



The revolution will be
collaborative

About Wiredcraft & myself	4
About Jekyll	8
Best practices	12
Multilingual	21
Dynamic Jekyll	29
Speed	35
Conclusion	38



Some context

We work with digital & innovation teams within larger multinational companies. We help them **move at the speed of startups & develop a product culture.**



Google



CNN



Etam

BASF
We create chemistry



Dior



Walmart ✱



Hilton



阿里巴巴 *Alibaba.com*



What does it mean?

We're the team behind the omnichannel strategies of Walmart, Nike, Starbucks, Hilton & many others, in Europe & China.

We ship software solutions
at high velocity...
For tens of millions of users.



Disclaimer

This talk may suck

- I just flew in from a Shanghai → Berlin → Dallas → NYC → Berlin → Paris → Berlin → Shanghai.
- I forgot about this talk and had to prepare my notes on the plane (with no internet, thanks Lufthansa).
- It will probably be shorter than 20 minutes.
- Also, I may swear a lot.

Story time

I'm the **Chief Janitor Officer**: I do whatever is necessary for our team to get sh*t done and move in the right direction.

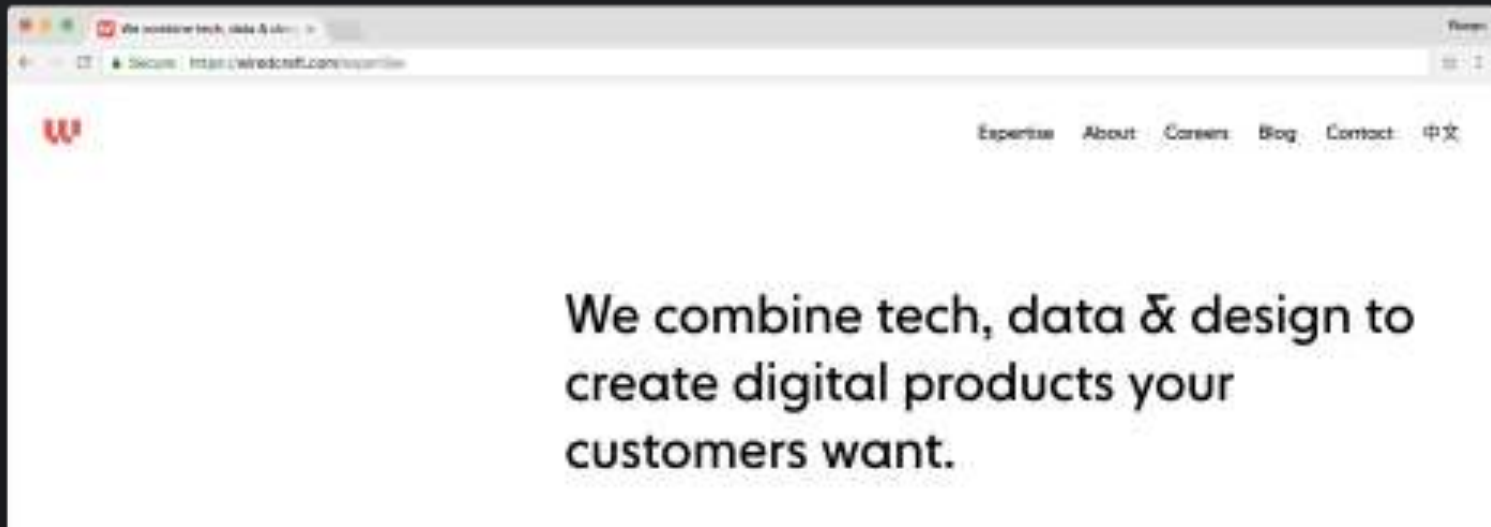


Ronan Berder

CEO & Founder

What do I know?

- Jack of all trades: design + code + marketing + business.
- I've been designing products since ~2001.



