

# How to Build an API Program That Doesn't Suck

Survival Guide for Digital Business



**ADAPT  
OR DIE**

#DigitalKnowHow  
by **apigee**

# Table of Contents

## API Mindset | 5

Pitfalls: A narrow old-school way of thinking

How-to: Have a broad, strategic view of APIs

## Leadership | 8

Pitfalls: Trapped by myopia

How-to: Adjust your vision and flex your muscles

## Metrics and Value | 11

Pitfalls: Measuring the wrong thing

How-to: Keep your eyes on the (right) prize

## API Team | 15

Pitfalls: The SOA mindset

How-to: Build the right API team

## Investment | 19

Pitfalls: Empty promises, old patterns

How-to: Invest in and value an API initiative

## API Design Principles | 22

Pitfalls: Waterfall ahead

How-to: Build APIs that developers love

## Governance | 24

Pitfall: The problem with perfection

How-to: Govern lightly





# Taking Your Business Digital: Pitfalls & Best Practices

You already know that APIs are the foundation for building and running a successful digital business.

But the path to success can be unclear, rough, and challenging. The game has changed, after all, and this requires tossing out many of the rules by which you've run your business.

**You're not alone.** Other businesses have struggled down the path toward increased competitiveness, agility, and opportunity—and we're privileged to partner with hundreds of customers across myriad industries as they gear up to go digital.

We've seen the successful decisions and strategies that have enabled companies to adapt to new digital realities. But we've also seen businesses make decisions that slowed, stymied, or, in some cases, reversed progress toward becoming an agile digital business.

There's no one path to travel. There are many ways to make your business digital, but there are also many pitfalls. This is the survival guide to help navigate a smarter digital journey.

Read on for the details of the best and worst practices in how a business organizes, leads, measures, invests in, and executes an API initiative.

**There's one bit of advice we can offer above all else: GET STARTED.** If you don't take steps to adapt to the new competitive landscape and to your customers' expectations, the very survival of your business is at risk.



# APIs: The Key to Surviving Rough Digital Terrain

APIs: don't leave home without them.

Mobile phones and other smart devices, the cloud, and machine learning have made some amazing things possible: physical assets can be represented digitally, digital assets can control physical ones, and digital assets can be combined to create something new.

Modern web APIs make all of this possible; they're how we build compelling experiences that generate, consume, and combine digital assets. They've become the foundation upon which digital business is built.

With the explosion of API-powered apps and experiences, and new, customer-focused IT organizations, companies across industries need solutions to manage APIs and their API-driven businesses.

An API platform is a system for efficiently and effectively managing APIs at scale. Its capabilities span design and development, publishing and operations, monitoring and analysis, monetization, and iteration and optimization.

API platforms help enterprises grow an ecosystem of consumers and API producers. They enable the secure sharing of data and services, give API providers the ability to quickly adapt to business needs, and allow producers and consumers alike to benefit from network effects—in other words, they can tap into a network of people and technology that expands their reach and increases opportunity.

# API Mindset

Many organizations understand the important role APIs play in adapting to the expectations of their customers, employees, and partners. Many have even implemented APIs. But the path beyond that first API is often full of wrong turns.

A project-based mentality, especially as an organization builds more APIs, results in a lack of visibility into how APIs are performing from a business perspective. You don't have a holistic view of which APIs are being used, by whom, and for what purpose. This can stymie your ability to make business decisions about what data and services to prioritize, which to combine for new product and service offerings, and which gaps to fill in order to deliver a compelling, unified user experience.

## Pitfalls

A narrow old-school way of thinking



Congratulations! Your IT department has completed an API project. Maybe you've even built a mobile app. You can check that off your organization's to-do list, right?

Not so fast. Sure, you've scored a short-term success. But by taking on the project in an ad-hoc, one-off way, over time your organization will stumble around in the dark and fail to leverage its assets.

## An inside-out technology-focused approach

Old habits die hard. In the last decade, enterprises invested in systems that solved the problems of the previous decade. But today's challenge is to evolve your business to adapt to an entirely new customer-centric, hyper-connected landscape. When an enterprise focuses on integration—connecting systems to systems—or solely on exposing data from systems, the people, processes, and technologies it selects are often inappropriate and ill-equipped to work on today's challenges. Just exposing the data doesn't help you enable secure, self-service, easy consumption of data and services by the developers who build the experiences and apps critical for a business to survive in a digital world.

