



2020 Legacy Modernization Report

Exploring the Connection Between Legacy
Dependence, Digital Strategy, and Achieving
Business Goals

Table of Contents

Introduction.....	3
Executive Summary	4
The State of Legacy Modernization	6
The Road to Legacy Modernization	33
Modernization Assessment.....	43
Conclusion	46
About the Authors	47

Introduction

Legacy modernization is replacing organizations’ antiquated technology systems, infrastructure, and processes. These modernization initiatives are typically driven by a company’s desire to reduce the risk, cost, and debt associated with an outdated technology environment. Legacy modernization has many benefits; it keeps some companies afloat by enabling them to achieve and maintain competitive parity, and gives others the ability to adjust their business and operational models during disruption—whether from market shifts (such as the movement from brick-and-mortar retail to e-commerce) or unprecedented current events, like the sudden need to support remote workforces due to the COVID-19 pandemic. And, for a few businesses, it also enables them to adopt industry-specific, emerging technologies that will elevate them to the top of their market. However, one of the most important benefits of legacy modernization is in how it affects a company’s ability to build, improve, and deliver its product.

When discussing modernization, the crucial connection between technology and a company’s business success is often overlooked—specifically, the correlation between legacy dependence and inefficient software development processes. Research findings indicate that the greater the dependence on legacy applications, hardware, technology stacks, and people/operational processes, the more challenging it is to develop and release product changes or features.

Unfortunately, it’s difficult to properly diagnose the true cause of product delivery issues and to understand how the symptoms connect to legacy dependence. Therefore, this report serves to simplify that diagnosis by outlining the symptoms, causes, and remedies of legacy dependence. It also breaks down the elements and use cases of legacy modernization initiatives, and offers a guide to building a modernization plan that will improve product development and achieve business goals.



Executive Summary

Primary Takeaways

Research conducted in the creation of this report revealed that the majority of North American businesses have begun to modernize their IT organization in some way, but few have completed full transformation. Those that have completed more modernization across different technology areas experience more efficiency, lower costs, and fewer disruptions (e.g., unplanned outages). Primary data insights include:

- Most respondent organizations are partially or heavily dependent on legacy systems (those older than 10 years of age). On average, organizations report dependence on legacy technology for **33% of their overall systems** and **29% of their business-critical systems**.
- Most respondent organizations have migrated at least half of their platforms to the cloud, but many are still migrating or are struggling to continue migration. Current cloud migration rate is at **70% (indicating those that have embraced or are in the process of migrating to the cloud)**, and **12% are fully cloud native**.
- **Security** is the **top concern** for the majority of responding organizations across IT management and in maintaining legacy systems.
- **Maintaining legacy dependence** is the **second-most pressing concern** for the majority of respondent organizations across IT management.
- Respondent organizations with heavy legacy dependence reported:
 - **higher overall IT management costs** than those with lower dependence.
 - **more unplanned outages** than those with lower dependence.
 - **slower software releases** than those with lower dependence.
 - **slower delivery times for product changes and features** than those with lower dependence.
- Modernization drivers reported by respondents in leadership roles revolve around **increasing competitive advantage, market agility, and technology / product innovation**.



Data Summary

For this report, approximately 500 professionals involved in or with knowledge of technology management departments and processes—including team members, management, and executive roles—were surveyed. These respondents represented organizations from various industries and segments, and with annual revenues of at least \$100 million. Sample sizes vary across data charts and visualizations, as logic dictated the questions that respondents received according to their familiarity and previous responses. Interspersed throughout this report (in green boxes) are quotes from survey respondents, lightly edited for grammar.

Modernization Personas

To better illustrate how modernization influences technology management efficiency, respondents are categorized into personas in some sections of this report. These personas—“Modern,” “Modernizing,” and “Not Modern”—were assigned depending on how respondents scored in six key areas. The scores are the aggregate of all six areas¹ and offer an indication of how widespread modernization is at each respondent’s organization. Among respondent organizations...

- 13% are classified² as not modern
- 49% are classified as modernizing
- 37% are classified as modern.

¹ Design approach (e.g., use/lack of DesignOps), SDLC automation (e.g., use/lack of testing or deploy automation), project management methodologies (e.g., use/lack of Agile methodologies), system/platform age (e.g., percentage of systems more than 10 years old), cloud migration (e.g., amount of workloads migrated to the cloud); SDLC management optimization (e.g., use/removal of outdated processes and workflows in SDLC)

² Modernization persona classifications are further detailed on pages 43-45



The State of Legacy Modernization

Defining Legacy & Legacy Dependence

To grasp the crucial connection and limitations between technology and a company’s business success, it’s important to understand all of the factors that contribute to legacy system dependence. Throughout this report, “legacy” refers to aging or obsolete elements of an organization’s technology environment, including hardware/software, infrastructure design, and management methods. Furthermore, legacy dependence is when companies have technologies, business goals, or organizational cultures that are out of sync with modern best practices and industry standards. This legacy dependence then prohibits full operational efficiency.

A company’s legacy dependence is measured by three main factors:

- **Age** — The age of the systems (applications, software, hardware, etc.) that are used to manage an organization’s core applications. In this context, legacy denotes systems that are more than 10 years old.
- **Design** — This is the way in which organizations have set up their IT infrastructure and tools. In this context, legacy denotes systems and infrastructure that are not designed to support modern technologies, processes, or best practices.
- **Process** — These are the processes in place for managing technology systems and teams. In this context, legacy denotes outdated cultures and processes, including software development and talent management methods.

Overcoming legacy dependence is a goal for many organizations, especially those that are running revenue-generating functions on outdated systems. Previously, it was possible to successfully function with traditional methods of leveraging technology—such as hosting applications and data on the premises or managing software development lifecycles with overview committees and multi-stage approval processes. However, the introduction of modern options like cloud platforms and testing automation have changed things. Today, leveraging antiquated methods slows business processes in a market that increasingly prioritizes rapid delivery and innovation. Legacy dependence stifles a company’s growth potential by impeding the product development lifecycle and time-to-market, preventing the company from adequately meeting customer demand and realizing their full business potential.



Legacy dependence paralyzes the software development lifecycle through challenges such as unplanned outages from code incompatibilities, lack of legacy-proficient talent to maintain system operations, and/or exploitation of security vulnerabilities that can lead to system-crippling attacks. Conversely, research shows that legacy modernization improves the results of the software development process overall, greatly enhancing a business' ability to deliver to its customers. For instance, legacy modernization in the form of an Agile transformation improves the engineering culture and collaboration between the groups responsible for delivery. When teams using Agile work together collaboratively and iteratively, they can deliver higher-quality, faster changes to products for both internal and external users.

