

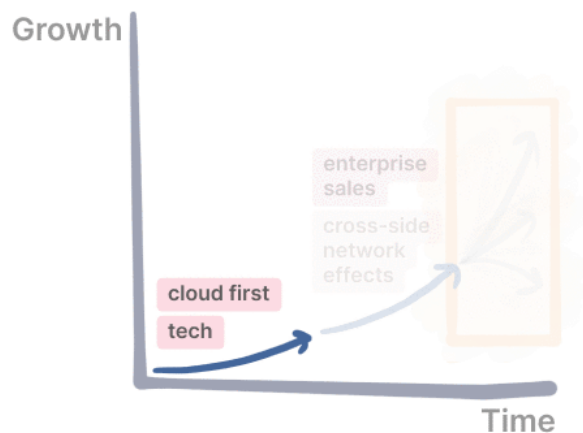
Why Figma Wins - kwokchain

Kevin

36-46 minutes

Companies are a sequencing of loops. While it's possible to stumble into an initial core loop that works, the companies that are successful in the long term are the ones that can repeatedly find the next loop. However, this evolution is poorly understood relative to its existential impact on a company's trajectory. [Figma](#) is a prime example of sequencing loops. They're now widely viewed as successful, but the key factors in their success and what bets they must make to get to the next level are less widely understood.

Sequencing Loops



bet you didn't know I could make gifs

The core insight of Figma is that design is larger than just designers. Design is all of the conversations between designers and PMs about what to build. It is the mocks and prototypes and the feedback on them. It is the handoff of specs and assets to engineers and how easy it is for them to implement them. Building for this entire process doesn't take away the importance of designers—it gives them a seat at the table for the core decisions a company makes.

Building for everyone in the design process and not *just* designers is also the foundation of Figma's core loop, which drives their growth and compounding scale. That network effect is made possible by Figma's key early choices like:

- Architecting Figma to be truly browser-first, instead of just having storage be in the cloud
- Their head start in new technologies like WebGL and CRDTs that made this browser-first approach possible
- Focusing on a product purpose built for those designing vector based digital products

Figma's compounding growth is not only due to product market fit, but is also driven by the alignment between their product and distribution. There are limits to Figma's success if it remains only valued and spread within companies. In order to break through that asymptote, Figma must build a global network effect across the ecosystem. Figma's value to new users should compound as Figma's adoption grows,

even for solo users outside of organizations. Figma has begun making its bets on how it will become a platform—namely centered around communities and plugins. While it's still early, these bets can be unpacked and understood.

Many companies are now at this inflection point. They have found success with their core product and are trying to push themselves to the next level of value to customers and scale. Our understanding of building platforms and sequencing towards them is still nascent. From how to shift the allocation of resources between a company's core business and its potential future expansions to how to structure a platform to catalyze its growth and more. Until the playbook is well understood, it is more art than science. There are many decisions to make, including which layers should be centralized, whether the ecosystem should be driven by open source contributors or profit-seeking enterprises, how broad in scope to allow the ecosystem to grow and where to not let it grow, and more.

Whether they can catalyze compounding productivity in design is a core question for the coming years. Engineering is one field that has exhibited significant compounding progress over time. Whether Figma's bets in building out global network effects will be able to push this level of progress in design will be an important question to watch.

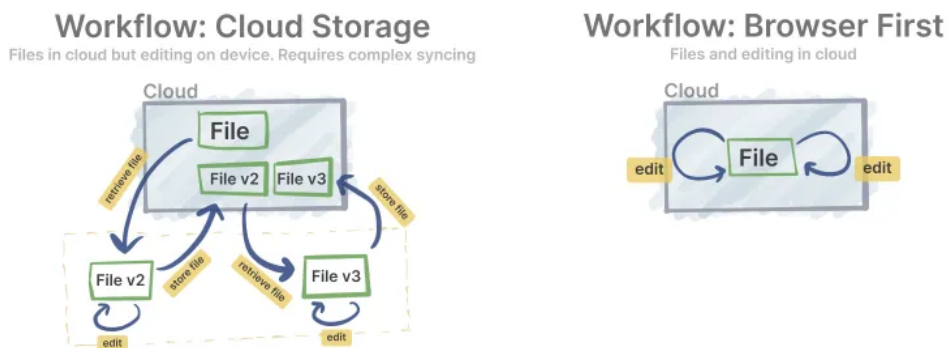
The Arc of Collaboration

When Figma first launched, its value proposition was primarily around making design collaborative. If design could happen in the browser, then designers could work together on the same projects. In fact, they could even work on a design at the same time.