For example, data extraction entails the cumbersome process of reverse engineering an app's internal structures and accessing databases directly. APIs, on the other hand, enable apps to expose data in structured ways, eliminating data extraction processes and opening the door for deep analytics and processes that empower developers to build new apps and experiences fast.

## APIs as tactics, not strategy

Even companies that were not born digital have hundreds, often thousands, of APIs. A business does not become a true digital business if it thinks about APIs only as a tool in the IT architect's toolkit. They must be a core part of the business strategy. The biggest sign of this pitfall? When there are no C-level enterprise goals or KPIs (key performance indicators) around using APIs to deliver new customer experiences, value, and revenue. Another clue? If there's no discussion about APIs as a way to streamline partner on-boarding, or as a path to optimizing internal processes.



**Companies do need assets like computers, servers, networks, and software, but those purchases are just the start. Digital leaders stand out from their competitors in two ways: how they put digital to work, especially in engaging with clients and suppliers, and how intensively their employees use digital tools in every aspect of their daily activities.**

- Harvard Business Review, April 2016

### How-to

Have a broad, strategic view of APIs



APIs have become a hot topic well beyond the boundaries of IT. Modern web APIs are how companies build compelling experiences that generate, consume, and combine digital assets. They enable organizations to engage in real-time, multi-party interactions with customers, partners, and developers. That's why successful enterprises manage their most important APIs as business assets.

The question is not how many APIs your enterprise has. Rather, it's how many APIs does it have that matter? The fate of your digital strategy rests on an accurate understanding of the APIs that have business impact. They are your options, contracts, revenue generators, and brand builders.

## APIs as products

Successful companies view their APIs as products. In today's digital market, it's to your advantage to treat developers inside and outside your organization as consumers you want to attract, delight, and retain. APIs should enable this. Imagine your API earning a great net promoter score.

## APIs as channels

APIs that succeed as products have the potential to become channels. An enterprise can maximize the potential value of an API or set of APIs by developing a go-to-market (GTM) plan, thereby turning product into channel. How do you manage your API as a serious channel play? Can you point to a defined value proposition and target customer for your API? What are the metrics for adoption, customer success, and economic value? Is someone accountable for driving those numbers?

## API ecosystems

Many APIs will lead happy lives as products or channels within a high-impact digital business portfolio. But building an ecosystem is where you can maximize value.

Regardless of what product or service experiences drive your business, supply-side economies of scale ("the more we produce, the less each costs us") are no longer enough to win. Mastering ecosystems has become the competitive advantage in many industries. Many successful digital businesses, whether digital natives or migrants, use network effects ("the more something is used, the better it gets") to get ahead.

You'll know you've unlocked the power of ecosystem dynamics when your investments shift the adoption curve from linear to geometric or exponential.



**Our sole purpose in life is to create great APIs. We spend a lot of time trying to put together a solid, easy-to-use experience for [developers] ... focusing on the best API experience.**

- Brian McMahon, Mapquest

# Leadership

An API platform thrives when it's sponsored by a highly engaged executive who has an "outside-in" perspective, a budget, and the power to implement organizational change. And this kind of person isn't always easy to find.

## Pitfalls

### Trapped by myopia



An "inside-out" or "build it and they will come" approach is a path to nowhere. API initiative leaders will fail if they focus on the inventory of data and services in the organization (and what can be built from them). The road to success is paved by partnering with the lines of business to identify the apps and digital experiences that will meet the demands of customers, partners, and employees.

Without close ties to the lines of business, the API leader can't really understand which APIs will drive the most value, and is usually unable to generate demand for APIs that are built.

**Hope is not a survival strategy.** The API initiative leader who exposes services or data in the hope that someone will discover and use them is doomed to failure.

Even when leaders have the proper perspective along with a mandate to assemble plans, they will flounder and fail if they lack the resources or authority to build APIs or change IT delivery models.

