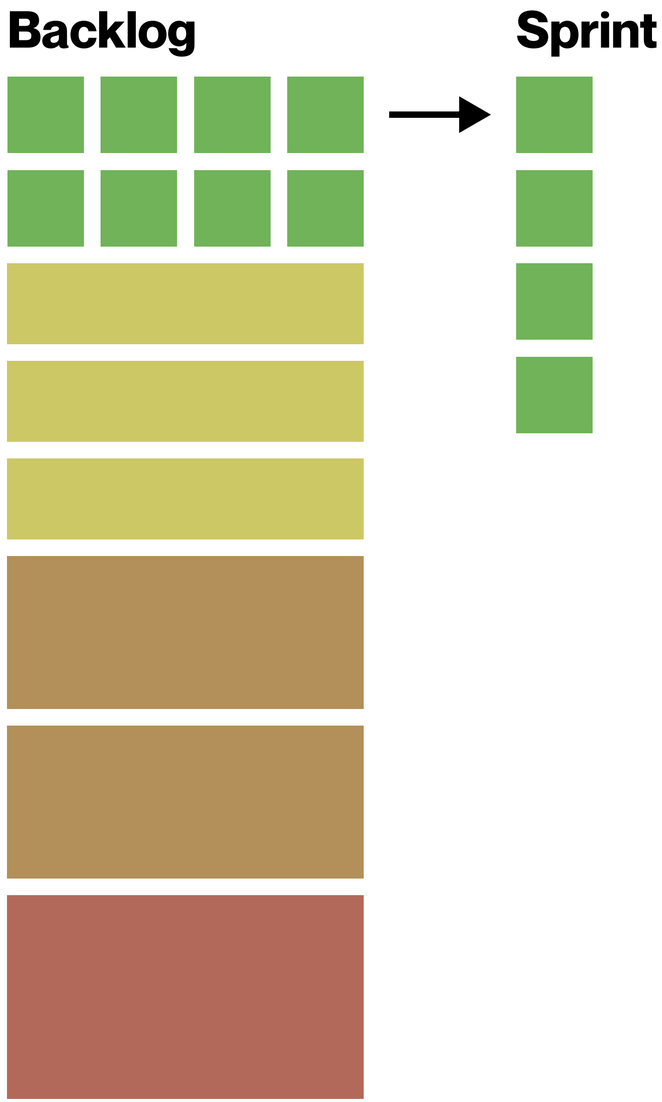
Breaking down User Stories

Sam Feller
awkwardengineer.com

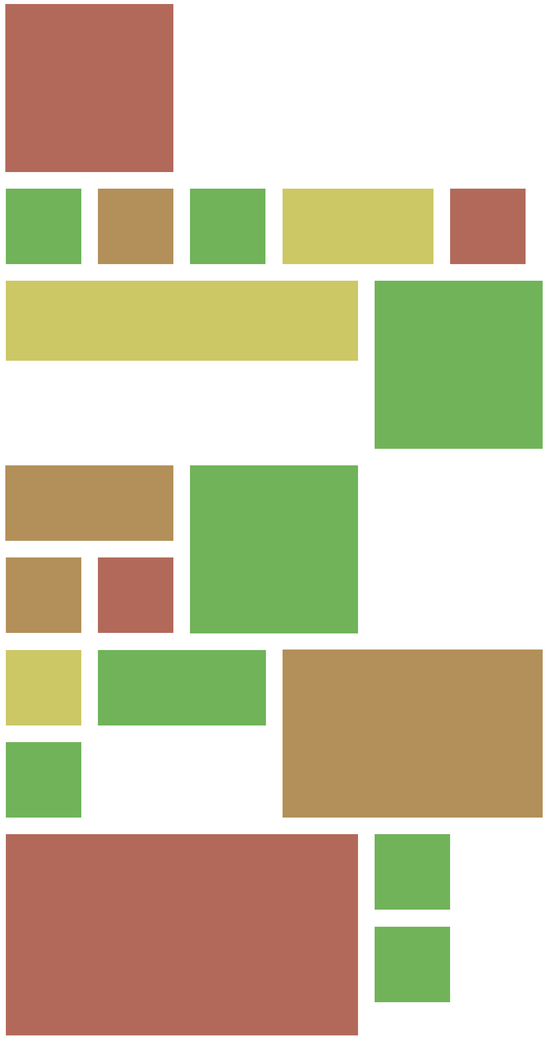


Notes

Every book on scrum I’ve ever seen has a backlog that looks something like this… big stories that get neatly broken down as the approach the top.

Then they get picked off and put into the sprint.

Real Life Backlog



Notes

But in real life, most teams I've worked with had a back log that looked like this.