In 2014 as I helped diligence Figma (*disclaimer: I worked on Greylock's investment in Figma, but I don't have a personal stake in Figma. sadly.*), I used to sit with designers at startups and watch them work. The top right corner of their screens were always a nonstop cycle of Dropbox notifications. Because design teams saved all their files in a shared folder like Dropbox, every time a coworker made a revision they would get a notification. And often there were complex naming conventions to make sure that people were using the right versions.

Figma solved this problem. Designs in Figma are not just stored in the cloud; they are edited in the cloud, too. This means that Figma users are always working on the same design. With Dropbox, this isn't true. The files may be stored in the cloud, but the editing happens locally—imagine the difference between sharing Word files in Dropbox vs. editing in Google Docs.

Any time a user edits a design, they are in effect checking out a temporary copy. This is why two users editing a file simultaneously creates issues. When designs are edited with Figma, there are no conflicts. And since revisions are a first party feature of each design in Figma, there's no need to have complex files with names like "profile_design_v23_final_draft2". Similarly, designers can comment directly in each other's files, instead of sending emails with feedback.

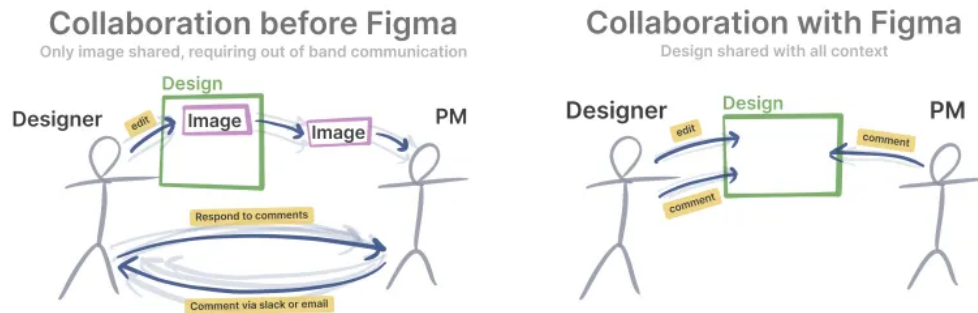


the one with less loops is better. I know, it's the first time I've ever said that

I used to be confused by the Figma's team consistent framing of Figma as browser-first. What was the distinction between this and cloud-first? Over time I've come to see how important this distinction has been. When many creative tools companies talk about the cloud, they seem to view it as an amorphous place that they store files. But the fundamental user experience of creating in their products is done via a standalone app on the desktop. Figma is browser-first, which was made possible (and more importantly performant) by their understanding and usage of new technologies like WebGL, Operational Transforms, and CRDTs.

From a user's perspective, there are no files and no syncing that needs to be done with others editing a design. The actual *experience* of designing in Figma is native to the internet. Even today, competitors often talk about cloud, but are torn over how *much* of the experience to port over to the internet. Hint: "all of it" is the correct answer that they all eventually will converge on.

Designers loved Figma and this drew initial distribution. And with features like team libraries, designers have incentive to pull in other designers on their team into Figma. But a tool designers love, while a prerequisite for success, does not fully capture the root of Figma's traction so far. While Figma has been building the ideal tool for designers, they've actually built something more important: a way for non-designers to be involved in the design process.



having to slide into the DMs is bad

Tools for Design, not just Designers

While Figma was building better tools for designers, their browser-first approach had a much more radical and significant impact on non-designers.

