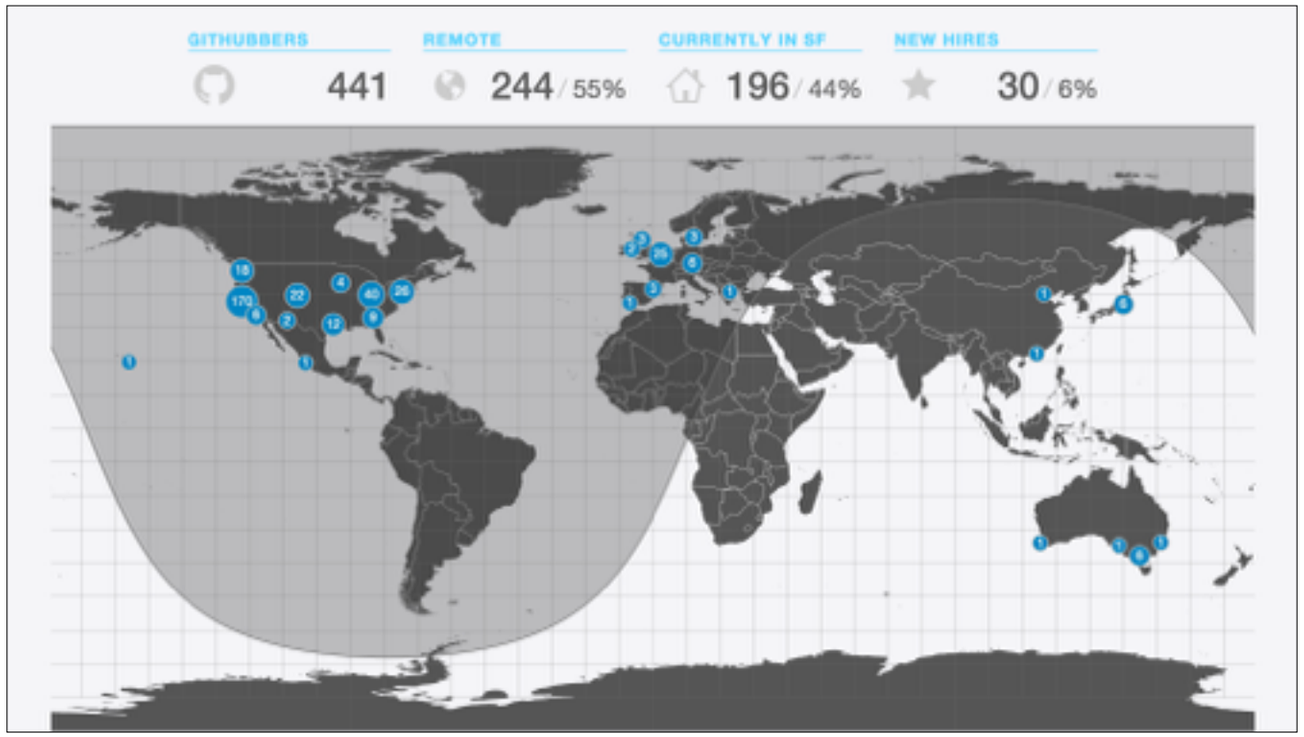
Using the product to design the product

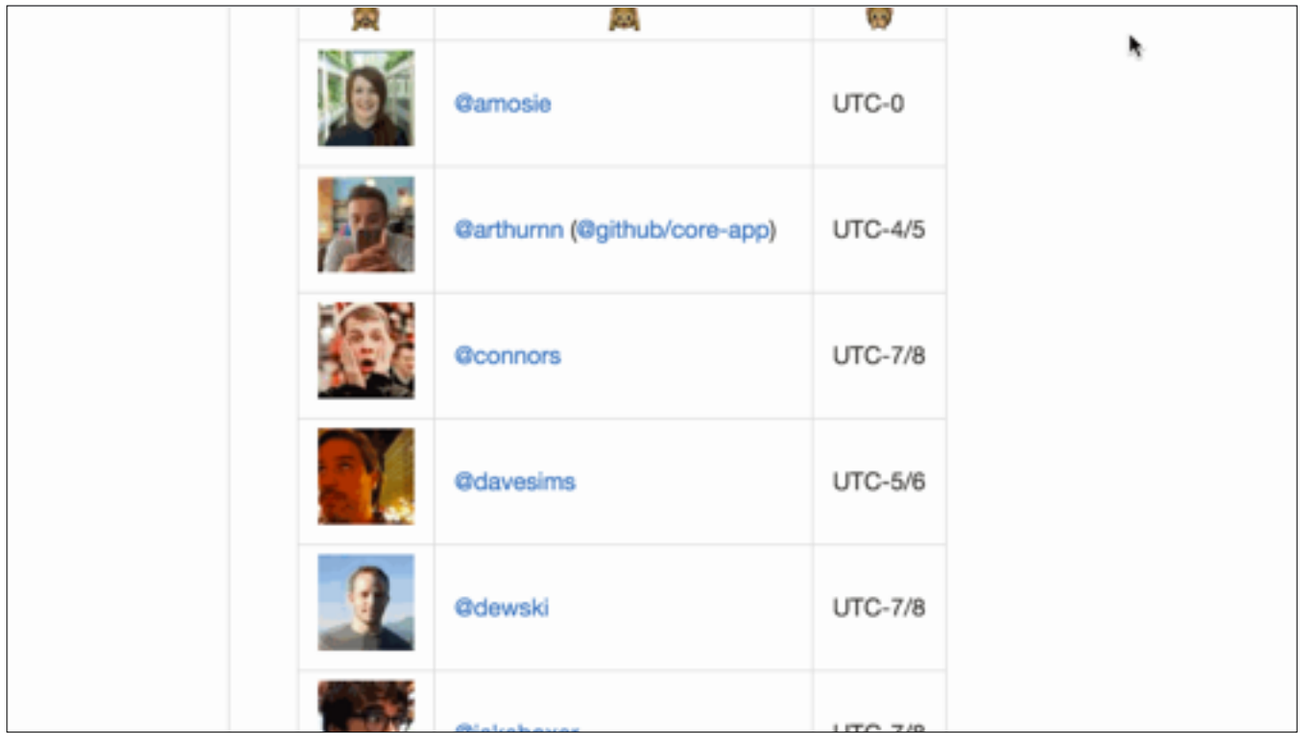
Welcome, friends!

@amosie

GitHub is basically the friendly front end of Git, GitHub allows you to store projects, manage your files and helps you work together with your team mates.

