Creating User Story Maps

Outcomes and Objectives

- investigate story maps and activities
- create a draft story map for the project
- leverage GitHub project boards

What are Story Maps?

(freely available resource: User Story Mapping by Jeff Patton)

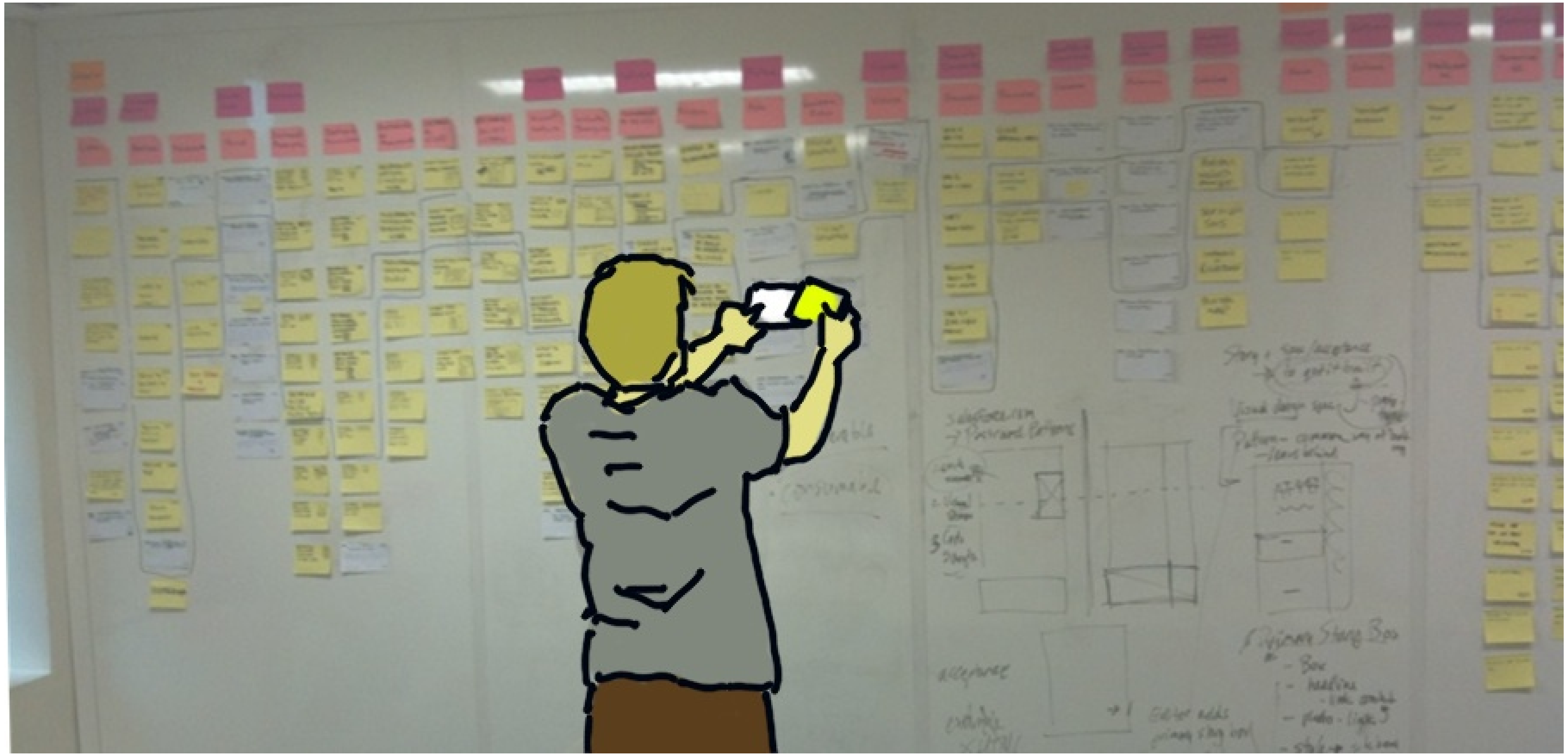
> Stories get their name from how they should be used, not what should be written.
>
> Story maps are for breaking down big stories as you tell them.

Backlogs are rarely *flat*, ranked lists of things to do. They are interconnected, dependent, multi-dimensional.

Story maps create 'narratives', scenarios of how the system should be used.