~~Adobe~~

~~Inkscape~~

~~Sketch~~

Figma

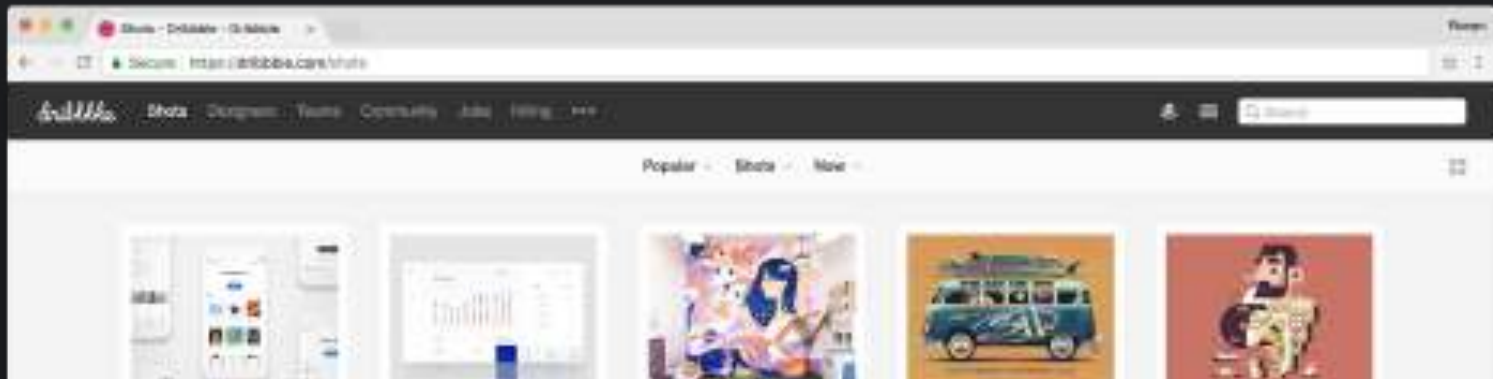
I started by learning with Photoshop on my own in 2001.

For the past 8 years though, I've used these tools mostly for colors & logos.

What's the problem?

Most designers are big crybabies

- They want to be left alone to play in their sandbox.
- They come back only when things are “finished” & expect a pat on the back for creating pretty pictures.
- Want proof? Go check Dribbble.



Why's that bad?

Designers waste the opportunity of sharing failures and process:

- That's where most of the learnings are for themselves and others.
- Additionally it skews the perception of what design work is.

I wrote about it before: wiredcraft.com/blog/engineering-creativity

Engineering creativity

Why we think most creatives are bullshit and how our team engineers creativity.

CREATIVES
are
BULLSH*T!

~

AKA → ENGINEERING CREATIVITY



Bad compared to what?

Software development

The tech community worked on collaboration issues very early on.

Proof is that Open Source is the big winner of the past few decades:

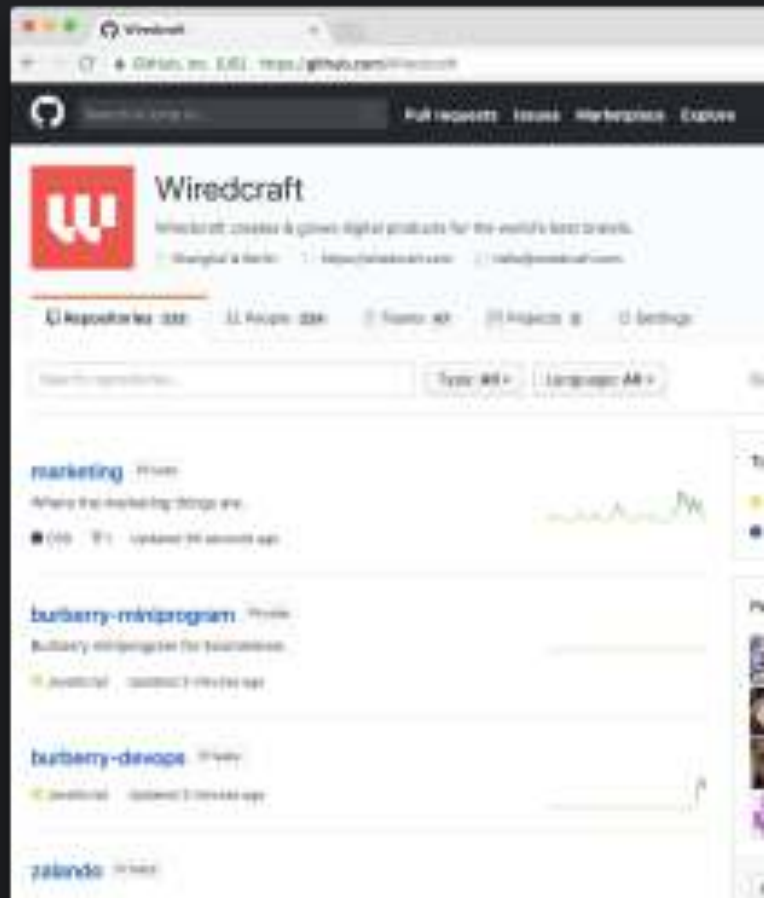
- It helps power most of the software you interact with.
- It tackles very tough challenges.
- It cross-pollinate innovation.
- It helps educate & grow developers.