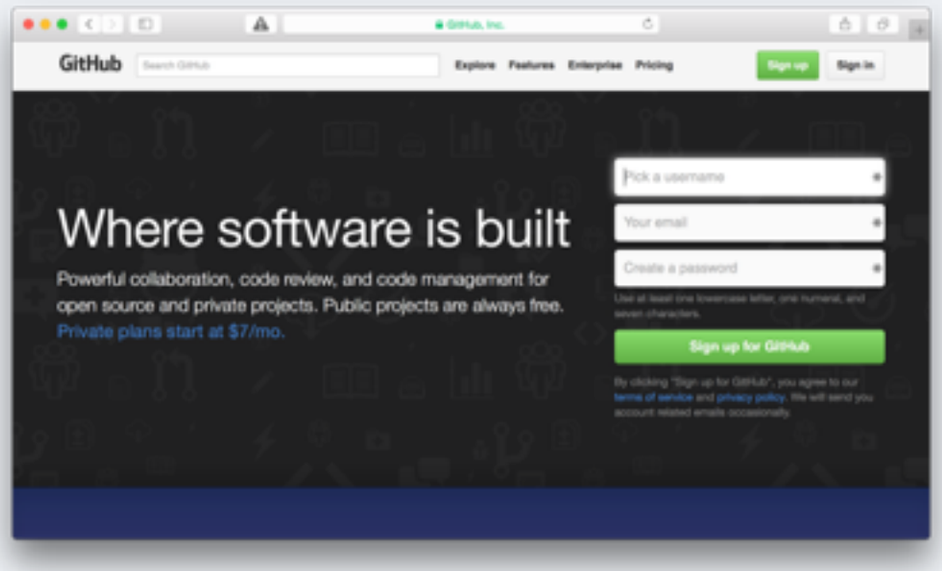


GitHub’s HQ is located in San Francisco, we’re pretty much a remote company as you can see from the map 55% of the company works remotely - I work in London.



	@amosie	UTC-0
	@arthurnn (@github/core-app)	UTC-4/5
	@connors	UTC-7/8
	@davesims	UTC-5/6
	@dewski	UTC-7/8

This is my team - I’m the only one in a European timezone. Most of my team are based out of SF so I only get 1-2 hours a day working overlap

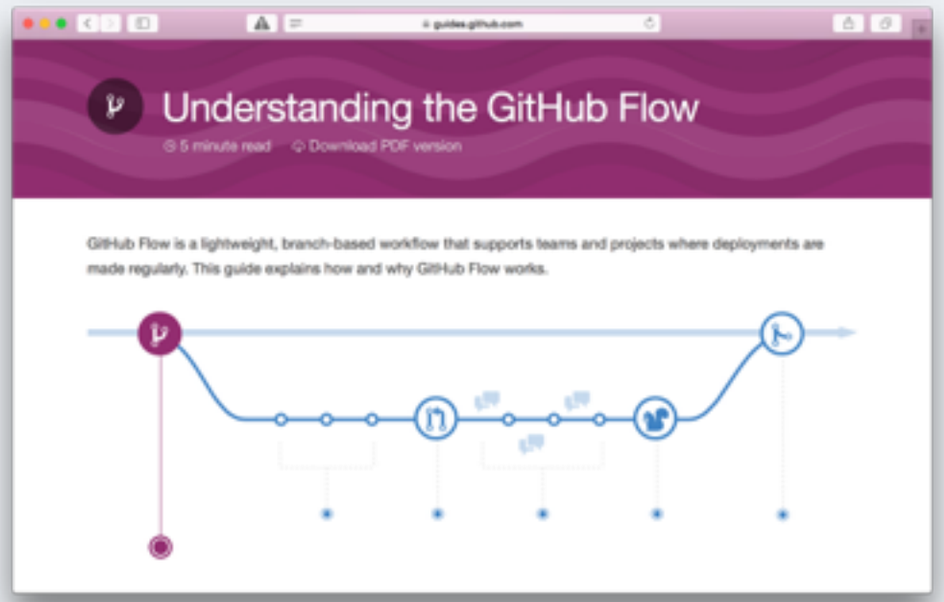


the best way to build and ship software

At GitHub we don’t have front end engineers, all the designers are responsible for maintaining the css and html. So that means I basically spend my whole day in either atom or GitHub

Designing something you use everyday can become weird

But when you spend so much time on one thing, and your designing the product you are using to make the product it can become pretty werid.



This is the GitHub workflow - it’s basically what Github is - it’s the development way of working. Ontop of this though, each team or organisation has their own way of working.

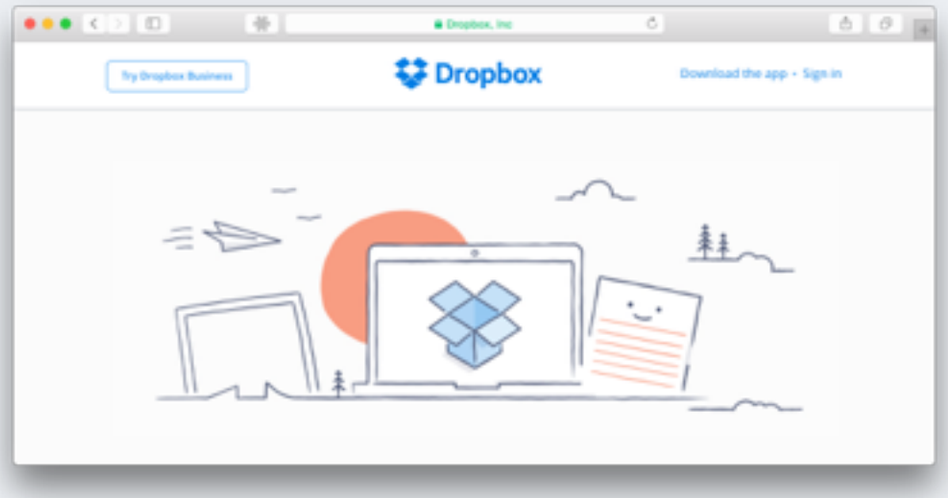
Sometimes we forgot we aren’t normal users

Organisations don’t work in the same way we do, our team is built around the product we are making. Sometimes we forgot that we aren’t normal users.

1.
Large file storage

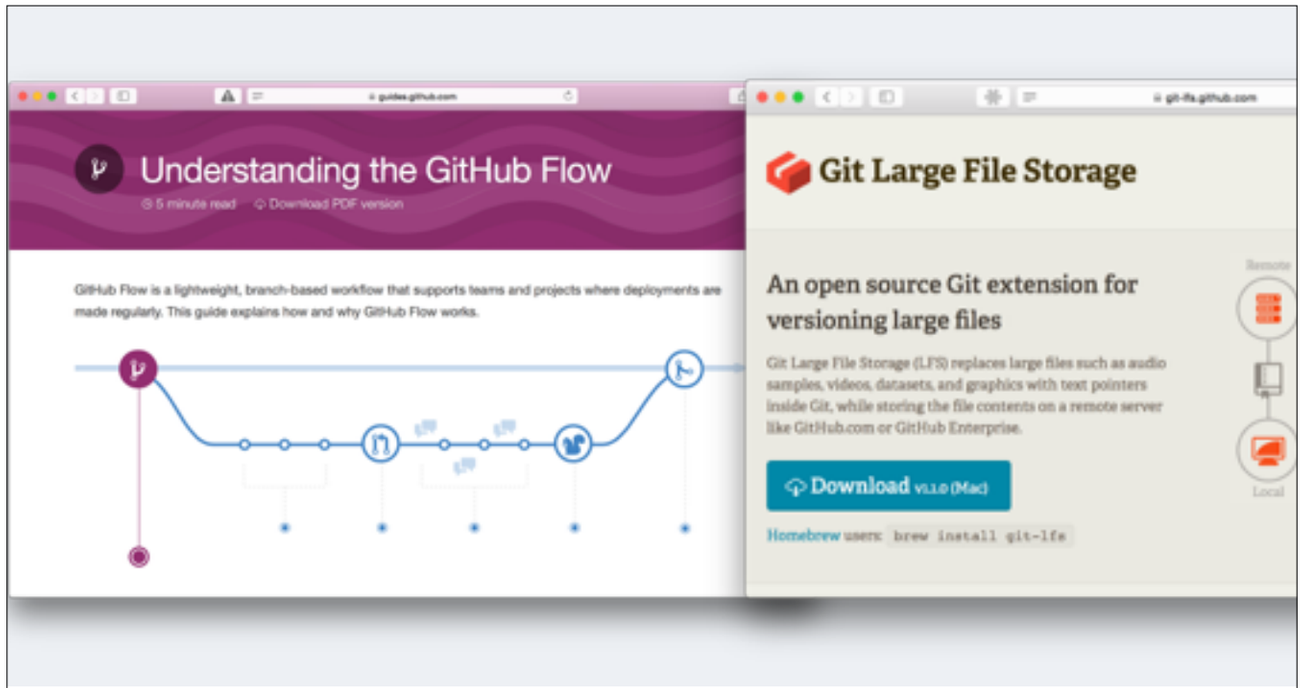
So I’m going to tell you a few stories now, and the first is about large file storage.

