Google Cloud

CIO Guide to Application Modernization

May 2020

What You Need To Know

The global pandemic has put unexpected pressures on businesses of all sorts — in ways no one was projecting at the beginning of the year. As a result, CIOs face a series of urgent challenges:

- How can they raise system visibility and system control over operations that are more dispersed and changing than ever?
- How can they cut costs, yet create a more agile and responsive IT system?
- How can they do more with older data, even as they understand better the data from a market that is changing every week?
- How can they help people work faster, with a minimum of change management, or set the stage for growth, while preserving capital?

In many cases the answer is a step-by-step deployment of cloud computing technology, tailored to meet the most pressing needs first.

Working with a comprehensive cloud provider, it is possible to create cloud systems that respect and preserve core assets, while enabling rapid modernization, in particular for the cost-aware agility and resilience of modern application architecture.

Why You Should Keep Reading

This guide offers a series of approaches to application modernization, from identifying needs and developing an action-oriented roadmap, to methods of identifying and effecting meaningful change in the most critical parts of your IT operations.

We have also included at the end a list of key solutions that Google Cloud and our technology partners have to give your organization fast results.

Introduction

Even before the current crisis, IT organizations saw pressure to be more agile and innovative. Customer demographics and expectations are changing. Competition is emerging faster and from unexpected sources. Business models are being reinvented. Digital technology was at the heart of many of these challenges, and its adoption was key to every company's response.

Similarly, some of the old challenges to change are still around. In particular, legacy IT systems inhibit change while consuming budgets. Although CIOs have successfully migrated some applications to the cloud, according to a McKinsey study, around 80 percent of them report that they have not achieved the agility or business outcomes they sought from application modernization.

Today's crisis has changed none of those things, but added a new urgency. In areas such as workforce reorganization towards working from home, ecommerce, and online logistics, we have seen how rapidly digital technologies can be deployed. Leaders are seeking ways to take this faster pace to other parts of their organization.

Frequently, plans include rapid adoption of public cloud computing technologies. Public cloud technology can reduce infrastructure costs and management overhead by consuming the provider's data center assets as an operating expense, and gaining a single view of that consumption. It is easy to scale up and scale down consumption as well. New technologies, as well as security upgrades, can be adopted in real time, as dedicated cloud providers offer new innovations. It's cheaper, it's flexible, and it's faster.

As these improvements, along with techniques like "serverless" deployment, are automated, customers are able to spend less on operations, and more on innovation. That's a better use of everyone's most valuable resources, people and time.

For all the ways the world seems different, however, organizational change can still be a challenge. Different business stakeholders still have contending needs. Critical operations can't be interrupted. No successful transformation is "one size fits all," nor does big change it happen overnight.

There are clear steps to take. Working with a cloud provider, you should:

- Assess your portfolio
- Figure out ways to drive new revenues by reusing existing services
- Develop your cloud architecture and modernization roadmap
- Establish a success-oriented modernization process
- Support your staff during and after the modernization
- Ensure efficient optimization for the long term

