tribecap.co /units-of-time-are-the-new-currency/

Units of Time are the New Currency - Tribe Capital

24-30 minutes

> *"The key to investing…..is determining the competitive advantage of any given company, and, above all, the durability of that advantage."*
>
> *Warren Buffett*

Today we're seeing that a moat—a barrier that protects a company from low-cost competitors or new, disruptive technology—isn't enough to build a lasting business.

A moat simply **buys a company time** to figure out the next great business. Just like the invention of field artillery in the 16th century rendered moats obsolete, technology today is grinding down barriers to competition.

Technology makes the moat of traditional business a mere jumping off point. For tech companies, it's always about what's next — not simply protecting what's here today.

For Buffett, operating as the low-cost producer in a crowded market helps a company build a moat around price. For tech companies, low prices trend towards free as a means to kickstart adoption.

GEICO competes with other insurance providers by selling directly to consumers, rather than relying on brokers.

NETFLIX

Netflix scaled licensed content to a massive user base through online streaming. That bought the company time to invest in a library of original and differentiated content.

For Buffett, creating economies of scale that drive lower per-unit costs protects a company from the competition. For tech companies, building one economy of scale allows you to build the next one.

COSTCO WHOLESALE

Costco delivers customers low prices at healthy margins through a chain of warehouses stocked with bulk goods.

amazon.com

Amazon created a low-margin, high-volume e-commerce business. That bought the company time to invest in an efficient warehouse, supply-chain, and delivery infrastructure, enabling the company to increase margins where appropriate.

For Buffett, “buying a commodity and selling a brand” creates a moat around mindshare. For tech companies, a great product and well-known brand need to be defended by an ecosystem of other products.

See’s Candies perfected the in-store retail experience to build an iconic brand and sell chocolates at high-returns.

Facebook created a social network that allowed everyone in the world to connect online. That bought the company time to win the messaging wars on mobile with Messenger, WhatsApp and Instagram.

Buffett’s not wrong, but technology has changed the nature of competition. While businesses were once considered only as valuable as the dividends they paid out, the “impenetrable” moats that let companies spit off excess cash are dwindling. A moat today is simply a temporary buffer that helps a company get ahead of the next innovation cycle.

The only way to build a lasting technology business

Moats protect you from competition, but they also cut you off from the outside world. To create a defensible business today, your product needs to be a **utility**. You have to build something that solves a user pain, and then scale until it’s so fundamental that it becomes a feature of other products.

Even if you have a hit app with hundreds of millions of users, you’re on a treadmill to produce content, add features and acquire new users. As Snap co-founder and CEO Evan Spiegel wrote in Snapchat’s S-1 filing:

> In a world where anyone can distribute products instantly and provide them for free, the best way to compete is by innovating to create the most engaging products. That’s because it’s difficult to use distribution or cost as a competitive advantage — new software is available to users immediately, and for free.

When you’re forced to innovate just to maintain your base level of engagement, you have no leverage on your time.

A utility gives you that leverage by spreading your product across other platforms and products. It’s so embedded in the world that most people don’t think about it — like energy. The social graph (Facebook), search (Google) and web services (Amazon) are utilities that power the consumer-connected world today.

I’ve spent a career building consumer products and now invest in early-stage companies. I’ve seen firsthand how companies reach the utility stage by going through three distinct stages. Utilities aren’t permanent, but they take a long time to grow obsolete. As the world moves faster, being a utility buys you time between now and the next big shift. That’s time you can reinvest forward.

How to get non-linear returns on time

When you compound time, you’re creating and recreating value faster than the current innovation cycle. That means you’ve leveraged early user growth and product innovation by reinvesting in the underlying utility.

Twitter co-founder and Medium CEO Ev Williams shares the first step for compounding time:

> Here’s the formula if you want to build a billion-dollar internet company. Take a human desire, preferably one that has been around for a really long time….Identify that desire and use modern technology to take out steps.

Saving time for other people allows you to compound time for yourself. Every minute that you save others is time that they spend creating additional value by using your utility, or building on top of it.

