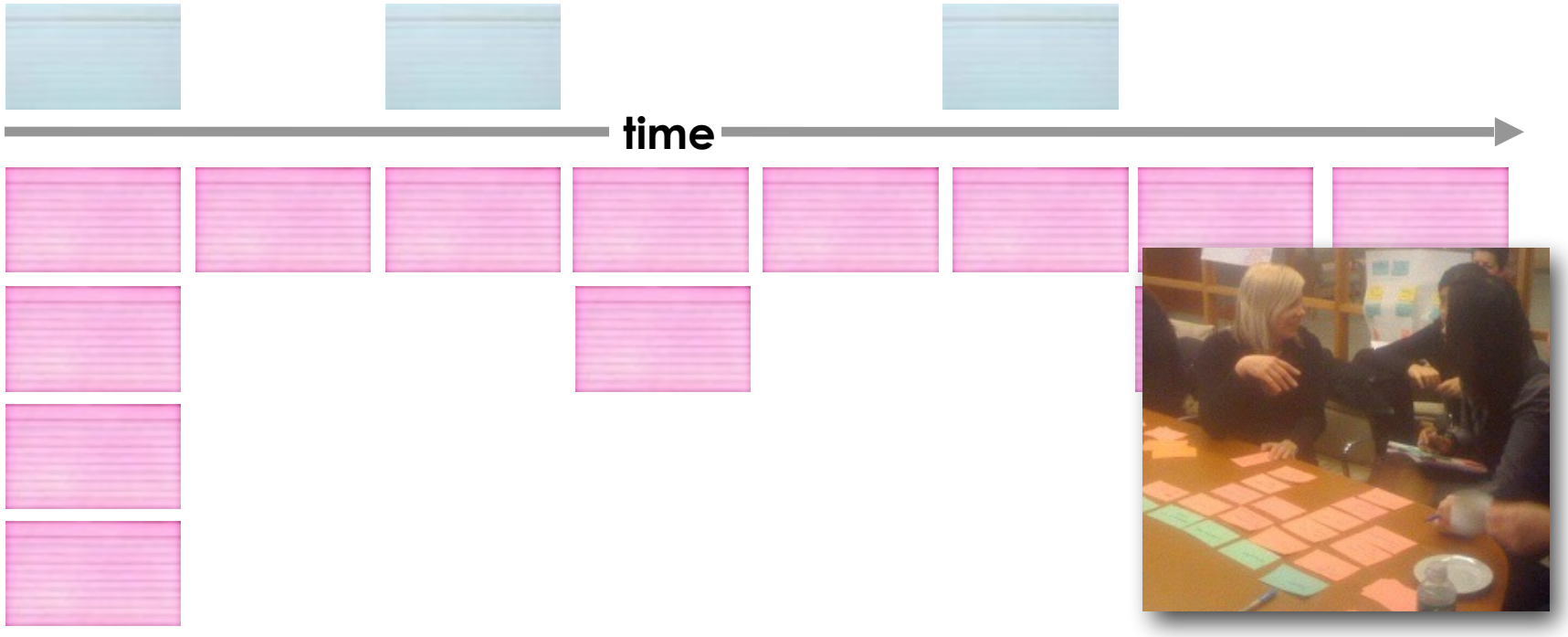


Assemble your harvested task-centric stories into a simple story map

Work as a team to organize your stories

Look for activities: tasks that seems similar, are done by similar people in pursuit of similar goals

Use a different color sticky to label activities

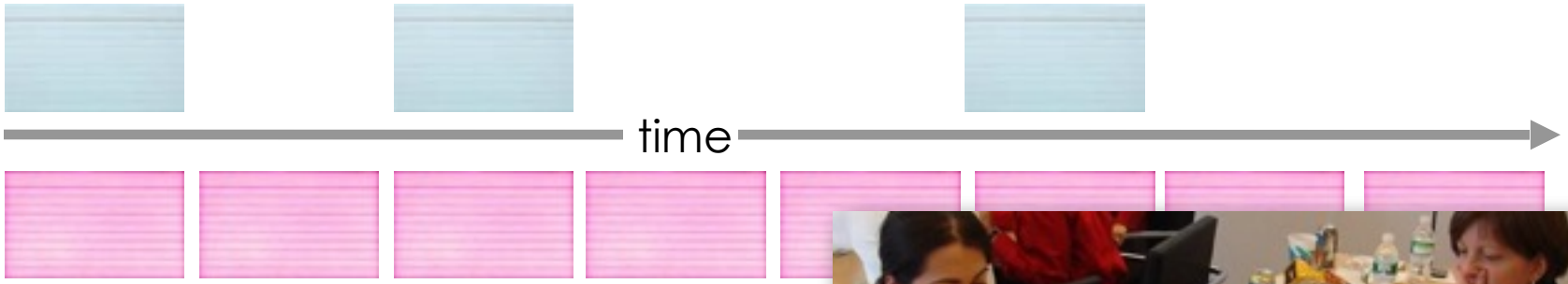


Discuss, fill in, refine the map, and test for completeness

Once you've got the idea down, it's quick to record stories as you discuss the experience with users