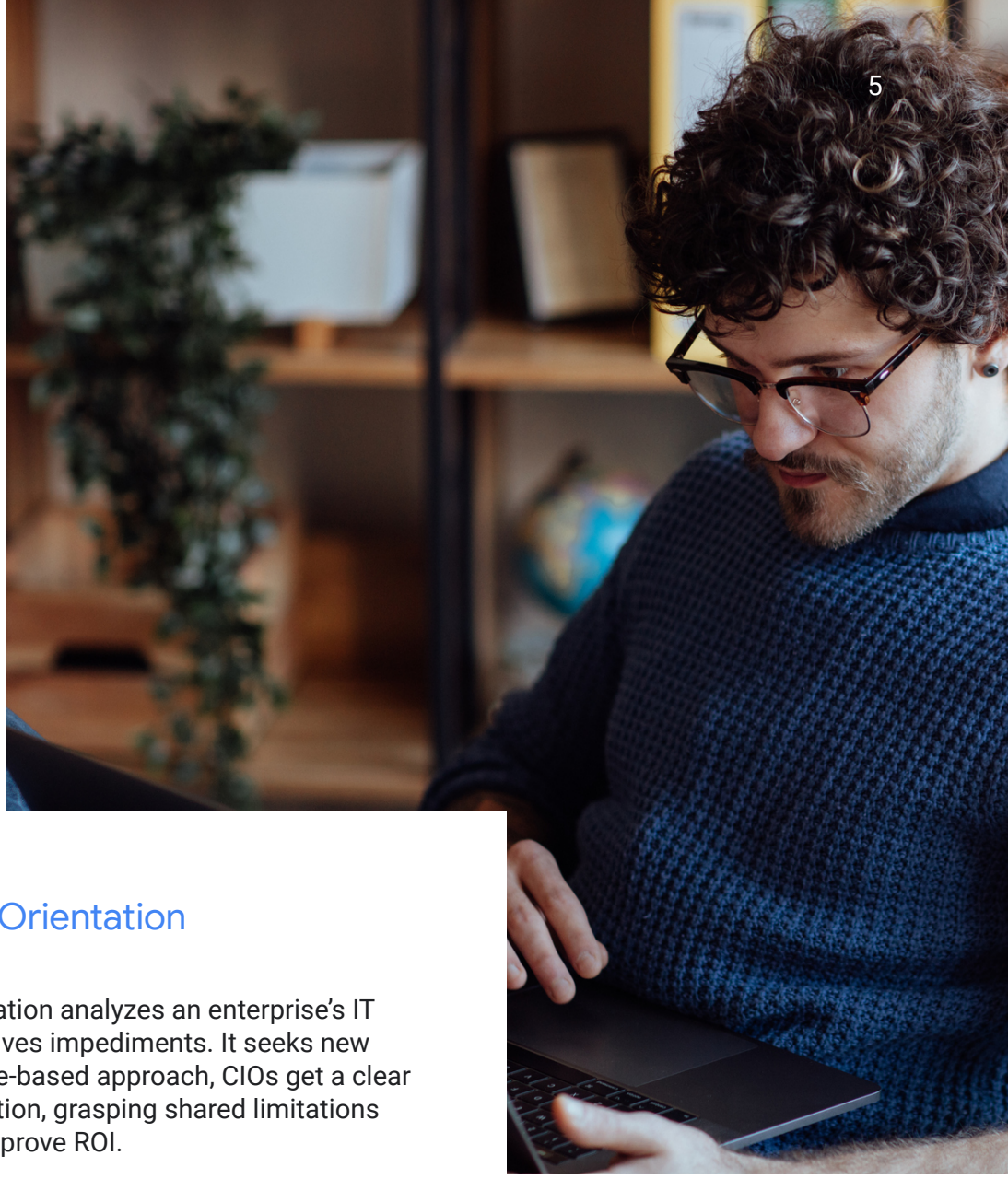
Starting Out

Most application modernization starts with application rationalization. Streamlining the existing application portfolio improves efficiency, reduces complexity, and lowers TCO.

The first thing to do is to inventory applications, identifying the core applications and their owners. Next, both the usage and lifecycle of these applications must be clearly understood.

The next step is application rationalization. There are many methods for this like the Gartner TIME Model, Forrester Wave Methodology. These methods were developed back when older IT technologies were the state of the art, however, and imposing an IT-centric vision for change that lacks a clear strategy for prioritization and alignment within larger organizational objectives. Results are too often slow, painful, and adversarial to other stakeholders.

With modern digital technologies, IT can be seen as a series of flexible services, architected so that different aspects can be adjusted without changing everything, while solving critical enterprise needs by organizing applications as a series of capabilities.

