

# A Roadmap to Agility

**To thrive in today's digital transformation environment, embrace tools and processes that guide the way.**



A close-up photograph of a man with a beard and sunglasses, wearing a dark suit jacket, white shirt, and purple tie. He is looking down at a white tablet computer he is holding with both hands. The background is a bright blue sky with some architectural lines.

## Table of Contents

Defining Agility	3
Agility's Requirements	3
Agility's Enablers	3
The Road to IT Agility	4
Adjusting Speed Along the Road	6
What to Avoid On the Road to Agility	7
The Importance of Traveling Partners	8

## Overview

Digital companies such as Uber, Netflix and Airbnb are a disruptive force, using software as a key competitive weapon in how they conduct business. This is causing more traditional companies to consider how they too can create compelling customer experiences.

In most cases, achieving these kinds of results requires a new way of thinking about how IT conducts the software development and delivery process—how to make it faster, more responsive. It requires experimentation in a way that enables developers to learn and adapt rapidly. And it requires a dramatic move away from the traditional waterfall development approach.

Transformation is not easy, yet the business and IT agility rewards are great, with multi-faceted, forward-flowing benefits right down the business line.

# Defining Agility

The best way to define IT agility is to look at what leading agile digital companies do:

- They have the unique ability to make IT changes and deploy them—often in a matter of hours.
- They have the ability to roll back changes if they result in unexpected or undesirable user impacts.
- They have the ability to do needed testing as well as try out changes with a few users before deploying broadly.
- They enjoy a strong feedback loop for the existing user experience as well as the ongoing impact of development changes.
- They have zero or nominal downtime during the maintenance and patch process.
- They have the ability to scale up and scale down in minutes and hours.

---

## Agility's Requirements

The Big Three tech trends—mobility, cloud computing and the Internet of Things—show that the world is truly going digital. As a result, organizations need to begin operating at the speed of digital, especially if the business is to take advantage of real-time, always-on connections within a data-rich environment.

Mobility in particular is at the heart of the digital customer experience, with users increasingly spending more time with their devices. And the mobile theme of always-on, always-available

further increases the need for organizations to embrace truly agile approaches to development, expanding the definition of becoming quicker and more adaptive.

Mobility also relies on an ecosystem of applications and systems to deliver desired, compelling customer experiences. It requires that front-end mobile apps as well as other applications in the ecosystem move at lightning speed.

---

## Agility's Enablers

As organizations grapple with mobility challenges, many are finding it logical to push the parameters of their cloud plans. They are recognizing the urgent need to become “cloud-native”—that is, ensure that their software, services and applications integrate smoothly in the cloud and work nicely together. And, by moving in this direction, IT can finally address the need for IT agility, excellent user experiences and seamless mobility.

### Embracing DevOps

A key factor for enabling agility is embracing the DevOps culture. Centered on teaming and integration of key IT stakeholders (developers, QA, operations, release management and business), DevOps helps organizations achieve greater IT speed and, thus, faster achievement of business goals.

A true transformation involves revisiting how people, processes and technology come together. It is about having the right culture and the alignment of all stakeholders and team members toward shared goals. It means changing the work environment to promote working relationships that are more trustful, collaborative and transparent. The right organization structure, incentives and team spirit need to be in place.

It also requires methodology and high-velocity processes that drive small, fast releases and experimentation. Here, we mean that processes and methodology must be automated and standardized to the greatest extent possible to achieve quality and speed.

### Key Enablers of Agility

As an organization starts its transformation into a cloud-native enterprise, it needs to focus on the key enablers, including:

- ▶ **Capitalizing on agile excellence through DevOps principles**
- ▶ **Effectively leveraging a dynamic, cloud-based platform**
- ▶ **Gaining more-efficient use of compute with containerization**
- ▶ **Making the most of microservices**

The more a business adopts these enablers and builds maturity by using them, the faster and more adaptive it becomes.

If you're working with Agile principles, as most of your peers are, DevOps is a key enabler. Unfortunately, few companies have realized Agile's full potential, because the delivery of end-to-end flow and speed is constrained, especially by downstream activities such as integration testing and release management. This is where DevOps makes the difference, dramatically moving away from the slow and siloed waterfall approach to a collaborative environment where agility ensures that tradition never stands in the way of progress.

