Bessemer Venture Partners

Harder, better, faster, stronger: A look back on our developer laws

Six years after launching our eight developer laws, we add our 2019 amendments and examine how B2D companies continue to run the world.

MAY 22, 2019

CO-AUTHORS
Ethan Kurzweil
Jenny Gao
Sakib Dadi

When we first penned the eight laws of developer platforms back in 2013, we didn't chisel them in stone. We always knew they were a living document. Five years and a dozen investments later, in companies like LaunchDarkly, HashiCorp, and PagerDuty, we recognize these laws as not just investing criteria that led us to some of the most successful developer-centric businesses, but also as practical case studies for developer-focused entrepreneurs who want to do the same.

Founders today face a completely different reality than founders had to navigate six years ago, and these three trends have impacted the way entrepreneurs build companies in today's era of enterprise IT:

1. **It's harder to cut through the noise:** With more than 22 million developers world-wide, the number of tools, platforms, and open source projects vying for developer mindshare and love has grown exponentially since the dawn of the iPhone. Cloud service providers, namely AWS, have gone from elephants in the room to wooly mammoths, making business-to-developer (B2D) markets far more competitive than ever before.
2. **We've reached a tipping point for best-of-breed tooling and disciplines:** In order to compete with these broad do-everything-for-you cloud vendors, it's more important than ever for startups to do one thing exceptionally well. There are some notable exceptions (e.g. HashiCorp), but we're seeing many breakout companies focus on one highly acute pain point. This way, they can go deep in a way that the cloud providers cannot.
3. **Containers and serverless technology are enabling a shift to microservices and distributed architectures:** This change in mentality is often dubbed "digital transformation" within enterprises and frequently meant a reorganization of teams, budgets, and responsibilities. Today, many Fortune 1000 companies are still in the transition period of these new technologies and we're seeing a greater willingness to purchase from outside vendors than build homegrown solutions. For well-positioned startups, this is great news as enterprises don't want to make a Hobson's choice to either waste precious internal resources keeping up to date or accumulate technical debt by not keeping pace.

After more than six years of seismic shifts that impact developers and the delivery of digital products and services, we recently stepped back and evaluated how our eight laws have aged over time.

Today's emerging startups and newly public companies are living testimony of how the laws have evolved into harder, better, faster, or stronger embodiments of their originals.

The developer economy has matured and so we're eager to refine our developer roadmap, especially as entrepreneurs pursue developers as their primary customers.

Here are the 2019 amendments to our developer laws:

1. Developer platforms based on granular and metered units of measure have more pricing power.

2. Well-designed developer platforms have natural upsell triggers inherent to the product.
(Original Law #1 & 2: Deliver a service that can be metered & grow as your customer grows)
Why pay for something you're not using? Delivering a service that can be metered and grows in tandem with the growth of your underlying customers is the gold standard of pricing strategies. And thanks to cloud service providers, companies now expect to only have to pay for what they compute, down to the nanosecond.

It's no coincidence that the developer-focused tools that have gone public in recent years all have greater than 100 percent net retention, sometimes much greater.

Company	Net retention rate	Timeframe
Atlassian	148%	Q1 2019
Twilio	147%	Q4 2018
PagerDuty	139%	Q3 2018 (S-1)
Elastic	130% (for 9 quarters)	Q1 2019 (QE call)
MongoDB	120% (for 10 quarters)	Q2 2017 (S-1)
SendGrid	113%	Q3 2018

*Atlassian's NDR source

But with greater granularity, comes greater flexibility on how to set up a pricing model. Tiered? By the minute? SaaS by usage or number of users? SaaS with the compute costs passed along to the end user?

Customers are now accustomed to metered pricing down to the smallest unit possible. Because of these pricing structures, startups can triangulate far closer to the value they deliver than ever before, and companies usually get a good sense of spend commitment based on usage and scaling needs.

3. Developer platforms tap into an existing budget or maximize engineering resources.

(Original Law #3: Developer platforms replace something companies already pay for.)

When developer budgets aren't clear cut, it's challenging to carve out new budget within enterprise spend. So, by replacing something companies already pay for, developer platforms can easily command the dollars associated with an existing budget line item.

However, two trends encouraged us to revise this original law.