We often forget that the purpose of the tools we use at work isn't to increase our individual productivity, but the entire team's productivity. Companies themselves often forget this.

The best tool for individuals may *also* be the best for the entire team. But it's the latter that matters. And beyond some level of enhancing individual productivity, it's the team aspects that increasingly matter. Word processing is a good example of this. There used to be lots of experimentation and customization around individual features like typesetting. But once this was good enough, the focus has shifted towards collaboration. And for most use cases, few care about typesetting today. Increasingly our tools must understand and align with how we collaborate. This was less important when collaboration was logistically difficult and prohibitively costly, but as collaboration becomes easier its importance has risen. People's work is less siloed—and their tools must reflect this.

The best tools enable collaboration that was previously unthinkable.

Historically it has been very difficult for non-designers to be involved *during* the design process. If PMs, engineers, or even the CEO wanted to be involved, there were many logistical frictions. If they wanted the full designs, the designer would need to send them the current file. They'd then need to not only download it, but also make sure they had the right Adobe product or Sketch installed on their computer—costly tools that were hard to justify for those who didn't design regularly. And these tools were large, slow, specialized programs that were unwieldy for those not familiar with using them. It was hard to navigate a project without a designer to walk you through it. Comments were done out of band in separate emails. Even worse, if a designer made an update before viewers had finished looking at the file, the file would be out of date—without the viewer being aware.

The experience was just as bad for designers. Even if they wanted their PMs and engineers to give feedback, they'd need to handhold them through the process. Designers would have to export the designs into images, send the screenshot, and later figure out how to translate the feedback into changes in the actual design. The feedback loop was so slow that they'd need to pause their process while waiting for feedback.

Of course, what really happened is that most of the time non-designers just didn't engage as much with the design process. The pain of reviewing designs wasn't the primary problem, but there was enough friction that reviewing often never happened.

Figma fixed this.

Sharing designs with Figma is as easy as sending a link. Anyone can open it directly in their browser. It's as easy as going to a website, because...well...it literally is going to a website.

Once they have the link, non-designers will always have the latest design. They can comment directly in the designs, without disrupting the flow of designers. And with collaborative editing, they can talk through changes in a meeting and watch as they are implemented in real time.

These technical changes are kindling for a far more important social norm shift. Figma makes it possible for non-designers to be part of the process earlier and throughout it all.

Figma made companies realize that non-designers should and could be more involved in the design process and how crazy it is that other design tools aren't built with the experience of and interactions with non-designers in mind.

Tightening the Feedback Loop

Figma allows closer collaboration between designers and non-designers, but the second order impact of this on the social norms of teams is far more impactful. Historically, there has been so much friction in the design process that design is brought in *after* most decisions have been made. And conversely, there is a limited set of changes non-designers can push for once the design is set.

Tightening the feedback loop of collaboration allows for non-linear returns on the process. Design can be drafted *simultaneously* with the product, allowing feedback to flow in both directions throughout the process. Aligning the assets used by design and engineering allow more seamless handoffs, and allows for more lossless and iterative exchange.

it's Figma's all the way down

Designers can get feedback continuously from engineers or PMs on their team. Some will have sensitivities about non-designers sticking their nose too deep into the design process. In some cases they might be right, but this cost pales in comparison to the benefit to designers.

Bringing non-designers into the process is what gives designers a seat at the table of product and business decisions.

For too long we've systematically siloed functional areas like design, sales, and customer service. But modern companies are internalizing that if their core loops are truly an iterative process, then functions like design must themselves be part of the feedback loop of the company. Design decisions cannot be entirely abstracted from the rest of a business—they are as intertwined as the decisions made in product and engineering.

Historically, design has been opaque as a business unit due to the logistical and technical difficulties of making the design process legible to others. But as these hurdles are increasingly solved by companies like Figma, we're seeing teams navigate how to best integrate design with the rest of a company's processes. This should be no surprise, as we have seen the same arc play out in engineering over the last few decades.

Means of Ascent

Much of Figma's current success is driven by its ability to spread within companies. Figma becomes more useful as more people within a company use it, driving advantaged speed and scale of penetration within companies.