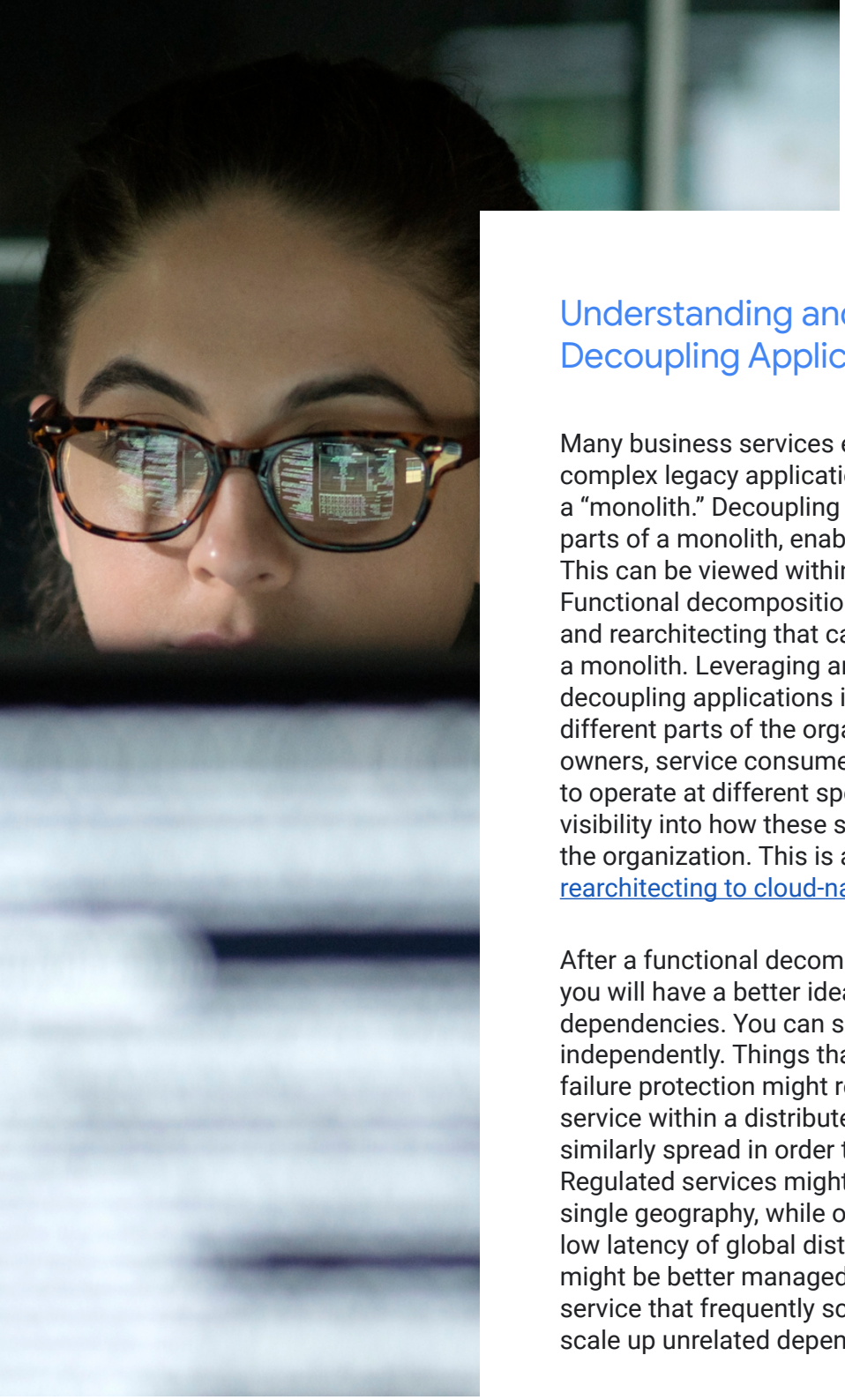


Business Service Orientation

Business Service orientation analyzes an enterprise's IT needs, identifies and solves impediments. It seeks new efficiencies. With a value-based approach, CIOs get a clear roadmap for modernization, grasping shared limitations that when solved will improve ROI.

It begins with thinking about Business Capabilities, from externally-facing customer features to internal tools like security, and data analysis. For example, Order Management is a Business Capability fulfilled by various Business Services, including Order Creation, Order Query, Order Updates, etc. Many times, capabilities might be running on a variety of platforms and in various environments, including different cloud services. Some were organically adopted by core IT, others by departments, or inherited through corporate acquisitions.

Ideally all Capabilities and Services should be identified and described, as a means to examining budgets, and as a basis for action. They can then be better managed as decoupled applications.



Understanding and Decoupling Applications

Many business services exist within a large and complex legacy application, sometimes referred to as a “monolith.” Decoupling is the process of extracting parts of a monolith, enabling them to run on their own. This can be viewed within the context of business need. Functional decomposition is one method of refactoring and rearchitecting that can be used to break apart such a monolith. Leveraging an API based approach for decoupling applications into business services, allows different parts of the organization including the service owners, service consumers, other business teams to operate at different speeds. It also provides clear visibility into how these services are being used across the organization. This is a part of a larger process of rearchitecting to cloud-native.

After a functional decomposition exercise is complete, you will have a better idea of your components and their dependencies. You can scale and partition these workloads independently. Things that need high availability and failure protection might require multiple replicas of a single service within a distributed system. Backends should be similarly spread in order to survive coordinated failure. Regulated services might need to keep their data within a single geography, while other services may benefit from the low latency of global distribution. Cost and performance might be better managed by decoupling a dependency of a service that frequently scales up, so there’s no need to also scale up unrelated dependencies and services.

Once you have an idea of the underlying domains, you can choose to decouple services as it makes sense. A single eCommerce monolith might contain customer authentication, account number lookup, customer rewards program, award loyalty points, product promotion system, customer purchase pattern recognizer, etc. Breaking these out as separate, interacting considerations, teams can map the various dependencies and failure domains. This enables choices about how to divide the logical services and dependencies, or which to keep coupled.