If you're building a company today, start by **assuming that you'll be successful**. What does success give you the right to do next? What's next is how you reinvest the time you've saved for nonlinear gains. Success is important, but reinvesting that success to get compounding returns is where many promising companies plateau.

This is the formula for compounding time into a utility and beyond:

1. Reduce friction for your customers and yourself. Use the time you save to build your utility.
2. Compound time by investing in the ecosystem and getting other companies to integrate with your product. Other companies will integrate with you to save themselves time, building on top of your platform and giving you time to invest in the next great business.
3. Buy other people's time to defend your utility and stay relevant. Smartly acquiring new products helps you maintain your utility.

This roadmap helps you determine what to focus on and how to invest your time in each step.

1. Simplify your product to build a utility

To build a utility, start by simplifying your product to reduce friction around an existing behavior. If you're able to do that better than existing products in the market, you give yourself a head start.

Reduce the amount of time it takes to get from point A to B, and that's time you can reinvest into building a utility. If you can transfer the time you save internally to other companies, pretty soon you'll start to see your product as a feature of other products.

Getting a head start doesn't guarantee success. But the wider your head start, the more breathing room you have to plan and execute on the next innovation cycle.

Amazon Web Services

The success of Amazon Web Services, Amazon's $12B run-rate business, came from leveraging a seven-year head start into compounding returns. In an interview with Charlie Rose, Jeff Bezos said:

> It appears to me just empirically that if you invent a new way of doing something, typically if you are lucky, you get about two years of runway before competitors copy your idea. And two years is actually a pretty long time in a fast-moving industry so that's a big head start. For whatever reason…. **Amazon Web Services got seven years of runway before we faced like-minded competition.**

Those seven years helped Amazon build AWS into a $160B business that accounts for 38% of the company's $480B market cap.

Back in 2002, hundreds of developers at Amazon worked on different parts of Amazon's monolithic code base at the same time. Changes took weeks to coordinate and deploy. To reduce this friction internally, Amazon transitioned to a services-oriented architecture. Jeff Bezos had sent a company-wide memo to this effect:

> All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

While this created more work in the short-term, it broke Amazon down into hundreds of micro-services that communicated via APIs. By making all services accessible via API, Amazon drastically reduced the time it took to deploy new features and functionality.

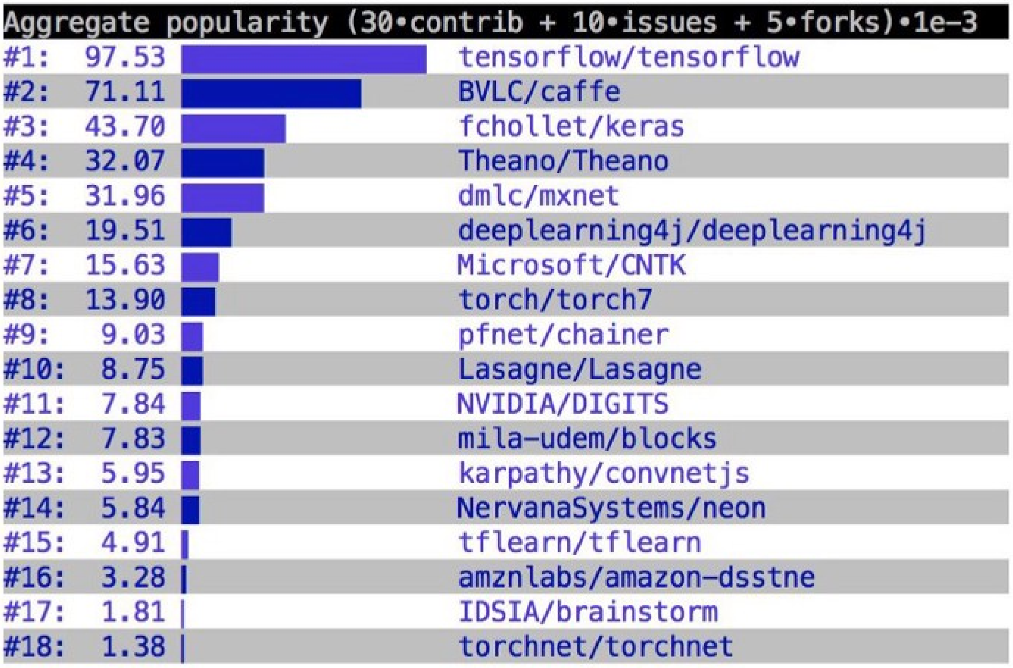
Contrary to what most people think, AWS didn'tsell excess server capacity externally to customers. Instead, AWS was saving people time. Amazon had done the hard work of operationalizing its infrastructure. By abstracting the complexity of managing these data centers, it was able to offer this as a service to external customers.