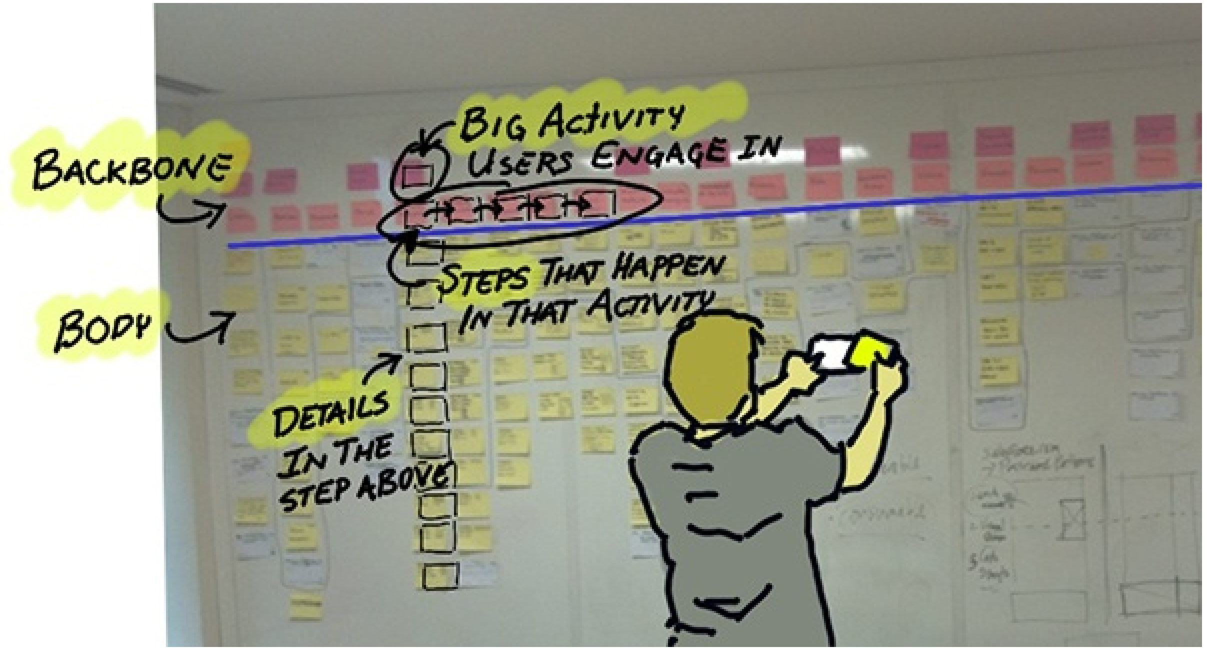
The problem

Imagine working on a system this size. Each yellow sticky represents maybe a day of work.

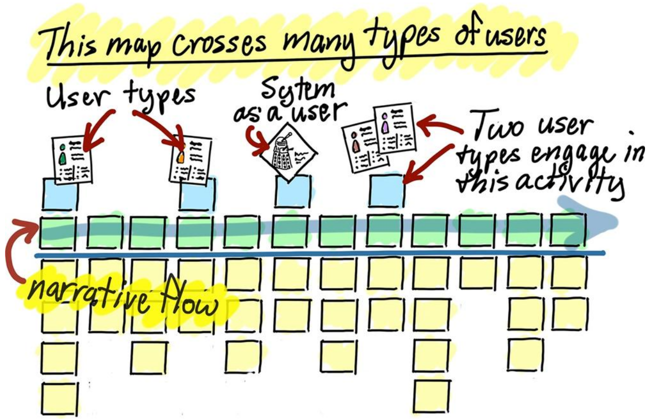


Backbone

From the top, story maps have **activities**, big 'chapter' narratives; then **tasks**, sub-sections of the chapter; and below that *backbone*, the different steps -*stories*- for that chapter.