Figma was quick to recognize that the constraints on design at companies is often not a problem of pixels, but of people.

Many of Figma's competitors are great tools for designers. But that's who they are for—designers.

Figma is a tool for teams to design. Not for designers alone.

now you're all stuck inside the orange box with me

By bringing both designers and non-designers alike into Figma, they create a cross-side network effect. In a direct network effect, a homogenous group gets more value from a product as more of them join. In contrast, a cross-side network effect involves two (or more) distinct groups that grow in size and value as the other group does, too.

Figma's cross-side network effect between designers and non-designers is one of the primary and under-appreciated sources of their compounding success over the last few years.

As more designers use Figma, they pull in the non-designers they work with. Similarly, as these non-designers use Figma, they encourage the other designers they work with to use Figma. It's a virtuous circle and a powerful compounding loop.

Cross-side Network Effects

Cross-side network effects often get less attention than direct network effects—partially because our vocabulary around network effects is less robust than it should be, but primarily because they are typically viewed as being specific to marketplaces. While it's true that cross-side network effects are most commonly seen in marketplaces, it's wrong to think that they *only* exist in marketplaces. Supply and demand is the most famous of cross-side network effects, but not the sole source.

Figma's direct network effect among designers helped it grow early on, but has limits on its ability to help Figma spread. The set of designers that a designer works with directly is a limited set—and unlikely to change often. While they may spread it via word of mouth and social referral to many other designers, the network effect has an asymptote on distribution.

Figma's cross-side network effect offers an additional vector. Designers who use Figma share their designs with engineers and PMs, introducing them to Figma. As these non-designers learn to appreciate Figma, they then evangelize it to other teams of designers they work with on different projects. These cross-side network effects jump across teams and help Figma metastasize throughout entire organizations.

more loops is better. Phew, all is right with the world

This impacts monetization and purchasing at companies. Paying for a new design tool because it has new features for designers may not be a top priority. But if product managers, engineers, or even the CEO herself think it matters for the business as a whole—that has much higher priority and pricing leverage.

Product-Distribution Fit

It's this cross-side network effect that has most shaped Figma's growth. Like planting seeds, Figma can start small with a scattering of individuals using it at each company—but each of them can lead to serious expansion throughout their companies. Once Figma is adopted by a few designers, all their colleagues across departments get exposed to the product and especially its collaborative features and see how much better the experience is for them. They then encourage its usage and adoption across their other teams and projects and so on.

This is what gives Figma the compounding growth rate it has. While it took Figma some time to ramp up its growth, there is now a predictability and compounding that is impressive.

The core of Figma's product *is* the core of its distribution loop. That is incredibly rare and powerful. True alignment between product and core distribution loops is what's needed to catalyze outlier ability to compound in growth.

Like many of their peers in this generation of companies, Figma has developed a two-step sales motion: landing and expanding via a bottom-up, product-driven approach, then doing top-down sales once usage of the product has metastasized. Preparing for this entails a whole slew of enterprise features, building up the sales machine, and much more. Over the last few years Figma has begun to build these out, though there is still a lot of work for them to do here.

yo dawg, I heard you liked loops...

This new type of enterprise companies is a mix of both product and growth-driven consumer and sales-driven enterprise. We are still early when it comes to companies understanding how to re-architect their org structure, GTM motion, pricing model, and more to best fit this model. And like SaaS before it, this bottom-up to top-down model will mature from art to science over the coming years. Though Figma has significant work to do to make their traditional enterprise sales process robust, it would be disappointing if they stopped at simply recreating the traditional enterprise sales process. Practitioners are starting to understand that these sales loops can be understood in the same way growth loops have been and that there are ways to drive significantly advantaged sales velocity and scale via product and ops. Figma is one of the companies on the frontier of this, and there is an entirely different series of essays to be written about the maturation of this nascent field.*

Building an Ecosystem

Where does Figma go next?

There is always more to improve in the core product. But there comes a time when companies need to start thinking about the next sequence of their growth.

up and to the right!