The Process of Modernization

Platform Design

With current applications and services fully identified, you can plan for a software delivery environment whose customers are other internal teams and stakeholders. Decoupling has done much of the work of making IT an internal business. Focus on a strong and sustainable outcome through areas like failover, business continuity, automation, security and scalability.

Your current applications and services can limit initial performance, particularly if applications have not been containerized, or manual processes haven't been automated. New technologies or processes can be introduced to grow these capabilities. Sometimes they directly replace limiting functions, or else they run in parallel with the old system.

It's important to have baseline metrics for measuring improvement. In thinking about modernizing and delivering software, there are well-understood metrics, which fall within three categories:

Speed

How long does it take to get code into production? How often do we push out changes?

Stability

How often do changes result in failure? How quickly do we recover?

Reliability

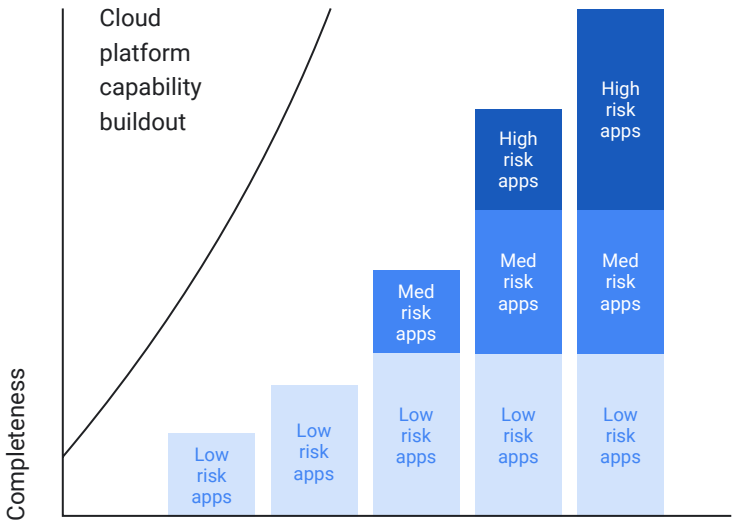
Are services available for customers? How well do we learn from outages?

These are the initial standard by which you can judge new capabilities to your platform, or the improvements to software technology, processes, or people.

Building Your Platform

In moving from the current state to your goal, think in terms of the order of the value you're getting. Every new capability provides new benefits or unlocks applications and services to be modernized. Some, like low-availability batch services, won't need as many resources. Listing out modernization needs in a prioritized order of costs and opportunities provides your team a concrete list of tasks to accomplish, along with clear success criteria and demonstrable impact.

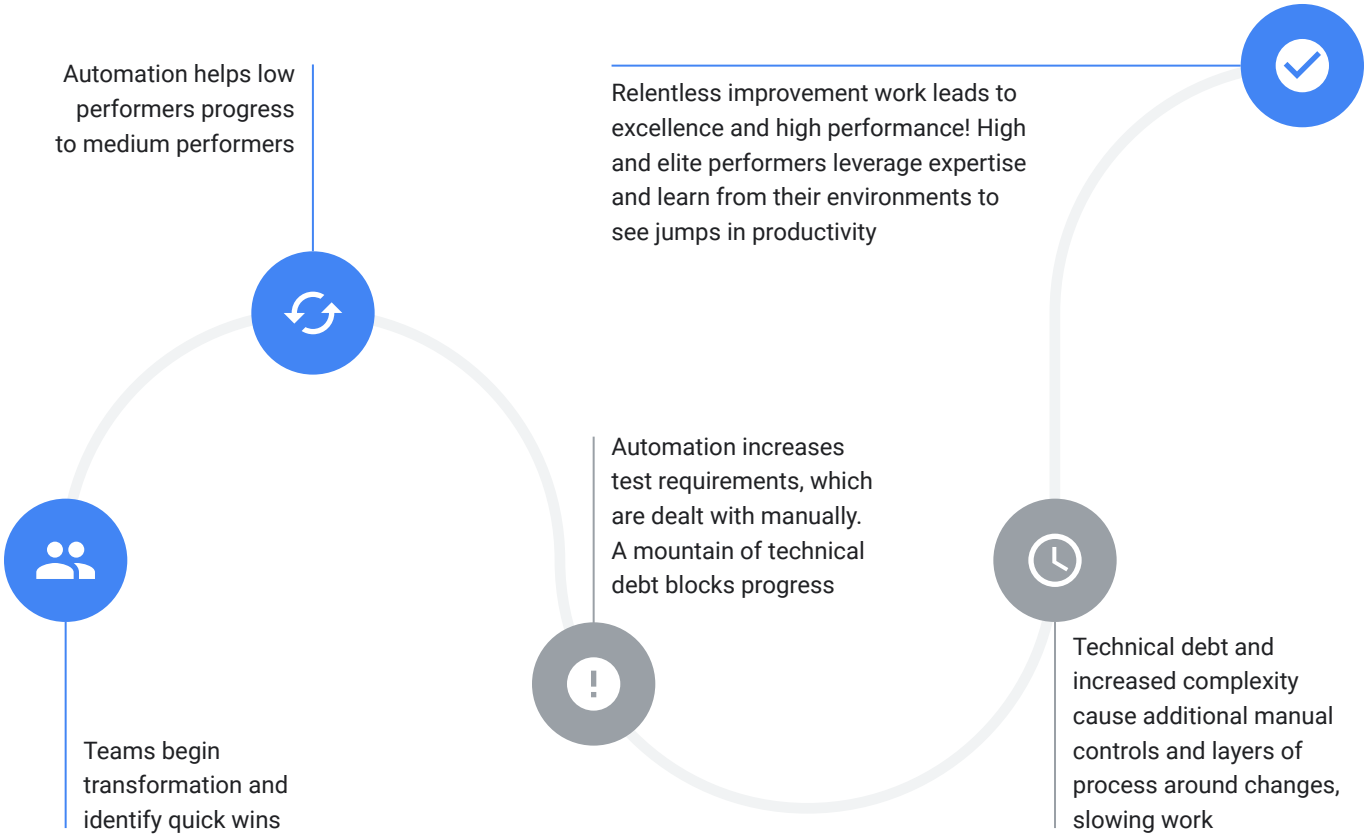
Migrations can also proceed based on factors like risk, usage, and urgency of need. Those with the least operational risk are a good place to begin learning, with little impact from failure, roll back, and re-trying. It's clearly good to work up to the most impactful service, based on things like utilization and business importance. Risks to consider include direct customer impact, loss of revenue, or negative brand impact.