Stories are wildly different sizes, and then what happens is there's no prioritization. New stuff gets added to the front, the backlog becomes stale, and lots of old stories linger.

Why does this happen?

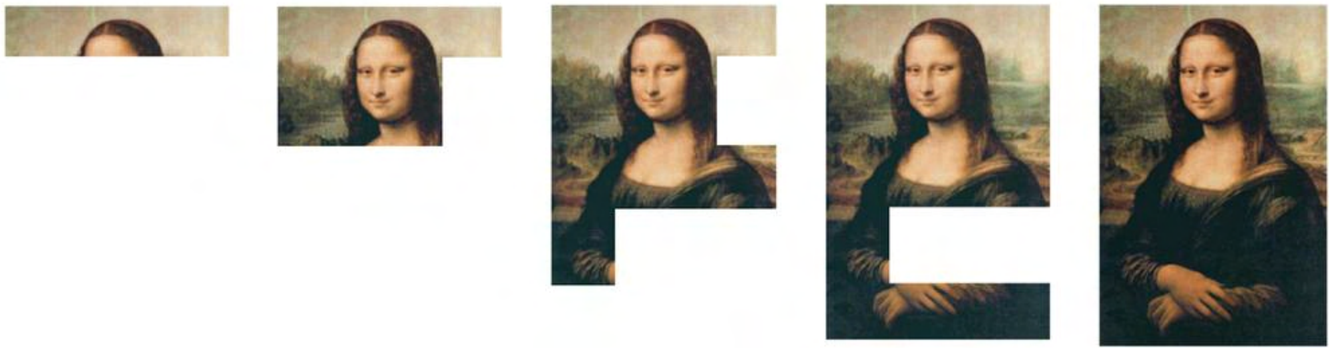
Writing a ticket in a tool freezes it.

Forming stories takes work

This is just one way!

Background / Agile Mindset

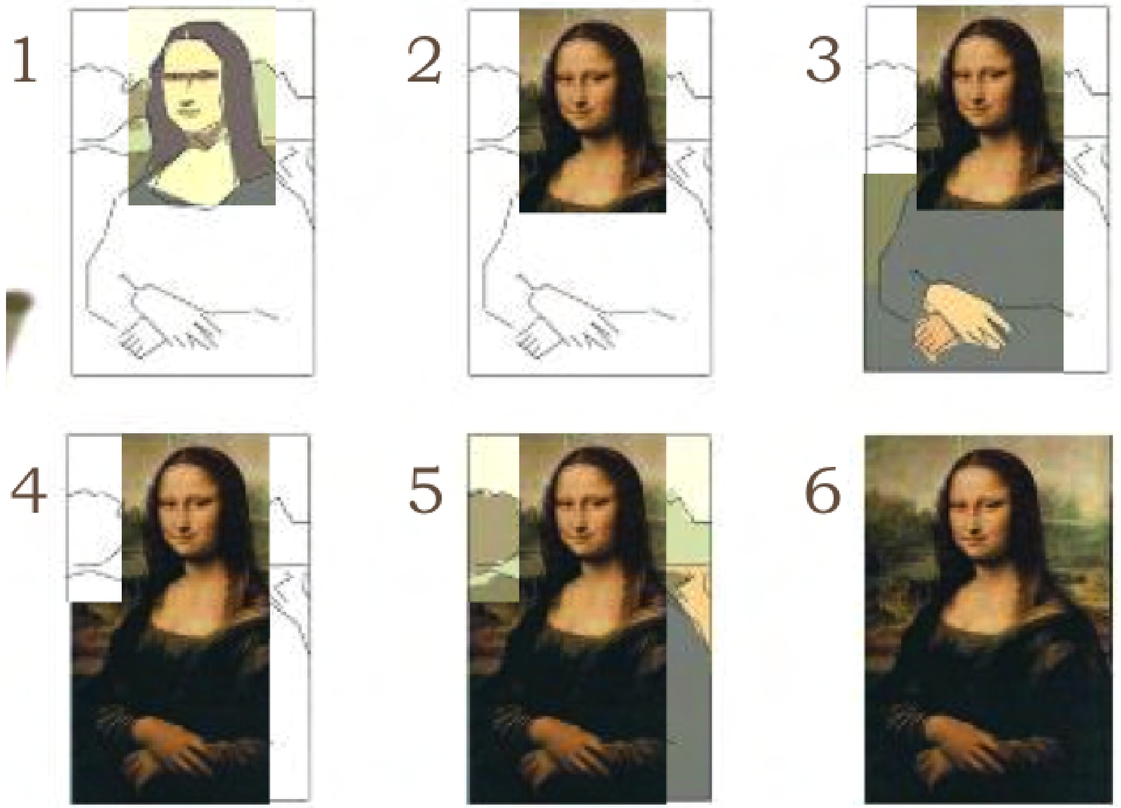
“Working software is the primary measure of progress”



Notes

There are different ways to draw the mona lisa.

