

# WHAT'S YOUR SKATEBOARD?

## EMILY STAMEY

© PHOTO BY KENNY LOUIE





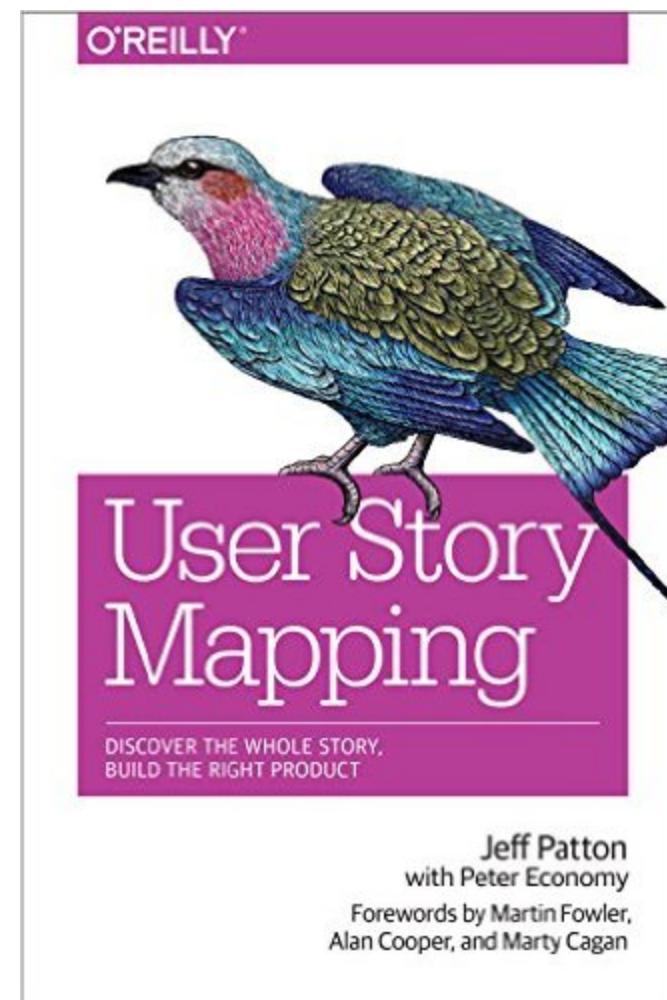
I'm a PHP developer

I've often worked directly with non-technical product owners and users

I workED at a University

**IN**QUEST

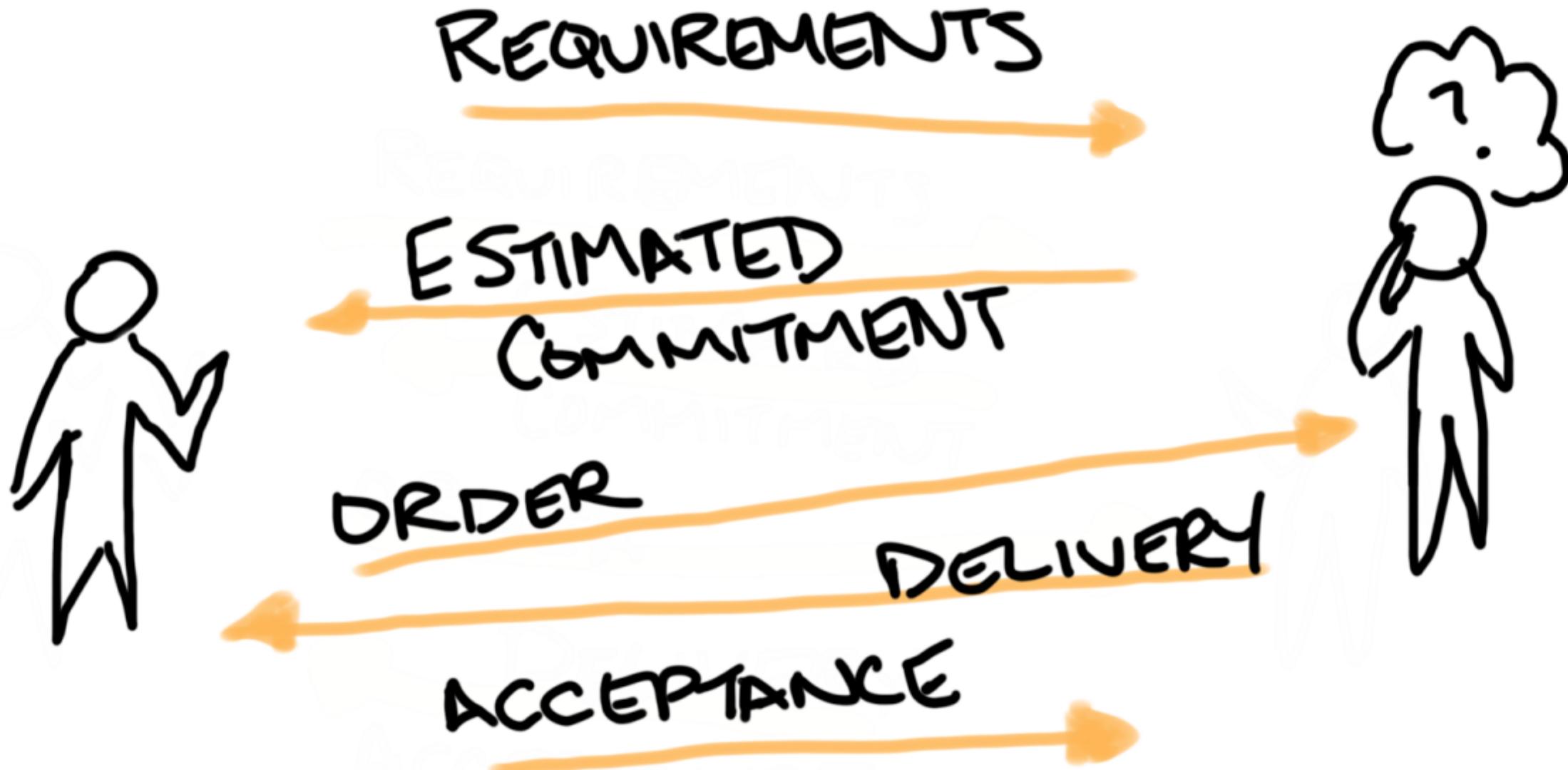
# INSPIRED BY



# A PROBLEMATIC PROJECT WORKFLOW

1. Research
2. Develop over several months
3. Deliver something
4. Learn if the big guess was correct

# THE CLIENT-VENDOR ANTI-PATTERN



# WHAT'S SO GREAT ABOUT THIS ANYWAY?

Communicate with non-technical users of the product