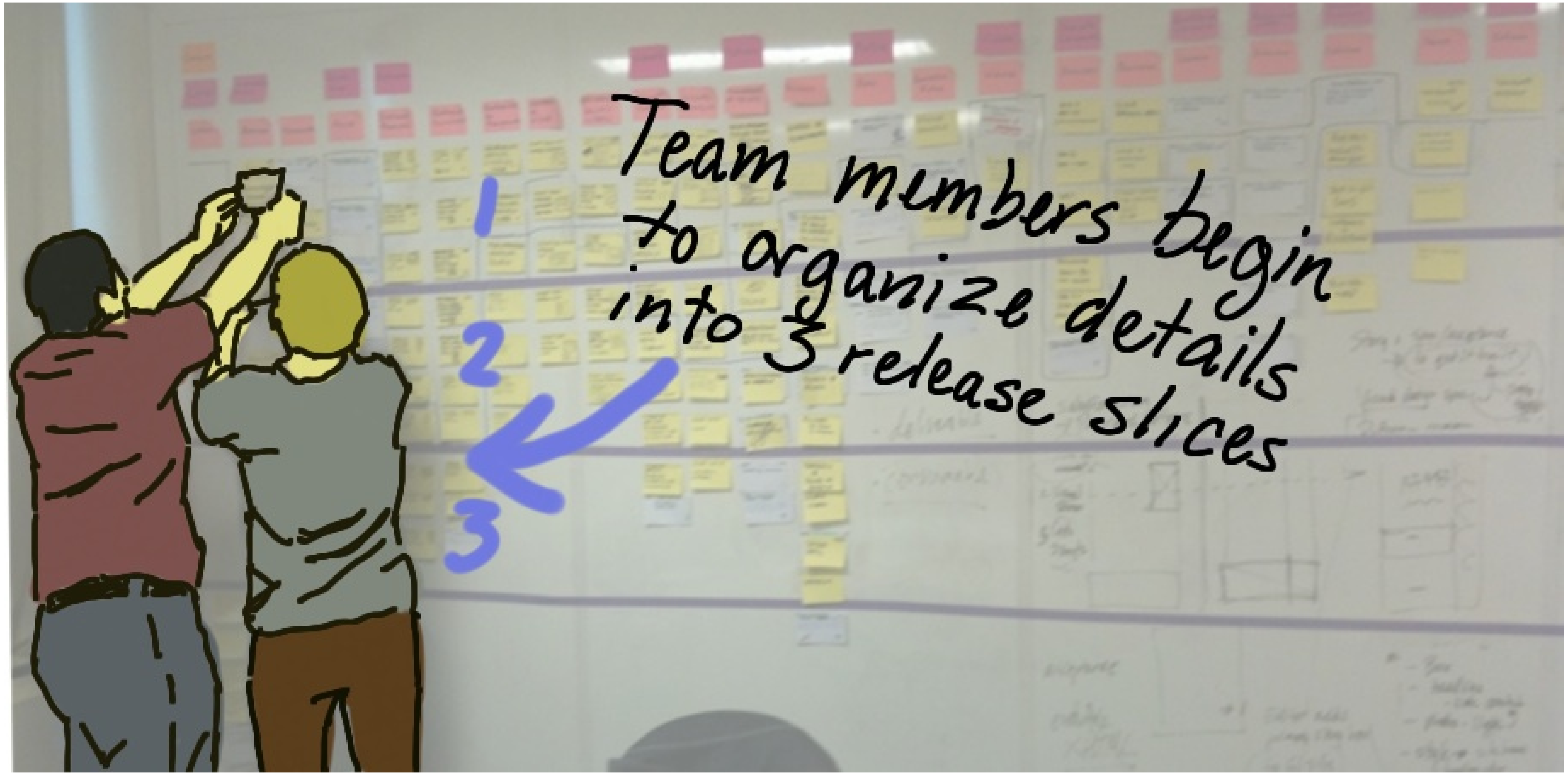


We can assign users to each 'chapter' or activity. The x-axis reflects the story flow (e.g., login to order completion email).


This map crosses many types of users
User types
Sytem as a user
Two user types engage in this activity
narrative flow

Release Planning

Horizontal slicing divides stories into release slices. The idea is that everything in the first slice is in the first release, and the product as a whole has a coherent narrative (the x-axis).