## How-to

### Adjust your vision and flex your muscles



Having a strong senior executive leader is the key differentiator between API platforms that thrive and those that never make it out of the project stage. This leader should take an "outside-in" approach to API development. It's the single greatest factor in delivering digital experiences with real business impact.

## Think “outside-in”

By “outside-in,” we mean that the first thing a successful API program leader should think about is what digital experience is needed by a partner, customer, or employee. Only then should the leader focus on the apps that will be needed to deliver that experience.

Once you know what the apps need to do, it’s time to consider the APIs required to create the desired experience. The very last thing to consider is which data or back-end services to leverage. We’ve seen the importance of this proven time and again in the field.

**The theory around the benefits of “outside-in” IT is well established. The Apigee Institute report, “[Lessons from the App Masters](#),” shows the positive relationship between IT departments that think “outside-in” and real business results.**

API initiatives are iterative, and need to be funded accordingly. API initiative leaders must have the clout to get the necessary budget. It’s the only way to empower the API team to deliver APIs as products. This often means freeing the API team from the constraints of traditional project-focused IT budgets and instead granting it the agility to build and improve APIs as products, not one-off IT projects.



We cannot take our foot off the pedal on our own marketing, internally, of the API program. We thought that based on the merit of the program, it would just work, it would just get adopted. But we really got traction when our CTO put a line item on his objectives, in his vision that APIs were going to be something for us.

- Saori Fotenos, Thomson Reuters

## Influence spending patterns

API initiatives are iterative, and need to be funded accordingly. API initiative leaders must have the clout to get the necessary budget. It's the only way to empower the API team to deliver APIs as products. This often means freeing the API team from the constraints of traditional project-focused IT budgets and instead granting it the agility to build and improve APIs as products, not one-off IT projects.

## Drive culture change

The most successful API programs often demand significant organizational mind shift and personnel realignment. The tools, techniques, and processes—as well as the perspectives of the people who work to make an enterprise digital—must shift to a new way of thinking. Executive leaders' authority has to extend to organizational structure, and they must be willing and able to understand, initiate, drive, and manage cultural changes.

# Metrics and Value

The measures that matter most to global enterprises are usually variations on the same short list: revenue and margin, customer satisfaction, market share, and innovation.

As the number of examples of digital transformation changing the game in every industry grows, it's become universally understood that digital capabilities can drive enterprise key performance indicators (KPIs).

For an API program to drive sustained and increasing value for a business, it must be aligned with the overarching objectives of the business. But how a business measures success with its API initiative must often depart from traditional metrics; the best metrics aren't always the most obvious ones.

## Pitfalls

### Measuring the wrong thing



“How many APIs do we have?”

CIOs beware. When this question is inevitably put to you, resist the temptation to pass it along to a virtual team that goes off to beat the bushes to find every piece of integration code across the enterprise in order to report back that “we have 47 APIs.” Or 923. Or 3,117.

It’s the wrong question. Publishing an API is not the same as creating value through the consumption of that API.

When the metric is the number of APIs, the APIs created can be of low value and quality. In many of these cases, the API might be exposed, but not used.

## What happens when you measure the wrong thing

IF YOU MEASURE THIS . . .	YOU'LL STUMBLE BECAUSE . . .
<b>Direct revenue</b>	Most APIs are enablers, so they need to go through a cycle of adoption before proving their relevance to revenue—and even then it's often indirect.**
<b>Number of APIs</b>	APIs must have observable utility. A plethora of APIs without relevance to apps are unusable and cause confusion—and, potentially, rejection of the program.
<b>Number of developers</b>	APIs must provide value to developers. A singular focus on marketing and on-boarding developers without paying attention to the value of APIs for those developers will result in loss of momentum as developers abandon the program.
<b>Cost reduction</b>	APIs themselves don't always reduce costs, but they are critical to broader internal refactoring of value chains, which do reduce costs.
<b>NPS (net promoter score)</b>	APIs themselves don't drive experiences, but they're critical for building and measuring the successful apps that do improve NPS.***
<b>Number of customers</b>	Customers don't experience APIs directly. Without careful systemic alignment (especially across app teams), the API program will fail.