The competitive landscape has also matured. Competitors are beginning to copy and encroach on Figma's core strengths. As they converge on Figma's views on the space, they pressure Figma to push forward. Sketch, for example, has shifted its pricing model to subscription and oriented around teams and has focused on moving more of the product into Sketch Cloud. In the last year, they've gone from being entirely bootstrapped to raising a venture round from Benchmark. This capital and support has fueled a more aggressive move in Figma's direction with a strong focus on collaboration, especially bringing Sketch to the browser and building team collaboration.

Adobe has made similar moves. Though powerful and widely used, Photoshop and Illustrator were not particularly specialized products for designers of digital products. In 2019, they launched Adobe XD as a direct competitor to Sketch and Figma.

Figma has been focused for the last few years on its value and spread *within* companies. Figma's next challenge is to improve its value and spread *across* companies in the ecosystem at large.

Global Network Effects

There are cross-side network effects that drive compounding value within companies using Figma. But how does it spread to new companies?

Given Figma's growth rate, it clearly does, but in less consciously compounding ways. Many people work across companies, especially agencies, and spread Figma to their clients. Similarly, when people leave their jobs and join new companies, they bring Figma with them. And of course, word of mouth helps spread it. These are all effective and amplified by Figma's intrinsic collaborative benefits, but they operate on their own natural cadence.

Thanks to Figma's hard work, the value in signing up is much higher for someone whose company already uses Figma. There are already teammates to collaborate with. There are assets and design libraries specific to their company. There are components that their team has built that they can reuse.

But for users who don't have co-workers using Figma, none of this value exists. They have the single-player benefits, but no direct increased value from the hard work Figma has done to add so many new

users over the years.

life is just an endless pursuit of trying to draw charts that go up and to the right

Figma's challenge is to create not just local network effects within companies, but global network effects that make Figma more useful to all users as it grows in scale. There are many directions Figma could expand its scope*, but our focus in this piece will be on the directions they have already begun to take.

say it together now: more loops is better!

In 2019, Figma began to plant the seeds for what their ecosystem loops across companies will be. With their launch of Figma Plugins last August and Communities more recently, they are starting to push the frontier of collaboration and productivity across companies.

this is my best graphic. It's all down and to the right from here

Empowering Creativity through Communities

Communities push their network orientation further by allowing users and companies to publicly share their designs.

Historically, when sharing designs on sites like Dribbble, it's often only the output image file that's shared. It's hard for others to see and use the full design themselves, including the layers and components that went into it. Even if the actual design is shared, the friction of opening it in whichever program and resolving any dependencies is non-trivial.

Sharing Figma designs removes this friction. Anyone can instantly open a design and start using it for their own projects. This enables users to frictionlessly become creators too, not just consumers.

Some designers have shared their UI kits, components, and design systems using Github. This has the right idea, but Github isn't meant for design. Forking a repo may be frictionless in engineering—but it is not for designs. The designs must still be downloaded and loaded into your app. For design, Github is more akin to a hosting site for downloading files. In many ways Figma's Communities are a reflection of Github's philosophy and intent, but built with design in mind. Duplicate a shared design, and a copy is instantly saved to your workspace and ready to be edited.

This frictionless process makes it easy for people to share and build upon each other's designs. But perhaps more importantly, including the entire design is a better abstraction layer that allows for new benefits.

context is king

Instead of just receiving the final output, recipients can see both the underlying components and assets within a design as well as more complex interactions and animations around designs. A great example is [this design](#) shared by Figma's design director, Noah Levin, of smart animations within Figma. By sharing the entire design with the community, others can not just see the animations—but play with and tweak the actual designs and animations directly. This makes it easier for them to learn how to use smart animations, or even just copy parts of Levin's demo into their designs.