Does anyone not know what it is? It allows you to version control large files such as pads, videos and audio files on GitHub - we released it in April last year.



For a long time people used things like....

We'd been telling users don't put large files on GitHub for years, instead we were telling users to use something else, something like Dropbox.



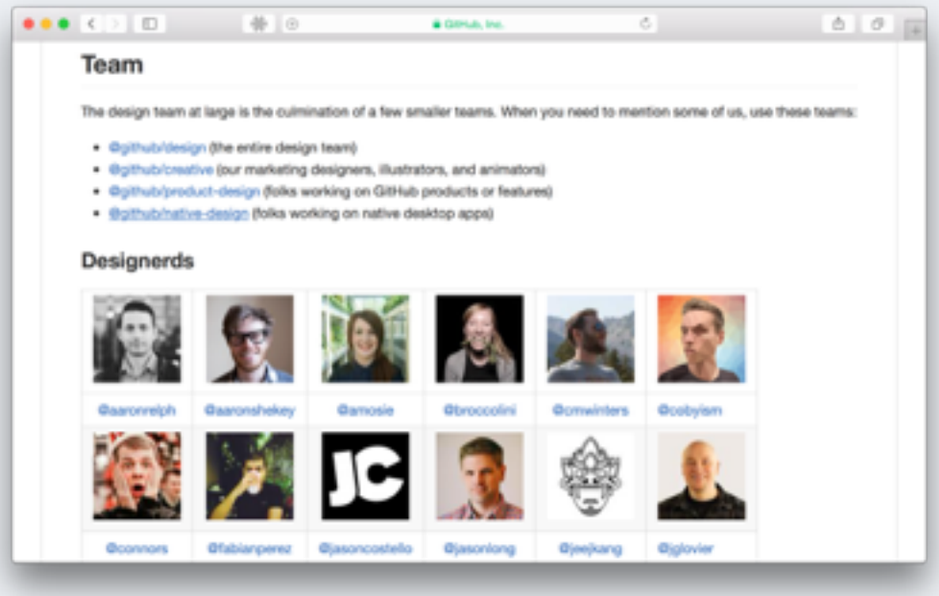
So in April last year, we released LFS. we waited, and waited and waited but people weren't starting to use it. And we wondered why. It was because for years we had been telling people not to put big files on GitHub, they had to alter their workflow and use other services. They were now used to that.

We forgot to design for behaviour change

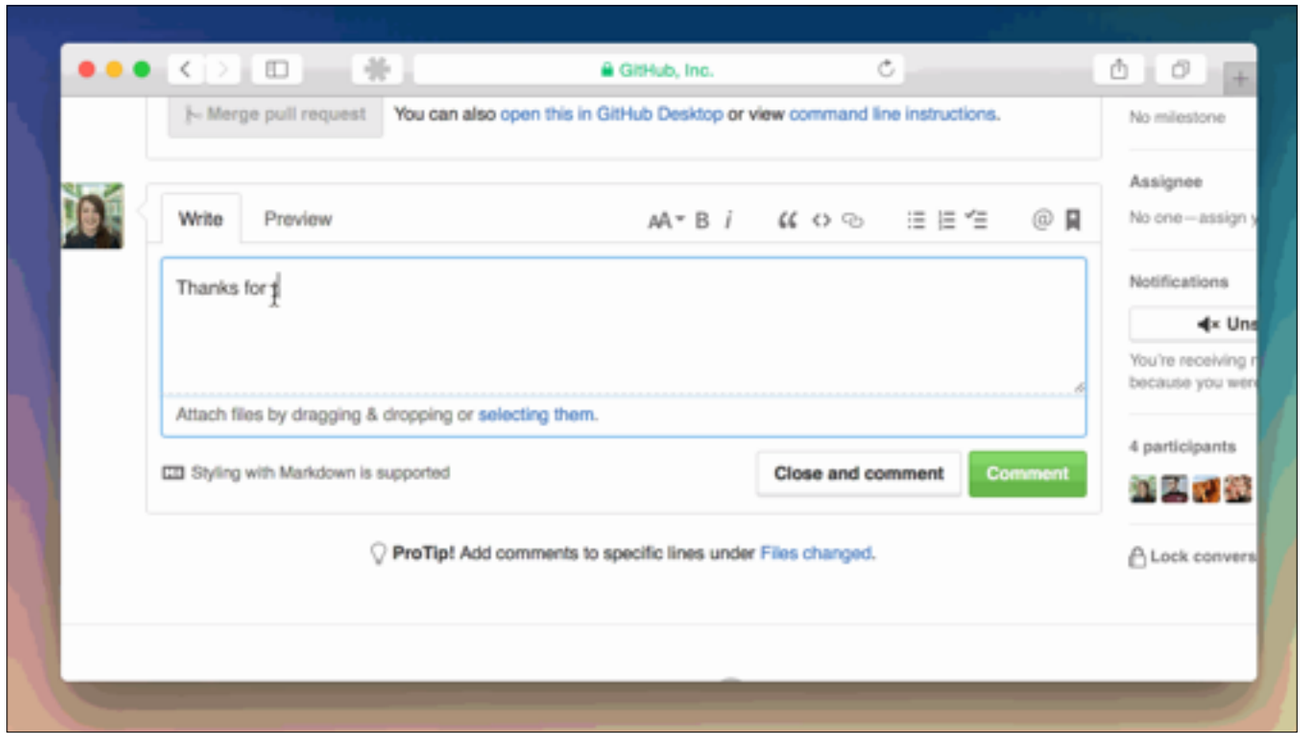
But were we really fucked up is that we forgot to design for behavioural change, we forgot to tell people at the point they needed it that they could now upload large files. We knew adoption would be slow, but we didn’t think know one would use it.