Discuss the steps of the process with candidate users

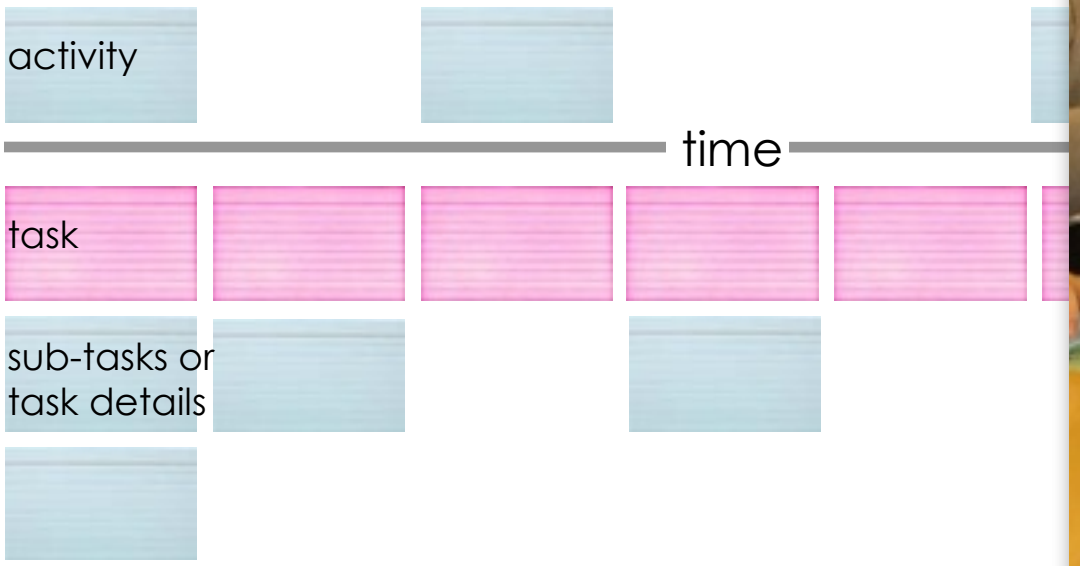
- Record tasks as they say them
- Rearrange tasks and insert tasks as you clarify the big story
- Add activities as you identify them from discussion



As details emerge in conversation, trap them under their associated task cards

Record details so they're not lost, and so those who you're working with know that you're listening

