

MARCH 2023

Lift and Shift vs. Transformation

Evaluation frameworks and best practices

Table of Contents

Introduction	3
Discover	4
Measure Business Needs	5
Data: Blocker or Enabler to Transformation?	6
Driving Cultural Transformation	9
Practical Steps to Deliver on the Vision	11
Next Steps	12

Introduction

When moving to the cloud, you face a critical decision on how to best migrate your applications. There are two potential approaches available to you:

- **Lift and shift:** Move existing applications to run in the cloud on the same architecture and technologies you used on-premises.
 - **Transformation:** Re-architect our applications to take advantage of new cloud-native capabilities such as elasticity, redundancy, global distribution, managed services, and built-in monitoring and logging.
2. Failure to achieve the agility benefits that come from new engineering processes enabled by the cloud – instead change is only made after tickets are bounced between teams, again just like on-premises.
 3. Inability to use higher level managed cloud services that remove the need for infrastructure and operational management.

Lift and shift is often seen as an easier and more predictable path, presenting both the least business risk and resistance from inside the organization. It can be the right path in some circumstances. But for critical applications, it rarely moves the business forward. You will be unable to fully exploit new cloud capabilities that enable your business to build, test, learn, and adapt.

It is rarely cost effective to simply lift and shift. In speaking with many customers, we have learned that costs can increase between 10% and 100% (or higher) when all they do is lift and shift to the cloud. There are three reasons for this cost escalation:

1. Existing applications are often not automated or elastic. As a result, they end up overprovisioned to accommodate high seasonal loads or transient usage spikes – just as they were on-premises.

The reality we all face is that every application is different, and so there is no simple or single right answer to choosing lift and shift versus transformation. In some cases, lift and shift can be the right first step, helping teams gain familiarity with operating in the cloud, before then embarking on a fuller transformation as they see everything the cloud has to offer. However this can also be a risk if your teams believe they are “done” in your cloud journey, and don’t then progress beyond that first step.

There are best practices that can help prioritize your efforts and mobilize your teams. By working with over 39,000 customers, including more than 50% of the Fortune 100, we have built evaluation frameworks that can be used to navigate the right path for your business.

“It’s your responsibility as the leader of your organization’s change initiative to find out all the right data about your current application estate and plan for the right outcomes, which could be years in the future. If you don’t do it, nobody else in your organization will. And your competitors are not standing still either.”



Discover

The first step on your cloud journey is to take an inventory of your application portfolio. This means diving into the details for every application. When we work with customers, we have them create tables and checklists that catalog each application. A simplified example of discovery might look like the following:

Application Name	mkt-expmt	Facilities Tracker	Payroll Analytics
Owner (Dept, Name, Cost Center, etc)	Joe, Mkt-001	TBD, facilities	Sue, Payroll, PYR-001
Current HW	4u HP 590e, 16gb ram, 4tb	Unknown host under a desk?	4 node cluster
Knowledge at Company	Yes	No source code, or people	Yes
Plan for Future (Stable, Expanding, Deprecated, etc)	Use new Experimentation Platform in 2021	Rewrite for new Work-from-Home	Stable, then deprecated
Cost (HW, Maintenance, etc)	\$3K	TBD	\$15K
Underlying Systems	SQL Server License expires at end of 2020	TBD	Yes, Oracle 12c, exp 2022
Budget to Move, and Amount Budgeted, if Any	\$5K 2020, \$25K 2021	Yes, part of COVID re-arch, \$50K	None
Priority to Move	3	1	2
Decision on Move Kind, and Date?	Move App in 2020, re-architect in 2021	Rearchitect	Lift and shift to cloud compute and storage at minimal cost

Once your discovery is complete, weigh the potential business impacts of moving each application to the cloud against the effort of actually moving it. To help make the determination, keep the following considerations in mind:

- Business impact is the most common pressure for change, driven by factors such as faster cycle times, improved customer reach, geo-resilience, and impending data center retirement. Normally, this is the top factor in prioritizing cloud migration.
- Feasibility is a function of both regulatory controls governing your market and technical constraints. The latter are especially important in the ordering and prioritization of your move. These include interdependencies between different applications, monolithic systems that need to be decomposed, the availability of underlying source code, and the existence of staff who have knowledge of the systems.
- Financial constraints are also key considerations. Departmental budgets, hardware depreciation, and the current state of long-term contracts with legacy software vendors are all important.

