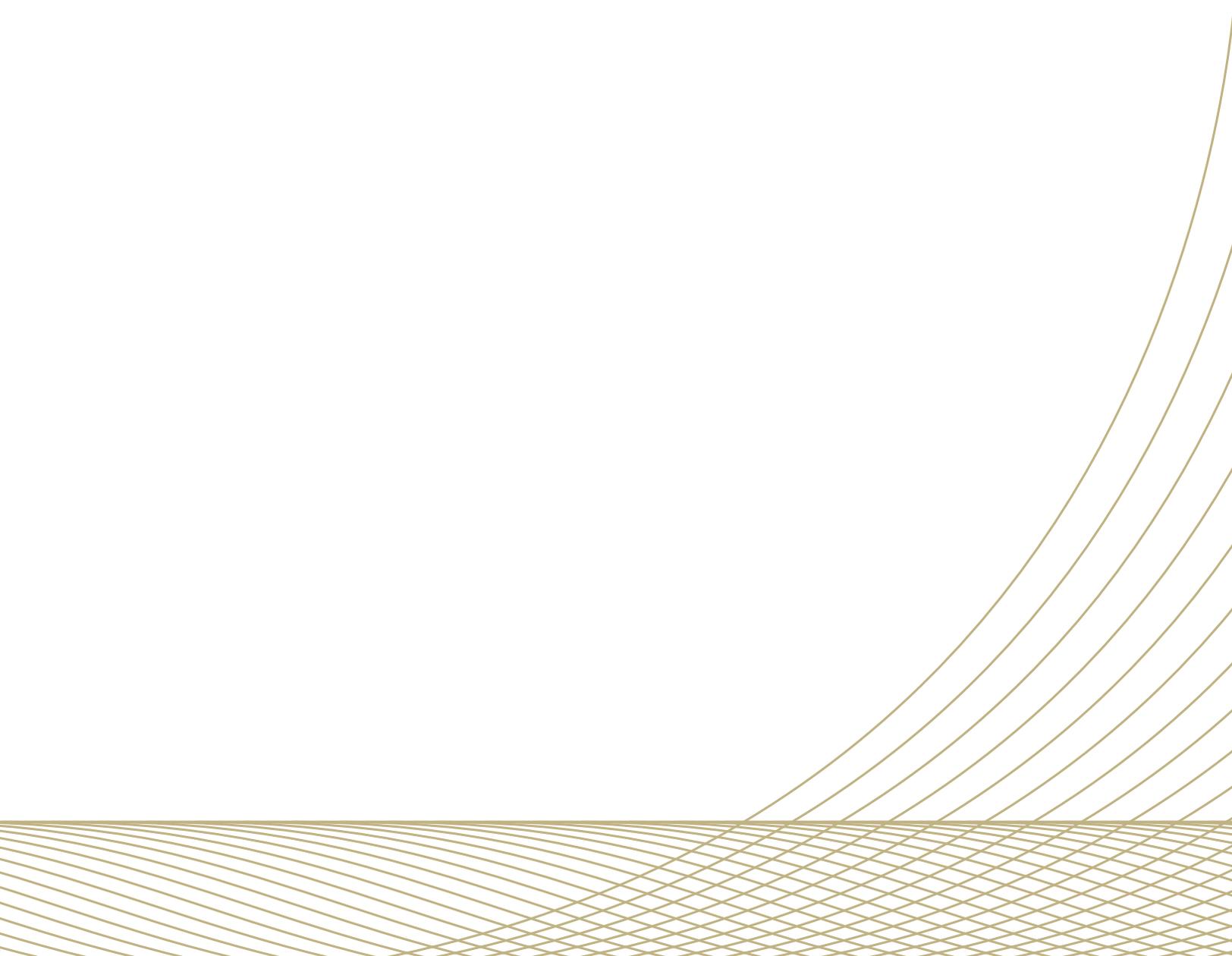




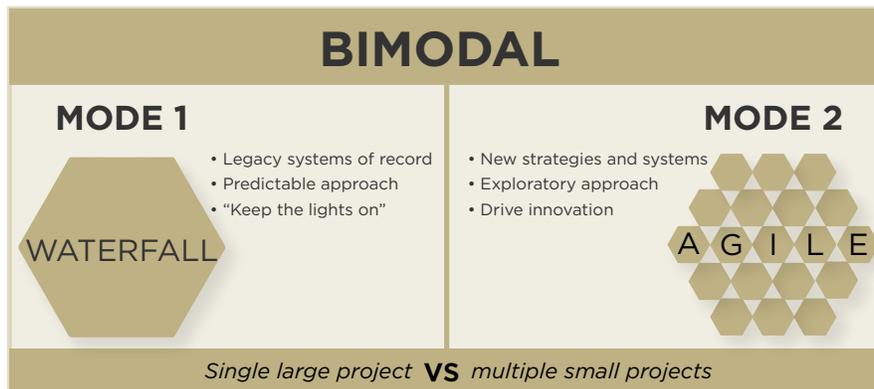
# Bimodal IT is Not an Excuse for Inaction: Modernizing Traditional Applications



## INTRODUCTION: BIMODAL IS THE REFERENCE POINT FOR IT

Gartner [defines](#) bimodal IT as, “The practice of managing two separate but coherent styles of work — one focused on predictability and the other on exploration. Mode 1 is predictable, improving and renovating in well-understood areas. In Mode 2, business and IT together explore and experiment to innovate and solve new challenges.”