The tech community is light-years ahead of the design community.

GitHub vs. Dribbble

- Work at all kinds of scales (individuals all the way to Microsoft & Google).
- Discussions, failures, criticism, learnings... all happen in the open.
- It values sharing over personal gratification.

This is a community geared towards the collective vs. the individuals.



Methodology

Openness

Collaboration

We addressed the first two in "Engineering Creativity".

They're necessary conditions for the 3rd one. But make no mistake, collaboration is where value is at.

Then what to do?

Collaboration with users

Collaboration with stakeholders

Collaboration across teams

Collaboration among designers

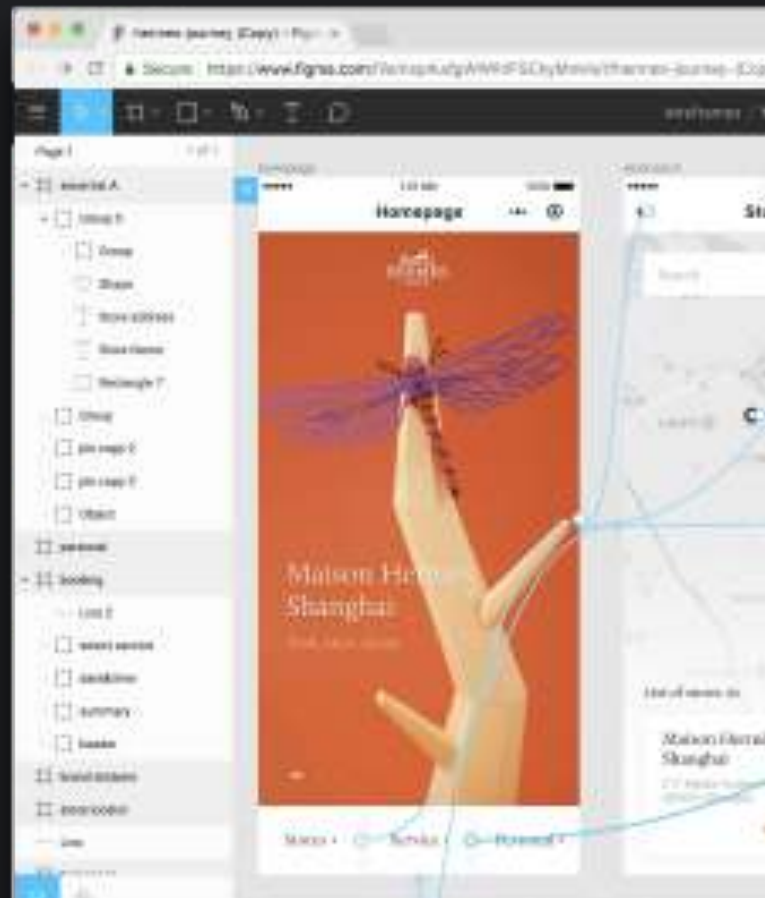
... and a secret weapon.