## Leveraging a Dynamic Platform

Additional agility comes from embracing a dynamic, cloud-based platform that seamlessly supports applications. A dynamically provisioned environment is critical to the speed of a DevOps-centered organization. It provides the ability to provision on-demand capacity and tear it down when you are done. This enables the team to create new dev, test or QA environments as needed, without wasting time. That means that features don't get delayed as people stand around, waiting for new environments to be provisioned. This reduces bottlenecks, improves the flow of functionality and improves quality at the same time.

Having a dynamic platform also benefits the operations teams, which can automate operational activities. This means that the entire flow of build and deploy can be treated as code that can be monitored and optimized.

## Capitalizing on Containerization

Agility is also enhanced when organizations utilize independent containers, enabling IT to quickly move, scale and install applications. Provisioning speed is a key component of containerization, because applications run the same in any containerized environment, which eliminates all the time commonly spent troubleshooting when an app moves throughout the stages of developing, testing and production.

Containers also have inherent compartmentalization and elasticity, which facilitates easy experimentation.

## Using Microservices

To increase agility at the application layer, organizations focused on becoming cloud-native need to adopt a more loosely coupled architecture using microservices. After all, architecting and deploying applications in a monolithic fashion results in long release cycles. With a more loosely coupled architecture and microservices, an application can be broken into smaller releases to enable parallel tracks of development and testing. A failure in one release does not necessarily delay the progress in other releases.

Having smaller releases makes it easier and quicker to debug and test. Also, these small releases can be reused by other applications. This results in a pipeline of small releases that deliver business value as soon as they are done.

# The Road to IT Agility

As organizations embark on the path to digital transformation, it is important to recognize that changes, especially using DevOps, take time and often follow a strategic roadmap to see truly transformational results.

## Step 1

### Commit to a philosophy

Going cloud-native and adopting DevOps are both philosophies and practices of collaboration and integration intended to continuously deliver fast development results to achieve business goals. More a journey than a destination, DevOps helps the business effectively combine people (culture), process (methodology) and technology (automation) components. It delivers high-performance teaming and execution among developers, QA, IT operations, project management, release and change management and the business.

Companies seeking to become cloud-native must be able to deliver custom-written software into production more frequently—weekly, if not daily or multiple times a day—to reap the benefits of agility. Hence the need for DevOps thinking.

DevOps requires a focus on microservices to enable independent development, testing, deployment and scaling. The operations team has to focus on building a self-service, agile infrastructure. And this approach is not just for new, digital apps, but it can also be used to break apart big monolithic legacy applications.

Data also has to be refactored and decomposed to align with a service pattern. Finally, app-to-app integration needs to be driven toward choreography versus orchestration. Choosing the right enterprise services buses becomes key as they may be sources of tight coupling.

## Step 2

### Secure commitment

If organizations want to achieve agility, it's crucial to have broad resolve and a commitment to change and continual process improvement.

This is especially true across key stakeholders—including developers, QA/testing, release and change management, operations, project managers, application heads and C-level executives such as the CTO and CIO. This level of commitment is often easiest to achieve when it's driven by a current pain or threat or when it effectively complements a strong vision or sponsorship from the top.

Critical Success Factor: There should be shared objectives and colocation whenever possible to facilitate collaboration and innovation. Leadership must think in nonfunctional terms right from the start, with a focus on business outcomes. Doing so helps everyone involved see business, applications and infrastructure together.



## Step 3

### Embrace proven tools and technologies

Pivotal Cloud Foundry (PCF), from Pivotal Software, is a dynamic platform that supports numerous languages and DevOps operation modes. PCF enables IT to develop, deploy, scale and manage software delivery more quickly and reliably. It provides developers an abstraction of the technology and infrastructure stack supporting an application and its data. It supports private- and public-cloud-based as well as on-premises environments, allowing flexibility in choosing the right hosting environment and further hybridization of applications.

The alternative is to build it yourself, and then you have two problems: Beyond building and maintaining your platform, you still have to work on the original application.

Taking a DIY approach ultimately monopolizes resources you could otherwise devote to application development. And doesn't IT best serve the business when it focuses on the enterprise's core, not on supporting technology? Devoting efforts to the core transaction is where the business can differentiate itself from its competition.