- Consider tucking them under tasks cards to "hide them" from the discussion

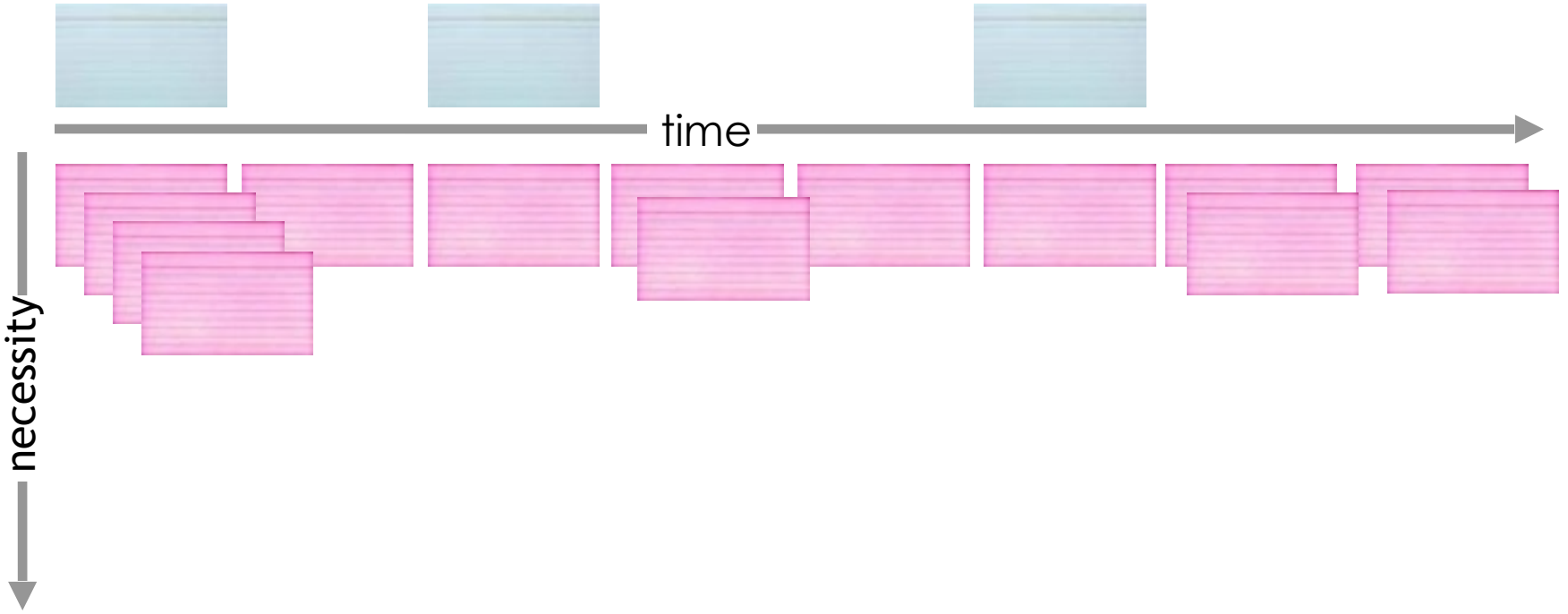


By arranging activity and task-centric story cards spatially, we can tell bigger stories

Add a vertical axis to indicate necessity

Move tasks up and down this axis to indicate how necessary they are to the activity.

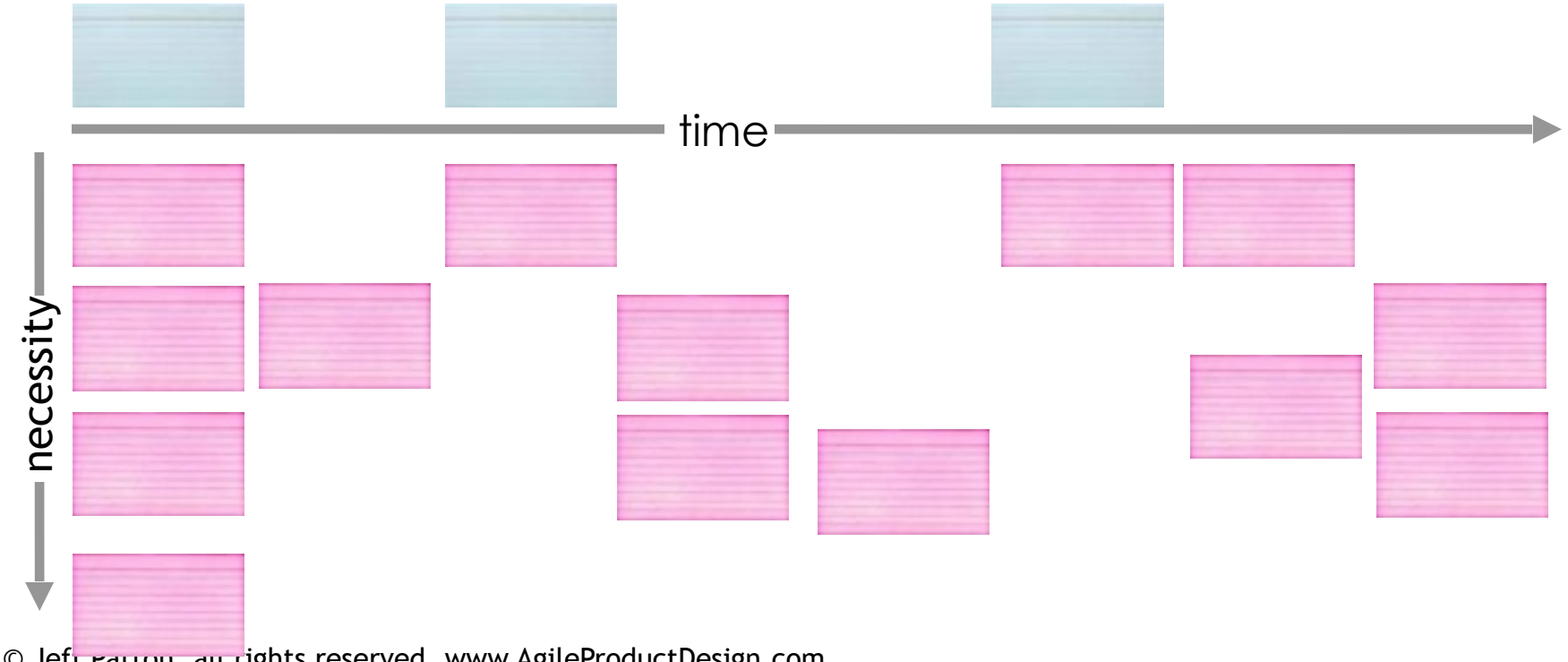
- For a user to successfully engage in this activity, is it necessary they perform this task? If it's not absolutely necessary, how critical is it?