Put a simpler way, bimodal organizations manage IT work in one of two fundamentally different ways: Mode 1 is the traditional way and, when applied to application development, often connotes waterfall methodologies, whereas Mode 2 connotes agile development. Mode 2 is often used for new, exploratory, and/or “digital” initiatives, while Mode 1 is presented as a less risky way to work with systems of record that run the business and “keep the lights on.”



Since its introduction in 2014, bimodal has become the de facto way to talk about everything from IT organizational structure and process to the evolving software landscape. Gartner is pointing out something that is obviously true—even though we know there are better ways to build software, most of the legacy systems we depend on were not written or architected with fast development cycles in mind. As a result, when we discuss IT modernization and competitiveness in today’s cloud-first world, the bimodal model provides an important frame of reference when setting enterprise strategy and execution.

## NOT EVERYONE IS SOLD ON BIMODAL IT

Despite its ubiquity in the world of enterprise IT, not everyone has jumped on the bimodal bandwagon. People who disagree with using a bimodal approach usually hinge their argument on the mistaken assumption that bimodal is a desirable end-state or organizational goal instead of an accurate assessment of its current reality.

Organizations, especially large ones, tend to resist change. For these enterprises, bimodal IT offers a way to confine changes that are necessary, yet painful, to discrete parts of the organization while allowing entrenched projects to remain status quo. Many organizations take this as license to only change their processes for newer, “greenfield” applications while leaving established processes as is for legacy systems.

Industry pundits have been quick to point out the flaws with this approach:

*“The implication of bimodal IT is that traditional IT is facing an awful future – one where it’s consigned to keeping the lights on for legacy applications but precluded from participating in what’s viewed as critical to the future of the company at large.” [Bimodal IT: Gartner’s Recipe For Disaster](#)*

*“This type of oversimplified and stilted approach has been failing to save innovation-hostile companies since Fred Brooks wrote about the infamous Silver Bullet. And this model will also fade into obscurity.” [Saying Goodbye to Bimodal IT](#)*

*“80 percent of CIOs that focused on so-called bimodal IT will accumulate a crippling technical debt resulting in spiraling complexity, costs, and lost credibility.” [Bimodal IT is mostly BS and will lead to failure, says IDC](#)*

 These commentators raise valid critiques—organizations that use bimodal as an excuse to stay static do so at their own peril.

## THE DANGERS OF MISUSING BIMODAL IT

In Gartner’s defense, bimodal has long been a reality, and the question now is whether it’s something we just live with or if it’s worth taking the time and effort to improve. Gartner [says](#):

*“A key misunderstanding of bimodal is the idea that Mode 1 processes and cultures do not need to change. But what CIOs quickly learn is that when scaling bimodal, Mode 2 cannot succeed without changes to Mode 1. Mode 1 needs enterprise agility as it focuses on renovating the core and on transforming Mode 1 business processes to deliver on digital business. Mode 1 also needs increased flexibility to integrate with newly developed Mode 2 solutions. Moreover, Mode 1 needs innovation, too, especially when it comes to building a competitive digital business platform.”*

### So why are organizations stumbling when it comes to bimodal?

Changing the way we work with existing systems is not simple. Modern software is built in small pieces and designed in a modular way with rapid change in mind. Legacy software was built in huge chunks and often involves compli-

cated dependencies, so simply figuring out how it works and what you want to change can be a daunting task. Large scale, manual regression tests are often needed to ensure these systems work after changes are implemented, and modeling these systems using elastic infrastructure can be a large project in and of itself.

The [theory](#) is the risk of changing systems of record is too high, and that you have to trade off quality and reliability for faster software development cycles. Thankfully, that is not true, and an entire body of [DevOps research](#) proves it.

Yet companies who fail to apply changes to Mode 1 systems take an even bigger risk— stagnating to the point of failure. Since Mode 2 systems are often dependent on Mode 1, how fast Mode 1 systems change determines and limits the rate of change overall. Enterprises that ignore or put off changes to traditional systems of record are effectively reducing their own ability to compete.