\*\* There are a few businesses that are purely API businesses, where a subset of their APIs can be directly monetized such as information feeds, brokerage services, and purchase/payment transactions.

\*\*\* There are partner scenarios where the “customers” of an enterprise are almost exclusively developers, in which case, APIs can directly result in NPS improvements.

## Standalone profit and loss functions

Another common challenge involves treating APIs or API initiatives as standalone profit/loss functions. This can lead to the perception that APIs are “exotic” and not a part of the core value-creation activities of an organization. This makes APIs far less likely to show material value to the business as a whole.

Viewing APIs through this lens often goes hand in hand with applying the classic project management triangle to API success, with the three constraints of scope, time/schedule, and cost/budget. This view might work in a traditional IT model applied to services creation, but it won’t enable delivery against the new realities of the digital economy.

## How-to

Keep your eyes on the (right) prize



Just as an “outside-in” approach is key to API initiative leadership, the most important metric to track when it comes to APIs is the velocity at which digital experiences are created.

## Proceed with speed

Focusing on agility and the speed at which digital experiences are created helps to avoid narrow, arbitrary, or API-focused metrics that distract from what’s most important: generating business value. Focus metrics on driving the growth, breadth, and speed of delivering the applications that depend on APIs.

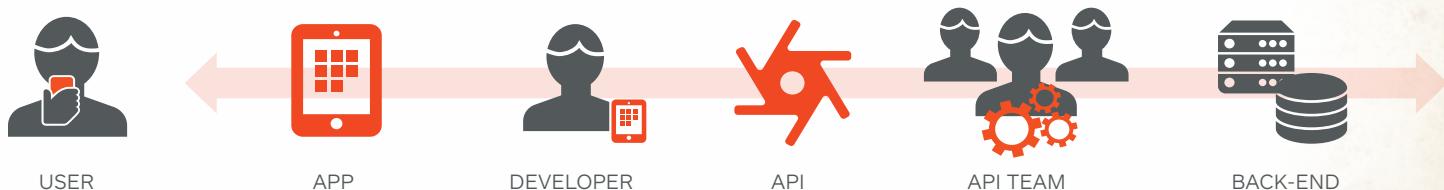
## Enable the entire digital value chain

APIs are the core technology in the digital value chain. An app calls APIs from within its code, invoking services elsewhere in a network that return data or perform processing. These apps include mobile apps, websites, and a partner’s servers, and are the products that deliver the experiences and create business value.

The APIs that power the apps must have certain properties in order to be valuable to the apps: good design, robust security, real business value, solid performance, and quick access to the right data in the back-end systems upon which businesses rely.

Finally, the back-end systems that do most of the work of the enterprise must be well-connected to the APIs, meaning they trust the APIs for access control, traffic management, security, proper identification of the user, coordination, and even lightweight orchestration of multiple back-end systems.

## The digital value chain



## Don't define metrics in isolation

Don't define metrics for APIs in isolation. To be effective, the metrics for the API program must align across the people and technologies in the digital value chain.

## Measure what matters

IF YOU MEASURE THIS . . .	YOU'LL SUCCEED BECAUSE . . .
<b>Speed to API</b>	API teams that deliver usable APIs quickly enable the creation of new customer experiences by app development teams.
<b>Speed of on-boarding</b>	Developers building front-ends (apps and websites) move through a portal-enabled, automated approval process quickly. They self-serve all subsequent on-boarding tasks: registering their apps, obtaining keys, accessing their dashboards, and discovering APIs.
<b>Growth of traffic</b>	You'll develop a strong devOps and runOps culture, continuously monitoring, improving, and driving value through APIs. This must be coupled with measuring the reliability and scalability of back-end systems.
<b>Number of products sharing the APIs</b>	The API team focuses on building APIs that matter to the business. APIs that unlock valuable data from back-end systems get reused often to build apps that drive value for the business.
<b>Number of active developers</b>	Active developers indicate a thriving developer ecosystem. When APIs deliver real business utility, developers build apps that drive value for the developer and the API provider. Signups for signup's sake are like empty calories: they generate heat and activity, but do not drive any business results.
<b>Number of partners</b>	A thriving partner ecosystem enables you to extend your brand with new products, complement and enhance existing product offerings, expand reach via new channels, and leverage new business models. Training events such as DevJams and hackathons for a partner's developers are good ways to jumpstart interest and drive adoption.
<b>Cost reduction</b>	Cost savings arise from understanding the utilization of back-end systems and emphasizing the creation of business value on top of a small set of commercial, off-the-shelf (COTS) platforms. Don't roll your own. It's hard to achieve the necessary scale, and it's expensive.
<b>NPS</b>	Your end user experiences the digital value change via an app. Measuring and tuning all aspects of the digital value chain results in a better NPS—and, hopefully, a higher rating in the App Store.