“Through the discovery process you can start to identify those applications that should be prioritized for migration.”

Measure Business Needs

With the prioritization of applications complete, the next step is to identify which applications should be lifted and which should be transformed. Though it’s easy to get tied up in all the different technical, financial, and regulatory issues, it’s best to start with business agility.

To make this assessment, a valuable framework you can use is to ask your teams the questions below. Each question helps you drill down into KPIs for each application.

Answering yes to any of the questions below indicates that transformation, rather than lift and shift, is going to help you better deliver the ideal end state – one that fully exploits the power of the cloud.

Questions to ask for each application	Yes/No
Do we need to be able to build, test, and deploy new features and services in days and weeks rather than months?	
Do we need to run experiments (i.e., A/B tests) at least weekly on the production application, and provide business analysts and product managers with real-time data from those experiments?	
When building and evolving our application, do we need to be able to change our data models (e.g., “roll schema” in legacy database parlance) on demand rather than wait weeks or months to make each change?	
Do we need to be able to scale the application quickly if and when we need to, without rewriting or rearchitecting it – particularly from a single-node solution to a fleet/distributed solution?	
Do we know the monitoring and availability requirements?	
Do we need the application to recover from failure or tolerate planned maintenance without causing a service outage?	
Can we instantly roll back a change if needed?	



Data: Blocker or Enabler to Transformation?

In order to move an application, you have to move the underlying data. It's often said that "data has gravity." This means that the more complex and large the data is, the harder it is to move. Why is it hard? Here are a few reasons:

1. Data in most applications is live – meaning that there is no way to shut down the applications and freeze the data while it is being moved. This complexity, requiring snapshots and Change Data Capture (CDC) tools, can make an application move almost impossible.
2. It's expensive to move data. You may have seen some cloud vendors offering to send an articulated truck loaded with shipping containers and storage to your data center. If you can move online, data networking charges can be expensive. In addition, if you run with your current network, the sheer bandwidth required to move a live application can impact the rest of your company, sometimes requiring you to upgrade your network connection.
3. It's more than just the data. There are complex, intertwined systems and graphs of dependent applications connected to the target data that potentially need to be moved with it. This can cause you to have to group applications together in order to move, or to at least undertake an initial decomposition while on-premises prior to moving.
4. Data in its source form is often not the same as you need it to be in its destination form. This includes silos or islands of disconnected datasets living in multiple relational and NoSQL databases, data warehouses, data lakes, caches, search engines, and more.

Data size, interconnection, liveness, provenance, and regulation can make it seem almost impossible to move an application and its data. In fact, this is one of the reasons why some companies choose to completely lift and shift their entire application and data estate to the cloud unchanged.

You may be asking yourself, "Doesn't my chosen cloud vendor offer me many of these same systems in their hosted environments?"

Our answer is a rather surprising but resounding "No." After working with thousands of customers over the last decade, our opinion is that complete lift and shift is not actually possible.

Why? Your on-premises infrastructure has many assumptions built into it: monitoring and alerting frameworks, deployment tools, machine/network/storage size and speeds, etc.

There is no cloud available that completely emulates the underlying systems in your data center. Nor do any of the clouds have all the features of your bespoke on-premises systems, which you have built up over years or decades.

This leads us to an important conclusion: Lifting-and-shifting your applications and data en-masse versus transformation is a false choice. As the leader of your company's move to the cloud, you must gather the right data, make the right decisions, and actively choose the amount of transformation you can and should accept, both overall and on an application-by-application basis.

So, if it turns out that every migration is on a sliding scale between doing the least possible – resizing, rebuilding, and recertifying at one end of the scale – and re-envisioning, re-architecting, and redeploying, then the key question you face is whether it is right to just replicate that same on-premise data architecture to the cloud or whether you have an opportunity to reassess your data estate.