1. The adoption of open source technology within enterprises translates to more teams building custom workflows and using open source software as building blocks to construct their infrastructure. As a result, we've seen a lot of commercial open source software companies successfully sell enterprise editions of their product that largely replace the internal work of teams that maintain open source infrastructure or build bespoke tools.
2. In 2013, containers and serverless were at the inception stage. Docker was the new darling and Kubernetes hadn't yet been released. As companies adopt containers and serverless technology, microservices, and distributed architectures have become even more popular. Companies that build developer tools around new technologies such as Kubernetes, GraphQL and new ecosystems may find themselves selling into the broad "digital transformation" project budget instead of a clear existing budget line item.

As distributed architectures become more complex, budgets for developer solutions are less about replacing exact existing line items and more focused on helping enterprises get the most out of their engineering resources, advancing innovation in the process.

4. Developer platforms strive to be as frictionless as consumer products.

(Original Law #4: Developer platforms have a fantastic developer experience (DX).)

Most companies creating tools for developers are hyper aware of developer workflows and making things as easy as possible for developers. The best ones create a delightful experience and are as easy to use and intuitive as consumer products. One of the many driving forces behind open-source momentum has been frictionless DX. Because these tools are easy for individual technical users to get started on, they immediately see the value of the product and share the love far and wide, creating a virtuous cycle.

For example, when we first invested in Cypress, the Javascript testing framework, we were blown away by the strong product feedback from so many developers who sought out a better open-source alternative to Selenium. On top of being a productivity gain, Cypress makes testing a delightful experience.

Here's what we noted in our original investment recommendation:

> "A small startup we spoke to said that Cypress has reduced their total testing time from 2 days to 30 minutes! This dramatic reduction in testing time completely changes how developers view testing. Rather than only testing apps rarely, when they have to push out a new deployment, Cypress now allows developers to run tests on every single code commit and thus get real time feedback on things that may be broken. Getting this rapid feedback through frequent tests on smaller changes to the code base substantially reduces the amount of time that developers spend debugging their code."

Auth0 is another fantastic example of how developer experience plays a critical role in helping new users navigate difficult domains and problems. For example, authentication and authorization have always been tough problems to solve internally, especially in the early days when homegrown solutions had no guiding principles and were becoming increasingly complex. Auth0's developer experience meant incorporating simplicity and accessibility through high-quality online resources and documentation. Today, this seems to have paid off as they have over 700,000 unique visitors a month and 16,000 members in their forum coming to them as a source of knowledge and expertise in the authentication space.

5. Developer mavens are usually the first and best marketing channel–their authentic evangelism drives adoption and sales.

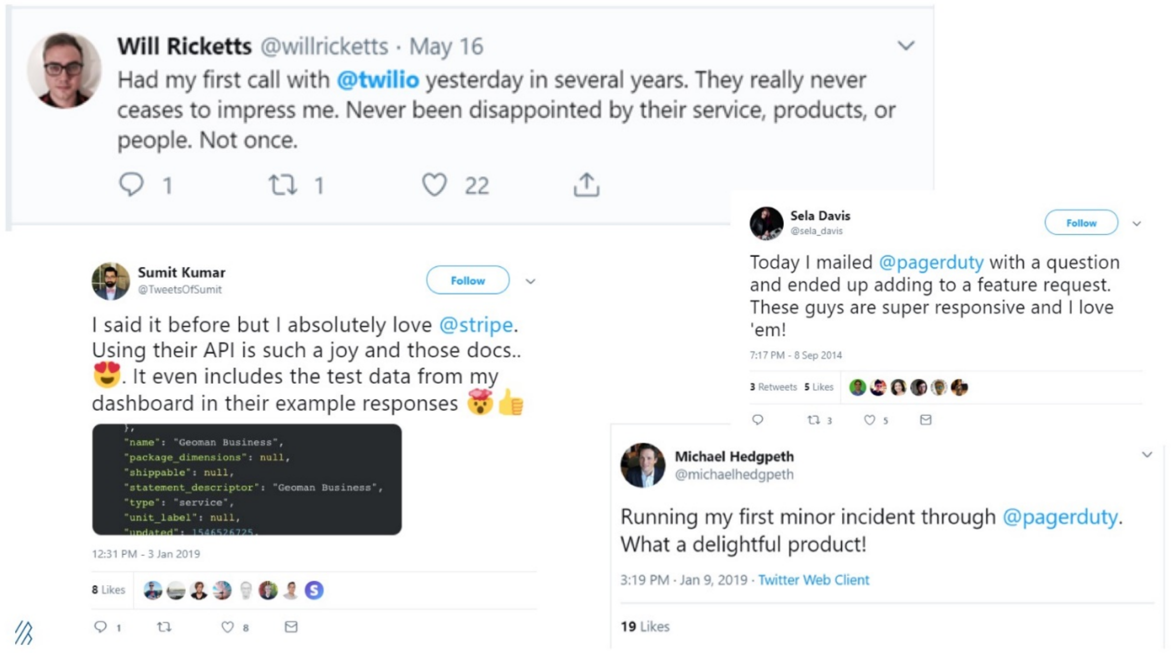
(Original law #5: Developers love it and talk about it, so they become your best marketing channel.)