The problem is compounded by incorrect assumptions about bimodal IT. Organizations who view bimodal as a goal are lulled into a false sense of confidence that incorrectly validates the view that Mode 1 systems don't need to be changed or modernized.

Jez Humble, one of the authors of the Continuous Delivery blog points out where this approach goes wrong in [The Flaw at the Heart of Bimodal IT](#), “In reality, the situation is even worse: the majority of enterprises are still forced to perform infrequent, big-bang orchestrated deployments even for their agile products and services, due to the tight nature of this coupling. Much of the benefit of using agile processes is thus wasted.” He shows through [solid research](#) that high performing organizations don't take the risk of standing still with their systems of record and are able to improve both agility and stability at the same time by driving smaller changes at a more frequent pace.

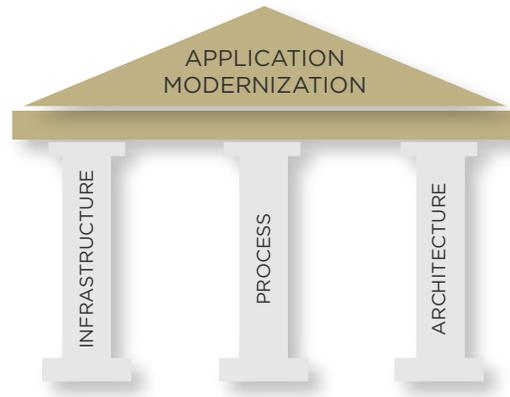
## **MODE 1 MODERNIZATION IS BOTH POSSIBLE AND PREFERABLE**

Existing, “intractable” IT systems can and should be modernized. When an enterprise chooses to apply changes to Mode 1 workloads, it maximizes the return on existing investments for its business, reduces the burden of technical debt, frees up resources for Mode 2 innovations, and accelerates agility overall.

Change, however, requires an honest and accurate assessment of an enterprise's current reality. Mode 1 processes and technologies will not transform overnight, and enterprises need to be careful not to exchange one set of risks for another. Throwing out mission-critical applications and systems of records and rebuilding them from scratch using modern architectures and processes is too risky, time-consuming, and costly to be effective. Instead, modernization

should include systematic, incremental changes that enable enterprises to take steps forward without jeopardizing what exists.

Three core pillars build on each other to make application modernization possible in a safe way that does not involve ground-up rewrites.



### *One: Infrastructure Modernization*

Much of what slows down legacy development is the time required to get infrastructure in place to test and deploy these large, monolithic systems. It can take weeks or months to just get ready for an experimental test.

This leads to a reluctance to try things, which, in turn, slows learning and speed of change to a crawl.

Modern clouds solve this problem by making infrastructure elastic, i.e. easy-to-establish and easy-to-discard at relatively low cost. The key to modernizing infrastructure is enabling organizations to make legacy systems elastic too. Various infrastructure-as-code systems such as Puppet, Chef, and Terraform help automate and accelerate the process of provisioning infrastructure resources, but more work is required to configure and provision systems with these tools.

### *Two: Process Modernization*

Modern systems are built using agile concepts, and this practice increasingly extends all the way through operations teams (DevOps). Taking an existing system that was built with a waterfall development process and beginning to modify it in an agile way involves breaking the work into smaller chunks, which makes the often-overwhelming work needed to produce a release something that can be done quickly and reliably.

These releases and small changes in code need to be tested independently, which makes the elastic infrastructure mentioned earlier critical. Each QA team and each change needs a fresh and validated environment, meaning once

the tests are completed, the environment needs to be discarded.

### *Three: Architecture Modernization*

Changing architecture is one of the most problematic things about modernization, as a new architecture often means radically different technology and general application profile (performance, scale, etc.).

One technique to change architecture is to do it in steps, like this:

1. Create a “harness” that operates under your new architecture (For example, Kubernetes/Docker with a noSQL persistence layer)
2. Find a way to run old and new systems in parallel
3. Select functionality from the legacy system that can be reimplemented as a microservice in the new system.

Keep doing this and, over time (often surprisingly fast), your legacy system will dissolve, replaced by modern services.

## **CONCLUSION**

To successfully compete in the increasingly cloud-centric IT world, enterprises must extend agile practices currently reserved for new applications to their legacy systems. The right approach is to apply Mode 2 practices to Mode 1 workloads, using agile ideas regardless of what system is being changed, and allowing applications to self-modernize over time. Successful application modernization is built on a foundation of modern infrastructure that enables improved processes and architectural changes to the applications themselves. This approach requires confidence and fortitude, but in the end pays for itself with increased ROI, agility, and competitive advantage.



**Skytap Headquarters**

719 2nd Ave., Suite 800  
Seattle, WA 98104  
1-888-759-8278

**Skytap UK**

60 Cannon Street  
London, EC4N 6NP  
+44 20 3790 9062