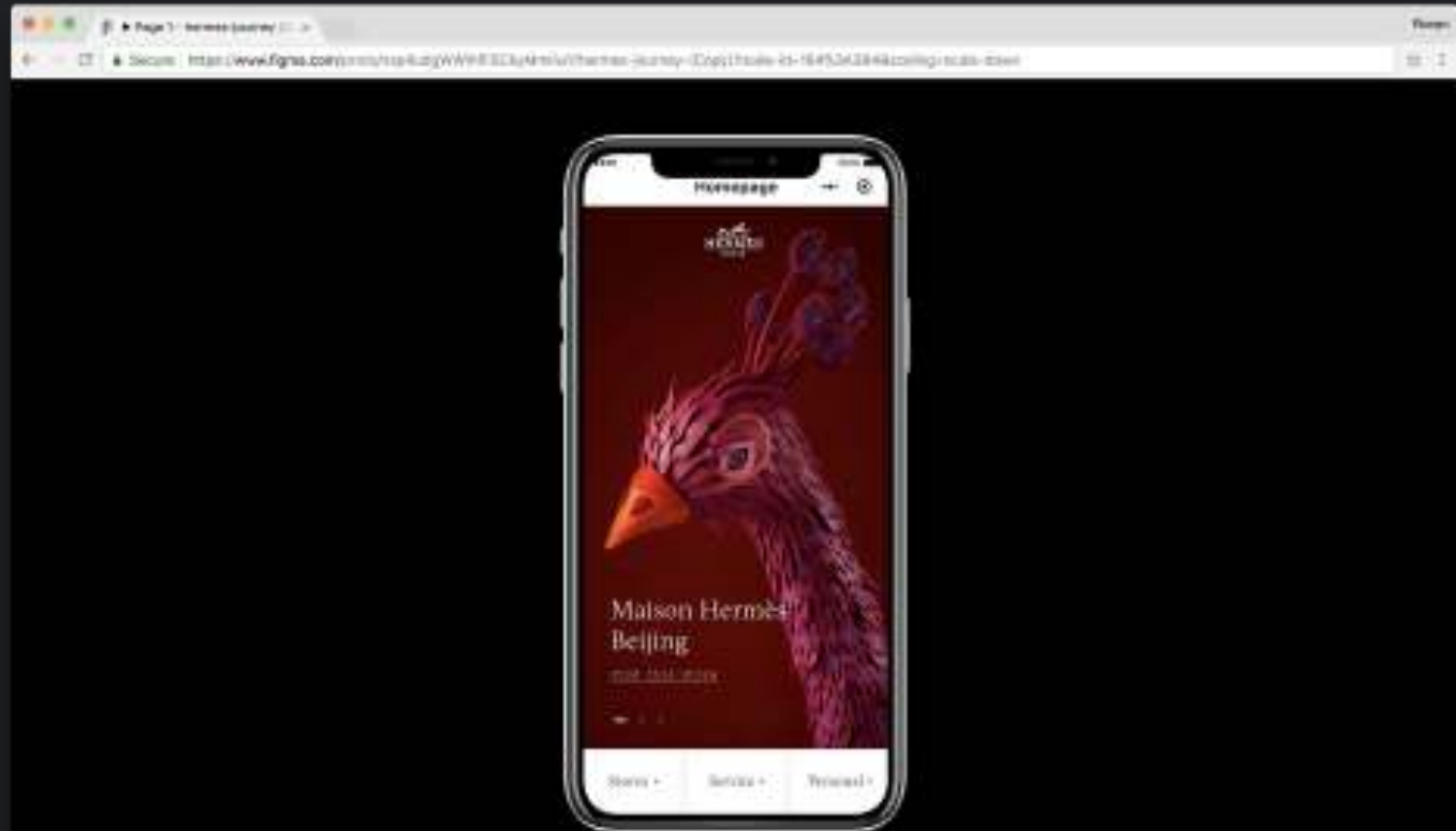
Prototyping

Collaboration with your users.

Get your design in front of actual users as soon as possible to validate your assumption.

This is the core concept behind our Product Design Sprint.







The Wirecraft Playbook | Product Design

playbook.wirecraft.com/en/inspiration/design/

Playbook

What is your question about?