You can always walk away from the bottom and say “there’s a working painting”. it may not everything you want, but it’s still the mona lisa.



Notes

Real life tends to be a hybrid of the two, you need the whole thing in place, but you do more work on the more important bits first.

Customers want cake, not flour or eggs.

Notes

This is the other analogy I use.

Cupcake

Cupcake + Frosting

Cupcake + Frosting + Sprinkles

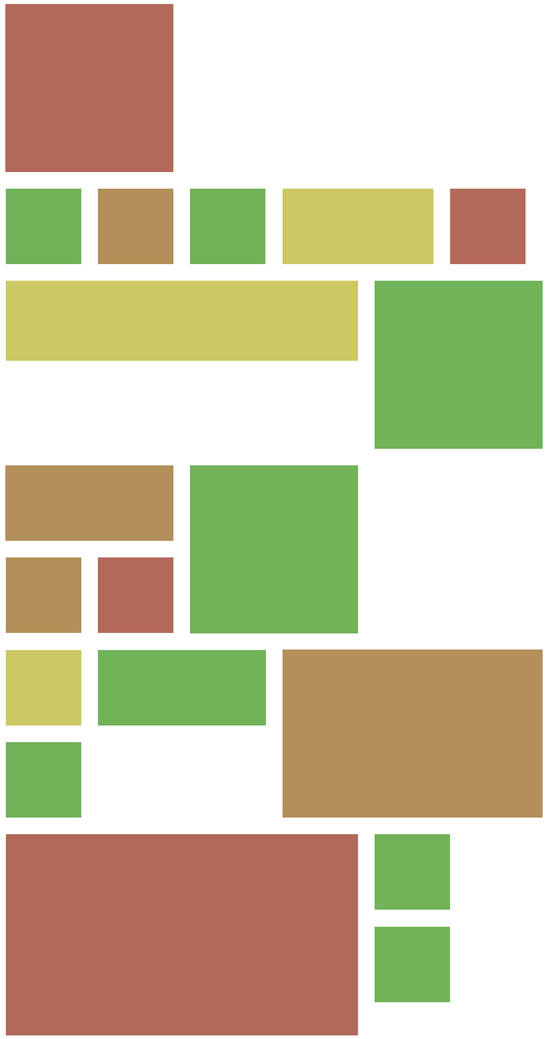
Notes

Customers can eat cake. Cupcakes, frosted cupcakes, and then the ones with the little sprinkles.

Get to the point where you can ship cake, then go back and add more.

That’s great, now what?