# API Team

Throw multi-month feasibility studies and detailed requirements documents out the window. You're not planning an IT project—you're charting your API initiative. Your team needs to focus on building quick prototypes, meeting with customers, daily release trainings, and A/B market testing with proofs of concept.

An API team needs to be agile. It should consist of a loosely coupled core of four or five members representing different perspectives, such as product management, architecture, API development, and service integration. It's tasked with creating and iterating the governance and development processes needed across the entire enterprise to quickly and flexibly deploy APIs at scale.

## Pitfalls

### The SOA mindset



Sometimes an API team doesn't have what it takes to be agile. We've seen teams that maintain a centralized SOA governance mindset, enforcing standards with contracts and strict, centralized control. This model creates bottlenecks—it's too slow to deliver apps and experiences at the pace of mobile.

Creating “centers of excellence” can also be a trap. The most successful API teams are decentralized, enabling them to be agile, empower individuals, and monitor and ensure standards.

But worrying about standards can go too far, and result in “analysis paralysis.” Beware the organization that deploys an API platform, and claims to launch an API program—but builds no APIs pending organization-wide agreement on governance, standards, and process.

The most successful API programs focus on getting started—on building APIs that make it easy for developers to access content and data from internal systems, with the right levels of access control, and in a way that shapes the data to exactly the size and format necessary for ease of app development.

Defining ‘agile’ from the top down doesn’t work. This is not a manager’s process.  
We decided to let the API team work it out.

- Daniel Wolff, Vodafone

## How-to

### Build the right API team



The API team is the core of any new API effort, and getting it set up, funded, and empowered quickly is a critical task.

The team's goal is to make the development and use of high-quality APIs pervasive throughout the organization, with a focus on enabling agile API development using pragmatic REST design principles and building APIs that developers will love.

#### **Successful teams have several things in common:**

- ▶ Strong support from an executive sponsor who has the ability to provide funding and other resources.
- ▶ A mandate to build APIs—not just create architectural and governance models.
- ▶ The willingness to train other teams to build APIs quickly and in accordance with design best practices.
- ▶ Direct access to lines of business to understand their goals and digital plans.
- ▶ Freedom from the constraints of existing services governance, SDLC, and funding policies, and the ability to do what it takes to create and deliver highly consumable APIs quickly.

#### **The API team has several responsibilities:**

- ▶ Building highly consumable APIs that provide business value (initially, these tend to be the core capabilities, such as identity, that provide value across the organization).
- ▶ Researching and testing API development and management best practices, and applying these lessons.
- ▶ Proactively engaging colleagues across the digital value chain—from business and finance professionals through testing, security, and app developers—to enable new development processes for APIs.
- ▶ Creating material (such as an API playbook) to rapidly scale learning across the enterprise.
- ▶ Leveraging analytics to identify problems and teams that are having trouble with their APIs. A team with lots of APIs but no consumption, developer churn, or poor performance is in trouble.

## Who is on your team?



### The API team leader

A product manager—not a project manager—leads the API team. Product managers tend to take the perspective of a businessperson managing a technical product.

They make trade-offs on features and functionality based on their knowledge of the market for the API they are creating. They think of their APIs as products and iterate them based on feedback from their developer customers. By contrast, project managers focus on delivering against a set of requirements, and their goal is to get to “done” with their product unless and until that product is discontinued.

The product manager should be aligned with the business leaders for whom the API is being built, and should have a relationship with IT leaders to ensure that no bottlenecks are formed in development, operations, testing, or security.