Removes implementation from the discussion

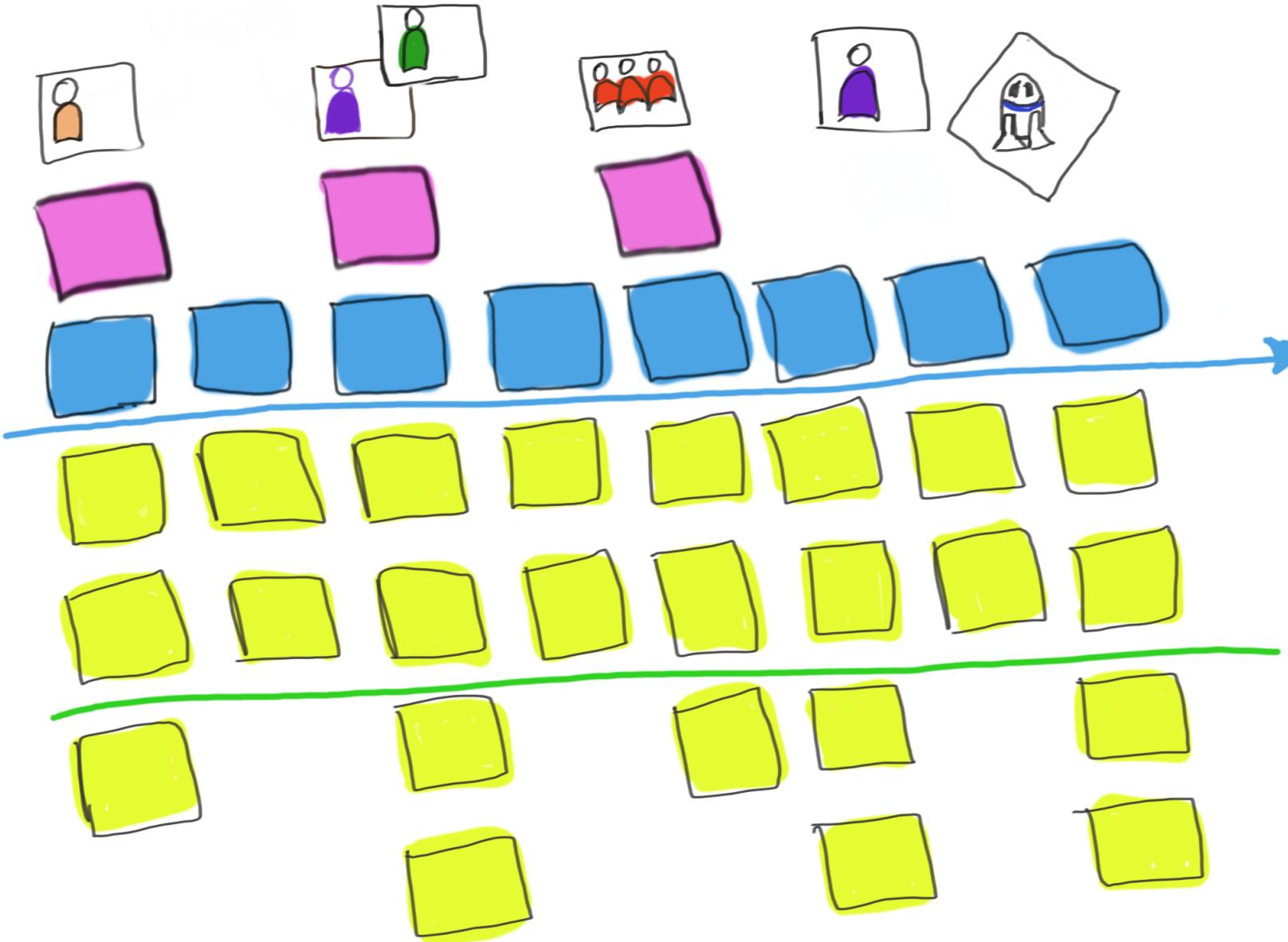
Objective-focused deliverables

# WHAT IS A STORY MAP?

A diagram of a project that tells the story of the people and systems involved in a process.

Detail is added as we learn more about the project

The map can be built for an existing application or a new application.

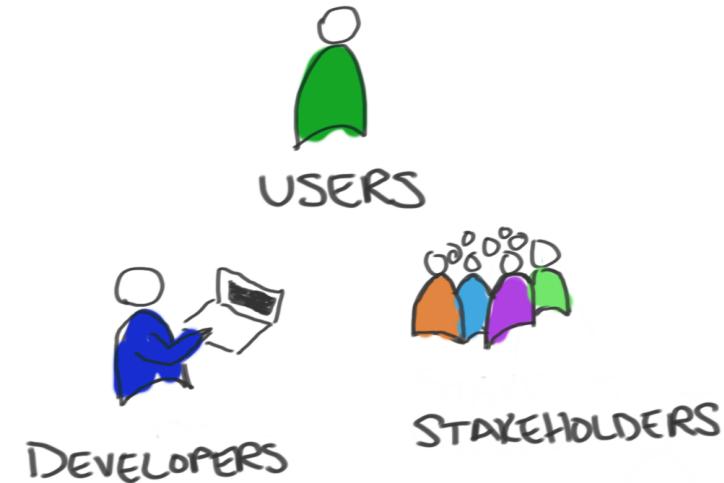


# WHO SHOULD STORY MAP?

Anyone who knows the process ...