When an organization embraces a cloud-native approach to development while leveraging DevOps principles, it facilitates and enables a transformation of the entire process by effectively breaking down silos. This approach essentially forces the creation of a true team environment in which it's significantly easier to collaborate and seamlessly integrate cross-functionality to achieve business and IT goals.

It also allows for the easier adoption of more-fine-grained, decoupled application architectures and development through microservices, which further accelerates the organization's ability to deliver increasing reusability and app portability.

Taking a cloud-native approach to DevOps also enables high quality and quicker velocity for delivering features, from concept to production, where those features can bring value to the business.

PCF supports developers with multiple languages (Java, Ruby, Python, Node.js, Go, PHP and .NET), frameworks (such as Spring) and middleware (such as MySQL, RabbitMQ, Redis and mobile notifications).

And PCF comes with operational capabilities baked in. These capabilities—including logging, role-based authentication and authorization, load balancers, autoscaling and governance

controls—give the ops side of DevOps everything needed for managing applications in production.

Critical Success Factor: It's important to view DevOps as a continuous loop and not as a one-way street. In considering tools, the critical success factors include modular, open, service-oriented, standardized environments. The goal should be to automate and integrate while seamlessly connecting with the build-release-run-repeat process.

## Step 4

### Invest in building the culture

Beyond technology and development, a crucial component looms on the road to agility: addressing the cultural change necessary to break down traditional silos and empower DevOps. Culture plays such a critical role that Gartner estimates that by 2018, 90 percent of I&O organizations attempting to use DevOps without specifically addressing their cultural foundations will fail.

Organizations need to enable disparate teams to work together, using continuous integration with continuous delivery to push their code into production. This enables these cross-functional teams to focus on capabilities and functionality rather than being siloed into thinking in terms of the tiers of a service-oriented architecture.

Likewise, organizations must recognize that DevOps is not just an automation-led transformation. Instead, it requires people, process and technology changes.

## Step 5

### Build the muscle

Implementation requires building on small wins by methodically developing and growing organizational muscle to achieve agility.

Start with one or two workloads, and build momentum from the bottom up. As with other key technology trend disruptors such as mobility, organizations trying to transform based purely on technology adoption fail to gain business agility.

# Adjusting Speed Along the Road (Moving Fast, Faster, Fastest)

As an organization takes the road to agility, it typically finds that agility can take many forms. Enterprises should be cognizant of not moving faster than the environment allows. It's important to understand when to move fast, faster and fastest.



## When to Go Fast

Think about which applications typically require monthly or bimonthly releases. For example, they often include most legacy applications and systems of record—ones that don't have to change in a matter of days or weeks—that have used waterfall development methodology.

Here the focus is on implementing a base level of DevOps. To achieve a fast speed, the key capabilities are strong virtual teams, automation for build and deployment and foundational application and environment monitoring.



## When to Go Faster

A good example of the need for faster speed is for digital and customer-facing assets. With such workloads, Agile methodology is typically used and is often constrained by downstream testing and release management activities that operate in waterfall mode.

The focus is on near-real-time app development with automated or one-click deployments to unclog bottlenecks and enable faster speeds with continuous integration, testing and delivery. And more-comprehensive application behavior and performance feedback are available. At this speed, deployments can be carried out within weeks—or even days.



## When to Go Fastest

If an application is the company's key competitive weapon, fastest is the desired speed. Also to be considered are those apps that drive the business and/or the customer experience. For example, mobile apps, because of their nature and ability to deploy directly to users, should be in the fastest mode.

This mode is where cloud-native organizations thrive and begin operating like a Netflix, Uber or Airbnb, with the ability to perform thousands of deployments a day. And here is where DevOps principles and PCF come together.

## CRITICAL SUCCESS FACTOR

**Digital apps don't operate at the same speed as the systems of record and middleware systems that support the apps. Clearly, it doesn't make sense to put systems of record on PCF, because there is too much to factor in. Automation partially addresses this issue—even if it's on a base level for systems of record or legacy systems. In addition, organizations should study slower legacy environments to identify natural themes or aspects where it may make sense to create and deploy microservices.**

# What to Avoid on the Road to Agility

## 1. Don't retain the status quo.

For the most part, organizations do not change their operations model enough. Instead of focusing on changing how they think about doing software, they think in terms of optimizing their current methods.