Developers have more choices today than ever before so when developers find a product with an amazing UX, it's like catching lightning in a bottle and they share that discovery with the world.

In 2013, developer news and open source projects traveled mostly through word of mouth or informal posts that redirected to a GitHub page or StackOverflow thread. Since then, formalized distribution channels and social media continue to vault developers into tastemakers that influence the pecking order of technology and solutions across all different categories of business. This shift in power impacts the way companies approach "developer marketing."

When companies open source a project or launch a beta, there's now a standard media tour: ProductHunt, HackerNews, Twitter, Reddit, Quora, etc. Developers, in turn, look to metrics such as npm downloads or GitHub stars to measure project health or how a technology is gaining traction within the dev community.

But here's the paradox to this law: While developers notoriously hate traditional marketing, a developer platform such as Twilio, Stripe or PagerDuty has never reached massive adoption or scale without developers advocating for the product in some shape or form.



Will Ricketts @willricketts · May 16
Had my first call with @twilio yesterday in several years. They really never ceases to impress me. Never been disappointed by their service, products, or people. Not once.
1 1 22

Sumit Kumar
@TweetsOfSumit
I said it before but I absolutely love @stripe. Using their API is such a joy and those docs.. 😍. It even includes the test data from my dashboard in their example responses 🤯👍

```
  },
  "name": "Geoman Business",
  "package_dimensions": null,
  "shippable": null,
  "statement_descriptor": "Geoman Business",
  "type": "service",
  "unit_label": null,
  "updated": 1546526725,
```

12:31 PM - 3 Jan 2019
8 Likes
1 8

Sela Davis
@sela_davis
Today I mailed @pagerduty with a question and ended up adding to a feature request. These guys are super responsive and I love 'em!
7:17 PM - 8 Sep 2014
3 Retweets 5 Likes
3 5

Michael Hedgpeth
@michaelhedgpeth
Running my first minor incident through @pagerduty. What a delightful product!
3:19 PM - Jan 9, 2019 · Twitter Web Client
19 Likes

Today adoption of cutting edge developer tech is one of the single biggest contributors to the brand equity of engineering teams and thus has a huge impact on a company's ability to recruit developers. Candidates take notice if old-school systems are in place and the tools an engineering team chooses to adopt are often key measures that the best and the brightest developer recruits pay attention to.

6. Developer platforms demonstrate network effects through community, collaboration, and data.

(Original law #6: Developer platforms exhibit strong network effects making the product better and more valuable as more people use it.)

PagerDuty, the IT incident management platform, is an illustrative example of network effects. As more DevOps professionals occupied a seat within a company's PagerDuty instance, the more value accrues for teams that are live on the system. Notifications, after all, are only as useful as the people who can receive them so everyone can get the right alert at just the right moment. More usage didn't just translate to product value, but also more value for the entire business.

As usage and data collection increased over time, other business systems had a greater incentive to integrate with PagerDuty. This sparks an additional flywheel of value for the business–new integrations lead to more accurate and timely alerts and signals, which all contribute to an improvement in reliability, responsiveness, and quality of digital output.

Another sort of network effect comes into play as PagerDuty and other developer-oriented businesses scale to massive footprint: one cemented with data.

As another illustrative example, Stripe is in the flow of millions of Internet commerce transactions. Through watching the attributes associated with these transactions, Stripe can start to make judgments about which are real, which are fraudulent, who should be upsold, and who advanced credit without even bothering to input a credit card. It is only through data that developers can build streamlined customer experiences like these. In the near future, developer businesses will be able to give customers a competitive advantage through the proprietary data they possess (e.g. the "data moat") and enable them to take the right actions at the right time based on the insights they derive from being in the application flow.

7. Developer platforms enable companies to focus on product differentiation and unique competitive advantage.

(Original law #7: Developer platforms eliminate non-core skill sets.)