Is this now the time to take a more holistic, platform approach – one that enables you to elevate data to a true, cross-enterprise asset class for the business? Data itself is certainly difficult to transform at scale but it rarely turns out to be the blocker – it’s what you do with it that is central to enabling business transformation.

From our experience, there are three paths you can take when transforming your data estate. They are not mutually exclusive, but rather used in combination, based on the application analysis from the discovery and measurement phases discussed earlier.

1. **Data layer augmentation of legacy systems:**

Create a flexible Operational Data Layer (ODL) on top of your existing legacy systems, federating data to make it available to serve targeted business initiatives. These initiatives could be creating new digital channels for customer engagement, or exposing data to AI/ML services in the cloud that can automate business processes and make better predictions.

The ODL path allows your teams to innovate without disrupting existing backends, for example moving to DevOps practices without touching the existing application stack. It allows you to take a more phased approach to cloud migration by blending the initial predictability of lift and shift with a path to fuller transformation as your teams’ skills and capabilities grow, and as business cases are solidified.

This path can be appropriate for systems that are locked on-premises for some regulatory

or practical reason; locked in to a database provider with hostile, inflexible, or expensive licensing, or that are to be deprecated soon enough that re-architecting the underlying systems doesn’t make sense.

2. **Decoupling with microservices:** Most companies have large monolithic applications with complicated business logic, and those applications call into large multi-table monolithic databases with mixed data structures. To achieve business agility, the company must decompose the on-premises monoliths by reimplementing core functionality as a set of microservices in the cloud, each powered with its own datastore.

The end state of the needed re-architecture is to hide all native data access behind APIs, giving flexibility to the datastore implementation behind those APIs. This is in contrast to the “raw” data access to each table given to many different business applications in a traditional legacy application-database pair.

By decomposing your monolithic applications into independent services, each powered by its own agile and flexible database, you simplify application logic, which is the first major benefit. You have also created agility for the data so that it can be deployed, manipulated, and updated by small, largely autonomous teams. Each of these teams will then have the freedom to make their own technology choices, and the business will be able to fund each one on an ongoing basis according to the business priorities at that time.

“The intersection of technology with culture and people is where the true benefit of modern applications and modern databases takes shape.”



Many companies believe that the way to move ahead is to re-architect both the application and the database at once. Our experience is that this is practically impossible. So you have a choice: You can start by moving the data into smaller separated data stores first, and then rewrite the applications to use those data stores, or start by separating out the logic into different applications and services first.

Typically, your teams will write microservices that implement the needed business logic first, working against the entangled data in the monolith. This sets your teams up for moving out of the database monolith.

Like the ODL approach above, this helps you make stepped changes to a fuller transformation over time. It is typically used when it is viable to decouple specific business

domains into isolated service boundaries. This can be a different and more desirable approach than creating an ODL because it helps you accelerate how quickly you exploit cloud-native features while pairing down technical debt.

3. Build cloud native: This approach is best applied to net-new services that are born in the cloud, delivering completely new business functionality. Typically they harness native technologies such as serverless functions, CI/CD pipelines, and event-driven architectures.

The first two approaches described above are more suited to the modernization and transformation of existing applications. This is because these paths connect the legacy systems that provide the foundation of today's business with the pathway to what's next.

“You can envision your existing systems — discovered and measured in Steps 1 and 2 above — as a tangled, decades-old ball of yarn of systems and business requirements. Your job is to tease apart the individual threads, one by one, pull them out to make them accessible, and then snip them off so that they can be transported to newer, more modern architectures.”



Driving Cultural Transformation

Beyond technology change, we have observed across hundreds of customers that the largest blocker to transformation is cultural change.

Research has shown that close to 70% of transformation initiatives are either slowed or completely stalled. Surprisingly, a major cause of failure is the human factor. After all, when looked at dispassionately as a set of prioritized steps and processes, it appears that change management should be merely an execution challenge, albeit a complicated one. Instead, what we have observed is that transformation rarely happens as the linear process our technical training would have predicted.

Almost all transformations are delayed by setbacks such as erroneous expectations, surprises in technical complexity, or cost overruns, to name a few. In addition, the lack of alignment with stakeholders on both the likelihood of these events and how to pivot and react to them can cause the failure of the initiative.