Scaling internally gave Amazon the right to keep growing what would become AWS.

Google's TensorFlow

In 2015, Google open-sourced its machine-learning framework TensorFlow. Open-sourcing some of the hardest parts of Google's machine-learning push and giving it away for free is part of the company's strategy to become *the* utility in machine learning and getting everyone else to integrate with it.

Since then, TensorFlow has grown rapidly in popularity:



```
Aggregate popularity (30•contrib + 10•issues + 5•forks)•1e-3
#1:   97.53   tensorflow/tensorflow
#2:   71.11   BVLC/caffe
#3:   43.70   fchollet/keras
#4:   32.07   Theano/Theano
#5:   31.96   dmlc/mxnet
#6:   19.51   deeplearning4j/deeplearning4j
#7:   15.63   Microsoft/CNTK
#8:   13.90   torch/torch7
#9:    9.03   pfnet/chainer
#10:   8.75   Lasagne/Lasagne
#11:   7.84   NVIDIA/DIGITS
#12:   7.83   mila-udem/blocks
#13:   5.95   karpathy/convnetjs
#14:   5.84   NervanaSystems/neon
#15:   4.91   tflearn/tflearn
#16:   3.28   amznlabs/amazon-dsstne
#17:   1.81   IDSIA/brainstorm
#18:   1.38   torchnet/torchnet
```

Deep Learning Frameworks by popularity. Google's TensorFlow is more popular than Theano, Amazon's MXnet, and Facebook's Torch.

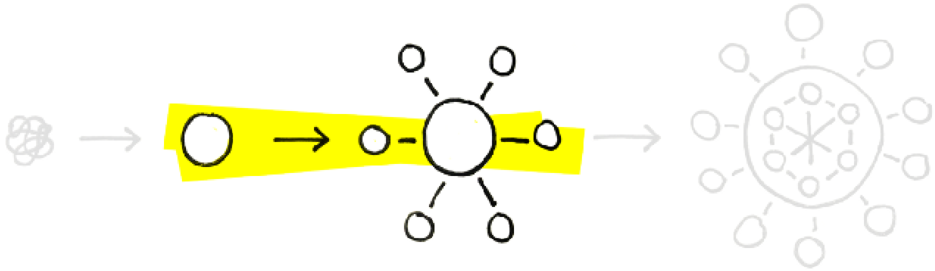
Google's machine-learning algorithms are reportedly five to seven years ahead of the competition. By keeping TensorFlow to itself, Google would have maintained its lead time — similar to how moats are created by stockpiling assets. But by taking the opposite approach and giving TensorFlow away for free, Google created a utility.

TensorFlow reduces friction for developers building machine-learning applications. In exchange, they build on top of Google's toolkit and utility. Google gets the next generation of developers building applications around TensorFlow, helping to improve it, along with a customer base they can iterate around.

At this year's I/O keynote, Google announced TPUs, or Tensor Processing Units — proprietary chips that Google has created to make it faster and cheaper to run machine-learning algorithms on the cloud. Via TPUs, people already building applications on top of TensorFlow can run them faster on Google's cloud. They've also released API access to image recognition speech processing products like Vision and Cloud Natural Language, which are also built on top of TensorFlow.

Seven years later, this is how Google is trying to compete with Amazon's strategy. Open-sourcing TensorFlow means developers are more likely to integrate TensorFlow into their products, which in turn means that they're more likely to move them onto Google Cloud. The kicker is that improvements developers make to TensorFlow translate directly to improvements in Google's suite of machine-learning products, from cloud-based APIs to consumer applications, like Google Assistant, as well as its new line of hardware products.