2.
Teams

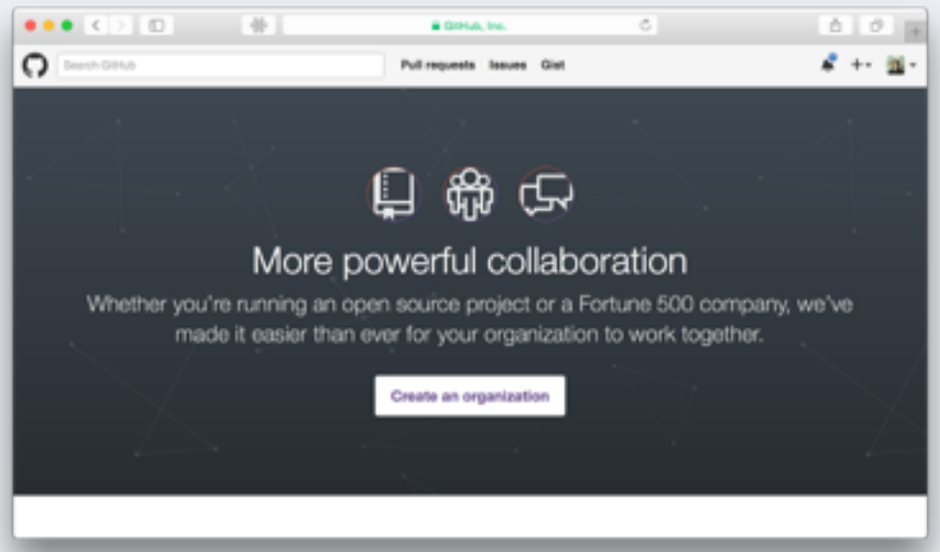
If your part of an organisation you have an access to a feature called teams



Teams are normally groups of people entered around one thing - that thing could be a product everyone is working on, or a discipline. The above screenshot shows the design team!



We use teams a lot at GitHub especially in PRS and issues - it means you can @ mention a group of people without having to type in their names individually.



Late last year we changed organisation permission so that you were in greater control of your users permissions. Before everyone had to have something called an owners team - after you could chose who was in whatever team you liked. Teams however are a new concept.

28% of existing orgs created a new team

We released the product with a select number of customers and started analysing the data, so 28% of existing orgs who had been transferred over to the new model created a new team

9% of new orgs created a team

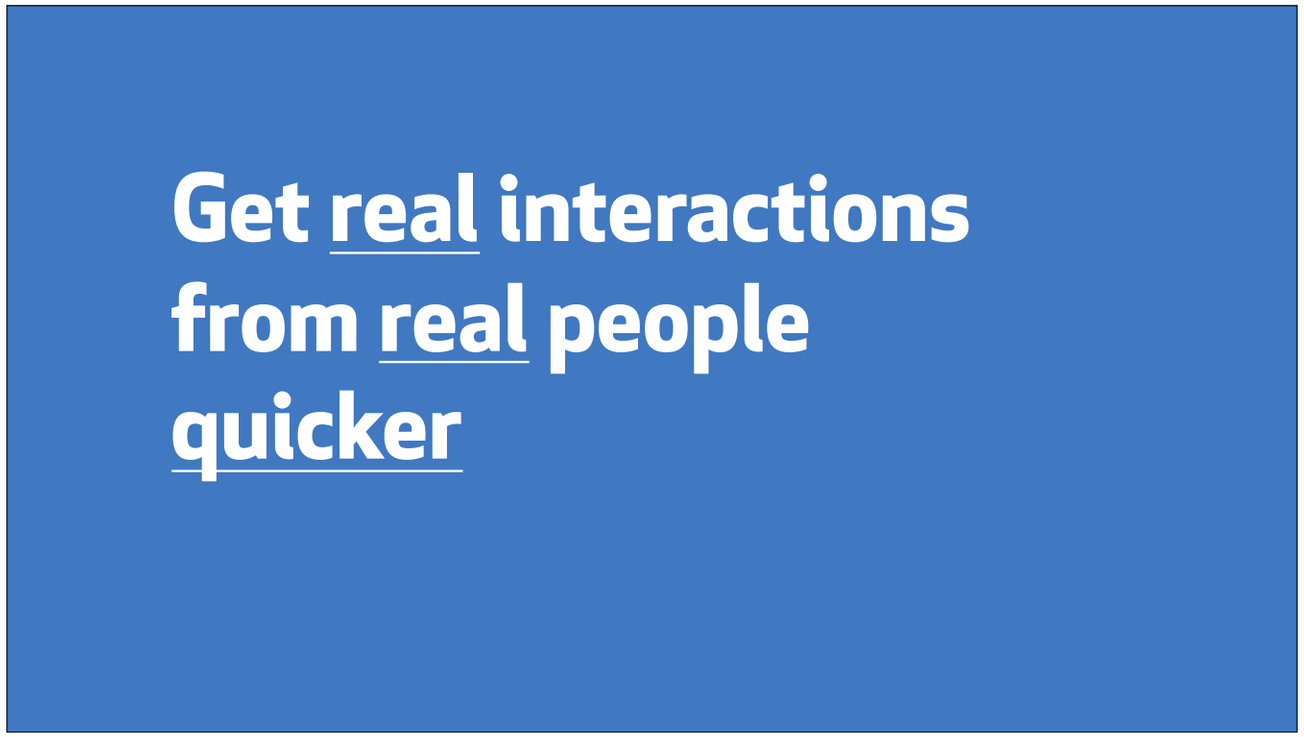
9% of orgs who are brand new to GitHub created a team

Barely anyone used @ mentions

And barely anyone used @mentions. Those numbers are all really low, they're don’t represent how we as GitHub use it at all.

~~We forgot to design~~ ~~for~~ behaviour change takes time

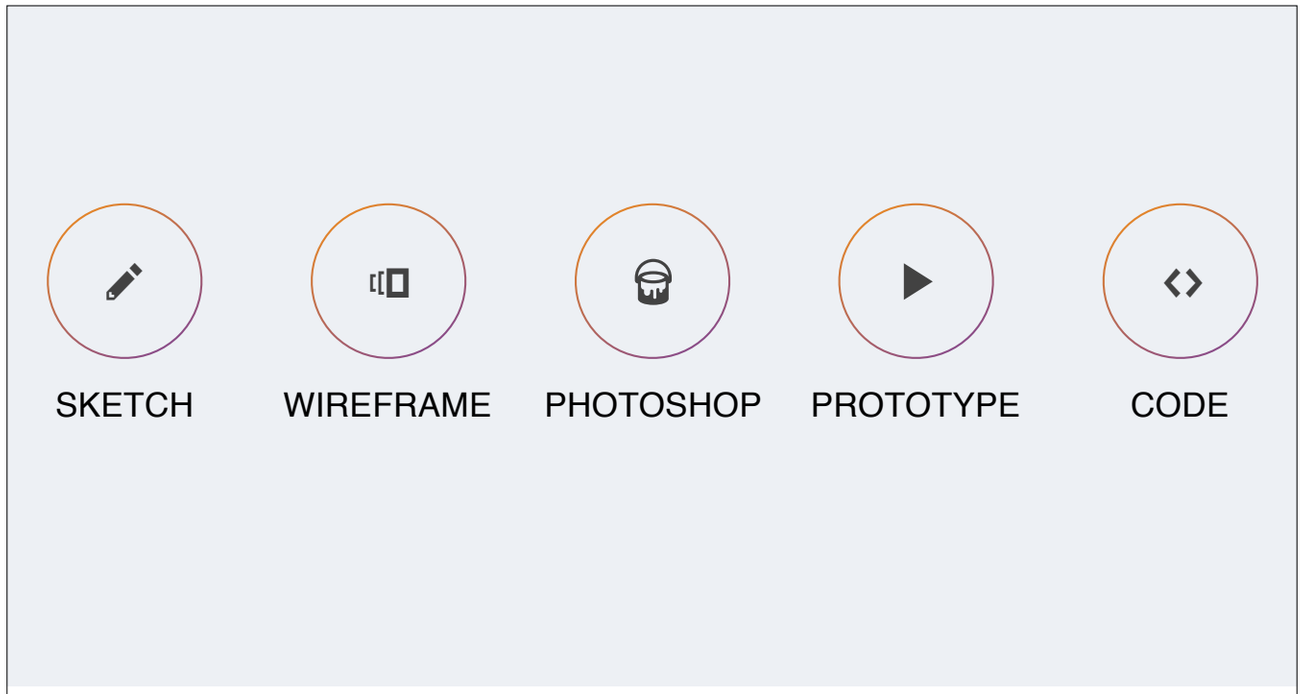
Unlike the first example of gave, we tried to design for behaviour change, but when your trying to change someone’s workflow it takes time for them to adapt and their behaviour to change.