Not Just Developers

At least one knowledgeable person  
from each group of stakeholders



# WHEN TO STORY MAP?



- When you have questions
- Before You begin development

# WHY STORY MAP?

- *Build shared understanding*
- Encourage full discovery before prototyping
- Prioritize work to be done as a group

**\*Lowers problems with estimates and feature creep**

# THE BIG PICTURE



© PHOTO BY BARNYZ

# FOCUS DURING DEVELOPMENT



# WHERE?

A large, clear wall or whiteboard.

A place central to the team, at least in the beginning.

# YOU'LL NEED

- painter's tape
- markers
- post-its (many colors & sizes)

# A SIMPLE EXAMPLE

Use a familiar process that is not your project

List five tasks you do in order to get to work. Put each one on a post-it provided.



DRIVE  
TO THE  
OFFICE



POUR  
COFFEE



CHECK  
EMAIL



PLAN WORK  
TO DO

- THING1
- THING2
- THING3

PRACTICE  
FOCUS  
MEDITATION



# FIND VARIATIONS

Pick 3 things that you did today that are different from your normal work routine.



DRIVE  
TO THE  
OFFICE



POUR  
COFFEE



WALK  
TO COFFEE



CHECK  
EMAIL



PLAN WORK  
TO DO

- THING1
- THING2
- THING3

PRACTICE  
FOCUS  
MEDITATION



CHECK  
CALENDAR



ATTEND  
A MEETING



# BUILDING THE NARRATIVE: USERS

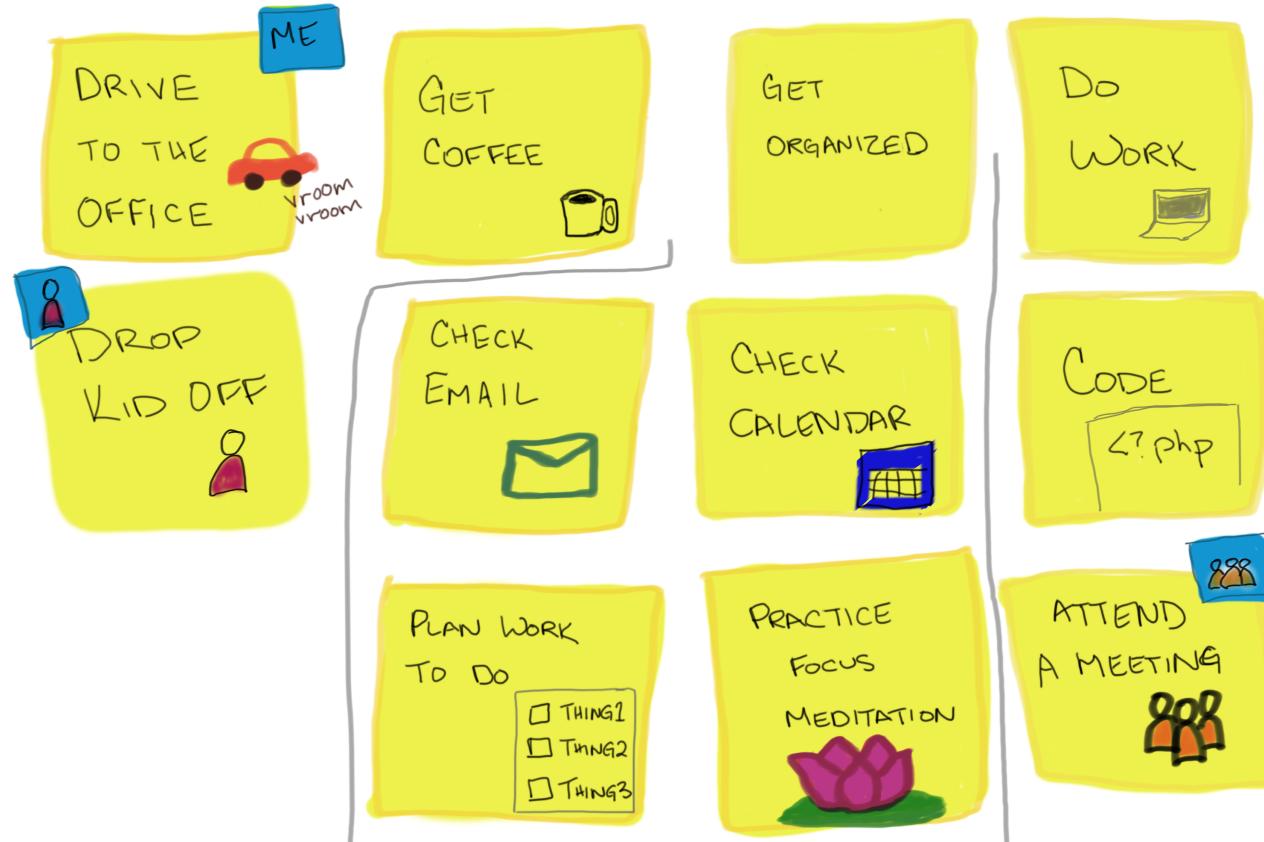


Who are the Actors/Users in your morning routine?