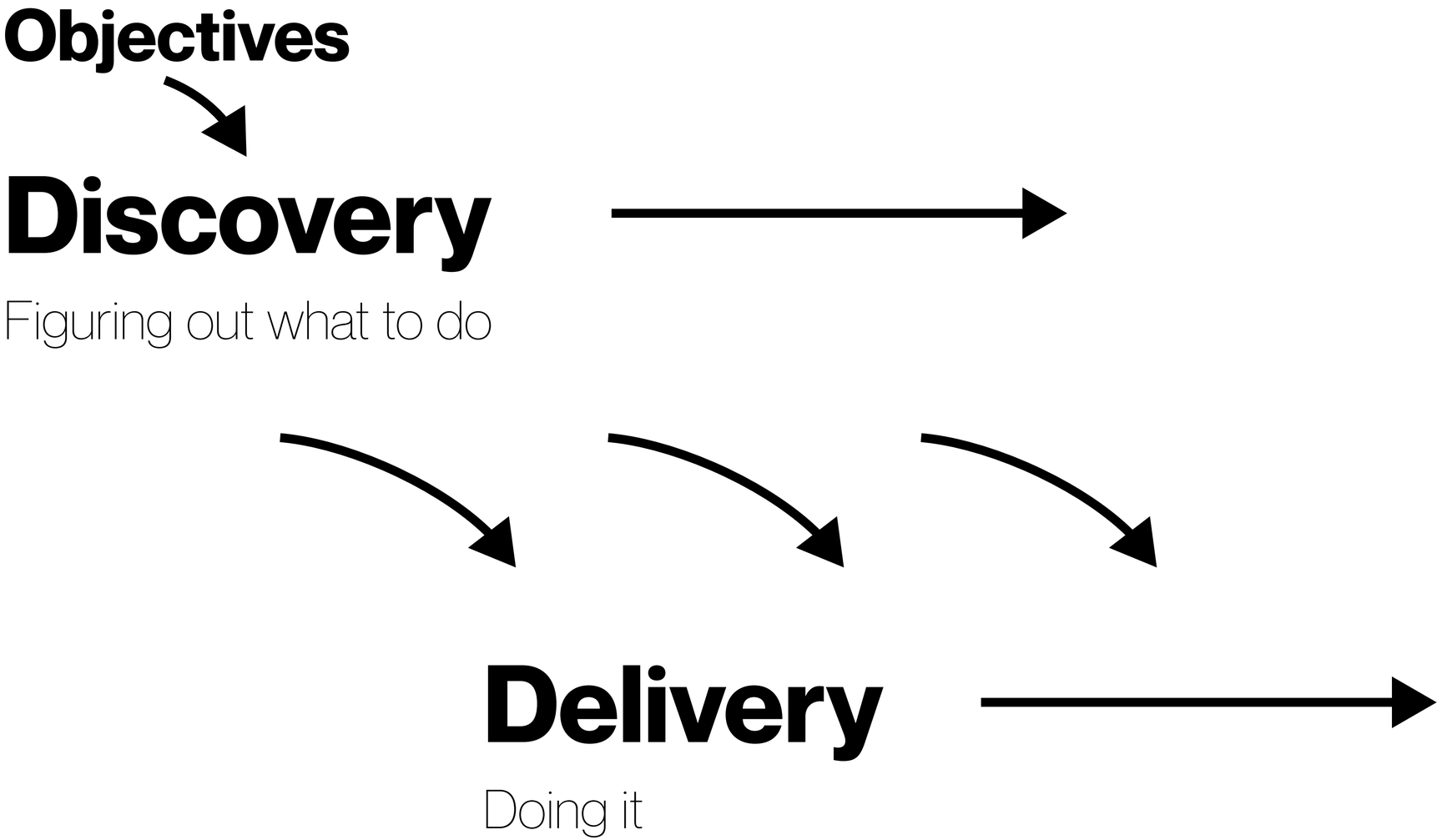
Real Life Backlog



Notes

Ok, so we get the cake analogy, but our backlog still looks like this. How we put this into practice?

Discovery / Delivery

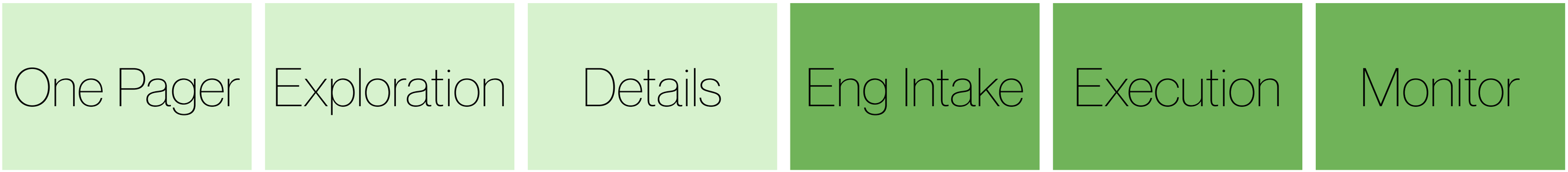

Objectives
Discovery
Figuring out what to do
Delivery
Doing it

Create a Pipeline of Work

(That Solves Problems for Customers)

Agility is the speed through the line.

→ Discovery Delivery →



Another, similar model.

One Pager

One Pager:

Plain english, **marketing consumable** description of what you're trying to do and why.

Plenty of online templates, (search for PRD) typically describe the problem, current state, proposed solution, success metrics etc.