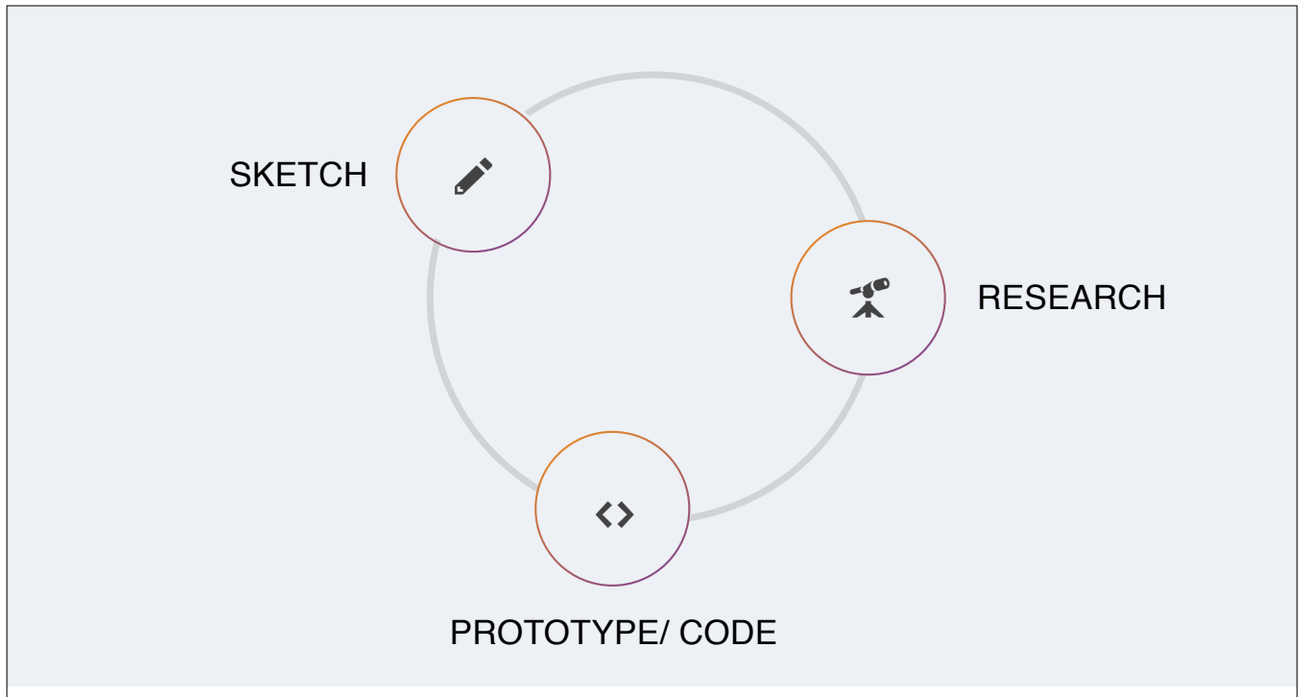
That’s why as designers, engineers and product people we need to put stuff infront of users faster we need to get real interactions from real people quicker.

So I’m going to talk a little bit about my process and why it’s so important to design in the medium that the product is built in.

So This is my Friend Wilson, he’s an engineer at Mozilla. The designers which he works with create spec documents for him, they upload the doc to Dropbox he downloads it and then starts working. If he then has any questions or there is a problem with a design they then have to go back and forth over email - stepping away from their workflow.



In my first couple of jobs I worked in design agencies and this was pretty much my process.

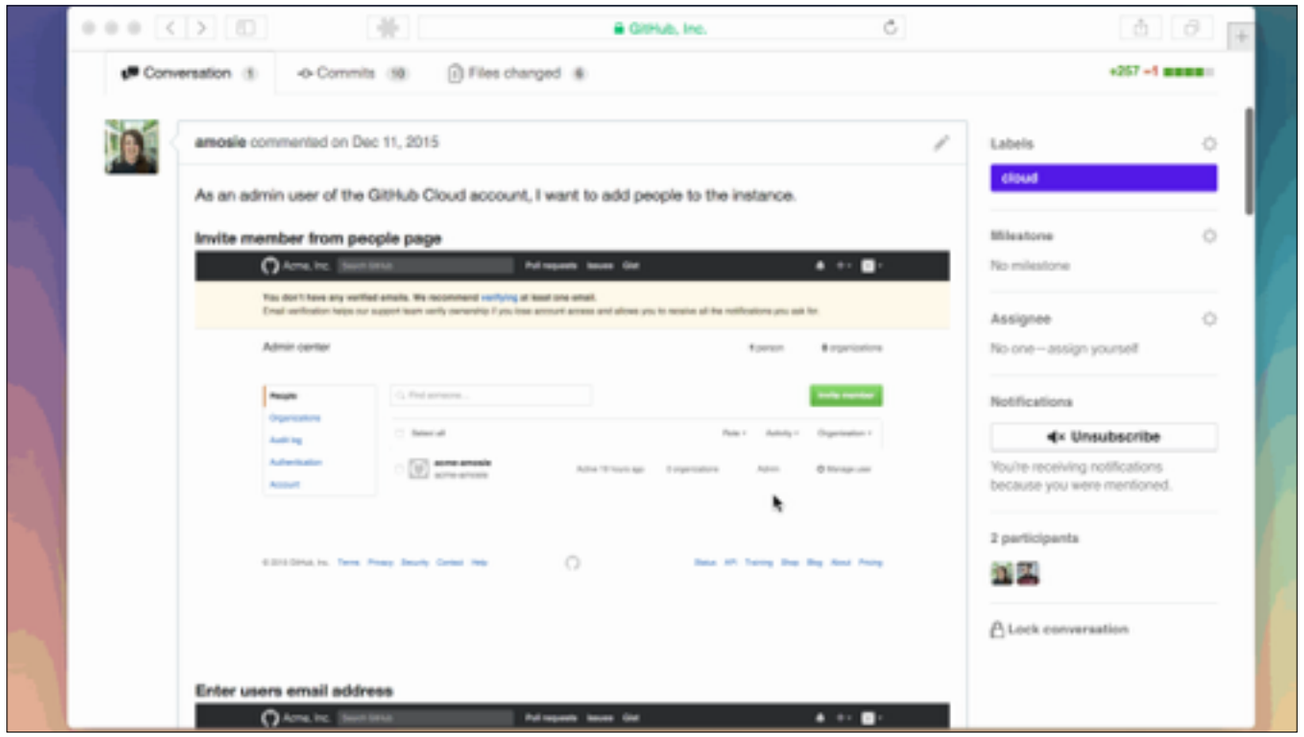


I knew that I could spend my time better, spend more time on solving problems than repeating my steps. Since then I have working of these three steps - design build test and repeat.