App migrations over time

The flexible and modular approach of modern architecture makes it easy to introduce new capabilities at the rate you need, even having many efforts working in parallel. That helps with learning capabilities in a low-risk way, then migrating to higher-risk services. Teams delivering modernization, either from IT engineers or in concert with product-owning teams, should study service adaptation methodologies such as Pace Layering or a Proxy Pattern.

Another benefit, compared with previous IT transformations, is that short- and medium-term gains can pay dividends even as the platform works toward realizing the benefit of long-term investment. Google's DevOps Research and Assessment[9] (DORA) calls this "The J-Curve of Transformation."



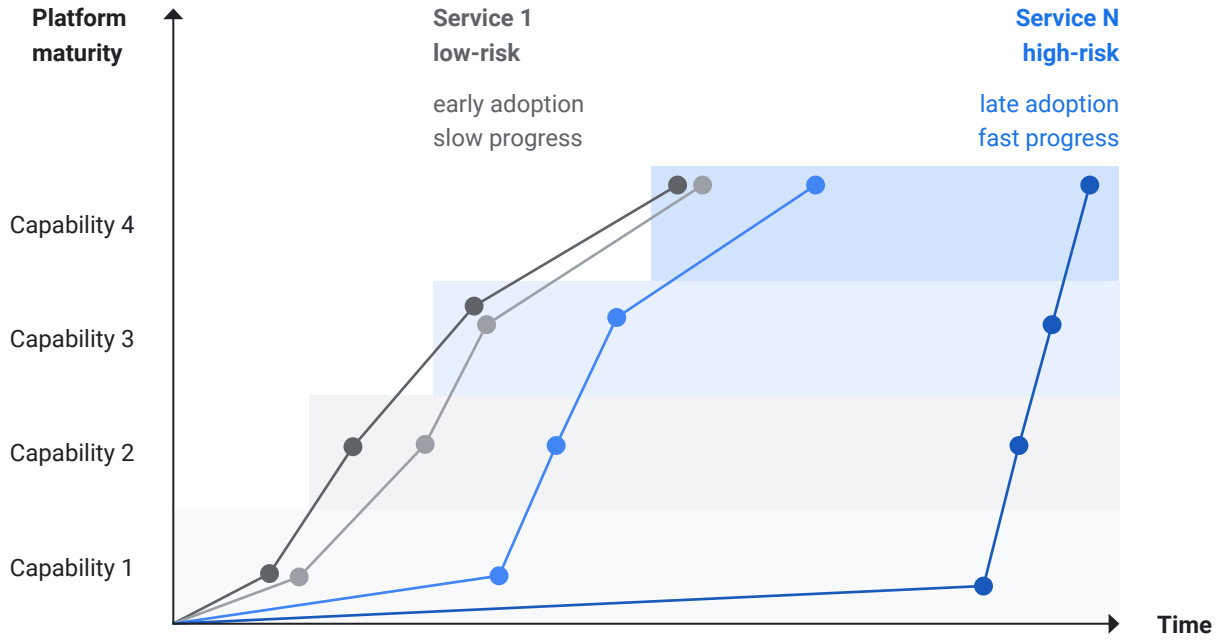
DORA also identifies several technical, process and cultural capabilities that lead to more performant software delivery and operations, summarized here:

Planning	Development and Test	Release and Deploy	Monitor and Optimize
Small batch size	Agile development	Standardized platform and process	Near-real-time feedback loop
Dedicated cross-functional teams	Continuous integration	Automated environment provisioning (IaC)	Loose architecture coupling between applications (APIs)
Continuous planning	Test automation	Automated release and deploy (toolchain)	Metric & reliability

These capabilities allow teams to more effectively deliver software within a team. Part of that software delivery is the development of your platform. These teams can then define and implement platform capabilities to be adopted over time, allowing your teams to use more effective processes and technology.

Using your Platform

As lower-risk services go into a Business Capability, you affect multiple capabilities at once.



As an example, consider a hypothetical company whose current platform looks like:

- VMs and SANs in a single on-prem colo, using VMWare.
- Windows servers running a single monolith web application, deployed weekly during a maintenance window.
- Large SQL Server databases exposed through web services and APIs

The company first identifies an idealized target environment to migrate onto. In this case, they choose:

- Containers in GKE in multiple global regions, using Anthos for Service Mesh
- A managed datastore: Cloud Datastore
- A managed relational database: Cloud SQL

In order to migrate, they introduce the following capabilities, following a pattern of “deploy a small instance or single service, then add more as you gain operational comfort”:

Limitation Identified	Capability or Technology Introduced	Benefits Gained	First Steps
Development environment and production are too different, resulting in bugs	Ability to “containerize” services	Development, staging, production environments become more predictable	Containerize one service and deploy, gain confidence in tools, procedures Develop a plan to containerize all services over time
Manual release process is error-prone and slow	Simple CI/CD pipeline for containers	Automated testing and deployment	First basic CI, then CD apply to services over time add more steps to CI and CD over time
Running many containers in current VMs is manual and risky	Introduce a single GKE cluster on GCP	Services can be orchestrated and easily autoscaled	Migrate one service onto GKE Add more over time
Can’t observe new distributed system or apply policy consistently without manual toil	Add Anthos Service Mesh and Anthos Config Management	Dynamic service discovery, config as code, service-level telemetry	Once several services are in GKE, add Service Mesh to observe traffic and deploy policy Add more policies over time
Need to avoid business disruptions to existing consumers of some services	Introduce API Management (Apigee)	Context aware routing, payload transformations, consumer specific API Facades	Per service, route less critical consumers to the modern version of the service and get feedback Route rest of the consumers in a controlled manner

Limitation Identified	Capability or Technology Introduced	Benefits Gained	First Steps
Some services must be within 1ms of legacy databases Some services require use of on-prem secure hardware	Introduce a GKE on-prem cluster	Ability to deploy within the same network as legacy services and data	Spin up a GKE cluster on-prem, deploy an existing on-prem service there Add more services, clusters with time
Customers in other regions are getting poor performance	Introduce a second GKE cluster on GCP	Regional resilience and lower latency to global audience	Adopt a multi cluster operational model: adjust deployments, incident response, load balancing, etc.
Our large amount of non-relational data is in an expensive relational database	Migrate initial data to managed datastore	Scaling becomes very simple	Migrate one service's data to datastore, deploy new code accordingly
Our relational database is expensive and hard to maintain and scale	Migrate remaining data to managed relational database	Licensing cost drops to zero	Migrate one shard, tenant or customer account to cloud relational database, repeat
We expose some commercial off the shelf applications which have maintenance and consumption complexities	Introduce vendor-agnostic endpoints via APIs, then move over the code into GCP via some method like Velostrata, Migrate for Anthos, etc.	Stop leaking vendor specific interfaces to consumers.	Create vendor-agnostic API endpoints and move the application to cloud
Now that most of our services are in the cloud, keeping the any on-prem is a costly, less-secure, less flexible	Migrate risky, data-heavy apps to GCP	Shut down on prem colo	Validate business continuity plans and data recovery do not depend on colo, remove all traffic but keep "cold" for some time as you gain confidence

By following this model, this company is able to **progressively add capabilities** to its software delivery and operations platform, adding capabilities to their business along the way. Each component added either allows more applications to be modernized, or delivers new value to those that have been modernized already.