Time to market is not the only reason organizations need to be able to make quick, quality changes to their software; companies also need to make adjustments in response to regulations and security vulnerabilities. Research shows that security is the top concern for many technology organizations, as well as their top-reported challenge in both IT management and maintaining legacy systems. In fact, Levvel has found that security issues can be one of the greatest signs of legacy dependence because legacy systems are not equipped to protect against modern cyber threats; legacy code can even exacerbate security vulnerabilities. Although strong product delivery and security management may seem like two separate goals, optimized software development is actually vital for both. Legacy dependence prevents organizations from being able to make both high-quality and timely product changes and responses to security breaches or unplanned outages.

The following sections explore legacy modernization trends among North American organizations and how each factor in legacy dependence affects an organization's ability to meet its business goals.

Legacy Dependence — Age of Technology

Permeation of Legacy Systems

Legacy systems are prevalent in most companies. They can be found across many departments and processes, and while outdated, it's likely that they still function. However, legacy systems are not nearly as effective as the solutions that have hit the market in the last decade. Newer systems have been engineered to take advantage of cutting-edge technology and modern frameworks, whereas incompatibility issues with legacy systems only worsen as technology evolves.

The potential damage of legacy dependence increases even more when the systems are business-critical or necessary for the execution of revenue-generating functions. Research indicates that 78% of companies are reliant upon outdated business-critical systems, although a large portion of these are in the process of modernizing (**Figure 1**).

Of those surveyed, the largest portion of respondents reported between 26% and 50% dependence on legacy systems (**Figure 2**). A smaller portion reported 51% dependence or greater, while very few companies reported no dependence on legacy systems at all.

Figure 1
Legacy Systems Supporting Business-Critical Functions

"Would you say your company is heavily dependent upon legacy systems for business-critical functions?" (n = 369)

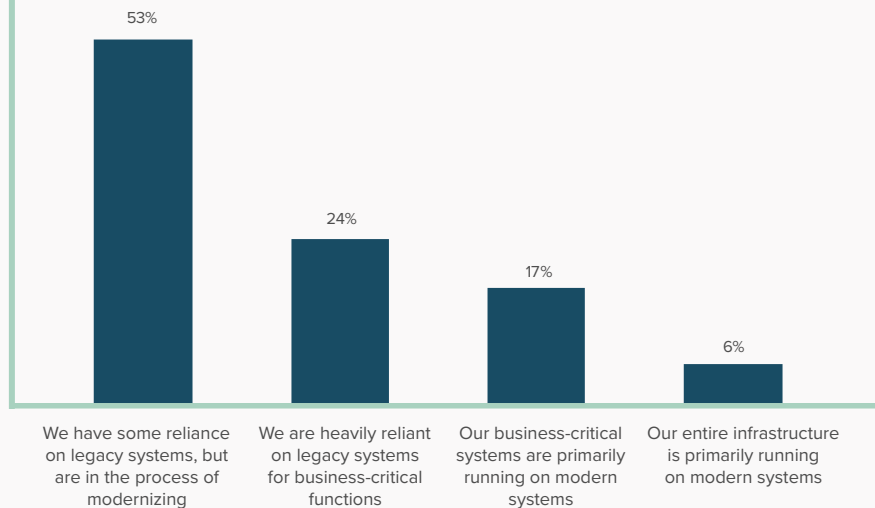
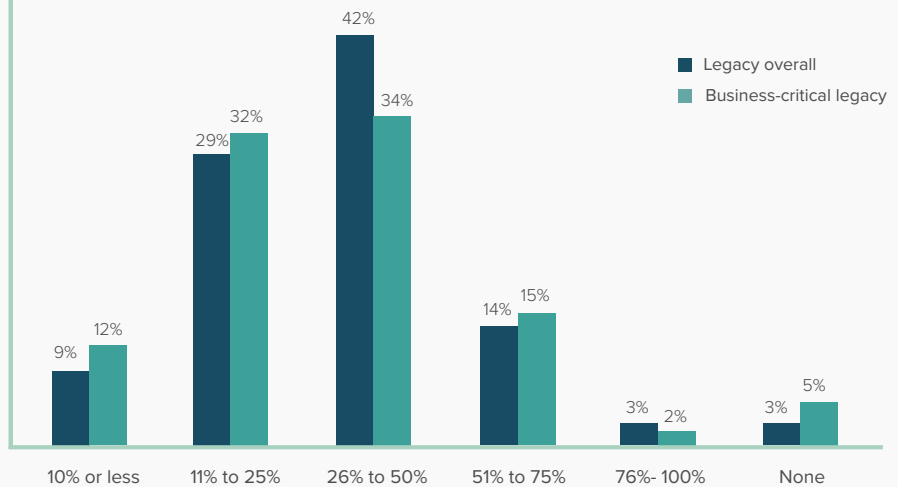


Figure 2
Organizations' Percentage Rate of Legacy Systems

"Approximately what percentage of systems are legacy?" & "Approximately what percentage of business-critical systems are legacy?" (n=396)

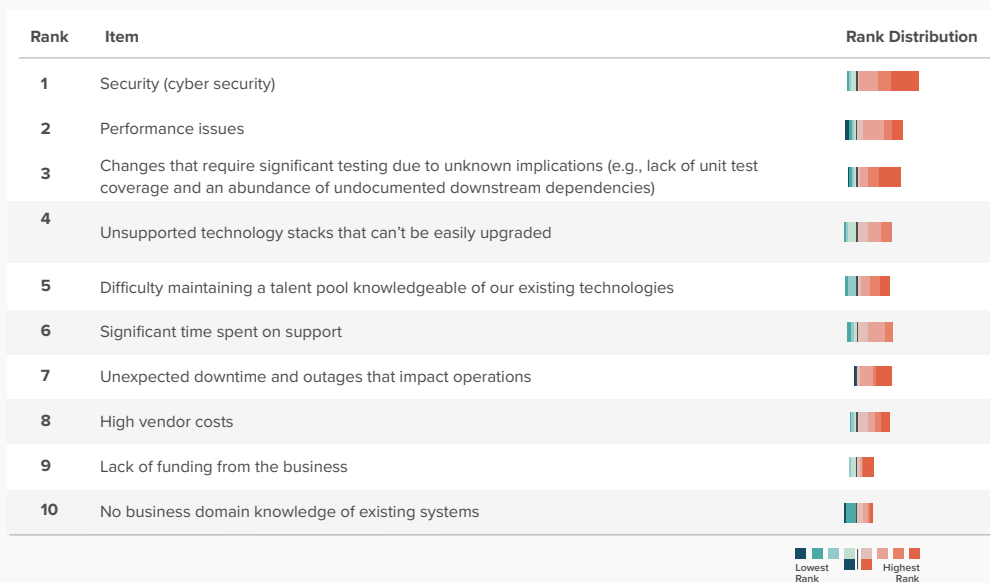


Technology Management Survey, Level Research, 2020

Challenges in Maintaining Legacy Systems

Figure 3 highlights the top issues that organizations face in maintaining their legacy systems. As previously mentioned, security is the primary challenge, followed by performance issues. Performance issues are things like non-responsive user interfaces (UI) and UI freezes, as well as latency and long wait times for requested actions or data. These issues often arise when a system’s infrastructure is not set up to scale, causing applications to break during periods of especially high volume, such as overwhelming website interaction upon the release of a new product.