The Promise of Plugins

I've been using Figma to create very niche memes to send friends. Figma is great, but as I've gotten more particular I've begun to add colored rectangles (with rounded corners because I'm not a barbarian) behind any text to make it more readable. This is a small but annoyingly rote series of tasks to do. Recently, I discovered the plugin [Substrate for Text](#) which simplifies this exact process. Now I can just select the text, activate the plugin and this text background gets auto-generated instantly. This plugin is made by a designer, [Andreslav Kozlov](#), who lives in Russia. On the other side of the world, Andreslav had felt the same problem I had, built a plugin to solve it, and shared it with the internet. And that improved the power of Figma for me, making it far easier for me to immediately spin up memes mid-conversation with friends.

though maybe he should also be blamed for the increased speed with which I can troll my friends now

this isn't how this meme is meant to be used either

This is a small example of the promise of plugins. Figma's plugins make it extensible so designers can augment their workflows, be empowered with new abilities otherwise impossible, and easily share them with others.

i hope you appreciate the precision of my squiggles in the productivity curve

Most Figma plugins today simplify tasks that would otherwise be very repetitive or painstaking. Today, companies can build private plugins to address their specific needs. For example, companies have built plugins that automatically generate dark mode designs, pull in external data and lint for common design errors, or make it easy to generate design assets in the right orientation. These plugins level up their entire teams.

The real power of plugins, however, is in making them publicly available across the ecosystem. Plugins are collective progress available to all users. Whether creating [charts](#), [custom maps](#), [pulling data into your designs](#), [redlining for engineering handoff](#), or [random blobs](#), plugins leverage up designers' productivity. And plugins like [Figma Chat](#) show that the frontier can be pushed out even further by enabling entirely new abilities for designers.

As companies scale, it becomes harder to sustain consistently increasing value to customers. With more customers, it becomes harder for one company to address all the unique use cases and needs. Companies must also increasingly service less ideal customers as they expand beyond their core audience. This is especially true for users who don't have all the compounding benefits of working with a team that all use Figma.

your local and global utility curves should look like synchronized divers. That's what everyone says right?

Plugins help Figma fight this drag. As they scale in users, more plugins will be created, making the product better for new users and spurring more designs to be created.

Architecting a Platform

Companies like Sketch have shown the importance of a robust plugin ecosystem (and of course, Adobe before them has a long track record of encouraging plugins). It's impossible for a single company to build all the features and tools needed by each user. Platforms are needed most when the diversity and scale of use cases is larger than can be built—or often even understood—by the company. Sketch's plugins allow the value to Sketch customers to grow at a faster pace than Sketch can ship new features. Figma will also likely stay focused on a generalized and core set of features, leaving a huge surface area of user workflows for plugins to address.

this graphic will probably get its own essay

Companies are still early on in understanding the nuances of becoming platforms. More and more companies are reaching the scale and dynamic range where this is a priority, but for the most part, they are individually recreating the wheel. In a decade, there will be clear frameworks, metrics, and supporting ecosystems for building platforms; today there are few. This is a natural maturation, and one that all business models go through, such as SaaS and subscription over the last decade.

they grow up so fast.

Because of our lack of shared vocabulary around building platforms, it's hard for many to understand the subtle but important delineations between the approaches of different companies. It's easy to assume, for example, that all plugin systems are built the same, but this impulse is often incorrect.

Sketch: a Case Study

Take Sketch's plugin ecosystem. The API is well documented and the plugin coverage is high. However, plugins are outside the scope of the core product. Users installing a plugin are most commonly directed to the Github page of the plugin, which they must manually download and install. This is another example of the gradients of what it means to put your product in the cloud. Even if Sketch is in the cloud, its plugins are local files. There is friction in downloading and installing them. This is compounded in a work setting as teams must manually make sure employees are using the same plugins if needed.

Because plugins are not first-class citizens in the Sketch product and handled out of its scope, plugin management is very decentralized. Plugins can register to be listed on Sketch's site and enable automatic updates, but otherwise Sketch is very hands-off. There is no official Sketch source for how popular a plugin is, nor do they have to approve plugins. Instead, users must rely on Github stars or reviews on third party sites. There is no oversight on plugins causing performance or stability issues.

This shouldn't be mistaken for criticism. The weaknesses in Sketch's plugin architecture show only because of how successful they have been at encouraging a robust community of plugins around Sketch.