### The architect

The API team includes an architect who guides the creation of initial APIs, as well as the development and testing of best practices for API design. Ideally, this architect will also develop APIs—we've found that the best API architects are those who code.

The design of APIs for the consumption (or northbound) side of the API platform—the side that delivers the apps and experiences to end users—is very different from the design and architecture of APIs for the exposure (or southbound) side of the API platform, which plugs into the back-end systems.

The architect needs to clearly understand the differences, and be able to speak both app/web/consumption and back-end/integration/exposure languages because alignment between the two concerns is critical. If such a person is hard to find, select a strong consumption side architect to work alongside a strong integration-side architect.

We've seen the most success when one person (initially, at least) is tasked with understanding the ins and outs of the API platform, especially when it's a cloud-based solution. This individual educates other team members on how to operate and leverage the platform for best advantage in the organization, in turn increasing adoption and providing even better leverage.

For an on-premises deployment, we suggest that you have a team dedicated to running the platform at scale.



### The evangelist

Another key role is the person responsible for outreach to the consumers of your API product. It doesn't matter if these developers are internal or external to your organization. They create the developer community that uses your APIs.

The developer evangelist supports the developers, monitors their progress, answers their questions, and brings feedback to the rest of the API team and the business. Common tasks for the developer evangelist include managing the developer portal—including responding to forum questions and writing blog posts—meeting with developers across business units and partners, running hackathons, and contributing to road-map discussions.



### The API developer

API developers need to know what modern app developers want and need in order to produce great apps quickly from a set of intuitive, highly consumable APIs.

This requires strong web development skills. They need to be well-acquainted with the API platform in order to implement the proper security, traffic management, and other policies that secure and scale the APIs per enterprise standards.

To effectively integrate APIs with the back-end services at the correct granularity and cadence, developers need to be strongly grounded in the enterprise's middleware. If at first a company finds it difficult to hire one or more developers with the ability to develop the northbound/consumption oriented APIs as well as the more traditional southbound/exposure APIs, form a team with developers who are strong in one domain but also enthusiastic about the overall API program. To enable speed and business agility, this team should have a strong bias toward the web/consumption side of the platform.



### The extended team

In addition to the core roles of leader, architect, API developer, and evangelist, it is critical to develop several auxiliary roles across the organization. These roles don't need to be filled by people working full-time on the API team, but it's critical to have a concerted effort to create API capability and understanding within business units and across the many IT silos—especially in development communities, security, testing, and operations teams.

While a central team is the best way to start for most organizations or lines of business, API development skills should become pervasive throughout the company. When a large number of developers are skilled at creating and consuming APIs, the organization reaps the benefits in terms of faster development times and market agility.

# Investment

When becoming a digital business is a strategic imperative, an enterprise needs to invest in an API platform to run the business at scale and to manage and secure an increasingly complex network of connected applications, interfaces, and back-end systems.

This is the moment when many organizations falter—just on the cusp of enterprise-level modernization. Without direct funding that matches the API initiative’s bold aspirations, the initiative is doomed.

It bears repeating here that APIs are about business, not IT. As such, they can’t be funded like IT projects, which are centered on traditional cost-related metrics, such as headcount and bottom lines.

## Pitfalls

Empty promises, old patterns



An overarching problem we’ve seen repeatedly is that of the unfunded mandate. Also known as “digital whitewashing,” bold statements about being a “leader in digital” and “acting like a startup” get tossed around without dedicated funding to support the vision. But having a budget dedicated to the API initiative is just the start.

Depending on the size of the organization, the portfolio of projects and the myriad tasks that can be serviced by a platform can be huge, covering a large swath of the organization and scores of stakeholders. Tradition holds that an enterprise takes all of its projects, evaluates them against ROI, and then tackles them based on that ranking. In this model, funding for the platform comes from each of the stakeholders as their project is initiated.

This approach is all wrong. It’s like building a factory as the orders come in, starting with a vacant lot.



## How-to

### Invest in and value an API initiative

The most successful API initiatives—the platforms for digital business—are funded as separate investments, using a more strategic approach such as organizational capital or real options valuation.

How a company invests in its future can be restricted by the uncertainty that crops up in periods of rapid change.