2. Integrate with other products to grow your utility



A head start buys you time to build a platform. To grow a platform, you need to cooperate with and invest in other companies in the ecosystem.

This runs contrary to the common wisdom that tech companies operate in winner-take-all markets and need to create monopolies to survive. Peter Thiel writes about this in his book *Zero to One*:

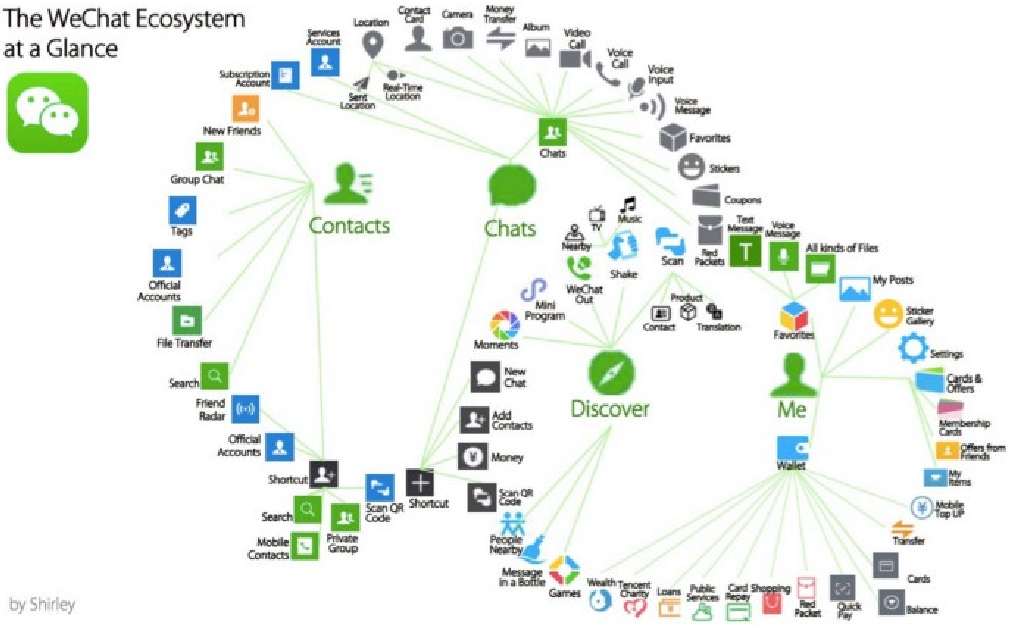
> In perfect competition, a business is so focused on today's margins that it can't possibly plan for a long-term future. Only one thing can allow a business to transcend the daily brute struggle for survival: monopoly profits.

According to Thiel, monopoly profits provide companies with a moat and the luxury of long-term utility. This philosophy is flawed because your ultimate goal isn't to create a monopoly. The goal is to create a utility. This can — *but doesn't always* — result in monopolistic profits.

By investing in the ecosystem and integrating with other products, you can save time by building new functionality from scratch. Because you have a head start and you're continuing to compound time, you receive asymmetric information that you can apply to your advantage. Because you've invested in other companies, you grow as they grow.

WeChat Pay

Over the past three years, WeChat's mobile payments has grown from nothing to 40% of the market because the company has co-operated with and invested in other products in the ecosystem. By integrating with these products, WeChat provides users with increased functionality without having to build it.



WeChat's novel red envelope feature helped trigger viral growth for its mobile payments product. WeChat Pay spread virally by making it easy to send and receive money within close-knit groups of family and friends.

What cemented WeChat Pay into a utility was a series of strategic partnerships with and investments in other companies:

- JD.com: Tencent owns a 20% stake in JD.com, one of the largest e-commerce sites in China and Alibaba's main competitor. 52% of first-time visitors to JD.com come from a Tencent-owned platform. While WeChat has access to user demographics and which public accounts users follow, JD.com has transaction data on what people can actually buy.
- Didi: Tencent invested $700M in Didi Chuxing in 2014, and another $2B one year later. By partnering with Didi, China's primary ride-sharing service, WeChat embedded Didi's services directly into its apps. It allowed WeChat users the utility of ordering rides — while remaining in WeChat.