# BUILDING THE NARRATIVE: ACTIVITY GROUPS



# BUILD THE STORY



# LEARN MORE

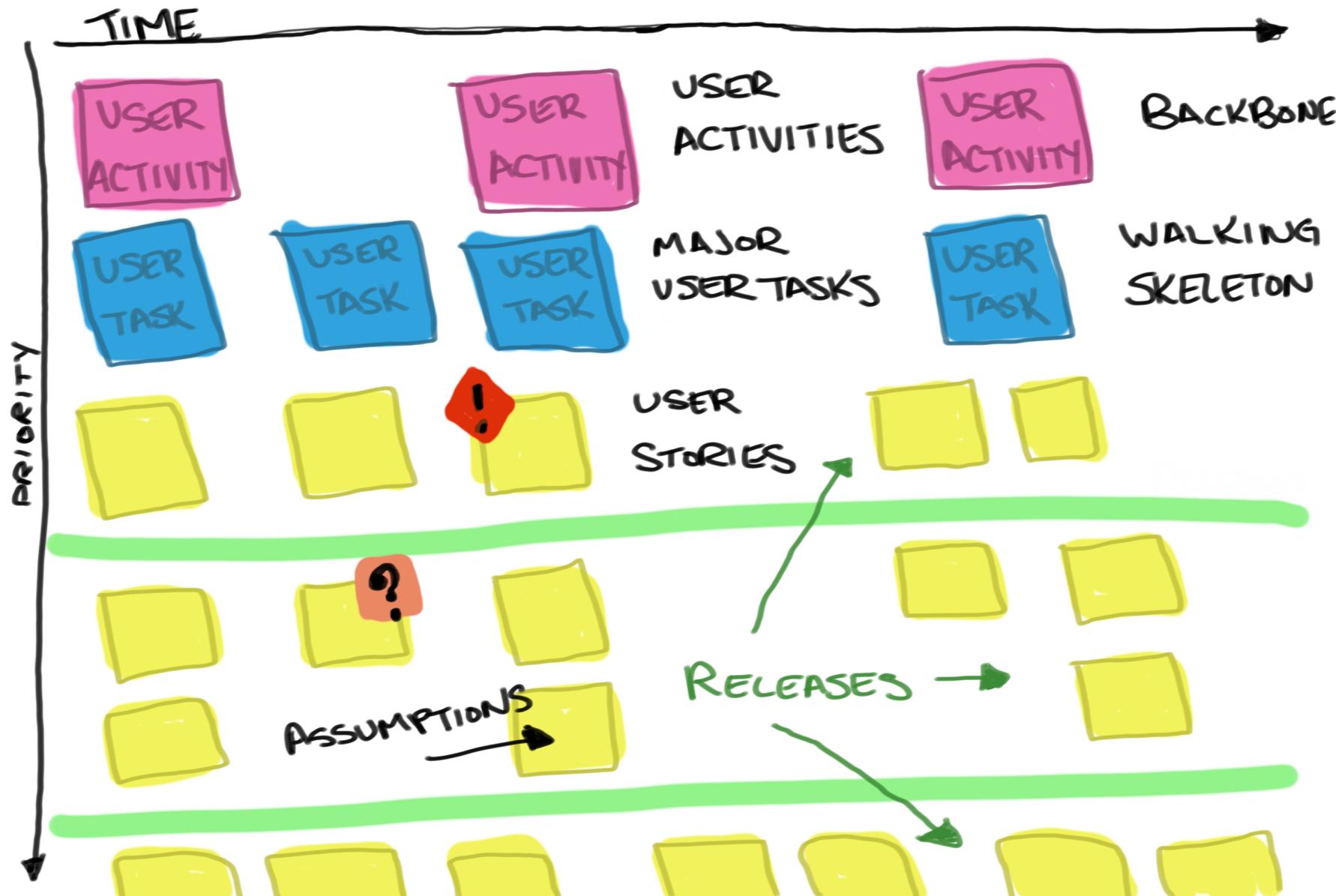
- Risk
- Assumptions
- Uncertainty

If you cannot elaborate, mark it and revisit

# CLARIFYING QUESTIONS

- What could we learn to replace risk with **REAL** information?
- Do we really know what has been mapped, or did we fill in **assumptions**?
- Are you sure about the story you're telling?

# LABEL THESE



# THOROUGH DISCOVERY

- Understand the full process
  - Understand "why" steps are needed in the process
  - Talk about things inside and outside of the app

# THOROUGH DISCOVERY

- Simplify and lower risk at implementation
  - Lowers the questions at the phase of implementation
  - Limits Feature Creep (beginning implementation w/o understanding, new features come in)
  - Better estimates

# MORE THAN JUST ADDING STORIES

“ If the only thing you create while making sense of a big opportunity is more, small stories, **you're doing it wrong.**