Personally, I prefer

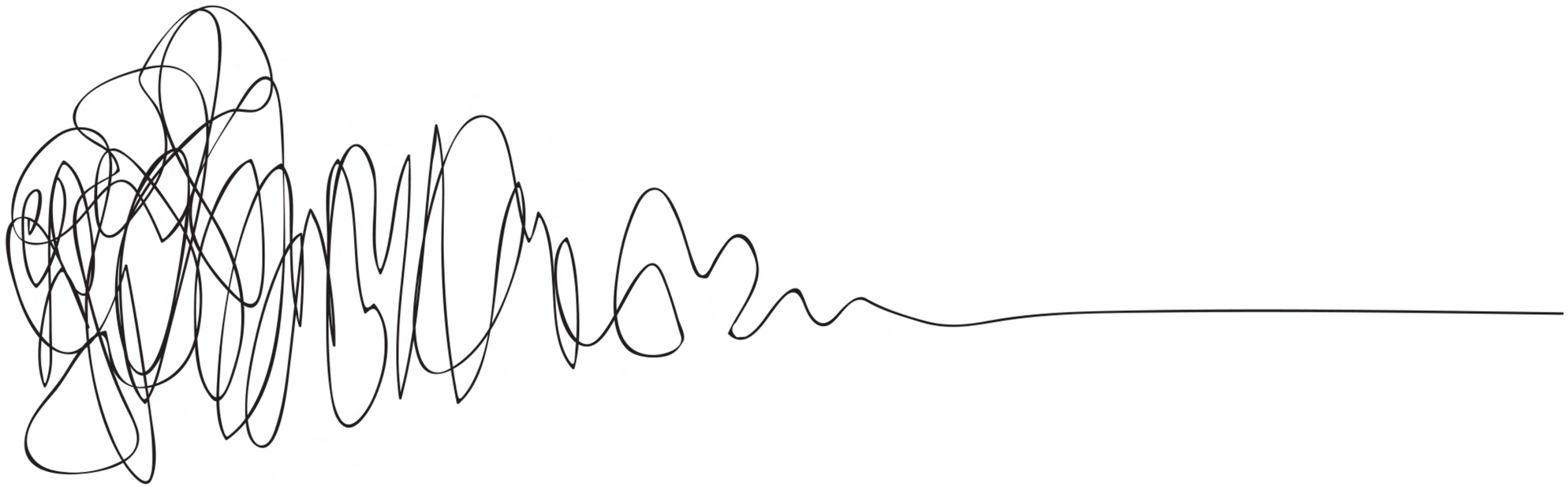
1. A single, plain english, marketing consumable paragraph.
2. Mock press release (search for Amazon PR/FAQ)

Notes

When I’m working really closely with Design and Engineering, I honestly find any more than a paragraph of plain english text (that marketing/ sales can understand) to be too much detail and too constraining on the solution space.

The rest shakes out through discussion and process. We’ll see what I mean soon.

Exploration



Notes

This is called the “design squiggle”. It’s a good analogy for the messy back and forth at the beginning and then clarity emerges.

Worked example:

The average home improvement project is 3 Home Depot trip. Let’s make an app to plan you shopping trip.

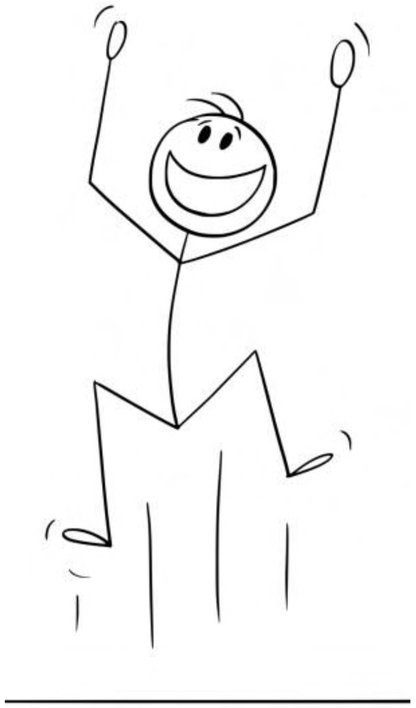
Notes

This is the example, because this is about story writing. This isn’t about problem discovery or user research to know that this is a problem worth solving. That would be a topic for another lecture.

Some tools:

1. High level journey / story map
2. Sharpie Level Sketching
3. Design mocks

HMW Pick a project	HMW Know it's time to shop?	HMW Figure out what we need?	HMW we track what we've bought?
Browsable project library	GPS near store	Inventory	Checklist
Project suggestions	Request from partner	Project "recipes"	Aisle by Aisle
Skills progressions			