Rather than trying to build a vertical stack of payments, e-commerce, and ride-sharing, WeChat has integrated with other companies that perform each function. WeChat can continue to grow its messaging utility, while offering users the ability to shop, call a car, and find someone to do their laundry all at the same time.

Not only does WeChat benefit by extending the reach of its mobile payments, but because it is invested in these other services, it actually benefits financially when they grow. WeChat for example, blocked Uber from its app, helping Didi. Meanwhile, it shares data with e-commerce businesses, like JD.com, to increase transactions.

By bundling together external services behind a single interface, WeChat controls the flow of data and usage, and benefits disproportionately from usage on the network.

Ethereum and ERC-20

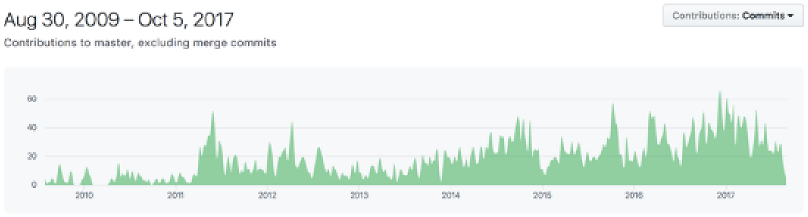
Cryptocurrencies like Bitcoin and Ethereum are bootstrapping the formation of new developer ecosystems through shared financial incentives on the network. With cryptocurrencies, developers are rewarded by the value of the token. The more miners, investors, users, and developers there are on the network, the higher the value of the underlying token. We're seeing exponential growth in crypto networks because they're able to compound time and send it back to participants in the network.

While the internet is built on top of open-source projects like Unix, TC/IP and Wikipedia, the majority of value created from the internet is captured by a handful of companies like Google and Facebook on the application layer.

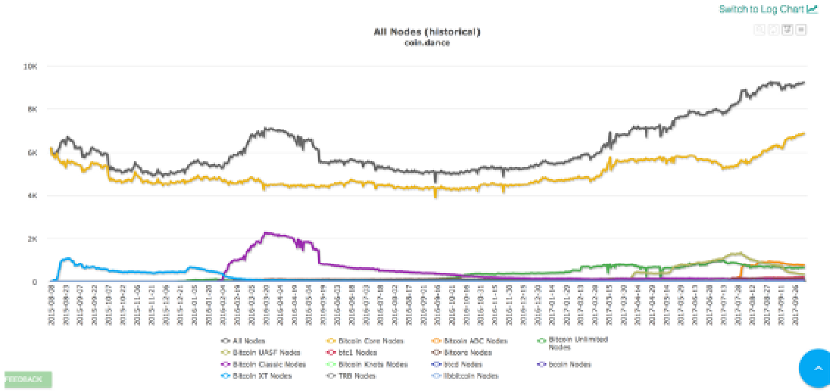
Developers contributed their time to open-source projects and were paid back in social currency. By contributing to the ecosystem, developers built up name recognition and community goodwill. They received job offers based on contributions — often from companies like Google and Facebook. With projects like Bitcoin and Ethereum, developers reap financial rewards as the underlying platform grows.

With ERC20, for example, Ethereum created a shared set of standards that make it really easy for developers to launch new projects on top of Ethereum. ERC20 tokens are interoperable with Ethereum and can be exchanged for Ether. That's one of the reasons why many recent initial coin offerings have been launched on the Ethereum blockchain. Creating a shared set of standards makes it easier to create, fund, and integrate new projects. These projects can integrate with each other without direct interaction on top of Ethereum. Each new project multiplies the existing value of the system.

Commits to Bitcoin Core over time



Growth in Bitcoin nodes over time



We’re still in the testing phase of cryptocurrencies and it’s unclear which today will remain tomorrow. While Ethereum itself has created a lot of value and utility in the ecosystem, it needs to figure out how to drive more contribution to the underlying protocol — rather than just act as a stepping stone for new blockchains. The protocols that win will be the ones that incentivize higher collaboration between all participants on the network.

Swift and React.js

Companies like Apple and Facebook build internal tools and development frameworks to buy time by increasing developer productivity. Open-sourcing developer tools and languages allow these companies to expand beyond existing platforms and benefit from the broader community of developers.