- Jeff Patton

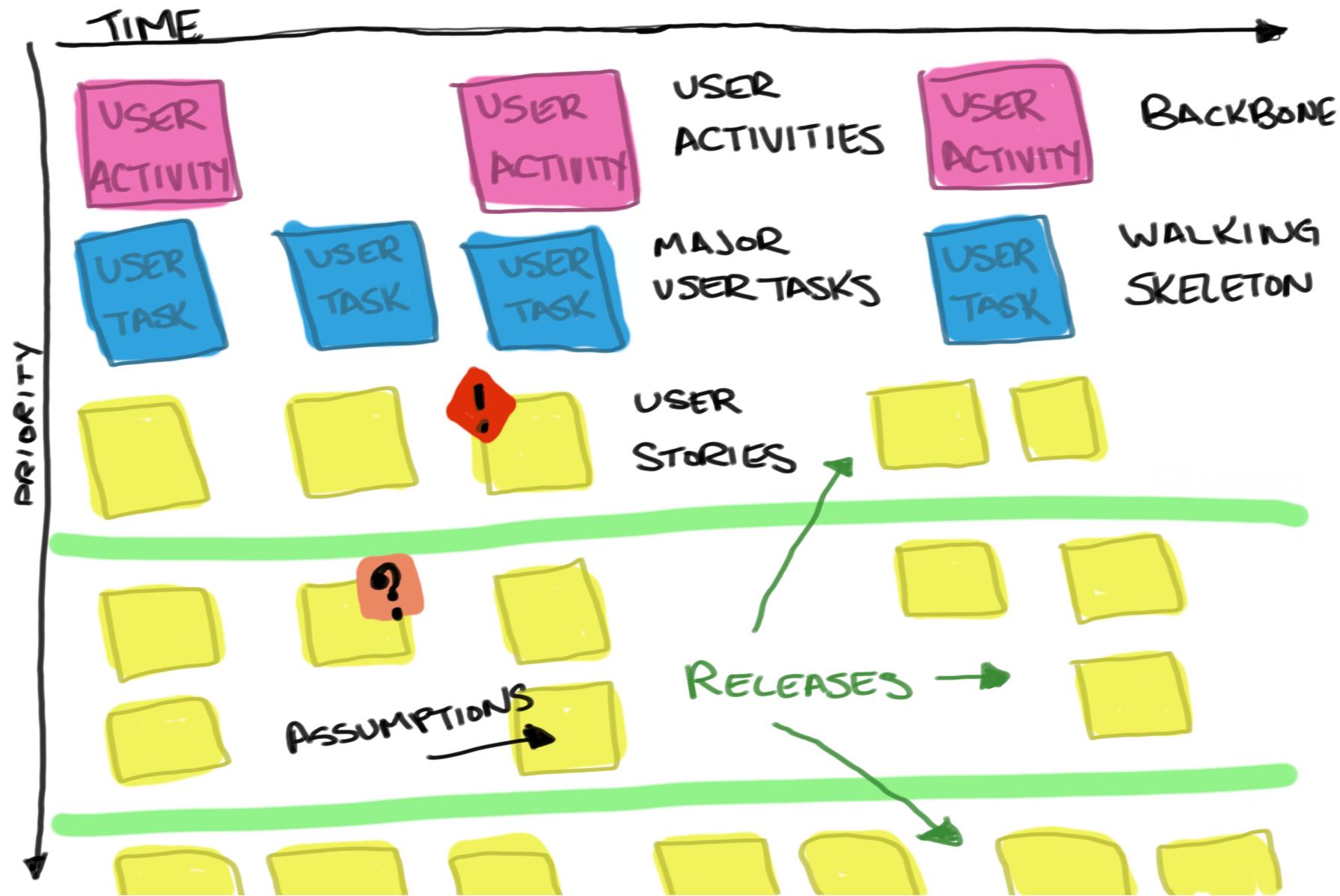
# FOUR STEPS TO DISCOVERY

1. Frame the idea
2. Understand Customers and Users

# **SIMPLE PERSONA**

# FOUR STEPS TO DISCOVERY

1. Frame the idea
2. Understand Customers and Users
3. Envision your solution
4. Minimize the plan



# PRIORITIES

# PRIORITIZING THE PROJECT

- Who will use this product?
- What steps must they accomplish for success?
- Remove/postpone the rest

# PRIORITIZING FEATURES

- **Differentiator** - feature sets you apart from competition
- **Spoiler** - moves in on someone else's differentiator
- **Cost reducer** - reduces organizational costs
- **Table stakes** - feature necessary to compete

# FOCUS ON OUTCOMES

- What are you hoping to do with your application?
- Prioritize features based on the problem they solve
- Implement only what solves the problem or meets the objective

# PROTOTYPING

# PROTOTYPING

- What is the smallest thing you could build to prove/disprove an assumption?
- Sketch & prototype to test viability of the solution
- Aim for less than minimum, get feedback, and iterate often
- When you give prototype to **development partners** you can include metrics to see what they actually do

# MVP VS. MOST VALUABLE FEATURES

Focus on releasing valuable features every time.

Sometimes we plan features in a chronological order

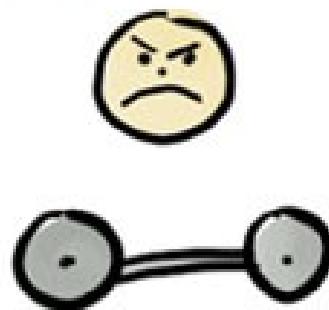
Or we divide the project into components

# WHEN WE BUILD IN PIECES

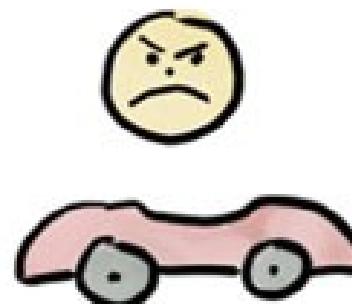
Not like this....



1



2



3



4

# WHEN WE BUILD IN PARTS



**INSTEAD WE WANT TO ITERATE!**



1



2



3



4



5



# STRATEGIES



# OPENING GAME STRATEGY

- When the number of features is too large, you can cut a slice across that gives you the minimum end-to-end functionality;
- It doesn't solve all user's stories, but it affects the largest number of stories.
- With this product, they can have test data, begin testing it for load, and see how it will work

# MIDGAME STRATEGY

- fill in and round out features
- support optional steps
- implement tough business rules
- continue testing the product usability

# ENDGAME STRATEGY

- refine: make it look more polished and efficient
- it's here that you'll have feedback from users that can be applied.



# SCHOLARSHIP APPLICATION

Applicant's Name:

Present Address:

Phone:

Home

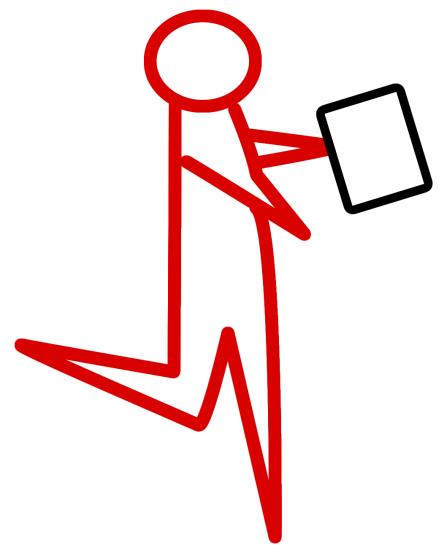
Mobile

Fax Number:

E-mail:

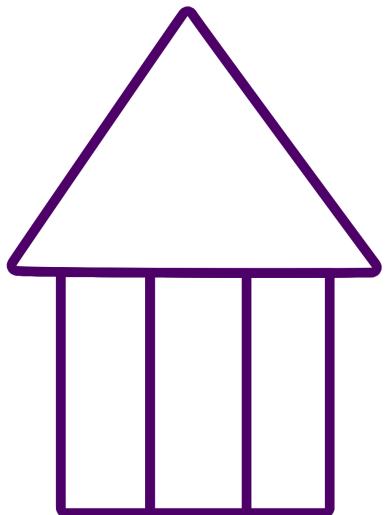
Place of Birth:

**FALL**



**STUDENT APPLIES**

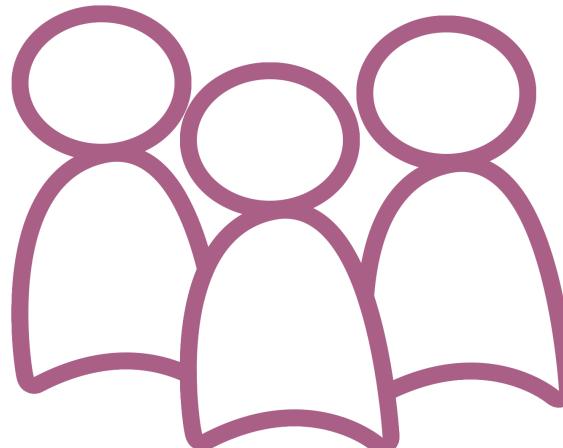
# JANUARY



## ENGINEERING FOUNDATION

**ORGANIZE SCHOLARSHIP CRITERIA  
ORGANIZE BUDGETS**

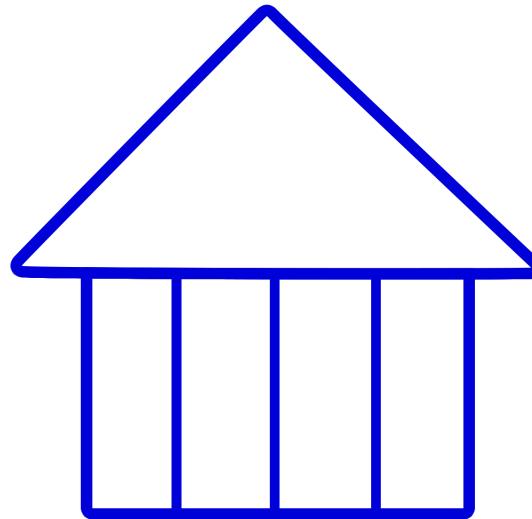
**MAR - APR**



**SELECTION  
COMMITTEE**

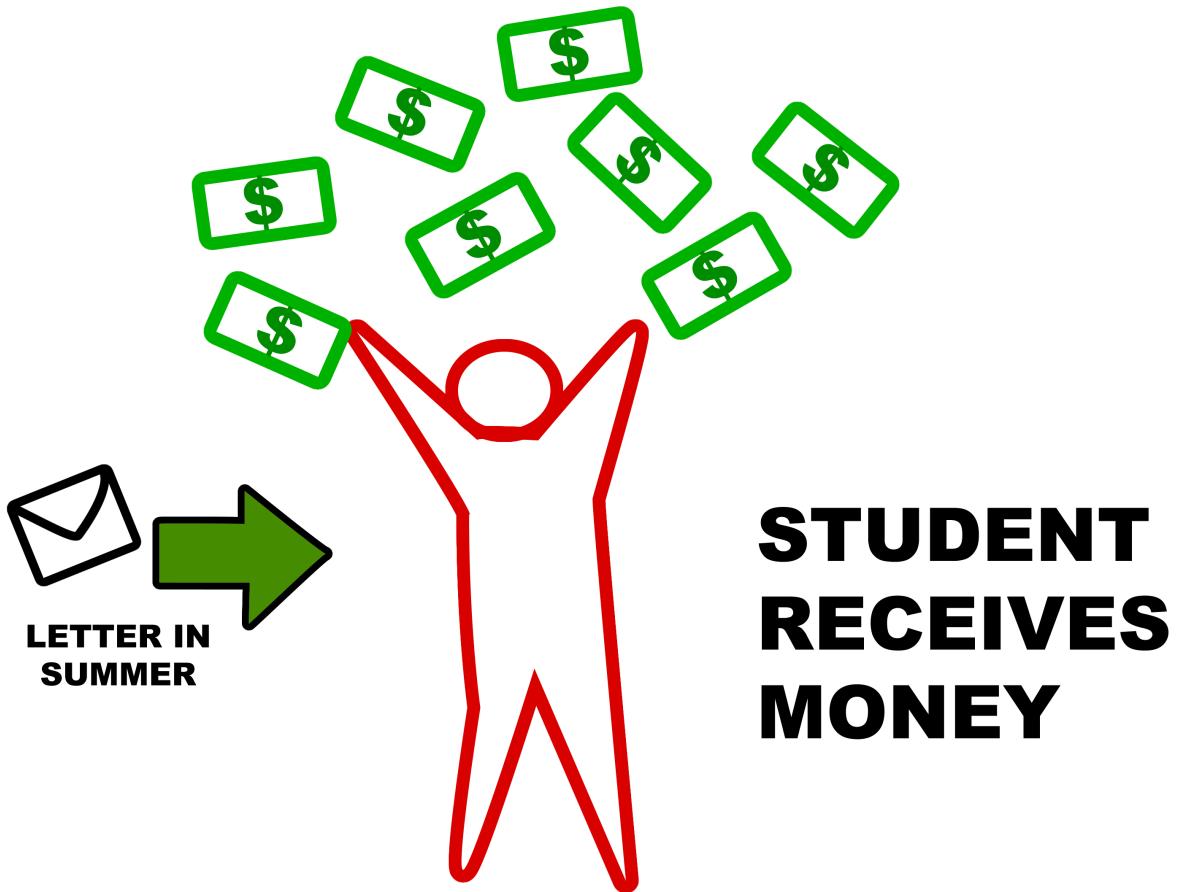
**REVIEW CANDIDATES  
SELECT RECIPIENTS**