Figure 3
Organizations’ Ranked Challenges in Maintaining Legacy Systems
“What are your organization’s main challenges in maintaining legacy systems?” (n=369)



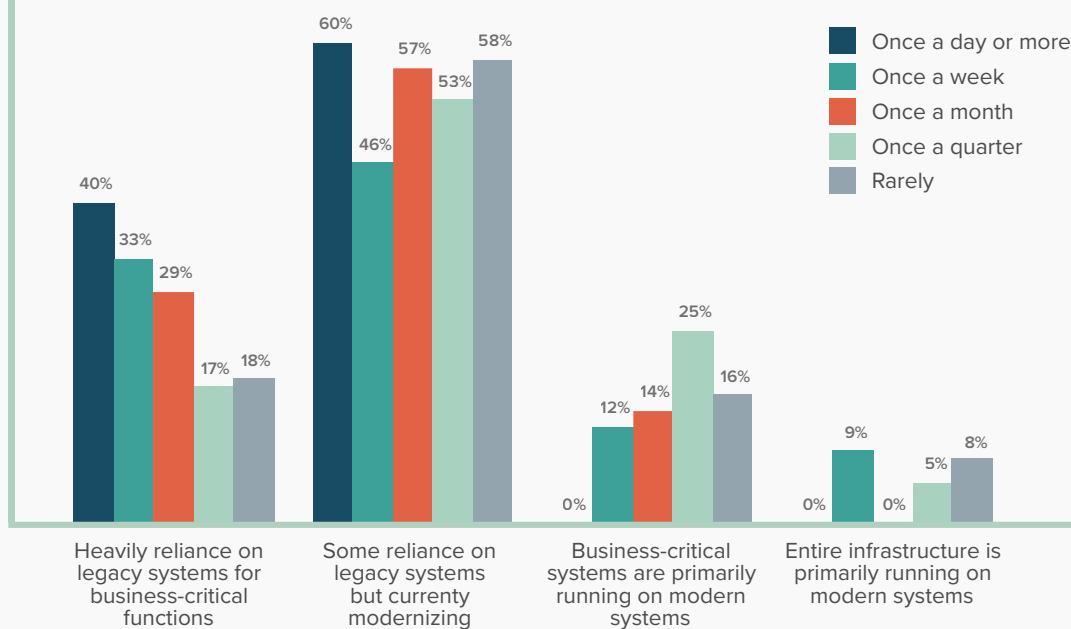
Technology Management Survey, Level Research, 2020

Other issues revolve around the implications of software development in complex technology environments, as well as how incompatibility affects the visibility of and control in managing changes. Outdated technology stacks and the difficulty associated with removing them is also a highly ranked challenge according to organizations, in addition to finding talent with the skill sets necessary to maintain existing stacks. Overall, the data reveals that the effect of legacy systems reverberates throughout many areas of IT and business processes.

Unexpected Outages As a Result of Legacy Dependence

The occurrence of unexpected outages is another common issue of legacy dependence. Outages are a reality for most businesses and the majority of survey respondents reported experiencing them. However, there is a correlation between running business-critical functions on modern systems and experiencing fewer unplanned outages (**Figure 4**). Outages often occur as a result of code that is not properly tested during software development lifecycle (SDLC), as well little to no business analysis on the impact of new code against the product team’s requirements. Testing automation allows development teams to test new lines of code against dependencies and business logic so that released code is strong and outages are rare.

Figure 4
Rate of Unplanned Outages By Reliance on Legacy Systems
“How often does your organization experience unplanned outages?” (n=299) &
“Would you say your company is heavily dependent upon legacy systems for business-critical functions?” (n=369)



Technology Management Survey, Level Research, 2020



Technical Debt & Legacy Dependence

Despite the difficulty of legacy systems and the issues that arise from heavy dependence on them, they are a reality for many companies—and one that is hard to escape. One factor that affects the percentage of legacy systems—as well as how long the business maintains these systems—is the company’s technical debt balance and its approach to managing that debt.

Technical debt is the cost of additional rework down the line when a technology group implements a limited solution in the short term, rather than using a better approach that would take longer to implement initially. It’s also worth noting that there is a relationship between a company’s technical debt and its tolerance for legacy systems. That’s because removing them is challenging and expensive, and decision-makers often deprioritize modernization in the face of competing initiatives, thereby allowing the status quo to persist. However, the status quo can be draining, and the older and more business-critical these systems are, the more difficult it will be for the company’s technology teams to maintain operational efficiency.



“ We have so much technical debt in legacy that it will be a long time until we can modernize it all.”

Eventually, though, this debt must be paid. Consequently, many organizations perform regular cost-benefit analysis to determine when and how to engage in more extensive improvement initiatives. Granted, large, transformative initiatives can be expensive. But, it often costs more to maintain technical debt in the long run; the longer an organization waits to perform a strategic overhaul, the pricier the project will be. In the case of legacy systems, the technical debt is more than just the price tag of future implementation projects; it’s also the revenue that the company is losing when it can’t get competitive products to market quickly. The time lost in inefficient and antiquated processes is money left on the table. Therefore, it’s the responsibility of prudent decision-makers to evaluate the long-term costs of tolerating legacy dependence and commit to modernization. The result is less technical debt years down the line, including lost market