It doesn’t need to be hard

Reduce the process, spend more time on things that matter

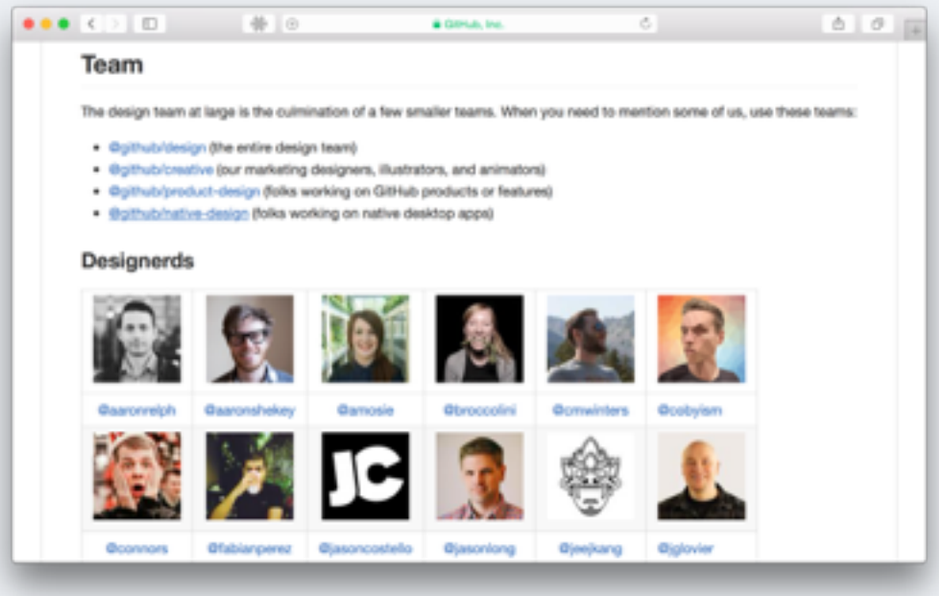
Reducing the process means that you can jump in to code so much quicker, you get to work with real data, real interactions you really get to feel how the product works. It’s super hard to design a transition state, a button hover in a photoshop document.



This is something which I have been working on lately, I did a few sketches but jumped straight in to the html, because we have pretty established patterns for things I don’t need to spend too much of my time repeating what has been done before. I put up something which we call a pull request so I can get an engineer to jump in and make it fully functional.

Design problems arise as it's being built

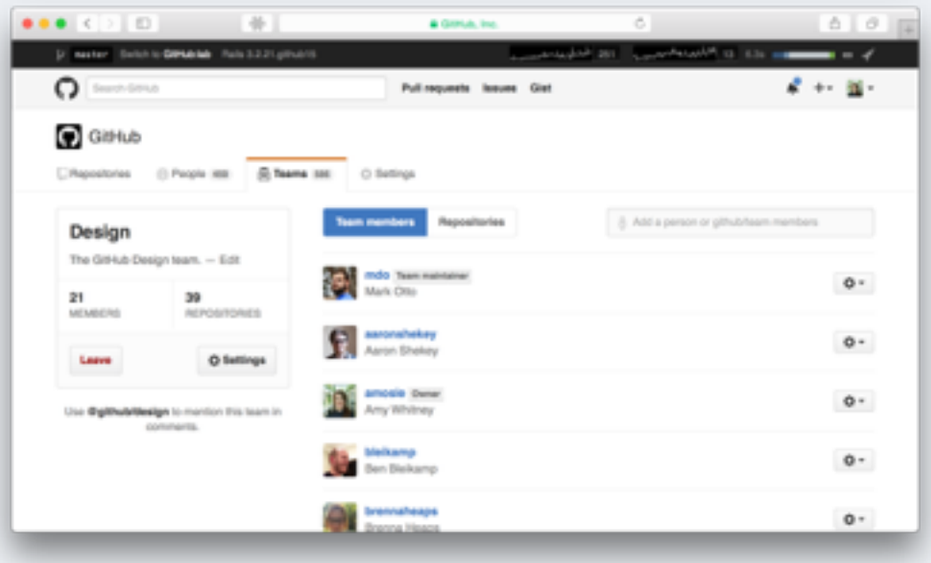
GitHub is a really technical product, and you don't always know problems are going to arise as it's being developed. If your not working in the code it's so hard to spot, fix and design for these problems.



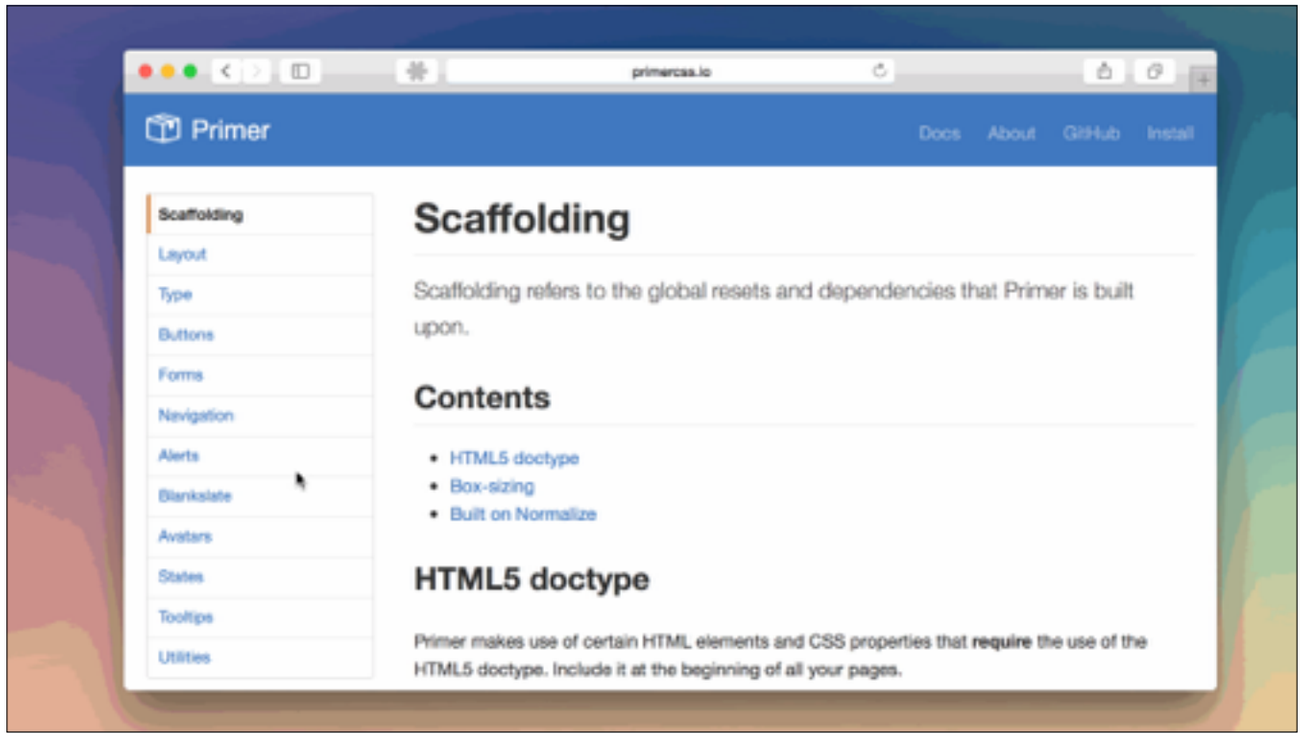
We’re really fond of constantly pushing information at people, they can chose what they read and digest rather than having to ask for it all the time. If you have an organisation account on GitHub there is a really cool feature called teams - when an engineer is working on a problem and wants your help you can just ping the design team or a specific designer - they can hop in to where the problem is and fix it.