By arranging activity and task-centric story cards spatially, we can tell bigger stories

Test the Story Map by telling bigger stories with it

- Choose an activity to start with
- When reading left to right use the conjunction “and then” to connect cards in the story
- With cards in the same row use “or” to connect cards in the story
- For cards below the top, “absolutely necessary” axis, use the phrase “might optionally” to communicate optionality
- Chose a concrete user name to help tell the story



By arranging activity and task-centric story cards spatially, we can tell bigger stories

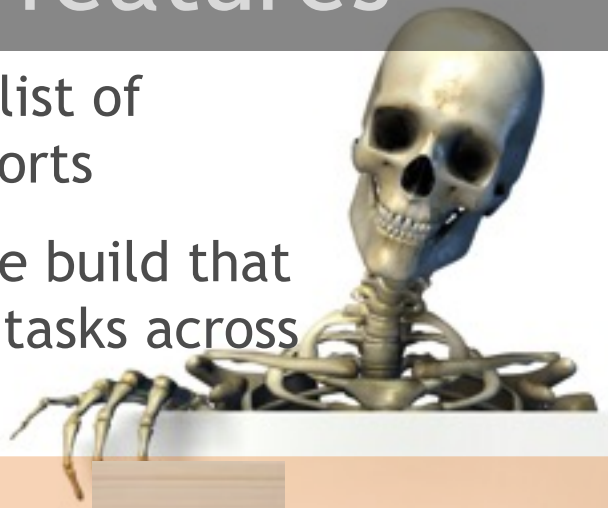
"Steve knows the title of what he's looking for. He steps up to the kiosk and **searches by title**. Optionally he might have **searched by artist**. After seeing titles that match what he typed in, Steve **views the price new and used**, and then **views the status** – whether it's in stock or not. He notices it's in stock as both new and used, so then Steve **views the location** in the store for the used title."

Discussions over story maps help drive out more details

The user story map contains two important anatomical features

The **backbone** of the application is the list of essential activities the application supports

The **walking skeleton** is the software we build that supports the least number of necessary tasks across the full span of user experience



Using discussion, fill in your story map

Work together as a team

Look for **alternative tasks**

- What else might users of the system have done that didn't come up in your scenarios?

Look for **exceptions**

- What could go wrong, and what would the user have to do to recover?