Home > Design

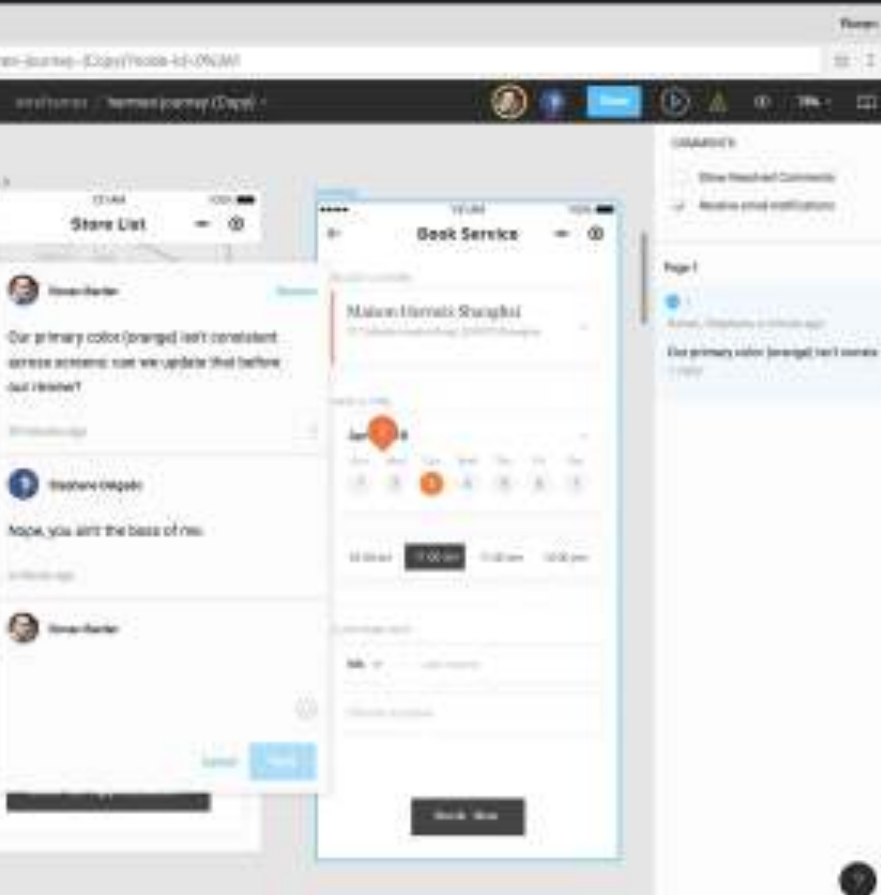
Product design

Our **product design process** helps us improve the odds of **making what we want** by focusing on the **idea/validation feedback loop**. We apply it to what we do; it helps us design marketing experiments, write business plans, and execute company strategy...

We took inspiration from **Google Ventures' Design Sprint**, giving iterative

- **Understand:** establish the context and problem (e.g. "What data do we need to solve this problem?"), list our assumptions and define how we intend to solve it.
- **Diverge:** generate multiple (potentially conflicting) solutions to solve the problem.
- **Converge:** align the team around one approach (e.g. a single user flow).
- **Prototype:** quickly build a prototype we can test.
- **Validate:** test the prototype, usually with actual users outside of our company, to see what works and what doesn't.

Each cycle helps us validate and invalidate our assumptions, and get a better understanding of what works.

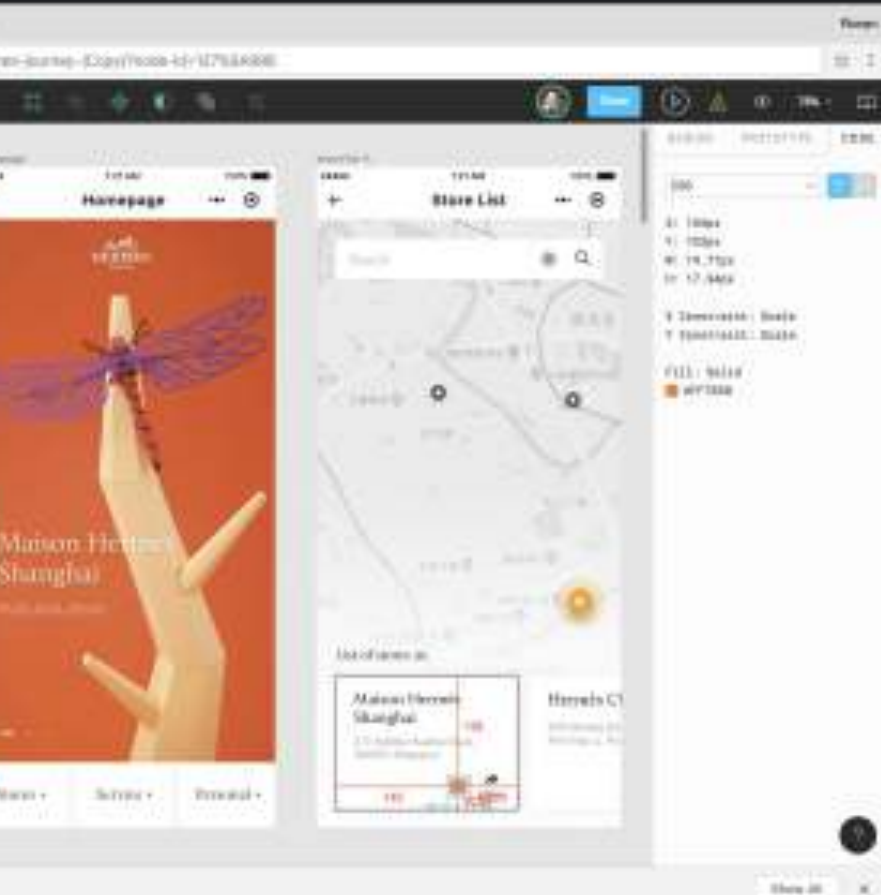


Commenting

Collaboration with stakeholders.