Back in 2014, Apple open-sourced the Swift programming language, an alternative to its Objective-C language. While Objective-C was hard to learn, required a Mac, and could only be used on platforms developed by Apple, Swift is open-source and platform-independent. For Apple, Swift is a way of future-proofing its macOS and iOS platforms. As Craig Federighi, SVP of Software Engineering at Apple, remarked, “we saw open sourcing as a critical element to make Swift reach its potential to be *the* language*,* the major language for the next 20 years of programming in our industry.”

Because Apple has already bought itself time investing in a healthy — and highly profitable — developer ecosystem, Apple guaranteed high adoptionfor its new language. Swift gave developers an easy and fast language that they loved. Because Swift is open and platform independent, Apple gets an even larger number of developers learning the new language, building applications on top of it, and improving it.

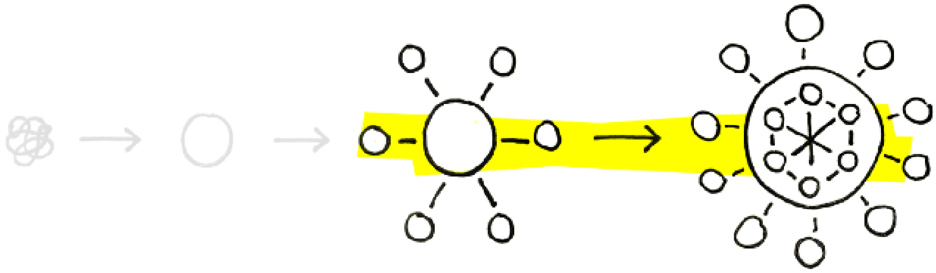
Similarly, Facebook open-sourced React.js in 2013, a Javascript library that powered Facebook’s News Feed as well as Instagram’s web app. Two years later, Facebook followed-up by open-sourcing React Native, a framework designed to make cross-platform development across Android and iOS easier. Within a year of open-sourcing React Native, 30% of commits were made by contributors outside of Facebook. By 2016 over 50% came from external contributors. Today, React is one of the dominant javascript libraries, used from giant companies like Walmart and Netflix to small startups.

Unlike Google’s Android or Apple’s iOS, Facebook doesn’t own a mobile operating system. Open-sourcing React Native allows Facebook to help developers build cross-platform mobile apps more easily, but it also helps them buy time and expand. Facebook might not own a hardware platform, but they own the dominant platform for user interfaces that engineers are building around today.

Where Javascript is an incredibly messy language, React Native provides developers with a framework for developing UI in a declarative way. It saves developers time from building their own internal tools for web interfaces. In return, developers pay Facebook back in terms of bug testing, pull requests — and building their applications in the image of Facebook. Controlling the roadmap of React.Js means that Facebook is literally shaping the next-generation of applications.

Today, it's paradoxically *harder* to replicate something that's open source even though anyone can clone the underlying source code. That's because of the developer community, the bug testing, and adoption that have been built around the project.