Any kind of IT transformation activity has to address three aspects - People, Process and Technology. We have addressed the Business Service Orientation process earlier. But without the right People structure in place, any process is bound to fail. So let's look at that next.

The Product Delivery Model

Most IT organizations deliver software through a Project-Based Model, that focuses on department goals like time and schedule, but often ignores business needs. Success is often better described using a Product Delivery Model.

In Product delivery models, teams own the entire lifecycle of software- definition, delivery and support. They also determine the lifecycle of major/minor releases, new features, deprecation

The funding of Product teams is not limited to the project but on long term strategic value. Product teams are driven by wider stakeholders, internal or external, depending on who is using the product. This facilitates iterating and identifying new capabilities for success. There is better alignment between business objectives and product releases.

In this model - teams are organized by capabilities. These include two kinds of teams broadly:

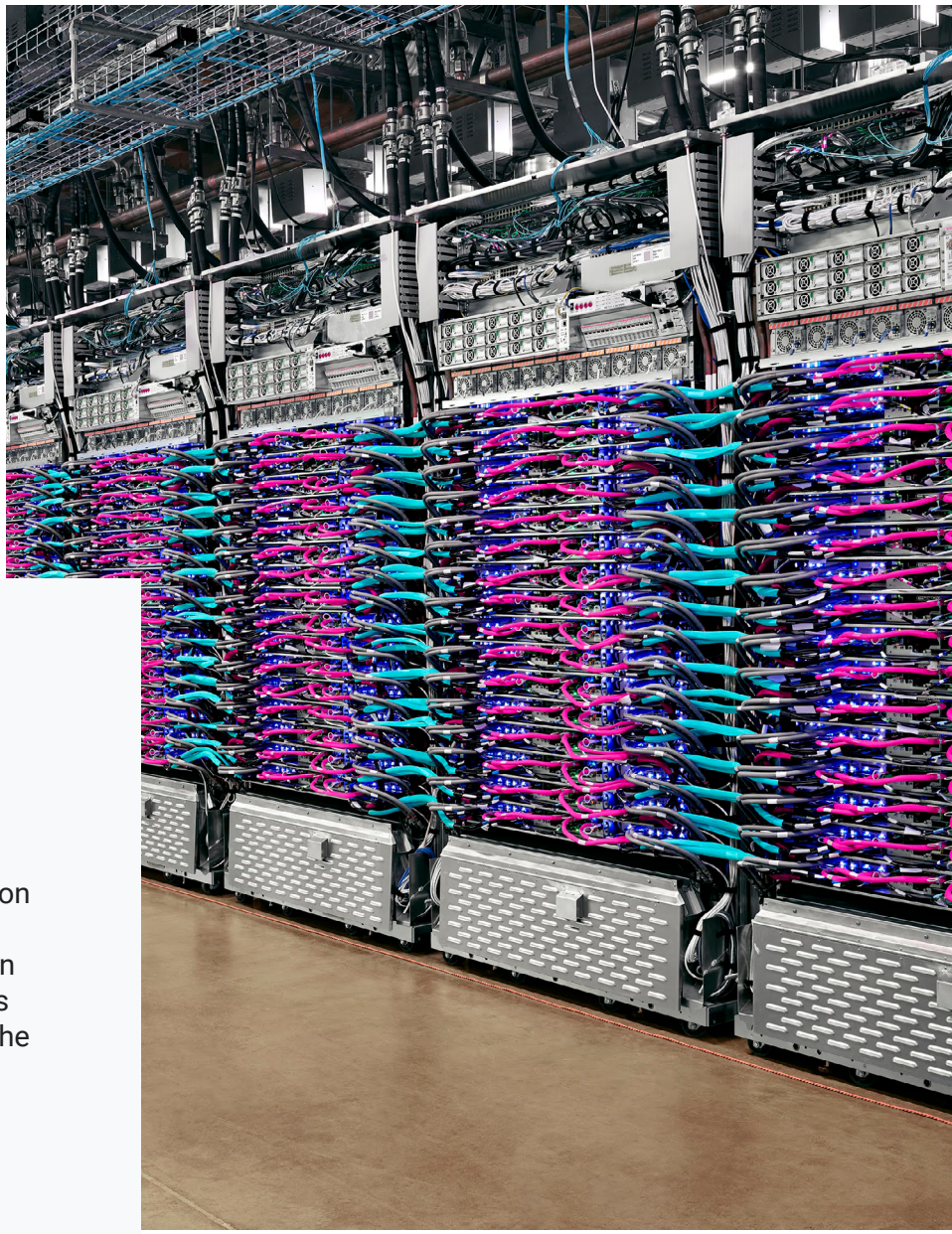
Business Capability Teams

Are aligned closely with business stakeholders and are focused on delivering business services. Their service delivery and feature roadmaps follow the same trajectory of the respective business units. These teams focus on servicing needs for both external and internal customers. In the above example these include the teams focused on Order Management, Fulfillment, Mobile Apps etc.