Get a discussion going on specific parts of your UI/UX.

Great for client feedback in view only mode.



Code

Collaboration with developers.

Important to note that this doesn't require additional seats or installing software.

Also, it works on Linux, MacOS and Windows since it's Web based.

A few extras...

- Vector networks
- Team libraries
- Version history
- Sketch import
- Constraints
- Price

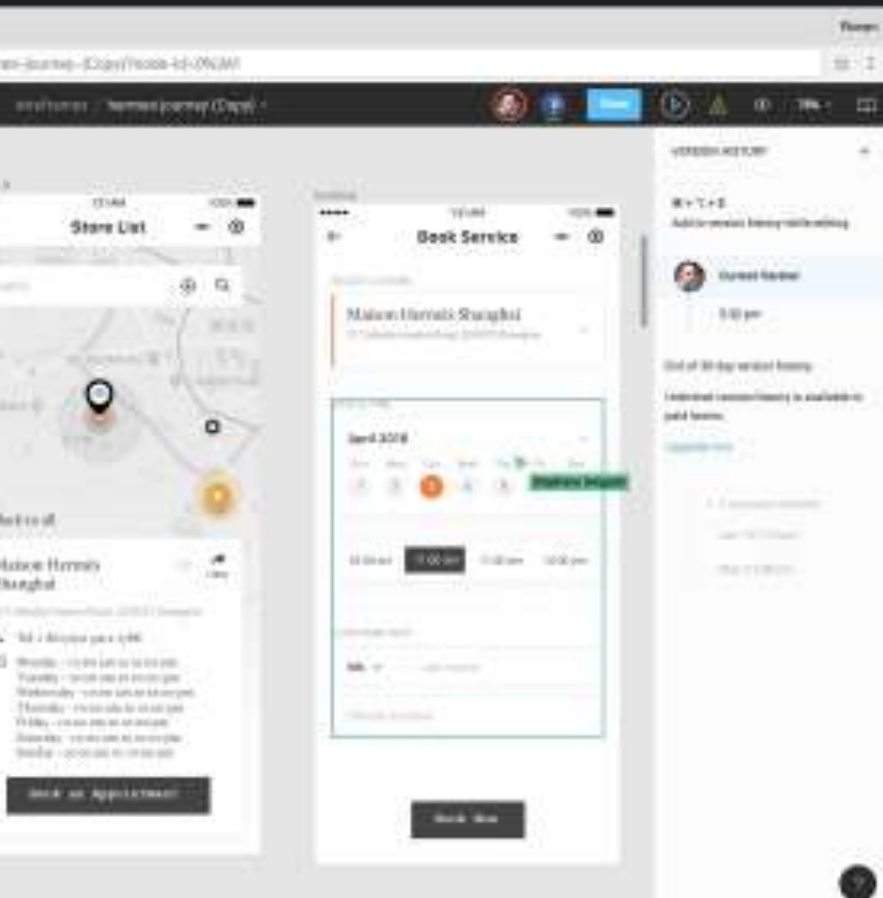
So what?

I can already do all of that with my tools...



Abstract

Why Figma then?



Secret weapon #1

Real-time collaboration is nice but the real value is in the canonical URL.

No more: “Is it the latest file?” or “What version do you have?”.

It’s the main barrier to entry for true collaboration.

Secret weapon #2

Figma is building an API.

This is the first step towards automation and sharing across services.

We're about to witness an explosion of innovation in the UI design field.



Hard to grasp the potential:

- Generative design
- Automated image manipulation
- Programmatic generation of alternatives
- Code generation
- ...



5. Syncing Figma designs to React components

One of the API opportunities we are most excited about is the potential for automatically converting Figma to other frameworks. We've already seeing it: PagerDuty introduced a React integration for others to see, and Sara Vieira, a front end developer from Portugal, built a way for React components to be rendered directly from a Figma file using the aforementioned GraphQL connector she worked on ([Link](#)).





GitHub



(The Figma design tool in Figma (left) and GitHub (right))

With the new Figma API, GitHub can push changes to their icon system —
Designs — from a single design file. When someone needs to alter an icon,
they can simply edit the Figma file, which serves as the source of truth. A



Wrapping it up

The future will be open

I'm sure designers will find many reasons as to why it can't work. People did the same when Open Source emerged.

But openness prevailed.

The competition between figma and the others is already bringing interesting innovation (e.g. Sketch Open Format).

An API-driven strategy will only accelerate this trend.

Questions?

Get in touch: **hello@wiredcraft.com**