Notes

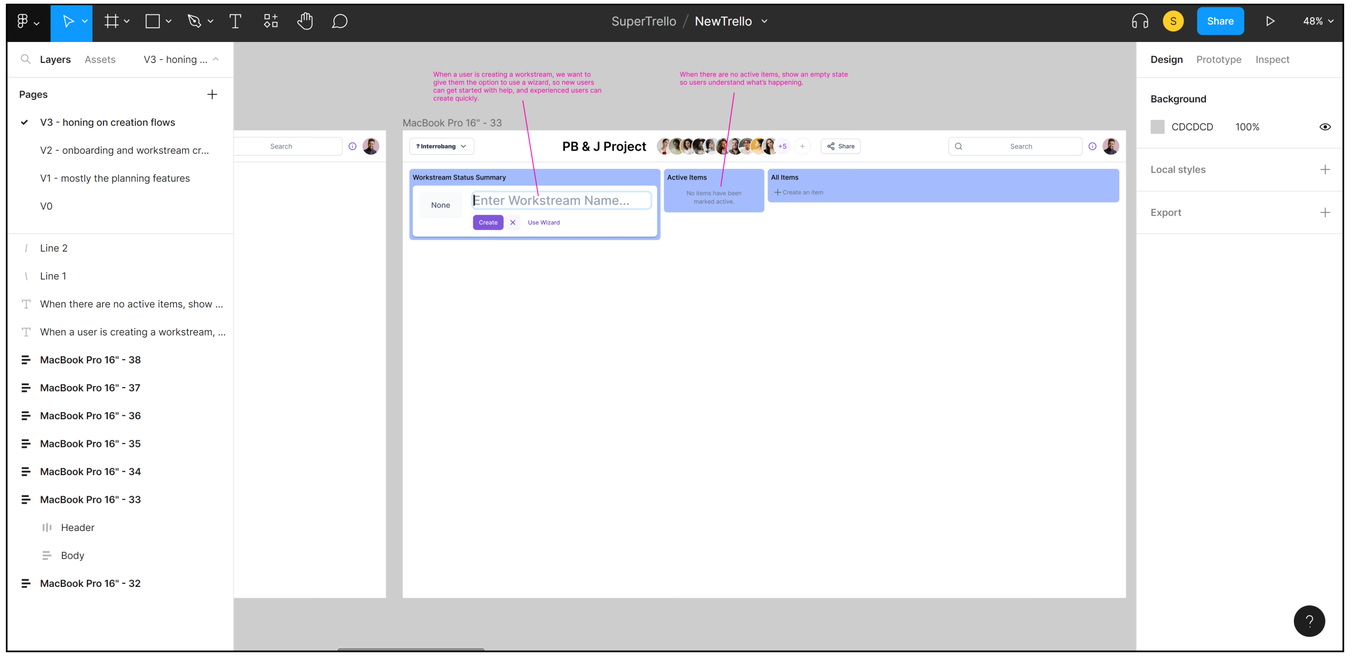
All software is about helping customers get from point A to point B, and solving something for them.

The journey map is a tool to model that. The top is the "spine", asking "how might we..." solve major steps in the sequence of solving a problem. The "ribs" that hang below are features to answer the "HMW?" question.

Notes

If you can't get all the way across the spine, you can't complete the journey and get the customer from A to B. You don't have the Mona Lisa and you don't have cake.

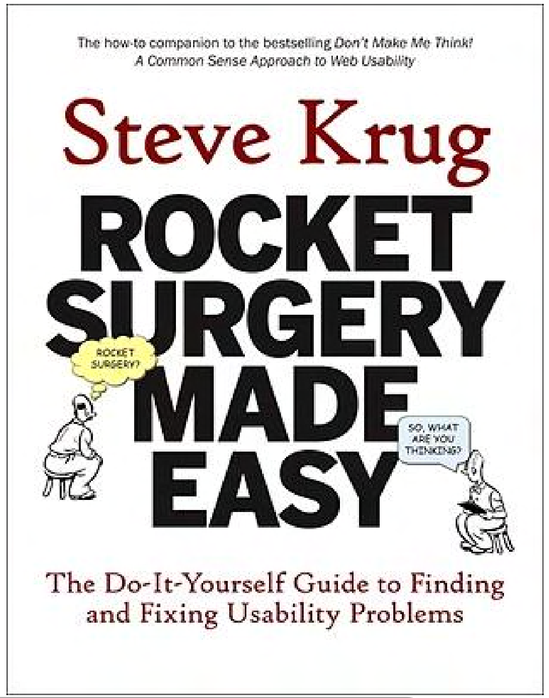
Create Designs, Test, and Iterate



Notes

I'll write/review story annotations with design and engineering together in the room.