In the face of these challenges, what we have seen successful transformational leaders do is create a vision for the future that aligns with the business, and that both empowers and enables their teams. In fact, we have seen them use a three-step approach:

1. **Business alignment:** Setting the vision is hard. It means brokering alignment across stakeholders in the business, forged around shared business goals. It means bringing your colleagues in Finance along with you. This involves getting them to accept not only that the cloud is a different cost model, but that getting there is going to take a lot of creativity.

Focusing on the value the cloud brings is key, with examples including:

- Faster cycle times for new digital initiatives that will generate new revenue streams
- Improving operational efficiencies, with costs tightly aligned to usage
- Reducing “regrettable spend” in maintaining aging data centers and dealing with pernicious enterprise software licensing audits
- Paring down technical debt accrued by on-premises infrastructure

2. **Set realistic and explicit expectations:** Your current systems, like Rome, weren’t built in a day, and they won’t be transformed in a day either. Like the remodel of a house where you have to keep living in it during the change, there will be surprises, cost overruns, and it will take time and energy from all stakeholders to succeed. That’s the first expectation we recommend you set.

The second expectation you should consider setting is around the product vision and roadmap for your company. You certainly already have a long-term product vision for the company, a vision that is unchanged by IT’s decision to move the company’s infrastructure to the cloud.

However, it is likely that as a transformational leader, you will have to negotiate with the business to delay or modify parts of that product vision for the short term, at least in order to have a more agile company and product vision for the future.

“The respect and support of the entire executive team is essential for you to be able to transform the technical underpinnings of your company. It is better that you get their buy-in that it’s as complicated and prone to pivots and replanning as any other major business transformation — and that you get that buy-in before the first challenges appear.”



Those are the expectations you should consider setting outside of tech, but you also need to set expectations inside of the IT and technical organization. You need to make sure your teams understand that in order to succeed, all processes and tools are fair game for change.

In particular, it's important to ask for, and expect, suggestions for modern, service-based, agile systems to change the underlying infrastructure of your applications.

Naturally, people who have been with your organization for a long time will be concerned about their place in the new world order of a cloud-based, agile, elastic infrastructure. This "new world order" will see old procedural and manual jobs replaced by cloud-based services and automation.

In addition, our talks with many companies have informed us that successful transformation often requires bringing in external employees at both high and low levels in your organization.

These employees need to be welcomed into the organization and valued for their new skills without causing an allergic reaction – one that will slow you down. You can help this integration by making sure that everybody – above and below you – is aligned around clear business objectives that have been agreed upon by stakeholders and sponsors across the executive team. By doing so, the newcomers can be seen as helpful change agents rather than as dangerous outsiders.

3. Foster experimentation: When your company was new, it experimented – perhaps wildly – to pivot to the successful place you are at now. For

new divisions in your company, they probably still do. The challenge is to bring that experimentation mindset into your move to the cloud. Think about it as zooming out and zooming in:

- Assess where your established market and customer expectations will be in 10 years time (zoom out).
- Then select two to three initiatives that you can execute against in the next 6-12 months that will move you towards meeting those expectations (zoom in).

For transformation, you need to test many hypotheses about how your products and applications will be received by your customers – and the cloud lets you do that far more easily. These experiments, by definition, will be new to your teams and they won't be as expert at them as their current work – which is a scary thing for them.

At the same time, you have an executive team that expects results - so you have to maintain your standard processes with milestones and regular checkpoints. The way to navigate this apparent conflict is to set the expectation that pivots will occur as a result of experimentation and the surprises they will likely reveal. It is important that pivots are valued as learnings and insights rather than as failures.

To make experimentation real, take advantage of the observability levels you can get in the cloud. Ask your teams to create real-time feedback loops on the effectiveness of products and features they are building. The cloud gives them ways to do this that were just not possible with on-premises systems or old organizational silos.

“Setting a bar of perfection for your teams will likely result in information secrecy, siloed behavior, and defensiveness when things don't go well. If you're not careful, your most senior employees may decide it's safer to delay the move to the cloud. So work to create a safe space for failure by positioning those failures as challenges and opportunities for learning.”