NPV (net present value) emphasizes linear solutions and impact, and ignores the value of enabling a range of options and potential future use cases, which an API initiative, with its reusable, highly consumable digital assets, makes available for the enterprise.

An API initiative enables flexibility, experimentation, and speed. This isn't only practically useful—it's strategic. You can count on three things about creating value using APIs: the optimal approach for doing so will evolve, change happens fast, and, on balance, the range of opportunities will only increase over time.

**Boards need to master a second language—one focused on digital themes, such as speed to market, agile product development, platform-based delivery models, and the benefits and challenges of analyzing various forms of corporate data.**

- McKinsey & Company, July 2016

More broadly, when it comes to the digitization of enterprise value chains and the commercial and social environment as a whole, we know that it will only increase, and it will happen fast.

There's an empirically proven model for investing in uncertain short-term conditions in the context of a long-term trend: investing in the stock market. Three principles have been validated as being optimal over the long run:

- ▶ Start investing as soon as possible.
- ▶ Don't stop investing.
- ▶ Invest in a portfolio, with the expectation that while some assets will under-perform, others will strongly outperform.

These principles are directly relevant to your API initiative and building a digital business. Unlike the Dow, digitization is never going to decrease (barring some epic global disaster).

## Think real options

Without resorting to complex formulas, the principle of real-options thinking helps set enterprises up for long-term digital success by projecting the value of an investment across different market scenarios. For digital investments, this should take into account the march toward greater digitization, the possibility of disruption, or new opportunities made possible by digitization.

[Empirical research by the Apigee Institute](#) shows clear evidence that a solid API initiative, aligned across business and marketing, whose progress is carefully tied to core enterprise metrics, must be supported by an investment strategy that uses a “real options” portfolio approach.

## The strategic implications are:

### Increased digitization

Embrace digital potential and make the best possible estimate of time- and industry-specific changes ahead, but don’t lose sight of the fact that the overall trend is toward more.

### Risk of inaction

Take into account the potential benefits across efficiency, customer satisfaction, and revenue, and the downside risks from inaction across these same areas.

### Fast fail and maximum reuse

Make a strategic decision that doing nothing is not an option. Pick the most promising projects, structure as much as possible for fast failure and maximum reuse, and keep pushing forward rather than standing still.

A few capabilities are likely to be assets that keep you in the game—even if they do so in ways that weren’t originally planned.

# API Design Principles

Many pages have been written on the intricacies of API design (Apigee offers several eBooks on this topic, including [Web API Design: Crafting Interfaces that Developers Love](#) and [API Façade Pattern](#)). But the success of an API design should be guided by one overarching principle: how quickly developers can get up to speed and start enjoying success using your API.

So it's key to think about design choices from an "outside-in" perspective—in other words, from the app developer's point of view.

## Pitfalls

Waterfall ahead



Developing APIs with a waterfall approach is in direct opposition to the goals of lightweight, fast, flexible innovation. Designing highly consumable APIs takes iteration.

APIs must be built using agile practices. Many organizations get caught in the trap of trying to decide by committee exactly how they will design APIs—before even writing any code.

Remember, your API is a product. Your API initiative will fail if you don't offer great products that developers love.

Don't overload your API with features; often the more features you have, the harder it is to communicate the unique and differentiating value of your API, and the harder it is to do a great job functionally meeting the key use cases.

## How-to

### Build APIs that developers love



An agile approach to building APIs looks different and includes: rapid, short iterations, continuous testing, and progress toward continuous deployment. There are several flavors of agile development, and, when executed well, any of them can work. (Apigee provides an agile framework, "[Accelerator Methodology](#)," to guide the process of setting up, building, and delivering a successful API program.)

The important thing is that API providers take a pragmatic approach that focuses on consistent application of a few core design principles paired with the application of technology to monitor and communicate with the developer ecosystem. This ensures that app developers have the best possible experience with your APIs.

We advocate a set of core design principles including using GET, PUT, POST, and DELETE; and using nouns and not verbs, plural and not singular when naming API resources. Consistent naming conventions for core functions such as customers and accounts will also go a long way in making a company's APIs intuitive and usable by developers.