3. Buy other companies to protect your utility



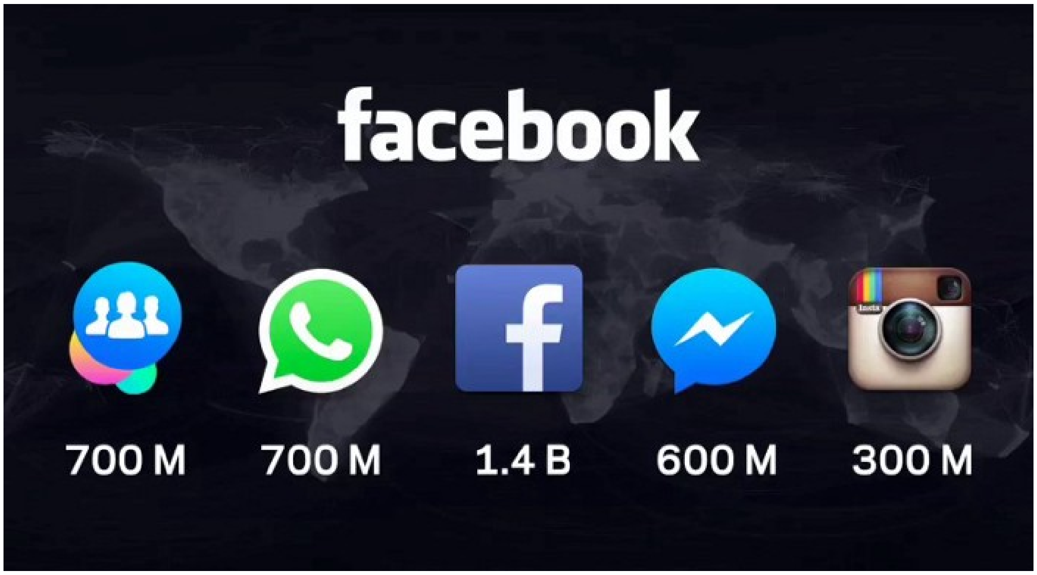
As your platform grows into a utility, the way you stay a utility is by protecting what you've built by acquiring new companies and their products.

It's hard to innovate on a single, successful platform. The success of a single flagship product and cash cow can blind you to more nimble competitors. You can acquire up-and-coming products, but integrating them into your core is often inefficient and costs time.

The "family of brands" strategy is a way that companies shore up their core while looking to the future. By buying new companies, you defend your utility and your moat of time by filling in gaps within your platform.

Facebook and Instagram

Last year, Facebook's CFO announced on an earnings call that the company was hitting its cap on ad load — the number of ads shown to users in the News Feed. Most of Facebook's recent growth has been driven by higher ad prices, which it will also eventually max out. By aggressively pursuing a family of brands strategy, Facebook can hamstring up-and-comers like Snapchat — and leverage these products to grow the platform.



Around the same time Facebook began its mobile advertising business, the company launched Messenger as a standalone app and also acquired Instagram. Two years later it bought WhatsApp.

I've written about this strategy before: Instead of one platform under perpetual attack, Facebook has five brands, each one still innovating. Facebook can work behind the scenes to test new innovations on specific products, aggregate data across its family of products, and apply those learnings across each product. Meanwhile, it doesn't have to waste time trying to fully integrate acquired companies into the broader organization.

Take Instagram's latest "Stories" feature. Instagram's Stories feed uses user data from Facebook to determine the order in which people see stories. Instagram also previously adapted its main feed around Facebook's algorithm. Today, Instagram and WhatsApp stories each have over 250 million active users.

As my friend Hiten Shah writes:

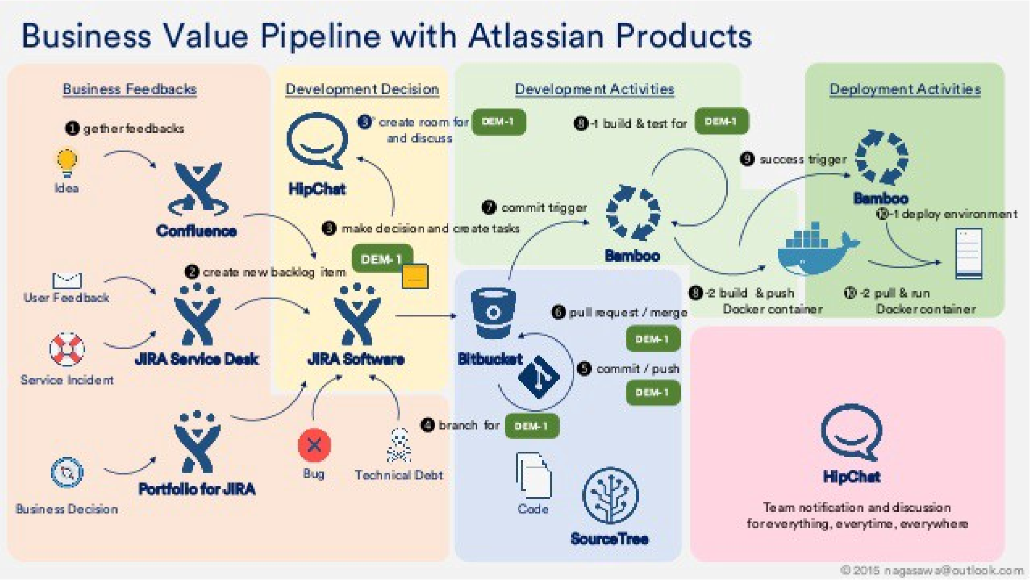
> You shouldn't be afraid to copy if you need to. For any company that lasts long enough, they'll face a period that they have to copy in order to get back in the game.