Practical Steps to Deliver on the Vision

The next challenge is how to deliver on the vision with your teams. An approach we have seen work is something we call the “5 Rs:”

- 1. Re-organize goals and people:** Get your teams to think more about products and less about projects. Consider the “job to be done” for your customers and what outcomes they are striving for. Organize cross-functional, long-lived teams around those outcomes that balance business, technical, security, and compliance needs. Also, think about the key people in your organization who might slow things down and bring in fresh members to the team – people with transformation experience and the scars to prove it. This may be needed to make sure all teams are exposed to the right vision and skills.
- 2. Re-equip:** It’s your responsibility to stand up a reskilling and training function to help your people overcome their fear of change through curated learning paths and certifications. It is also important to reward them as they make progress. There are mutual benefits to training – new skills help the organization transform faster and training boosts retention by demonstrating the value you place in your teams. Finally, engineers can become more loyal to you and your company as you help them modernize their skills and even become more valuable on the market.
- 3. Recognize:** Agile and DevOps are not new. However, over the past 20 years the industrialization of processes governed by heavyweight project management methodologies such as ITIL – along with the growth of functional silos – got in the way. Understand that this makes teams resistant to change. Regularly restate that it’s ok for people to be scared – by putting it out in the open, you take away its power.
- 4. Reset:** Emphasize that with the cloud, all processes and tools are candidates for change. By working directly with the business in cross-functional and tightly aligned teams, engineers can get a clean line of sight into how the work they do directly impacts customer experience and business outcomes. Don’t rely on project managers to drive alignment. Otherwise you’ll end up with a slightly more agile interpretation of a traditional waterfall workflow.
- 5. Re-Tool:** Embrace modern architectural patterns and technologies such as microservices and containers, Infrastructure as Code, on-demand and elastically scalable infrastructure, cloud-native databases, and fully managed SaaS. With these approaches your teams can connect technology to product in a way never before possible, cultivating a new passion for engineering innovation. With new tools they can build, test, learn, and adapt faster than has ever been possible.

“Whether you lift and shift most of your applications or transform every one, you and your org are going to go through major change, and will face challenges along the way.”



Next Steps

We've talked about the steps you can take to discover your application estate, prioritize the steps of your transformation, build a change initiative and get the buy-in of stakeholders all around you, and finally to put your plans safely into place. MongoDB has worked with companies across the globe, of all sizes and from all industries, to help them embrace the cloud.

Through [MongoDB Atlas](#), the industry's first developer data platform, we enable organizations to build innovative and highly scalable applications faster than any other platform. The benefits of MongoDB Atlas are amplified by the platform's versatility to address almost any workload. Atlas combines transactional processing, application-driven analytics, relevance-based search, and mobile edge computing with cloud sync. These capabilities can be applied to almost every type of modern applications being built for the digital economy by developers. An intuitive document data model and unified query API provide a first-class developer experience, delivered in the only multi-

cloud developer data platform built for resilience, scale, and the highest levels of data privacy and security.

This document has been a high level overview of the challenges and opportunities you may face as you continue your modernization journey. If you'd like to get more specific and targeted assistance with modernizing your business, contact your [MongoDB Account Executive](#). We will come and discuss how we can partner with you to drive meaningful transformation in this once-in-a-generation shift to cloud computing – one that provides the optimum balance between maximizing your speed while minimizing risk.

About the Author

Chief Product Officer, Sahir Azam, is responsible for product management, growth marketing/ operations, and GTM strategy across the MongoDB portfolio. He joined MongoDB in 2016 as SVP, Cloud Products & GTM to lead MongoDB's strategy ahead of the launch of MongoDB Atlas and helped grow the cloud business from zero to over \$150 million annually. Sahir joined MongoDB from Sumo Logic, a SaaS machine-data analytics company, where he managed platform, pricing, packaging, and technology partnerships. Before Sumo Logic, Sahir was the Director of Cloud Management Strategy & Evangelism at VMware.

Build on MongoDB Atlas and see the transformation. [Get Started Free](#)