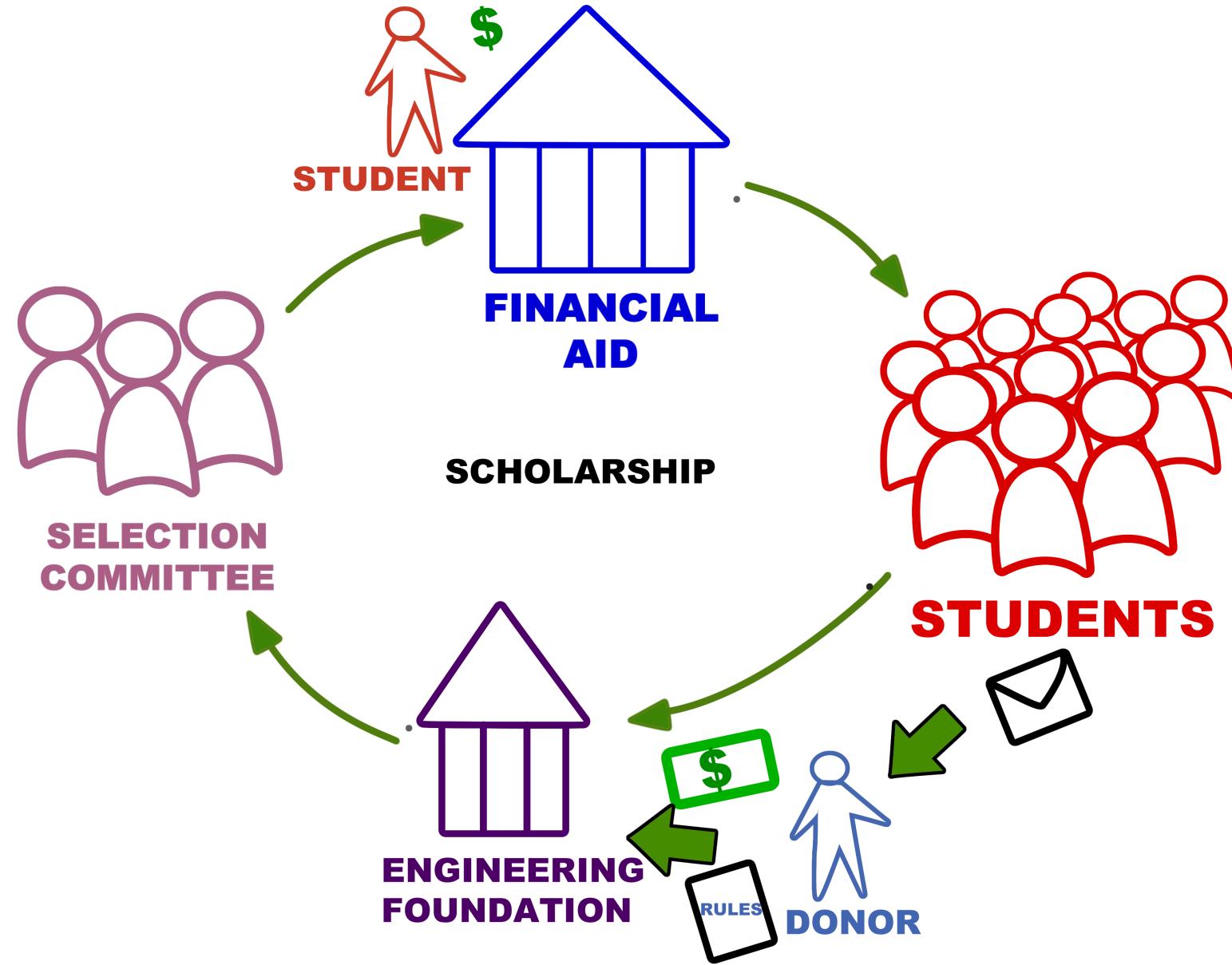
**AUG & JAN**



**FINANCIAL  
AID**

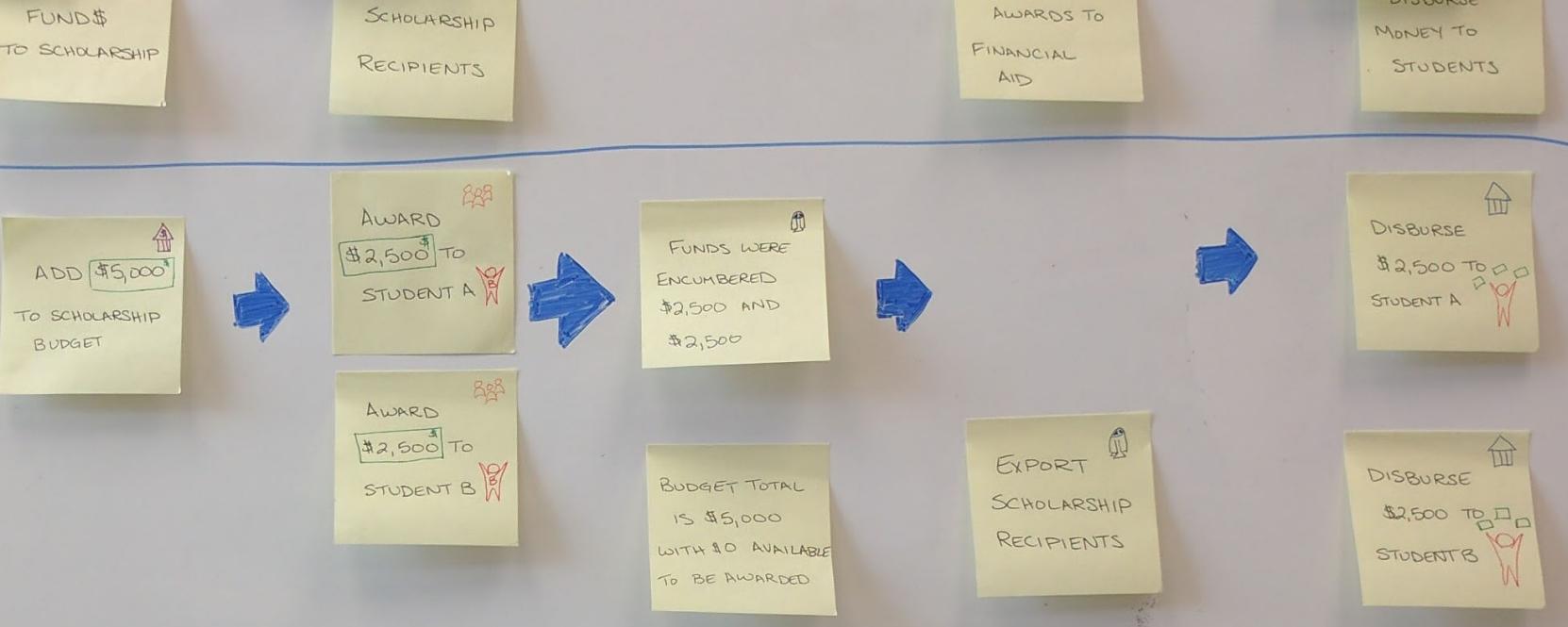
**RECEIVES AWARD DETAILS  
DISBURSES MONEY**