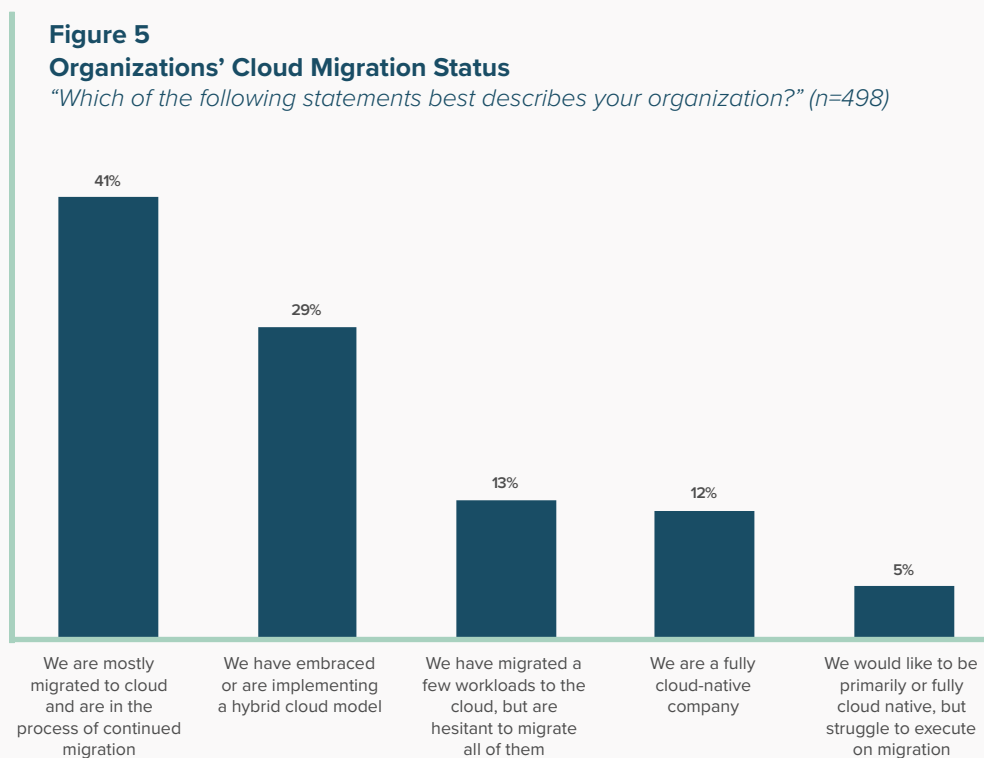


Legacy Dependence — Design

Outdated system design is a key reason why legacy systems struggle to keep up with a modern company’s high-speed and high-quality product delivery expectations. Design refers to the manner in which a system was built, such as an application built to operate in a specific environment (like a cloud environment as opposed to technology on the premises); the technology stack on which workloads run; or the system architecture, like the use or lack of APIs.

Cloud Migration Rate

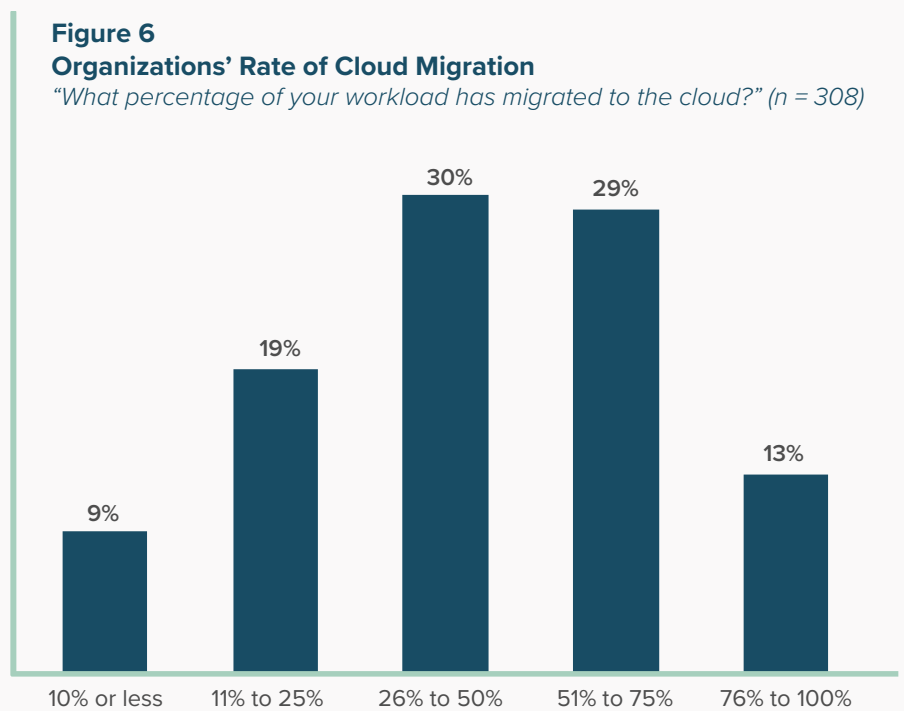
The use of on-premise systems is a strong indicator of legacy dependence. Consequently, migrating those systems to the cloud is one of the most beneficial legacy modernization initiatives that an organization can undertake to achieve the savings and improved efficiency associated with cloud computing. Cloud migration consists of moving workloads off premise to a more flexible, scalable infrastructure. **Figure 5** shows that the majority of respondents are still in the process of migrating workloads to the cloud, and that a small percentage are fully cloud-native.



Technology Management Survey, Level Research, 2020

Even if few are fully cloud native, the majority of the market has embraced the cloud and/or is in the process of doing so. Notably, companies that are less than five years old were significantly more likely to report that they are fully cloud-native than older companies; this is likely because most young companies have used cloud-based systems and applications since their inception. The implementation of a hybrid cloud model was also popular among respondents. In this scenario, organizations leverage both public and private (on-premise) cloud environments to house data, and use specific cloud provider services (such as AWS Lambda). Some companies choose public or private cloud computing for certain business workloads due to ease of use, accessibility, or the complexity of the applications. Similarly, others are influenced by the nature of their business models or industry, and thus adopt a hybrid approach to more easily meet regulatory requirements and/or to secure customer data within their own firewalls.

Typically, an organization will migrate workloads to the cloud in phased approaches, determining which to do first based on company priorities, such as the costs required to migrate them or the security regulations involved. **Figure 6** shows the distribution of workloads migrated to the cloud as reported by the survey respondents.