Design reusable patterns

Designing reusable patterns, instead of having multiple variations of patterns



Here you can see there is a list of users and each user has an avatar (there photo), their real name and their username. There are loads of similar looking user tables across GitHub, now if each table was even the slightest bit different that’s a hell of a lot of extra work, code and upkeep. We try and not do this.



Instead we have a front end style guide. So if you go to primercss.io you can see ours - it’s opensource. Design patterns which work across pages, which you can test in research and keep iterating on once not 1000 times.

Get stuff in front of users faster

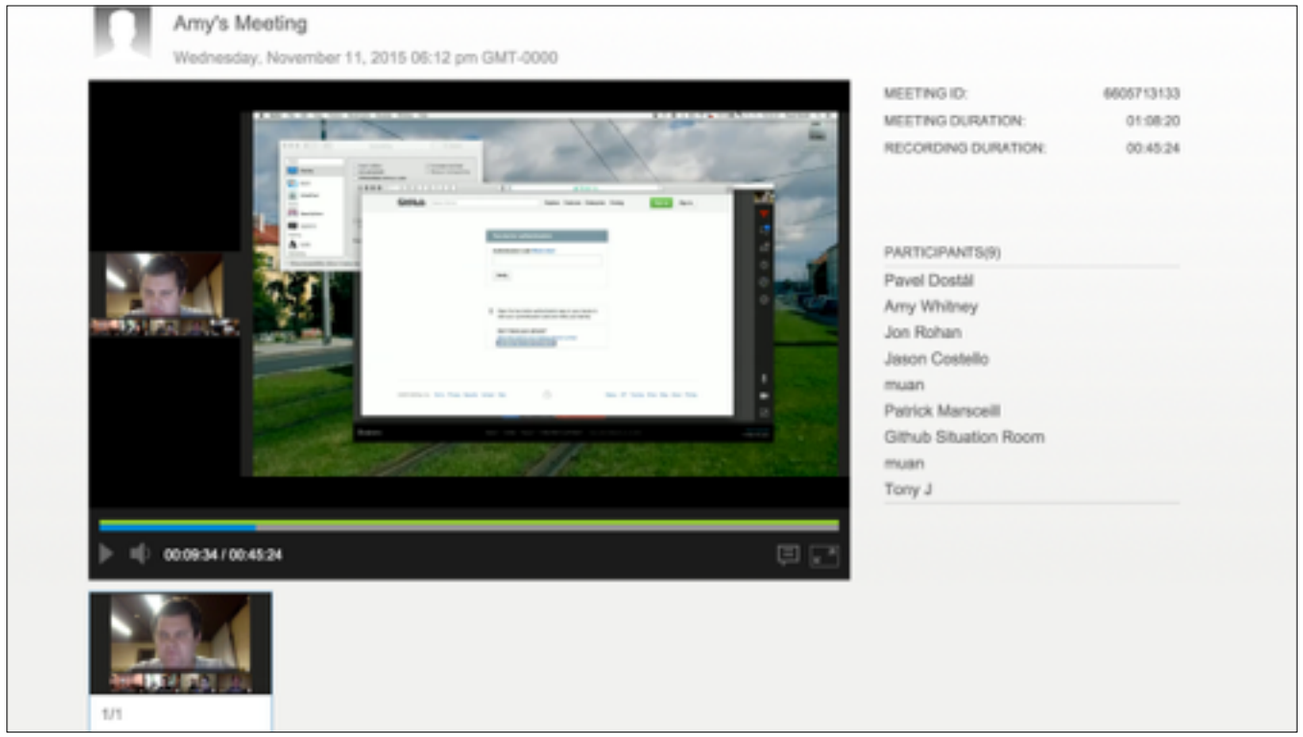
Fail early and often, it’s so much better to put something in front of users early than spend months working on something that your not quite sure if it solves user needs.

Prototyping
Staff ship
Early access

We have a few ways that we do this at GitHub, we make rough html prototypes and put them in to user testing. As we are a remote team we do this over video.
We staff ship a lot of our features before we release them to get rid of any teething products we may have, if it doesn't work for us then it's not good enough to go in to production.
We get groups of users to sign up to something which is called early release - we work closely with the support and engineering team - the customers which sign up know that there might be a few teething problems but are happy to join early and give feedback.

Better to comment on how the thing works rather than how it looks

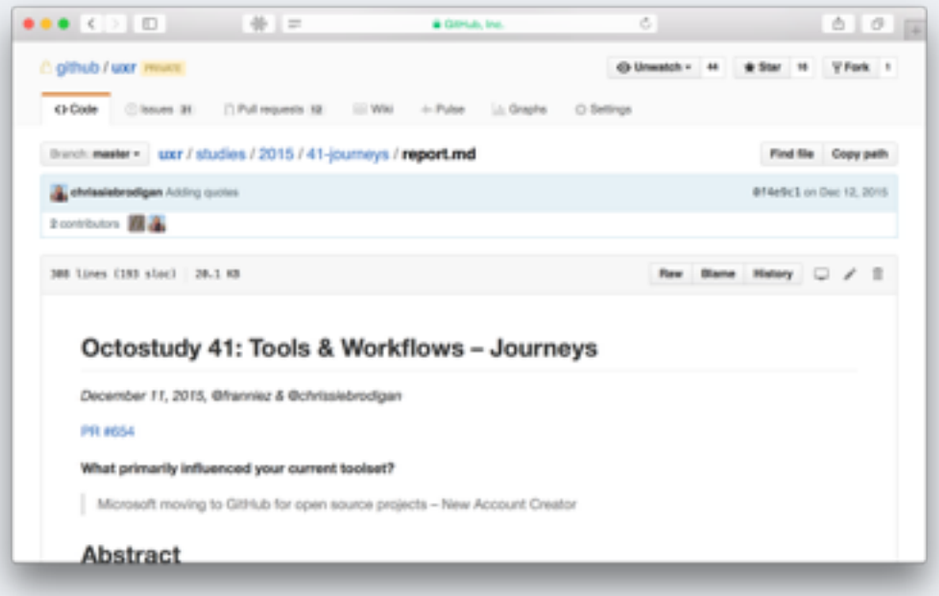
If I create a page in photoshop and then send it to someone then they are going to comment on how the thing looks rather than focusing on does it actually work for them.