Consider **other users**

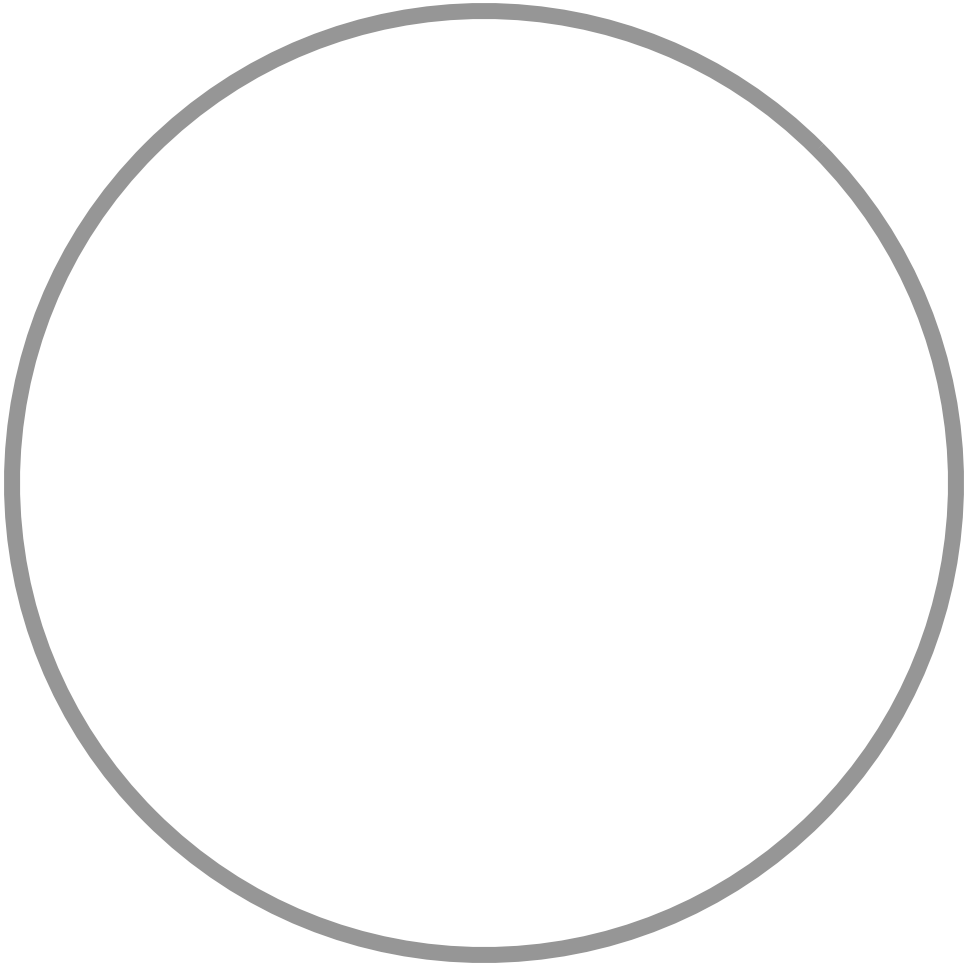
- What might other types of users do to reach their goals?

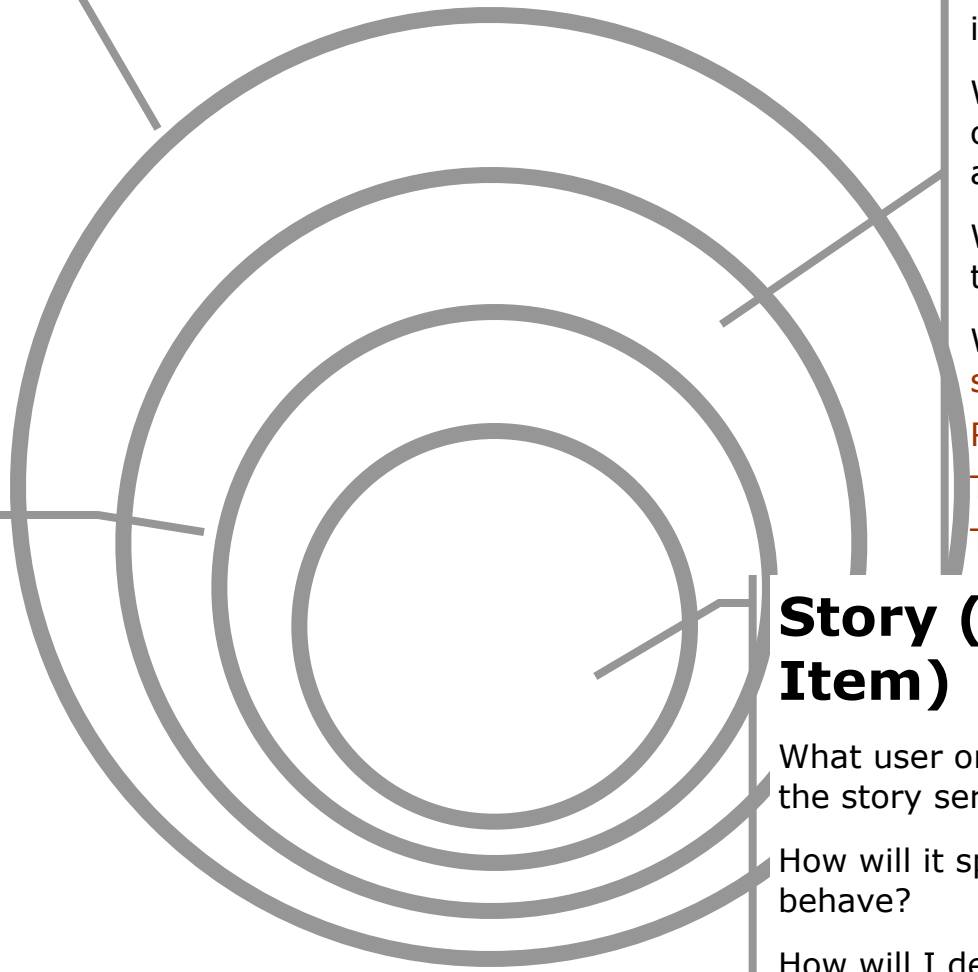
Knit al these additional stories into your map