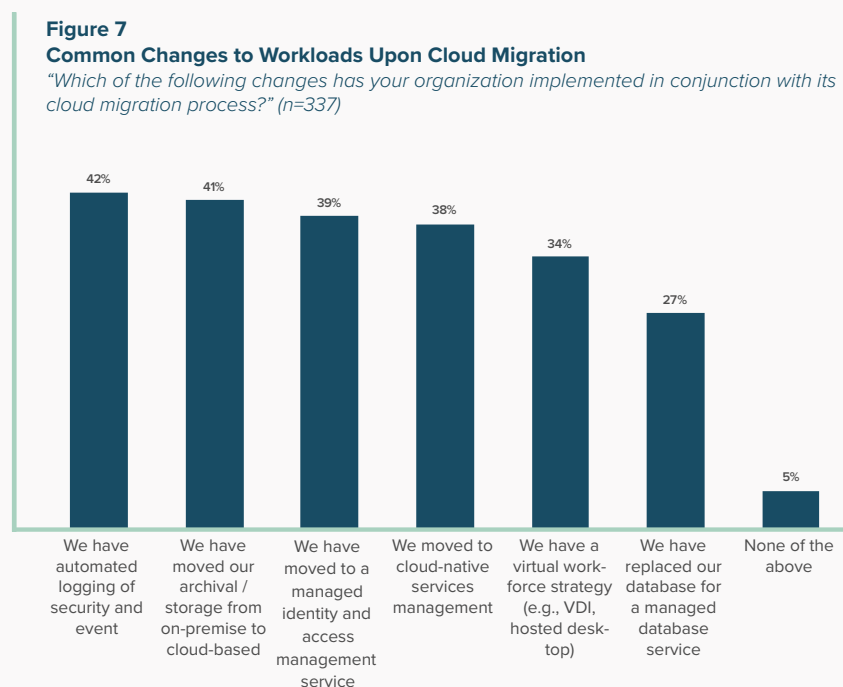
Technology Management Survey, Level Research, 2020



Rearchitecting for the Cloud

Optimizing solutions for the cloud is a vital part of the migration process, and without it, organizations miss out on the full advantage of cloud computing. Some organizations simply “lift and shift” their legacy applications to a cloud environment, essentially changing only how the application is hosted. In doing so, they don’t optimize these applications to operate effectively in the new environment, such as by refactoring them to connect to cloud-based APIs, or take advantage of cloud-native services and tools, like a managed database service. Organizations also dismiss efficiency with a lift-and-shift approach. For example, when companies move their security and event management to a cloud environment, but fail to automate or control it through a cloud-native software specifically designed for the process, they’ll face the same challenges as they did with their previous manual and/or on-premise solution or method.

In this context, a more accurate measure of modernity is whether the application has been rearchitected to leverage the cloud environment. This would bring the advantage of improved scalability, data analytics, and efficiency associated with cloud technology. **Figure 7** shows some of the architecture and refactoring activities required during cloud migration to fully optimize the systems for the cloud. Most organizations rearchitect solutions for storage and archival, as well as for security management, such as logging events. For cloud migration to truly improve technology management and product delivery, applications must also be successfully redesigned in the ways shown in **Figure 7**.



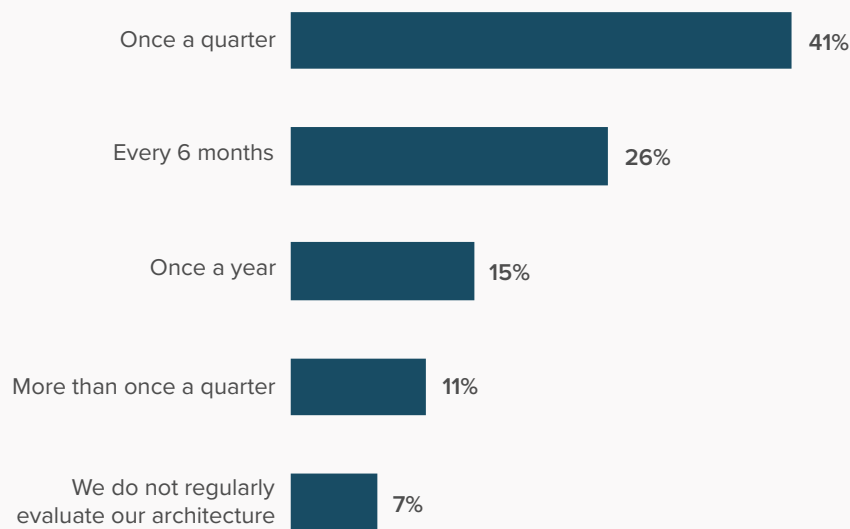
Technology Management Survey, Level Research, 2020



Architecture Evaluation

In addition to rearchitecting its applications and workloads for the cloud, another element of modernization and maturity in an organization’s cloud management is whether the technology staff regularly evaluates their architecture. **Figure 8** shows that most organizations perform evaluations at least every six months; best practice is to evaluate at least once a year. Evaluations can entail revisiting how recent advancements have affected previous designs, as well as identifying, applying, and maintaining new cloud patterns.

Figure 8
Organizations’ Regularity of Performing Architecture Reviews
“Please estimate how frequently your cloud architecture is evaluated for best practices.” (n = 337)



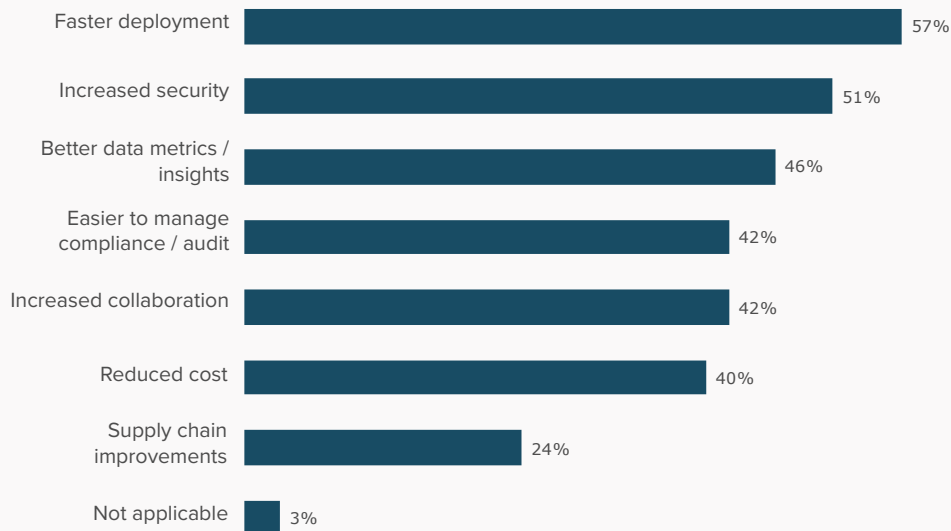
Technology Management Survey, Level Research, 2020