Platform Capability Teams

Are more applicable across business domains. For example, the Data Analytics team is responsible for maintaining Data Warehouse, Data Processing Pipelines etc. They are not responsible for attending to broken builds of higher tier teams. Thus, the higher tiers may have one or more dependencies on lower tiers.

Both these kinds of teams operate in product mode in the sense that they are focused on outcomes, long lived and manage the entire lifecycle of plan, build and maintain. The lower tier teams treat the higher tier teams as their customers in providing services required by those teams. This allows for a decoupled operating model while driving towards common objectives. For a more detailed discussion on the product vs. project delivery model, please refer to this.



Building on Success

In the previous sections we have defined how to orient your organization's process and people to drive business value. Understanding an organization's application landscape in the context of business function and priorities is an integral step in the modernization journey. Aligning teams to operate in a product delivery model is the vehicle to enable this journey.

Both of these aspects may also require change management among your teams. That way these people and process changes can become a continuous process of improvement and refinement. Over the course of time, the definition of what constitutes a business function in the enterprise will change based on factors like market conditions, product strategy, competition etc. The composition of your teams will also have to change accordingly. But by having this transformation approach as part of your operating model, you ensure that your resources are focused on maximising business value.

How Google Cloud Can Help

Modernization should not be a solo exercise. Your Google Cloud team, along with our trusted partners, is here to help. We take pride in the strong partnerships we build with our customers as well as providing solutions that meet their needs at every point in their journey.

Whether you are an enterprise trying to untangle the challenges of a legacy Java environment or looking to adopt modern development principles, we can help you with all the three facets of your transformation - People, Process and Technology.

As you encounter parts of your system that needs modernization, you don't have to solve every problem on your own. Google Cloud provides a host of solutions for addressing various challenges that face enterprises. Below is a brief list of them.

Technology Objective	Solution Details
Run your critical business applications on our secure global infrastructure	SAP on Google Cloud
Run Windows workloads on a secure and cost-effective cloud environment	Windows on Google Cloud
Migrate and run your VMware workloads natively on Google Cloud	VMware as a service
Providing hardware to run specialized workloads with low latency on Google Cloud	Bare Metal Solution
Migrate VMs to quickly get started in Google Cloud	VM migration
Extend legacy applications and modernize alongside new cloud services	Modernize legacy applications
Enable organizations to incrementally modernize their existing applications by migrating to containers and reduce operational overhead and licensing costs	Modernize Java Applications with Anthos
Get innovative software into the hands of your customers faster than ever with automated tools and expert guidance from Google Cloud	Modern CI/CD with Anthos
Attract and empower an ecosystem of developers and partners	New business channels using APIs

You will also find that Google Cloud has several unique attributes which we believe set us apart.

Reliability and Availability

Google Cloud's infrastructure was built to serve billions of users who rely on several Google services every day. Google Cloud makes this same world class infrastructure available to our customers so that they can get the same reliability and availability.

Open Platform

Most organizations looking to modernize their IT landscape, wish to avoid lock-ins and leverage open source platforms. Due to this, Kubernetes has become the platform of choice for application modernization. Google is the leading contributor to the Kubernetes project. Through Anthos, we provide a managed platform for Enterprises to adopt Kubernetes.

Greater Flexibility

Google Cloud's Anthos platform provides the ability to build and manage applications whether its On-Premise, Google Cloud or Other Cloud providers. This allows customers to build their modernization strategy in a hybrid/multi-cloud manner with minimal overhead. Also the same Google Cloud tools can be used for both on-prem and cloud infrastructure which minimizes the operational overhead during the transition. Leveraging Anthos as your App Modernization platform has shown to have an ROI of upto 4.8x.

Secure by Default

Google invests close to a billion dollars annually on building security products and related research. That same tools and approaches have been built into every layer of the infrastructure and application stack. From providing encryption by default at rest and transit to providing cloud-based HSMs to services for De-Identification of data, Google Cloud provides all the security tools and capabilities necessary to operate a modernization platform in a secure manner.

So before you venture down the path of solving problems from scratch, check out cloud.google.com/solutions or ask your Google Cloud representative. We would also like to invite you to check out our companion whitepaper - CIO Guide to Application Migration.