Slice the map to find ideal incremental releases

Agile teams plan product construction in layers

Agile teams plan product construction in layers





Product or Project

What business objectives will the product fulfill?

Product Goals

Product Charter

Customers

User Personas

Iteration or Sprint

What specifically will we build? (user stories)

How will this iteration move us toward release objectives?

Iteration Goal

Development or Construction Tasks

Release

How can we release value incrementally?

What subset of business objectives will each release achieve?

What user constituencies will the release serve?

What general capabilities (big stories) will the release offer?

Release Roadmap

Target Customers

Target Personas

Story (Backlog Item)

What user or stakeholder need will the story serve?

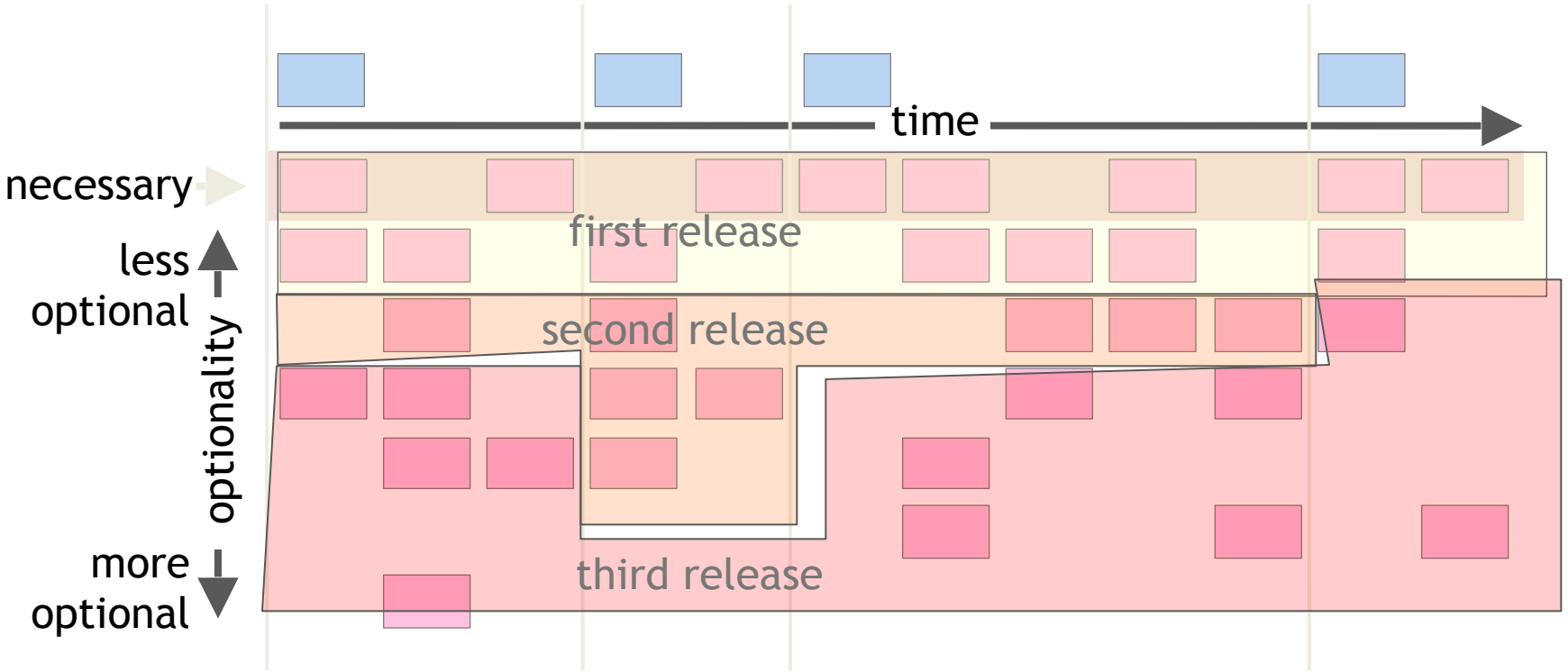
How will it specifically look and behave?