I write stories as annotations in design mocks



Run hallway usability tests first
Bring the engineers
Test with customers

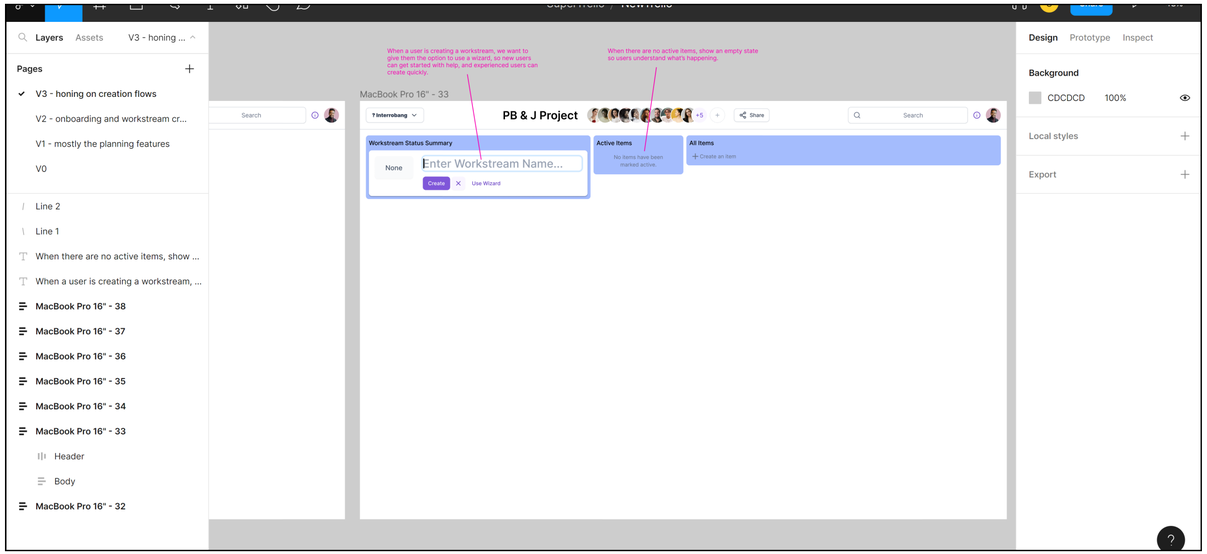
Notes

Great, easy to read book about user testing.
There's real evidence in live reactions.

Details

Now it’s story writing time

But it’s actually more like feature writing time.


Layers
Assets
V3 - honing ...
Pages
V3 - honing on creation flows
V2 - onboarding and workstream cr...
V1 - mostly the planning features
V0
Line 2
Line 1
When there are no active items, show ...
When a user is creating a workstream, ...
MacBook Pro 16" - 38
MacBook Pro 16" - 37
MacBook Pro 16" - 36
MacBook Pro 16" - 35
MacBook Pro 16" - 34
MacBook Pro 16" - 33
Header
Body
MacBook Pro 16" - 32
When a user is creating a workstream, we want to give them the option to use a wizard, so new users can get started with help, and experienced users can create quickly.
When there are no active items, show an empty state so users understand what's happening.
MacBook Pro 16" - 33
PB & J Project
Workstream Status Summary
None
Enter Workstream Name...
Create
Use Wizard
Active Items
No items have been marked active.
All Items
Create an item
Design
Prototype
Inspect
Background
CDCDCD
100%
Local styles
Export

When [situation], [Feature] so [Outcome]

Works for back end systems, too!
I prefer this to As an [X] I want [Y] so [Z]

Working with Eng

Engineering has been in the room from user story mapping, initial mocks, and usability testing.

For each story, talk about the technical approach, especially **good/better/best**

Notes

Asking specifically for good/better/best approaches invites creativity, and more importantly, helps make it clear what the various OPTIONS are, which is what makes negotiation, tradeoffs, and good decisions possible.

Now you can make tickets

Stories can now be estimated and loaded into sprints.

Home improvement project list example:

How might we...

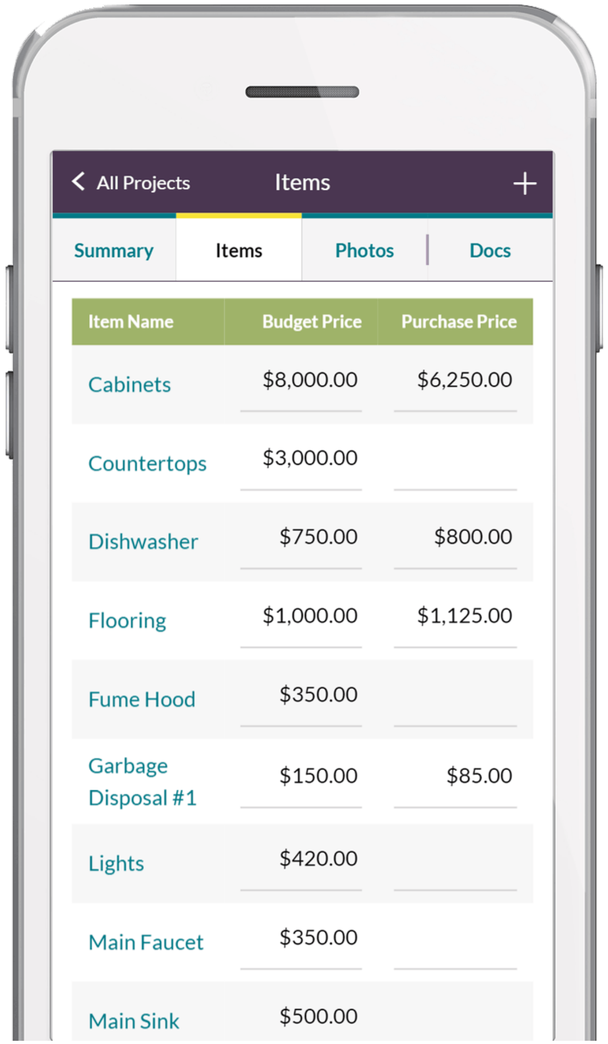
Help the customer get started with a home improvement project?