Platforms are emergent ecosystems, more akin to building a consumer social network than a traditional enterprise sales company. This is one of the core dilemmas of building platforms. They are complex organic systems that have to be carefully cultivated (gardening vs. engineering mindset) and it's hard to predict beforehand what direction they will go and the scale they will achieve.

Sketch cannot be faulted. To this day it's still unclear how ambitious plugins can and should be. By defaulting to a relative hands-off approach, they allowed the community to flourish unhindered by them. And it's by seeing Sketch's success with plugins, as well as their struggles, that others like Figma and

Adobe XD have been able to have increased confidence in the importance, potential, and levers of a plugin ecosystem.

But while Sketch cannot be faulted, their choices have stunted the full potential of plugins built around them. Sketch understands this. They are rearchitecting their plugin system right now to be fully in the cloud, which is a necessary step in the right direction.

Complex systems do not absolve companies of their need to make clear choices about the architecture, policy, and norms of their platforms. If anything, they magnify its importance. The structural choices made by companies ripple forward as the ecosystem emerges around them. And the direction and scale of the resulting platform are a function of the physics set by their initial conditions. We shape our abstractions, and thereafter they shape us.

Figma: Forming Foundations

Figma's plugins are very early, but promising. Natively built in and browser-first, when you click to install a plugin, it's available instantly. There's nothing to install after all, just access privileges to activate. That's magical.

stick figures included for scale

Like all good magic, the work to make this feel effortless is quite arduous. Figma's plugins must be secure, performant, and stable, especially since they are creating a plugin ecosystem for a browser-first system. Users should be able to trust that using plugins won't expose them to security risk or hurt Figma's performance. And both developers and users should feel confident that the APIs the plugins depend on won't be suddenly deprecated or broken. Without these preconditions, plugins will always be at best a small aspect of Figma. This trust and stability is the bedrock of a strong ecosystem.

i think the takeaway is that plugins are like rainbow cake?

This difficulty is best seen in Figma's engineering blog posts on building their plugin system. Within a month of [their post](#) on how they decided the architecture for plugins, they [learned of vulnerabilities](#) in their approach and had to switch it out.

Ensuring the platform can be trusted is not just a matter of technical architecture. Figma doesn't just host plugins, they also have a centralized [approval process](#), more similar to Apple's app store than to Sketch's approach. Plugins that want to be listed must pass Figma's policies around safety, business, usability, and legal.

The Path to Platforms

The business policies are worth noting in particular. While safety, usability, and legal are about maintaining the integrity and trust of the platform, Figma's business policies are about shaping what they think the plugin ecosystem *should* look like. For example, they allow monetization but prefer plugins be accessible to all users. Choices like how much to encourage an ecosystem of plugin businesses vs. a more open source community are important ones. There is no obviously correct answer, and it can and often needs to change over the life of a platform. One need only look at the makeup of Uber drivers, WordPress plugins, or Airbnb hosts over time to see this.

this will also be its own essay haha

Building platforms requires many hard choices like these. How should you balance encouraging growth today and building towards the ideal long term vision? To what degree should platforms influence which plugins get built or even which they should build themselves in the early days? How should you decide which features actually should be part of the platform itself vs. standalone plugins? How wide is the scope for what plugins can build? These are only a few of the core questions.

Perhaps Figma's most interesting choice is the heavy focus on ease of plugin creation. Figma's plugin system is explicitly designed to turn designers into developers by enabling them to create plugins for their own workflows. In most platforms and marketplaces, the ecosystem tends to split and professionalize over time. This is the natural gravitational pull of ecosystems*. To bet that a significant amount of the platform's leverage will come from individuals improving their own workflows is a bold one. It's a bet on induced demand, which is always the most interesting type of bet.