The fastest moving companies and teams wisely allocate their engineering resources to projects that move the needle in terms of unique competitive differentiation. Building internal tools is often a distraction. The Facebooks of tomorrow would be foolish to build much of the tooling they need to operate their organizations and infrastructure from scratch.

Some of the best developer platforms we have invested in have taken over non-critical skill sets for developers, like Cloudinary for image and video management. Previously, developers had to worry about how their sites rendered digital media and adapted to the myriad of devices used to access web content while Cloudinary allows developers and designers to ensure the right responsive design with only a URL call. This eliminates the need for developers to worry about whether potential customers can see a product image or sales demo and focus on content instead of the formatting.

Perhaps the most extreme enabler of this law in recent years is the rise in popularity of serverless technology. Serverless promotes the idea that all infrastructure except for the code logic can be abstracted away, freeing developers from non-core tasks of provisioning infrastructure, load balancing, and keeping the server OS running. Now, services such as AWS Lambda or Azure Functions will handle all of the scaling issues and eliminate this tedious setup work for developers.

8. Developer technologies empower all workers in an enterprise to contribute to product development, freeing up precious developer time.

(Original law #8: Developer platforms democratize development.)

Over the past few years, developers have become the cool kids of the enterprise. When we first conceived of our laws six years ago, developers were still on the outside looking in. Now they are the ones driving the decisions around product scope and delivery. As a result, other functions including knowledge workers like marketing, sales, and research professionals and non-coder technical disciplines like product managers, designers, data scientists, dev-ops practitioners, and security engineers want to get in on the action and have the same impact. We've seen the emergence of low-code or no-code functions as a result. These are software solutions that give their users powerful functionality without requiring development resources or the user knowing how to code.

One powerful example of the democratization of development is in the data realm for users like data scientists and analysts, who can now spin up a cloud GPU or provision the correct infrastructure to run their data pipelines and machine learning models. Both scrappy startups (e.g. Algorithmia, Floydhub, Paperspace) and cloud service providers (e.g. Amazon Sagemaker) offer one-click deployment for data scientists after they've created a model.

While these users are technical and highly experienced in their own domains, their time is not well spent learning how to provision infrastructure or optimize run-times.

When we first invested in Zapier in 2013, connecting and automating disparate business systems required technical skill–specifically the ability to parse the APIs specific to each system and then extract, normalize, and transpose the resulting data. Zapier made that as intuitive as a consumer app–without masking the ability to do more sophisticated manipulations–connecting all common business systems with a common interface layer and syntax.

In addition to being able to move and interact with data independently, business users are getting more input into the product development lifecycle itself.

Atlassian's Jira has made it easy for anyone from a product manager, customer support rep, or salesperson to submit tickets on software bugs, customer issues, and feature requests relating to an existing product that goes directly into a development sprint. Product teams no longer have to spend time collecting feedback across various parts of an organization as they have it all in one central system of record.

This trend is perhaps the one we are most excited about. We believe strongly that more and more functions of the enterprise, from marketing and sales to security and operations, will be "developerized" and transformed through platforms that allow more people to extend their capabilities and have greater and quicker impact.

What's next

With Elastic and Pivotal's IPOs and the two monster acquisitions of GitHub and Red Hat, 2018 was the year of the developer and perhaps more specifically, the open-source developer as one common thread across all of these companies is that they're commercializing open source projects (more on that in a subsequent post).

We don't expect these trends to slow down anytime soon (e.g Fastly's recent IPO), which highlighted the programmability of their content delivery network), and in fact, are expecting them to compound over time. As developers become increasingly powerful decision makers within organizations, our eight laws (and their amendments) are a testament to the progress developer-focused entrepreneurs have made and signal nothing less than an expanded universe of massive opportunity ahead.

Amendments to the eight developer laws:

1. Developer platforms based on granular and metered units of measure have more pricing power.
2. Well-designed developer platforms have natural upsell triggers inherent to the product.
3. Developer platforms tap into an existing budget or maximize engineering resources.
4. Developer platforms strive to be as frictionless as consumer products.
5. Developer mavens are usually the first and best marketing channel–their authentic evangelism drives adoption and sales.
6. Developer platforms demonstrate network effects through community, collaboration, or data
7. Developer platforms enable companies to focus on product differentiation and unique competitive advantage.
8. Developer technologies empower all workers in an enterprise to contribute to product development, freeing up precious developer time.