How will I determine if it's completed?

Story Details

Acceptance Tests

Given story map organized vertically by necessity, we need only slice to plan

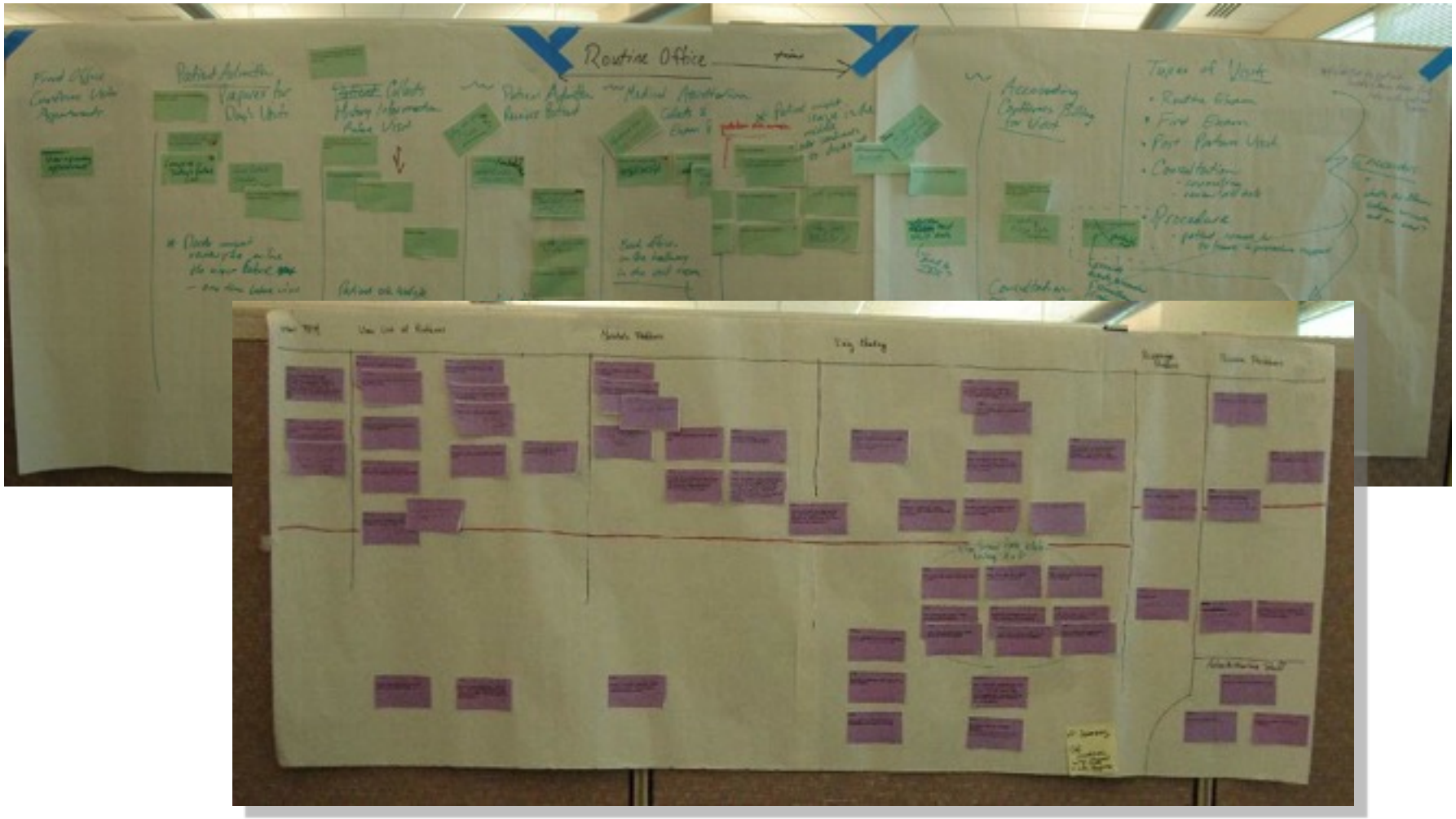


Choose coherent groups of features that consider the span of business functionality and user activities