Most importantly, once you've designed your initial APIs, leverage the analytics capabilities of an API platform to monitor the adoption and performance so you can make adjustments and reach out to developers as needed. Developer portals help a business communicate with the developer community, gather their input, adjust your API offering accordingly, and even monetize your API products.

# Governance

Many early SOA implementations were done by central architecture teams enamored of the elegance of a unified architecture, but removed from the realities of day-to-day app development. Today, we're at a point where an entire generation of developers has come of age in the internet era.

Services creation is a natural part of the software development process. It's ad-hoc, and it's prolific. API design and development is every developer's job. Thanks to the proliferation of virtualization, putting an API into production can often be done with little or no oversight.

## Pitfalls

The problem with perfection



Often the result of the SOA focus on reusability, a focus on governance too early in an API program will cripple the effort. Waterfall development, multiple-month, big-ticket projects, a focus on IT control, and funding methods that emphasize a “one chance to get it right” mindset will put success out of reach.

You're doomed if you put in place a centralized services governance process owned by a special architectural team in IT with an iron grip on agile and decentralized API-first architectures.

## How-to

Govern lightly



Successful API programs are built by doing—by iterating designs and processes to optimize the digital experience as quickly as possible.

Speed in delivering business value is critical. Making this shift—first in mindset and then in practice—is the most challenging aspect of launching an API program.

To make it happen, it's important that the API team is freed from existing governance structures so that it can discover and develop the processes needed to dramatically accelerate the creation of digital experiences through APIs.

A common response to this proposal of lightweight governance is the concern that security will be compromised, and that APIs will be used in ways that damage the enterprise brand. There are several good responses to this concern:

- ▶ If your organization already has apps, you already have APIs. If those APIs are not being run on a platform with a consistent security pattern, you're already exposed. The remedy is to implement an API platform.
- ▶ No one will be able to use an API unless they are provided a key. The provisioning of keys is completely within the control of the enterprise.
- ▶ Once a developer has a key, the organization will be able to use powerful analytics tools to know exactly what is being done with the key, how many apps, and how much traffic is running against the API, and other key data points. If at any time the enterprise wants to bar a developer's access to the API, it simply revokes the key.
- ▶ APIs offer a number of additional security methods that add depth to the existing security approaches of the enterprise, including authentication, threat protection, and traffic management. A well-implemented API security program adds depth to an enterprise security strategy. It doesn't weaken it.

Lightweight governance doesn't mean no governance. For many organizations, security provisions such as authentication and policy settings should be agreed upon and enforced, and API design guidelines should focus on consumability and design consistency.

However, API teams should start with a very short list of standards (a "bill of rights") and resist creating long governance processes with many standards (a "tax code").

Trying to plan governance for every instance of API creation in advance will kill your program. It's far better to get started building, learn what increases speed and agility, and leverage the developer, analytics, and policy tools in the your API platform to mitigate risk, then iterate and adapt as the needs of the business evolve.

A final thought on governance: it's much easier to enact legislation than it is to remove it.



# Ready to Create an API Program That Doesn't Suck?

It's not hard to build and operate an API initiative that doesn't suck. But it takes a mind shift for most companies. It is a different way of doing things from people, process, and technology perspectives. Thinking digitally means avoiding some of the pitfalls of IT tradition and seeing the challenges and opportunities from your users' perspective, measuring what matters, and structuring development and operations to support new needs.

From empowering leaders, to embracing the right funding models, to building the right teams, it's about working differently and leveraging the power of APIs across the digital value chain. Armed with APIs and an API platform strategy, you can do more than just survive. It's time for your digital business to thrive.

# About Apigee

Apigee® (NASDAQ: APIC) powers the APIs that make every business a digital business. Apigee provides a leading API platform that helps companies—from disruptive start-ups to the Fortune 100—rapidly adapt to the business and technology requirements of the connected, digital world. Many of the world's largest organizations select Apigee to enable their digital business, including over 30 percent of the Fortune 100, four of the top five Global 2000 retail companies, and five of the top 10 global telecommunications companies.

**For more information, visit [apigee.com](http://apigee.com)**

## Share this eBook