Figma's plugin ecosystem is very nascent. From their [list of supported features](#) and what's to come, it's clear that there is still a long way to go to open up their platform to more advanced plugins. And choosing the right abstraction layers for the plugin ecosystem is crucial. So far they have adopted a very strong stance on what the core technical decisions and promises around safety, performance, and stability need to be, but have been very hands-off on what plugins are built. This is to be expected early on. When it's unclear what should be built, it's not a bad policy to see where the creativity of the community leads. It often brings really amazing stuff, like this [demo](#) of a plugin. Over time, however, we should expect to see more focus. Right now their [plugin page](#) is unsorted other than popularity and some that are featured. This works when the number of plugins is low enough, but eventually they will have to decide how to categorize plugins and what they want discovery to look like.

As plugin categories begin to crystallize, Figma will need to form crisp views on which areas should be absorbed into their core product, what they want each category to look like as it matures, what essential plugins don't yet exist that they must help catalyze, and what new APIs they should allow plugins to address. The choice of how much to encourage monetization of the plugin ecosystem, as discussed above, is a perfect example of the kind of key decisions Figma will need to make (repeatedly) as they build out their plugin platform. Perhaps most importantly, Figma must decide the meta-framework by which to make these decisions intentionally rather than capriciously.

Pushing Progress

Has design been improving? Are we getting better at designing, not as an art, but as a functional practice?

The answer is certainly yes. We have tools that would have been unfathomable a decade ago, much less pre-computer. It is easier to design. It is easier to begin designing. Design is more scalable.

But how does design's rate of improvement compare to the rate at which engineering has been improving as a process? Design as a meta-process is less impressive by this benchmark.

Engineering is almost unparalleled in the rate at which it commoditizes itself and pushes the frontier of progress out. The best practices in frameworks, languages, and infrastructure are always rapidly—and sometimes tumultuously—evolving. What used to take entire teams to build before, requires fewer and fewer people every year.

this is the true up and to the right curve. and the one that needs to be figured out for every discipline

As disciplines evolve, they figure out the social norms needed to operate better, build tools that can be shared across the industry, and invent abstractions that allow offloading more and more of the workload. They learn how to collaborate better, not just with each other but with all the other functions as well. Disciplines are not an end to themselves; the degree to which they contribute to the larger organizations and ecosystems they are part of is the final measure of their progress.

Design appears to be inflecting in the direction of engineering. Figma is in pole position to drive this evolution. As a tool, it makes designers both more efficient and more collaborative by breaking down the walls between design and the other teams they work with.

the dream

But Figma's true potential is if it can make the transition to becoming a platform. If Figma can, they'll push progress in design as a *discipline*.

Which companies are successful in a field is decided by many factors, not the least of which is a good measure of luck. But when disciplines undergo tectonic shifts, the companies that thrive have outsized influence. The choices they make in abstraction layers, social norms, architecture, and more have large ripples for a generation. This is even more true for platforms, whose loops become core to their ecosystems. Like wet clay, the choices they make eventually set and become the underlying substrate that defines how the entire ecosystem grows. That is both a tremendous opportunity and responsibility for the companies, like Figma, that take on this mantle.

Acknowledgements

Many thanks to [Keila Fong](#), for the many discussions about this topic and help with this piece. As well as unceasing pressure to publish it.

Additionally, thanks to [Max Bulger](#), [Michael Dempsey](#), [Kane Hsieh](#), [Boris Jabes](#), [Dave Petersen](#), [Ryan Petersen](#), [Kevin Simler](#), and [Eugene Wei](#) for their discussions, edits, and help with this piece. [John Lilly](#), who led Greylock's investment in Figma, deserves most of the credit for seeding all my views on productivity and collaboration. His investment memo from 2014 is still prescient in ways that took me years to fully appreciate.

Thanks also to [Casey Winters](#) and [Brian Balfour](#). Building the [Advanced Growth Strategy](#) course was the origin of many conversations about Figma's loops. And I still teach the Figma case study every semester.

All graphics in this piece were created with [Procreate](#) and [Figma](#). Procreate is a fantastic drawing app for iPad. If you have made it all the way through this essay and don't know what Figma is then I don't know what to tell you. An integration between these two might have a target audience of only me. But I would love it.

* Further pieces to be written on these subjects