Benefits of Cloud Migration

The top-reported benefits of cloud migration influence both product delivery success and security. These include improvements in deployment time, security, and data management **(Figure 9)**. Respondents also reported better collaboration across teams, and improved compliance—which could include internal policies and standards, as well as formal external frameworks like PCI. It’s important to note here that security and compliance concerns are often the primary roadblocks that organizations report when considering whether to migrate workloads to the cloud. That’s because many companies fear that hosting sensitive data in public cloud environments could leave them more vulnerable to breaches and data loss. However, cloud platforms are created with advanced security architecture, tools, and controls, which enable users to prevent even more threats than their antiquated, legacy systems are capable of preventing. Once again, companies should recognize the importance of taking a transformative approach when migrating. The lift-and-shift model of migration can occur in security management, as well; some companies use on-premise security measures in the cloud, such as network DMP. But, security and compliance groups must understand the protocols, tools, and methodologies in native-cloud security and adjust their GRC standards to optimize these environments.

Figure 9
Top Benefits Experienced With Cloud Migration
“What benefits have you experienced since migrating?” (n=387)



Technology Management Survey, Level Research, 2020

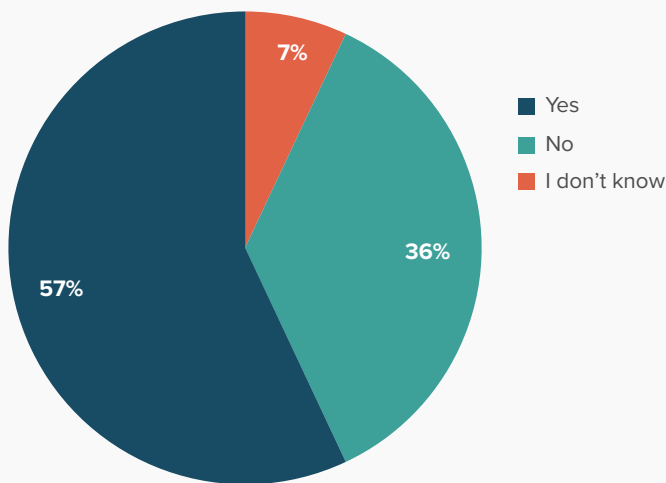


Cost Savings of Cloud Migration

Reduced costs is another key benefit of migrating to a cloud-native environment. **Figure 10** shows the percentage of respondents who reported a reduction in overhead expenses. This is a clear indication of the inherent benefits in reducing technical debt; in fact, most organizations that

become cloud-native experience a rapid reduction in overhead costs. The data also revealed a direct correlation between an organization’s cloud migration rate and its cost reduction. Namely, the majority of respondents who reported no cost reduction had only migrated some of their workloads to the cloud. In addition, the lack of cost reduction after migration is often a result of improperly optimizing the legacy architecture for the cloud environment prior to migration. When monolithic applications and processes are combined with cloud technology, computing costs can actually increase.

Figure 10
Reported Cost Savings After Cloud Migration
“Have you seen a reduction in overhead expenses (e.g., costs, staff involvement, etc.) since installing cloud-native services?” (n = 216)



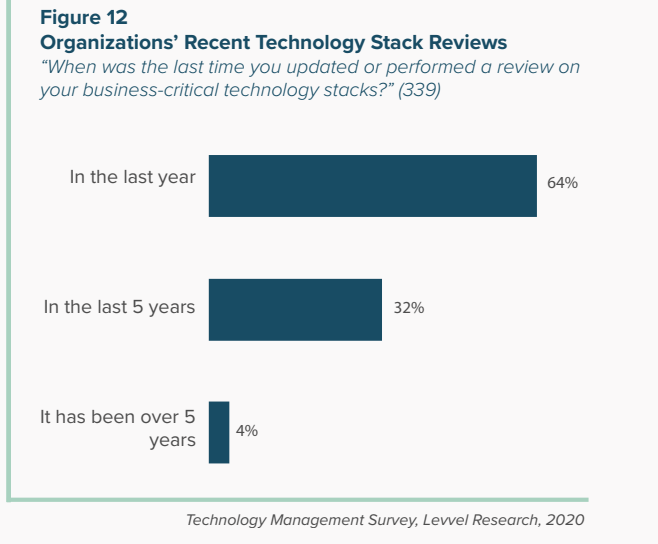
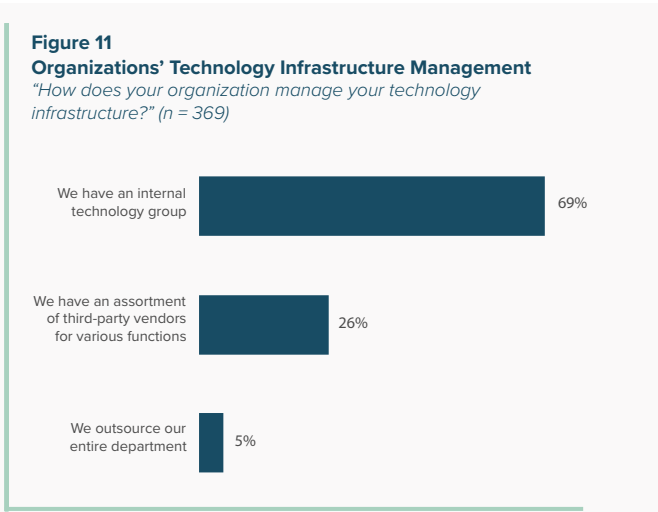
Technology Management Survey, Level Research, 2020

IT Infrastructure Management

One indicator of modernization is how an organization manages its IT resources, mainly whether its resources are in-house or handled by multiple third-party vendors instead. Relying on third-party applications or teams to manage various IT functions can result in a lack of coordination and customization for the needs of the business. For example, if an organization relies heavily upon an outside technology company to manage its website, this presents a potential bottleneck and delay when the organization has a sudden bug to fix. It could even expose an increased risk in the event of a security breach that needs to be addressed immediately. Conversely, an in-house team often results in improved management of IT challenges and greater familiarity with business needs for better coordination between business goals and IT.



Figure 11 shows that about one-quarter of respondents outsource their IT functions. Level has found that organizations that are dependent on vendors or outsourced IT also tend to have less market agility and ability to innovate with their products. They typically need more guidance on making substantial technology transformations, including modernization, as they often don't have the skills on hand to perform and manage these projects. Sometimes outsourcing IT functions is a result of a lack of understanding of the true role technology has on revenue generation. Therefore, unlocking the value of modernization requires more than just bringing some or all IT processes in-house — it requires a culture change.