Too often IT has been rewarded for thinking about optimizing the status quo. For example, over the past decade, the focus has been on optimization through virtualization to increase capacity usage, save on investments and improve management capabilities without having to really change what happens “up the stack” in the application layer or how IT operates.

However, transforming into a cloud-native enterprise involves realizing that technology is the easiest part of the transformation. This is especially true if the organization embraces an all-in-one platform such as PCF.

## 2. Avoid people and mission inhibitors.

The hardest part will be changing company culture to fully take advantage of what these new tools can do to speed up the software release cycle. For instance, organizations must be ready to think through what new changes and experiments they would like to put in production each week to make the customer experience better.

In fact, each individual area within the organization can either contribute to DevOps success or act as inhibitors. For instance, IT leadership can prove detrimental if it:

- Lacks a clear vision
- Assumes that DevOps is something the organization can buy
- Fails to provide adequate room for change
- Neglects to engage with other business units
- Does not recognize DevOps as a journey

On the other hand, if IT leaders develop and communicate a clear mission, it can pave the way to success by ensuring that everyone understands the end goal.



## 3. Don't dive in without counting heads.

At the business level, not giving cross-functional teams a seat at the table would be a mistake. Include everyone from governance boards to test groups that represent business, security, governance, legal, regulatory, development and operations. Having everyone involved will help when demonstrating achievements and enhancements, especially as they align with key business goals.

From a development perspective, the commitment needs to be consistent. For instance, simply hiring a DevOps engineer isn't sufficient. After all, DevOps is not a new role; it is a way of getting quality, reliable software to market faster.

Yet at the same time, diving headfirst into DevOps everywhere all at once can prove disastrous. Development will end up with decreased productivity and every team taking its own approach. This lessens the opportunity for steady learning and growth. Start with one workload, drive DevOps deep, analyze it and correct it—and then do it again.

## 4. Don't neglect the operations side.

DevOps requires new processes, new tools, new governance, new approvals and a noticeable change in control management. Operations cannot embrace DevOps without changing how it currently does business.

Often development and operations struggle to get along. DevOps is getting both sides together to understand each other and start solving problems as one cohesive unit. This happens only when each group begins to understand and appreciate the other's needs and challenges. If operations wants development and QA to understand the impact of the code it builds and tests, it will need to provide a view into what is running and what is occurring in production. This will help decrease feedback cycles and increase transparency.

## 5. Don't go it alone.

IT leadership needs to recognize that teams cannot transform if they are too busy running the business. Building a center of excellence and leveraging consultancy assistance can ease the path to change.

Technology can be the culprit if the organization fails to embrace dynamically provisioned environments such as Pivotal Cloud Foundry. Technology partners such as Capgemini and Pivotal can prove instrumental here by leveraging their expertise in assessing potential workloads, developing and performing detailed proof-of-concept architectures and solutions and building a well-defined roadmap to IT agility.

# The Importance of Traveling Partners

Every organization's journey toward becoming a digital, cloud-native enterprise is different. And being able to effectively determine the best speed of operation—whether to go fast, faster or fastest—can be intimidating.

This is where the expertise of partners such as Capgemini and Pivotal comes in, by offering organizations a unified approach focused on transformation and enabling IT agility.

Capgemini has the experience and expertise needed to help organizations reach DevOps success. It understands that DevOps is a journey that starts with a few workloads focused on building a solid foundation capable of scaling and extending to handle additional workloads quickly and efficiently. It's this approach that

ultimately facilitates unconfining standardization. DevOps thrives on its ability to continuously improve and encourage learning, sharing and adapting throughout the organization.

Capgemini brings end-to-end capabilities to help clients with their IT agility journey, from assessment, strategy, proof-of-concept, cloud-native architecting and DevOps implementation expertise to helping design, implement and run a transformed operating model.

Pivotal's powerful cloud-native platform, Pivotal Cloud Foundry, continues and supports the DevOps transformation. Pivotal has a host of agile development experts in Pivotal Labs who regularly integrate into existing environments to serve as mentors to your IT team throughout the product design process. ♠



## About Capgemini

---

Now with 180,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.573 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Contact us at:

[infra.global@capgemini.com](mailto:infra.global@capgemini.com)