Team members begin to organize details into 3 release slices
1
2
3

Release Planning

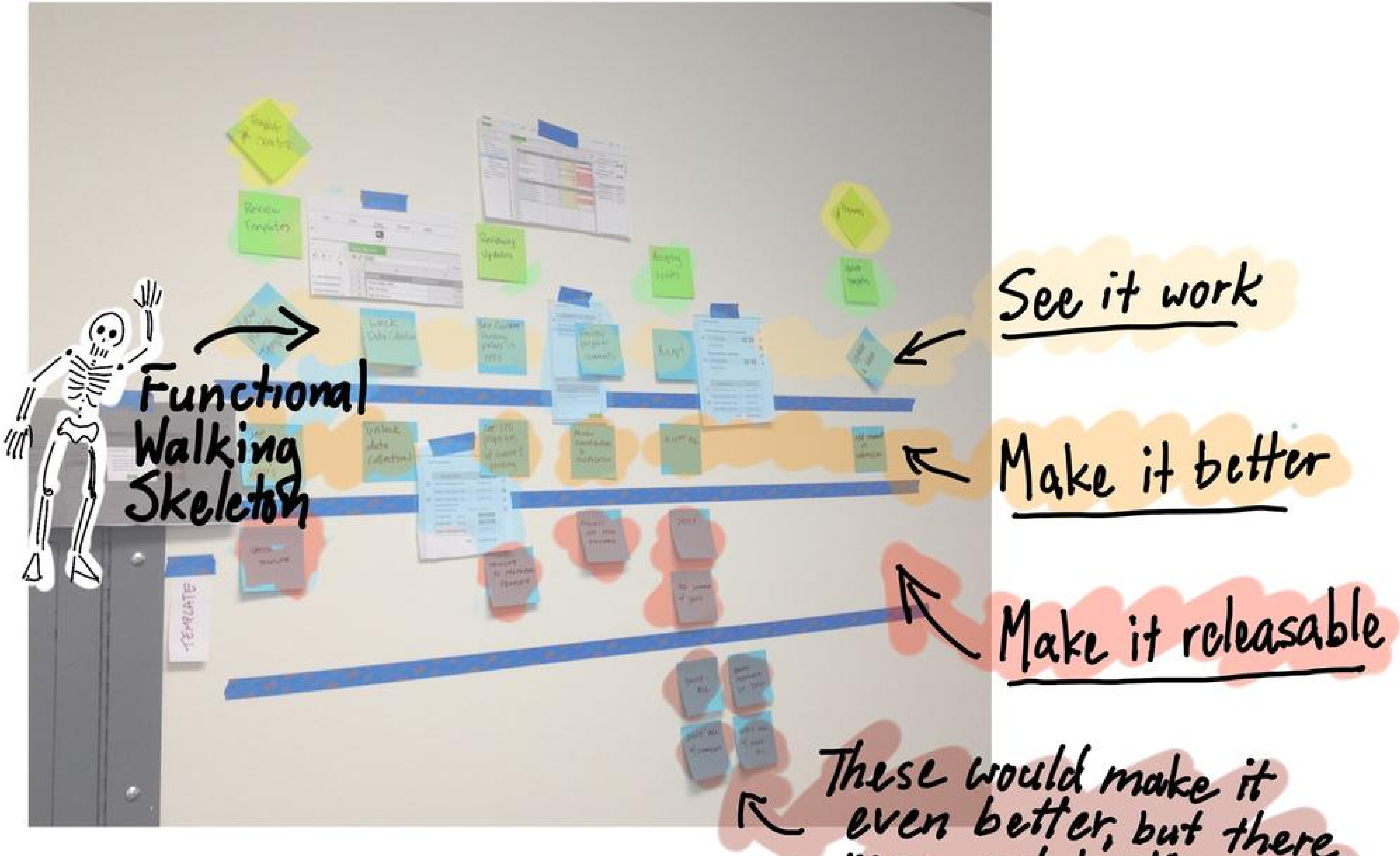
In our project, we have 2 "releases": Assignment 4 and 5. We are artificially separating them into backend (data store) and front-end (interface).

In practice, releases are much more involved.

Concepts (from Patton's book)

- Tasks are short verb phrases that describe what people do.
- Tasks have different goal levels.
- Tasks in a map are arranged in a left-to-right narrative flow.
- The depth of a map contains variations and alternative tasks.
- Tasks are organized by activities across the top of the map.
- Activities form the backbone of the map.
- You can slice the map to identify the tasks you’ll need to reach a specific outcome.

The skeleton idea is we target the first release to have a bare-bones idea of what we need in the product. We then increment from there in subsequent releases.


Functional Walking Skeleton
See it work
Make it better
Make it releasable
These would make it even better, but there
TEMPLATE

"see it work" - Must Have

"make it better" - Should Have

"make it releasable" - Could Have

"not enough time" - Won't Have

MVPs

> The minimum viable solution is the smallest solution release that successfully achieves its desired outcomes. (Jeff Patton)

and

> A minimal viable product is also the smallest thing you could create or do to prove or disprove an assumption. (Eric Ries)

You should be aiming for an MVP by Assignment 5.
The task in Assignment 2 is to figure out what "desired outcomes" are.

Creating USMs

- Install the VS Code extension `TextUSM` or visit https://textusm.com
- Write the activities and tasks quickly - use some stickies or notepad
- THen link the user stories in to your map.
- Export the result to a PNG.

Sample

see here

make sure to use the `.usm` extension

Syntax

see https://app.textusm.com/help

each indent creates a new sub-level

User Activities

User Tasks

User Story Release 1

User Story Release 2...

Add Images

image:https://app.textusm.com/images/logo.svg

Change font size, font color or background color.

test|{"bg":"#CEE5F2","fg":"#EE8A8B","pos":[0,0],"font_size":9}

Other resources

https://junwin.github.io/posts/2017-01-27-storymapping.html