Another indicator of legacy dependence is a team's technology stack management, including the software and coding languages the organization uses and how frequently it reviews its stacks. **Figure 12** shows that more than one-quarter of respondents reviewed their stacks in the last 5 years. In this context, legacy dependence is apparent when organizations use outdated coding languages and don't regularly review and revise their code to stay in sync with modern technology. Outdated technologies include Cobol, Fortran, JEE, and VB.NET.

It's also important to note that regular updates or reviews include not only outdated tech stacks, but also older/outdated/unsupported versions of a programming language and/or framework. For example, an organization may have a modern framework like Angular, but it may be using a version that is no longer supported. In addition, certain systems have security vulnerabilities that their providers are not going to proactively fix because they have moved beyond support for that legacy framework.



Modern organizations review their stacks and underlying software frameworks regularly, and use current technology, such as Kubernetes, Docker, React, NodeJS, Typescript, distributed NoSQL databases (e.g., Cassandra), AMQP-based messaging software, Go, .NET Core, Spring Boot, and Java 12. This not only helps with talent acquisition and interoperability with internal and external systems, but it can improve security, code quality, and responsiveness to bugs and outages.

Data Management

Data analytics is becoming one of the most important facets of a successful business, as the ability to leverage analytics improves an organization’s understanding of its customers and its role in the market. Organizations can potentially translate data into actionable insights — and make more informed decisions than their competitors.

Legacy dependence in data management is not just about how the data is collected, stored, and leveraged, but also how the organization views that data and its potential to affect the company’s business decisions, products, and competitive advantage. While some organizations may put effort into setting up infrastructure and analytics for their data, this doesn’t guarantee that they are using the results in their strategy and decision making. Therefore, shifting away from legacy data management is as much about redesigning infrastructure as it is about changing mindsets.



“ We have many problems managing consumer data to drive product innovation.”

When organizations are not managing their data effectively, they are simply collecting data. Traditional data management leverages data warehouses that are built on Relational Database Engines (RDE) not able to effectively scale with increases in data volume and data variety, or to handle the increasingly complex uses organizations have for that data. This infrastructure is also costly to procure and maintain. Modernizing organizations may use analytics tools like Hadoop, Amazon Redshift, and NoSQL to better manage the data, and while this is a good strategy, it does not address the underlying issues with RDE infrastructure overall.

Modern companies leverage cloud-based data warehousing and management tools to optimize their data and enable strategic and innovative analysis. These platforms are more scalable, and take much of the cost and maintenance burden from the organization that owns the data. Specifically, modern companies have successfully created a culture of data-driven results and metrics, and, beyond just cataloging customer data, they continuously take the insights drawn from their research and implement them into their product releases.

Security Management

Security management is a critical aspect of technology management and legacy dependence; it was also the top challenge cited when respondents were asked to rank their issues related to the maintenance of legacy systems (see Figure 3 on page 9). In fact, the design and management of security is closely tied to the systems it manages; if an organization is heavily reliant upon legacy systems—even with strong controls and processes in place—it will have difficulty truly lowering risk. Plus, the risk of security breaches and cyber attacks is even greater when vulnerabilities remain unchecked and unaddressed, as well as when an organization is reliant upon older systems that were not designed to protect against modern cyber threats.

Figure 13 shows that most organizations implement a designated team and network restrictions to manage their security. Certain methods—including having a security team—are more influential than others. But, even devoting employees solely to protecting an organization’s technology infrastructure is not always effective if the systems they’re managing and the measures they’re

using are outdated or manual. Rather, the strength of security also depends on the depth and trust of an organization’s use of technology. Overall, organizations are more likely to use high-level measures and processes—such as network restrictions and regular audits—than more specialized tools and automation, like secrets management and SIEM solutions. While security teams

are valuable, they’ll be fighting an uphill battle if they’re not leveraging the best tools to help them control and protect their systems and data.

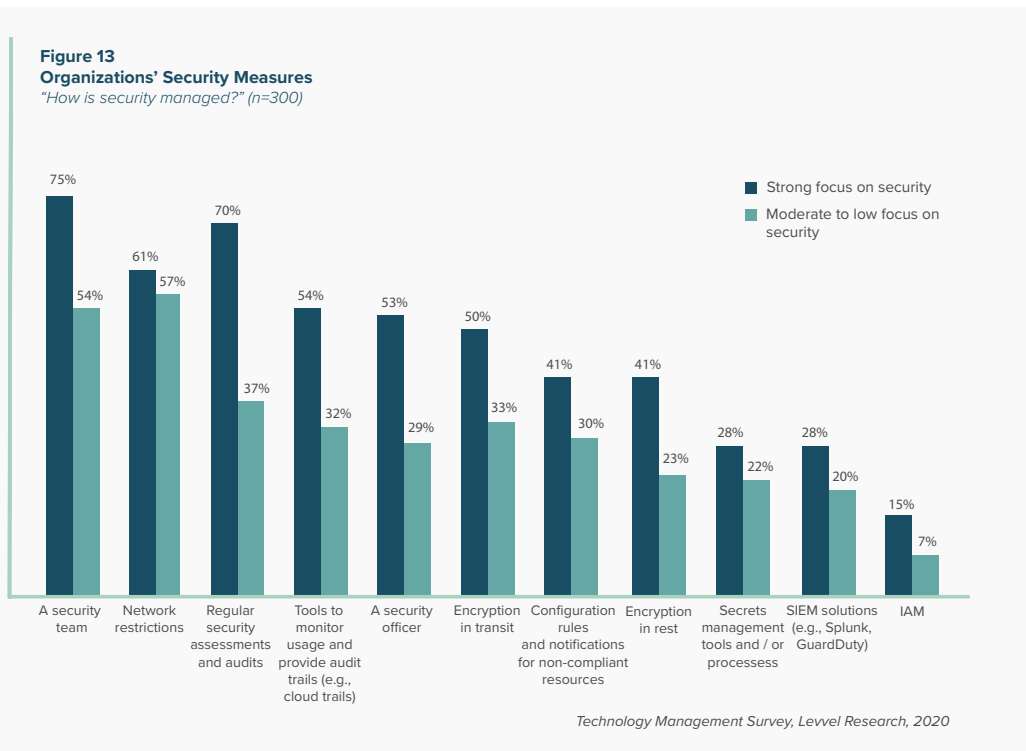
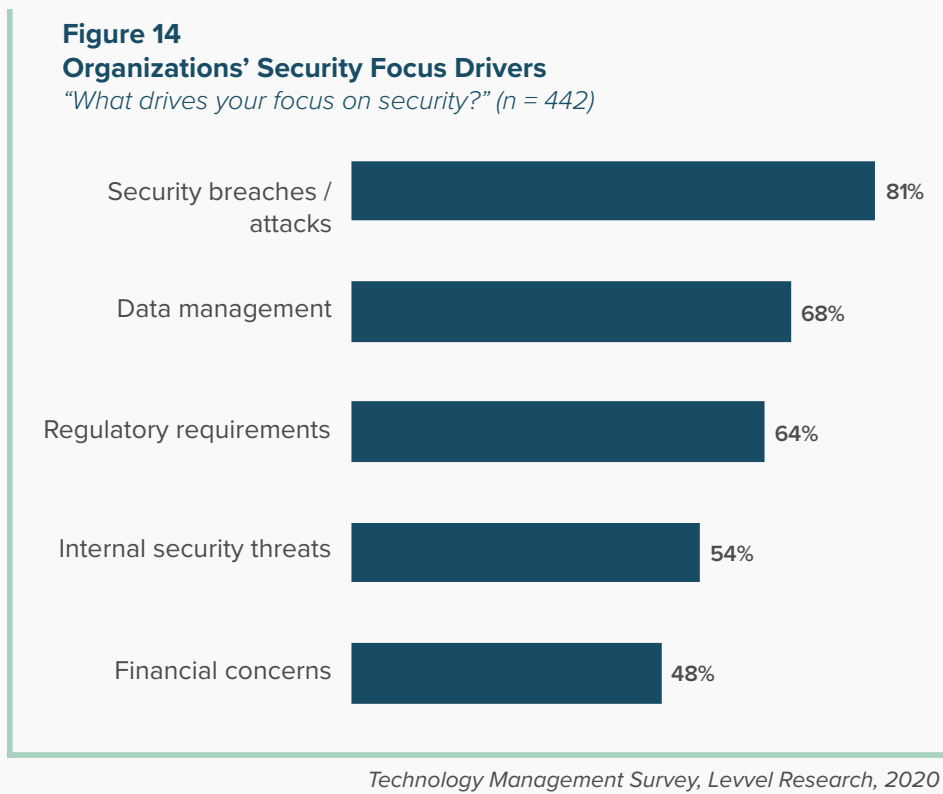