User Story:

When a user goes shopping for a project, we want to create list of tools and materials, so we can be confident when going to the store.

Notes

This is a simple, one sentence description of what we're trying to do. But as you'll see on the next slide, there is a LOT of detail and room for negotiation in this.



Items

- are they alphabetical order? resortable? editable? deletable?
- what about pictures or brand names?

Search?
Store Sections?
Barcodes?
Ads?

Notes

Creating the mock makes all sorts tangible for discussion with engineering.

Some features may be really cheap to implement, some may require all sorts of development to make happen. You don't know what you don't know, but discussing the mocks makes it tangible.

Notes

Remember, my goal is to take the high level user story, and then from that, extract FEATURE TICKETS. Those are the pieces of "cake" that I'm going to be asking engineering to build.

Things to Keep in Mind

Epics - A collection of features that together creates a unit of business value.

Feature - A working unit of software.

Task/Subtask - Sweeping the floor, mixing flour and eggs

Notes

Again, there are no “user stories” here. The user stories were for working with design and giving them space to solve a problem, but it’s the features that we’re actually going to build.

It’s ok to ship fewer cupcakes in the epic. Make another epic!

It’s ok to put frosting and sprinkles on later!

(In the same epic, if there is time, or in another one)

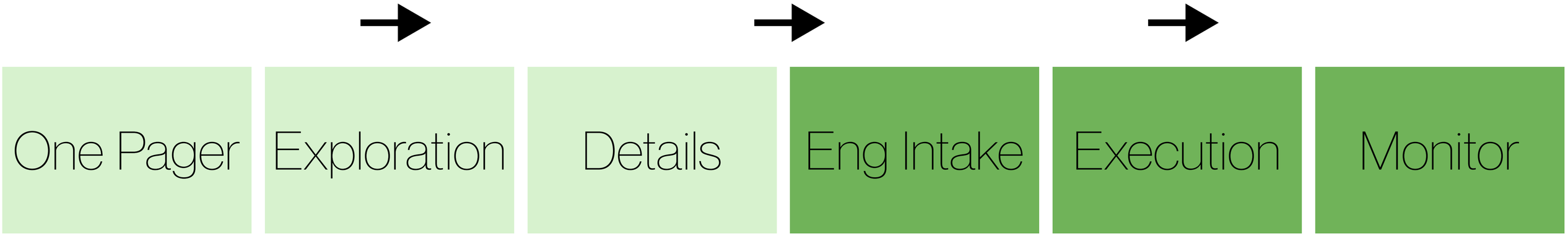
Common mistakes with Story Writing

Writing engineering focused tickets (flour / eggs) vs features

Writing tickets to soon

Siloing eng/prod/design while developing stories

Review


One Pager
Exploration
Details
Eng Intake
Execution
Monitor
Marketing consumable description.
Story map, Mocks, Testing, Iteration
Feature writing with good/ better/best
Estimation Sprint Planning
Build it!
Learn!

Making it happen

Discovery / Story workshop:

1.5 hr meeting, 2x per week
5 people max, ideally 2-3
PM, Design, Eng

Estimation

45min, 1x per 2wk sprint

Sprint Planning

30min, 1x per 2 wk sprint

One Pager	Exploration	Details	Eng Intake	Execution	Monitor
Marketing consumable description.	Story map, Mocks, Testing, Iteration	Feature writing with good/ better/best	Estimation Sprint Planning	Build it!	Learn!

When should you skip this?

If it’s faster. For small things, sometimes just jump to a ticket.

For interactions, sometimes it’s easier to just code it.

Questions?

questions@awkwardengineer.com

Notes

Box cake mix is.... almost cake. You still gotta do some work to be able to eat it. I use the box cake mix analogy for initial threads that connect a system end to end.

Maybe there's a lot of work that's still done manually, or you're faking it with a spreasheet, or you're using Postman. Cake is the goal.

Notes

Wedding cake is still cake, it’s just expensive.