Today, Facebook is busy monetizing that user growth by launching ads across its family of products. Facebook COO Sheryl Sandberg says:

> We do think that the advertiser base and the targeting and the measurement we offer once we figure out the format will be a very considerable advantage….one of the reasons we were able to scale into Instagram ads more quickly is because we were building off of the Facebook advertiser base. And similarly, the work we've done in Facebook and Instagram and Audience Network will help us expand to Messenger.

Facebook could have easily fallen victim to the success of its flagship product. Instead, it focused on building a family of brands to protect and expand its utility.

Atlassian's Suite of Products



Atlassian's Family of Brands

Today, free consumer products like Facebook typically emerge with a family of brands as the initial core product expands to hundreds of millions of users. In the B2B space it's extremely rare, if not impossible, for products to achieve this type of viral growth. By investing in a family of brands, Atlassian has reinvested the success of its early developer tools into a platform for workplace collaboration.

Atlassian launched its flagship product Jira in 2002, an issue-tracking tool for software developers. It followed up two years later with Confluence, a wiki tool for team collaboration. By growing around a cash efficient and self-service sales model, Atlassian has been able to invest in building out a family of products. They began with developers, expanded to teams in IT, and finally, to all the teams in a company.

For its first seven years, Atlassian grew incrementally with Jira and Confluence. Then, customer growth really took off as the acquisition of new products like Bitbucket and HipChat expanded the company's core customer base.

Atlassian operates in a crowded, fragmented market. Products like BitBucket (code repository) and HipChat (team chat) face competition from smaller, nimbler companies like Slack and GitHub. Meanwhile, Atlassian is also under attack from enterprise competitors upmarket like Microsoft and IBM.

In an interview, CEO Scott Farquhar highlighted Atlassian's family of brands as the company's competitive edge:

> The way we think about that market is that chat is one leg of the stool. There are multiple products that need to be used in order for collaboration to work. We've never seen chat as the be-all and end-all, it's one of a suite of products.

Continuing to invest in its family of brands allows Atlassian to do multiple things at the same time. 1). Continue to build out core offerings like Jira and Confluence with features for the enterprise. 2). Acquire new products like Trello to leverage word-of-mouth and build brand. 3). Earn time and capital to expand their utility.

For enterprise and IT software, the family of brands approach helps companies expand by attacking new verticals, while upselling existing customers more deeply into their suite of products.

Looking forward

Founders hoping to build the next Google or Facebook, are acutely aware that what they are building today will look very different in just a few years. A bookseller becomes the everything store, a search engine builds autonomous vehicles.

Building a traditional moat will be antithetical to building a great business. The only way to survive is to extract the core of your business and spread it out to compound returns on time. First, you have to save time for your customers and even yourself. Then, you have to invest it forward by co-operating with other products in your ecosystem. Finally, you have to acquire new innovation to maintain your lead.

With the rise of large platform companies like Amazon, Apple, Netflix, and Google, the future often appears less competitive than in eras past. The tech giants of today possess a combined market cap 10x greater than the Wintel duopoly of the PC era. Each of the internet giants has survived by being relentlessly paranoid about survival. They aggressively overturn their own business models by investing in new platforms from machine-learning to the blockchain. By managing chaos within existing systems, these companies establish new beachheads and new systems in a generative process.

Although these companies seem unassailable today, remember that nothing lasts forever.

A company might begin with a single product in a new and exciting space like the blockchain or autonomous transport. Pretty soon, they have to extend the utility of the original product to competitive products and platforms. Finally, they have to build or buy newer companies that threaten their utility and start the cycle again.

Thanks to Ana Braskamp and Ian Mendiola.