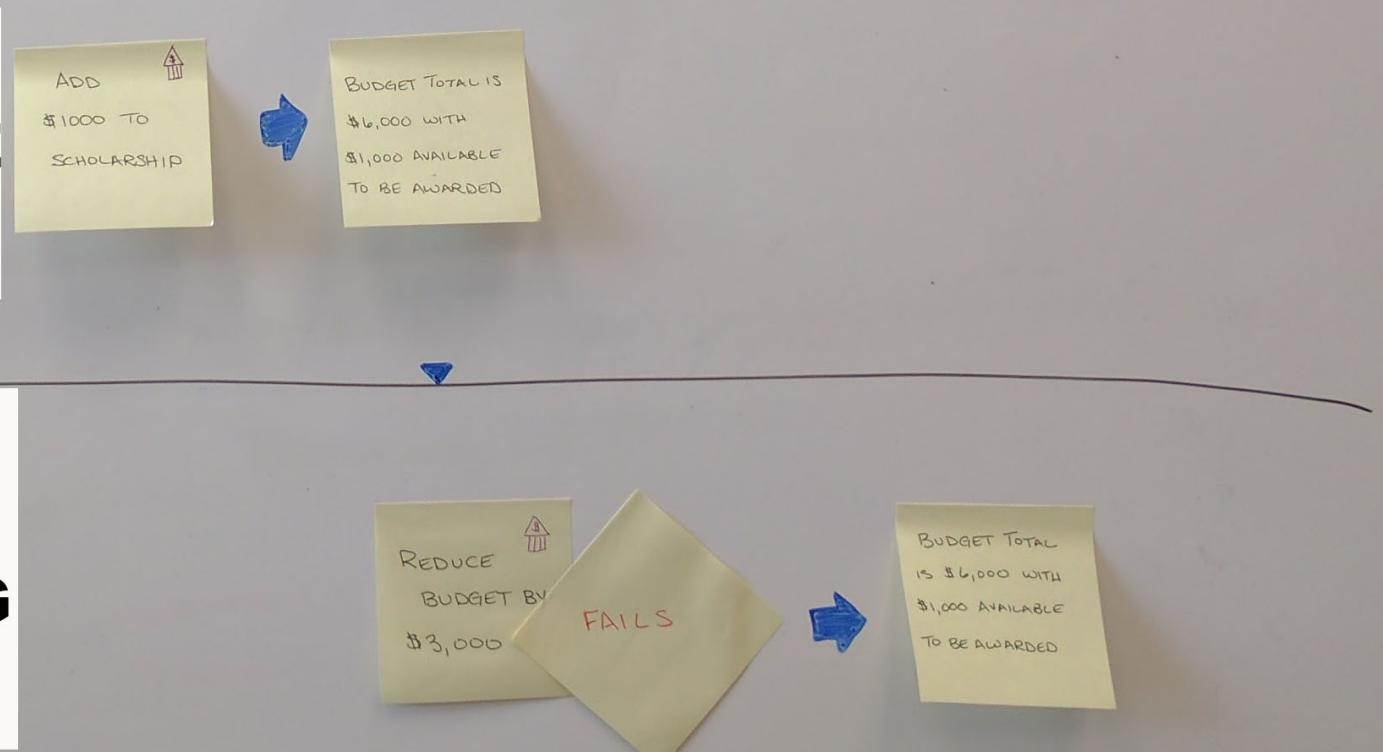
# OBJECTIVES

**SELECTION COMMITTEE  
SHOULD BE ABLE TO  
AWARD SCHOLARSHIPS**



**ENGINEERING FOUNDATION  
SHOULD BE ABLE TO INCREASE  
SCHOLARSHIP BUDGETS**

**ENGINEERING FOUNDATION  
SHOULD BE ABLE TO REDUCE  
BUDGETS WITHOUT AFFECTING  
SCHOLARSHIP RECIPIENTS**



SELECTION COMMITTEE  
SHOULD BE ABLE TO  
AWARD SCHOLARSHIPS

ADD  $\$5,000$   
TO SCHOLARSHIP  
BUDGET



AWARD  
 $\$2,500$  TO  
STUDENT A



AWARD  
 $\$2,500$  TO  
STUDENT B

FUNDS WERE  
ENCUMBERED  
 $\$2,500$  AND  
 $\$2,500$

BUDGET TOTAL  
IS  $\$5,000$   
WITH  $\$0$  AVAILABLE  
TO BE AWARDED

ENGINEERING FOUNDATION  
SHOULD BE ABLE TO INCREASE  
SCHOLARSHIP BUDGETS

ADD  
 $\$1000$  TO  
SCHOLARSHIP



BUDGET TOTAL IS  
 $\$6,000$  WITH  
 $\$1,000$  AVAILABLE  
TO BE AWARDED

ENGINEERING FOUNDATION  
SHOULD BE ABLE TO INCREASE  
SCHOLARSHIP BUDGETS

ADD  
\$1000 TO  
SCHOLARSHIP

BUDGET TOTAL IS  
\$6,000 WITH  
\$1,000 AVAILABLE  
TO BE AWARDED

ENGINEERING FOUNDATION  
SHOULD BE ABLE TO REDUCE  
SCHOLARSHIP BUDGETS WITHOUT  
AFFECTING SELECTED RECIPIENTS

REDUCE  
BUDGET BY  
\$3,000

FAILS

BUDGET TOTAL  
IS \$6,000 WITH  
\$1,000 AVAILABLE  
TO BE AWARDED

REDUCE  
BUDGET BY  
\$1,000

BUDGET TOTAL  
IS \$5,000  
WITH \$0  
AVAILABLE TO BE  
AWARDED

# SUMMARY

- A tool you can use with non-technical subject matter experts, customers, etc
- A visual guide for managing your workload
- Focus on objectives when you prioritize
- Plan to deliver a usable product at each deliverable

**SOOO...**

# WHAT'S YOUR SKATEBOARD?



# THANK YOU!



# EMILY STAMEY

@ELSTAMEY

[HTTP://ELSTAMEY.COM](http://elstamey.com)

[HTTPS://ELSTAMEY.GITHUB.IO/SKATEBOARD.HTML/#/](https://elstamey.github.io/skateboard.html/#/)