Here the design team are talking to a visually impaired computer science student - he is using GitHub using a screenreader. We'd never be able to design stuff accessibly without working in the html.

Putting research next to the product

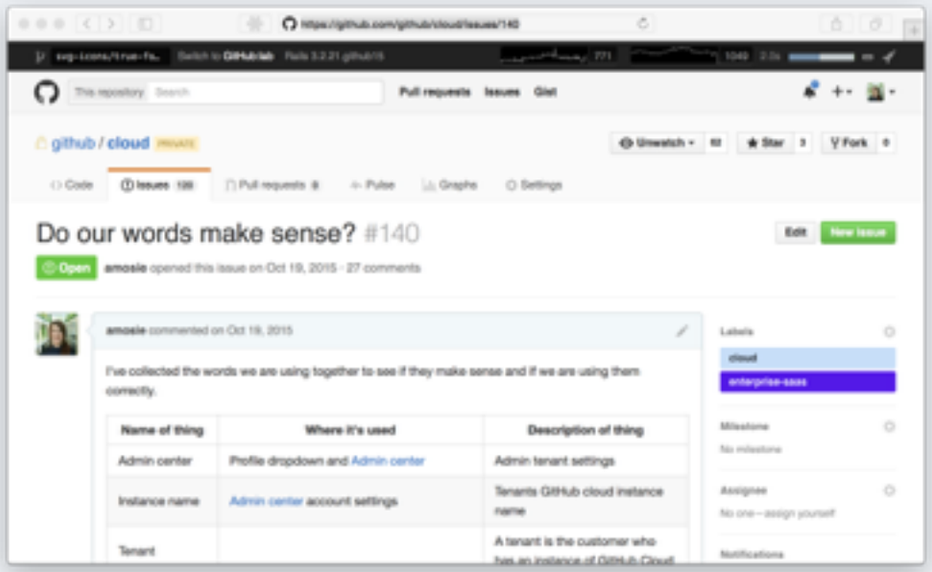
I think this is my favourite thing about GitHub, so I've worked in other tech places and the designers have been using GitHub and working with the developers but not the researchers too.



So the research team - yay! Works on GitHub too, all the research lives on and next to the product. This includes case studies of specific products and features, we also use labels to take problems so for example users really struggle with dropdowns across research studies so we can get easily searchable and better understanding of what patterns work and what don’t. It’s also a really great way for engineers to read the research as they literally can’t get away from it.

Everyone speaks the same language

I’m sure this has happened to you before - your designing a product that is somewhat technical and that technical word slips out in to the product.



Working together in the code forces you to speak the same language, something which is plain and you both understand.

Having a design history is ace

Put your hand up if you have had a file which contains the word final before?
Git and GitHub is version control - it allows you to see who has done something and go back to the piece of work we call them pull requests to see who has done it and why they did it. 1 file but multiple revisions.

Designing asynchronously

So earlier I spoke about being a remote employee.

Issues

Pull requests

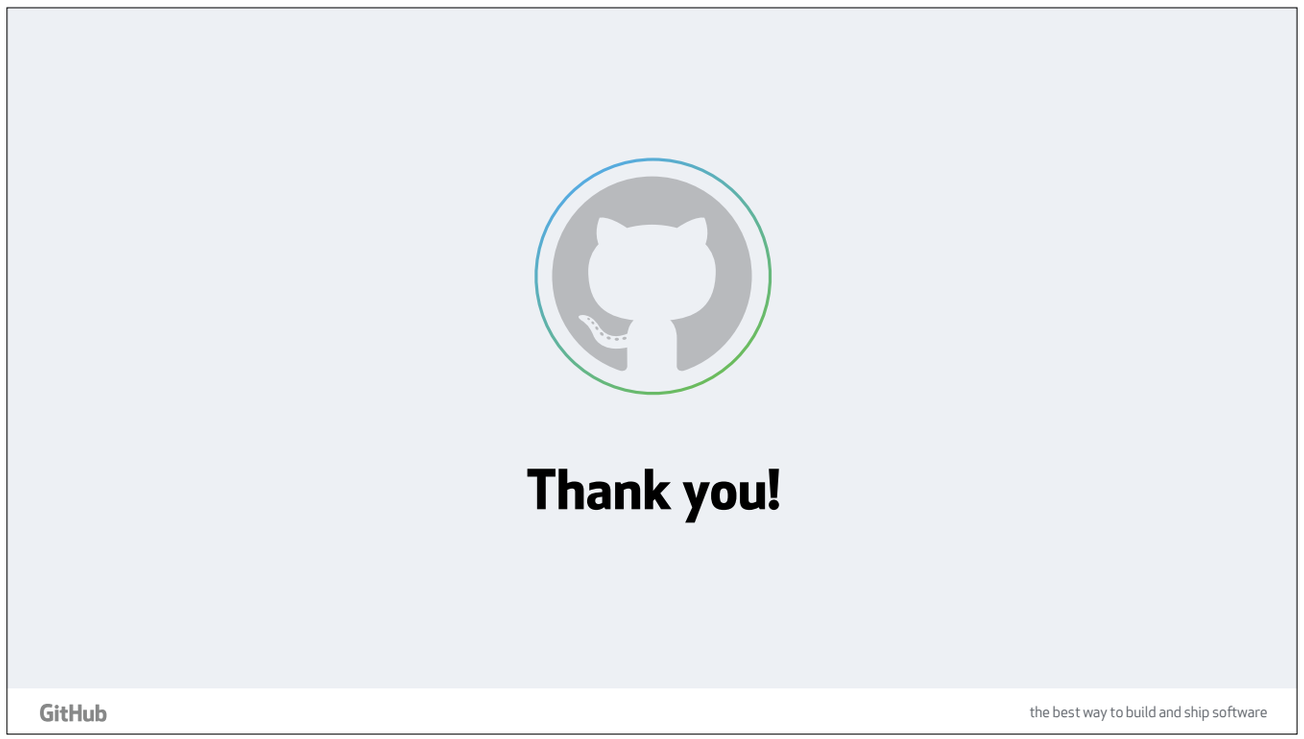
Communication is the absolute key to this - when sketching I write up my flow and ideas in something called a GitHub issue. I need to make sure I am so clear that someone who is 8 hours ahead of me can understand it without having to come back and forth asking me questions. When my flow is in a good enough state, I jump in to code. My code editor of choice is something called Atom - I open up the PR super early so the engineers can jump in with the back end and the team can give me any feedback they have. The less time I spend, wasting my time on things that won’t work the more time I can spend iterating and making things better.

Even the best engineers code isn't perfect

No one's code is perfect, it can always be improved. Don't be scared of jumping in to the code and opening up a pull request if you don't know how to fully do it. Engineers can hop in, they can pair with you, walk through what you don't know and teach you so you can do more next time. Be it on your work or a project in the open source community.

We're hiring!

https://github.com/about/jobs



Thank you and any questions?