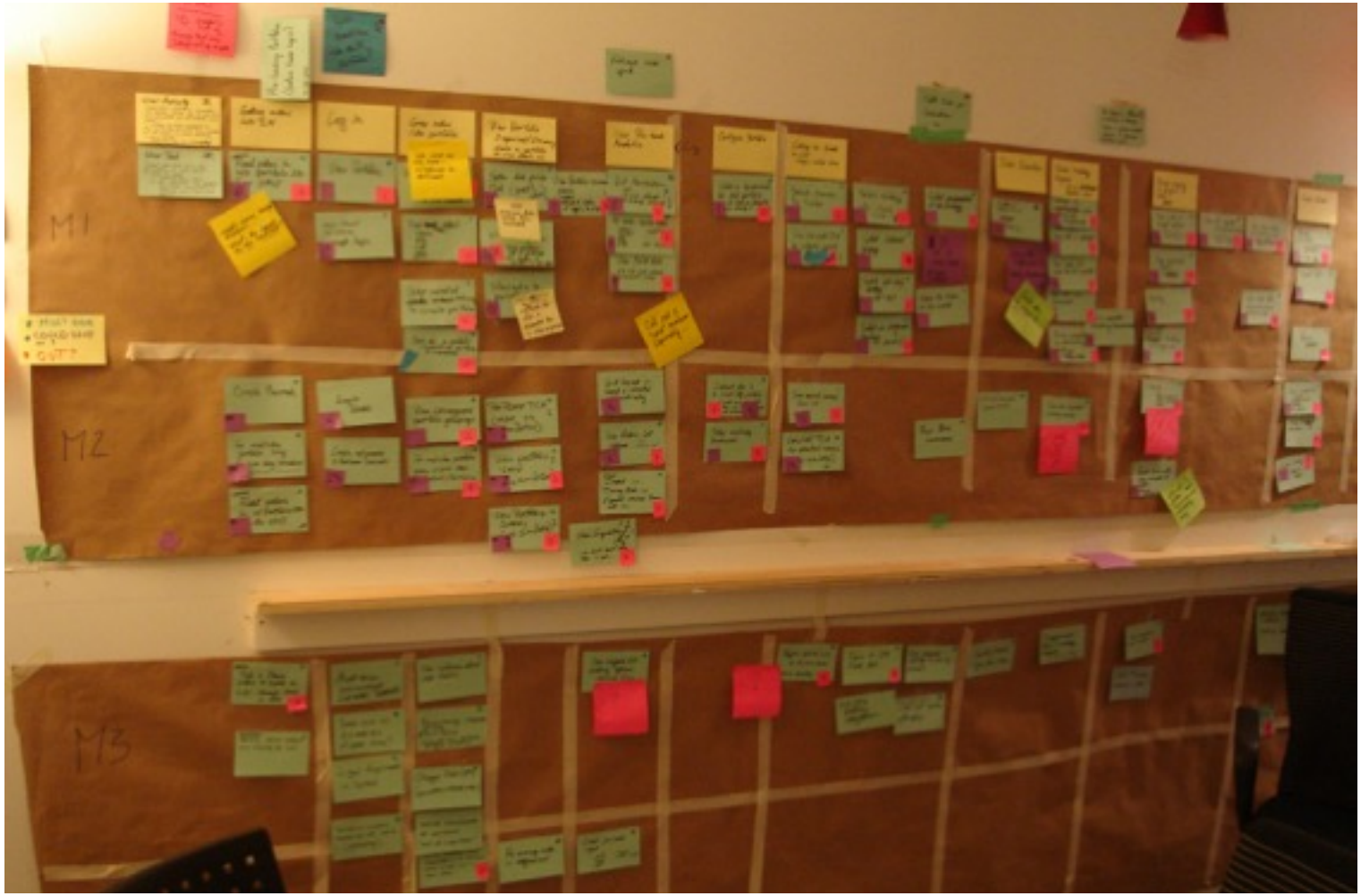
Support all necessary activities with the first release

Improve activity support and add additional activities with subsequent releases

Given story map organized vertically by necessity, we need only slice to plan



Adding tape lines to the wall lets participants organize stories into layers



Adding tape lines to the wall lets participants organize stories into layers

Planning incremental releases can be facilitated as a collaborative event

Guidelines for releasing on time

Thin stories aggressively during early sprints to build all essential functionality early.

Build up functionality only after all necessities are in place.

Protect time in the final sprints for product refinement.

Assess release readiness at the end of each sprint as part of product review.

Building Better Products Using

User Story Mapping

Jeff Patton
jpatton@acm.org
www.agileproductdesign.com