Figure 14 shows that drivers that inform organizations' focus on security; breaches and attacks were the top driver, followed by managing data and compliance. Data indicates that organizations with a strong focus on security are much more likely to use various security measurements overall. The data also indicated that the weaker the organization's focus on security, the less modern they were in other areas, too.

As systems change and external technology—both friendly and otherwise—grows more sophisticated, the danger only grows for antiquated systems and the data they hold. Simple antivirus measures are no longer enough to defend against cyber threats; instead, a modern approach to security management is to align proactive security measures with stronger systems.



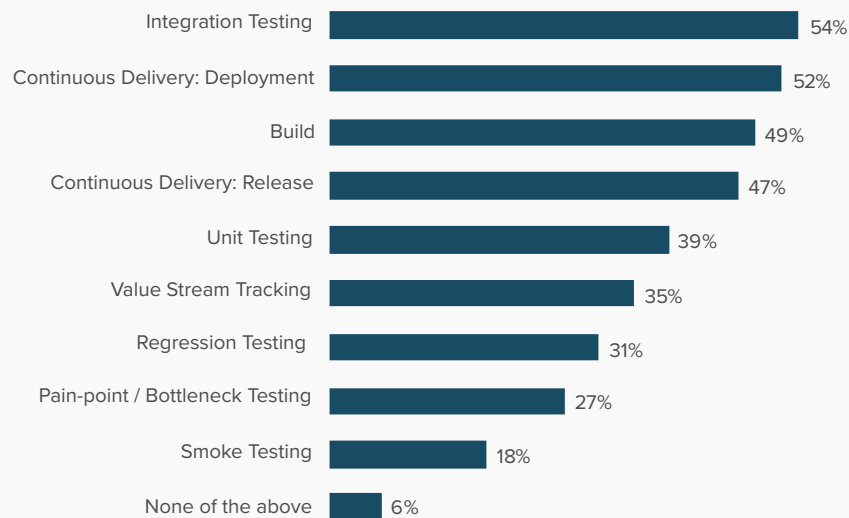
Legacy Dependence — Process

SDLC Automation

Automating the SDLC encompasses development, testing, integration, delivery, and deployment. Companies with legacy dependence typically use manual-based methods for preparing and releasing their code. For example, code deployment typically involves manually copying code to servers and recycling the application or server. Modernizing or partially-modern companies typically automate some testing and build, but true modernization entails fully automated testing, build, and deployment / release to the point of Continuous Delivery, effectively enabling Continuous Integration/Continuous Deployment (CI/CD) methodology in their software pipelines.

Figure 15 shows that about half of respondents have implemented at least one aspect of Continuous Delivery.

Figure 15
Automated SDLC Methodologies and Tests
"Please specify which of the following software development methodologies and tests are automated." (n = 307)



Technology Management Survey, Level Research, 2020

CI/CD is a core tenet of a DevOps lifecycle. DevOps principles allow software teams to quickly respond to customer needs in their software development; therefore, CI/CD pipelines are designed for continuous testing and release of high quality code. Continuous integration provides tools and frameworks for building and testing software in isolation before introducing other collaborators, ensuring that defects are detected quickly, while continuous deployment entails coordinating software releases into the appropriate environments (i.e., non-production and production) to ensure readiness and success. Modern organizations understand the value of CI/CD for maintaining competitive, market-responsive products, as well as for quickly responding to unforeseen challenges such as security breaches.

We don't know if changes can be done without breaking something.

Testing automation is important in unlocking the value of SDLC automation, as it ensures the code that is being developed and deployed is less likely to negatively impact other applications or pieces of code. Manual testing is also time-sensitive and susceptible to human error. Alternatively, testing automation reduces the chances of bugs and outages by validating the code against a set of unit and integration tests, ensuring it continues to yield

the same results although changes have been made. Figure 15 shows that the most commonly automated test is integration testing, which is used to provide smooth cohesion between multiple pieces of software. Unit testing³ is valuable for its ability to quickly check and provide feedback on small units of code against their dependencies within the same application and against business logic, while regression testing enables swift confirmation that new changes to released code have no negative impact. In all, testing automation provides speed to the SDLC process without sacrificing control, constantly ensuring that team's changes are compatible with strong, secure, and competitive products.

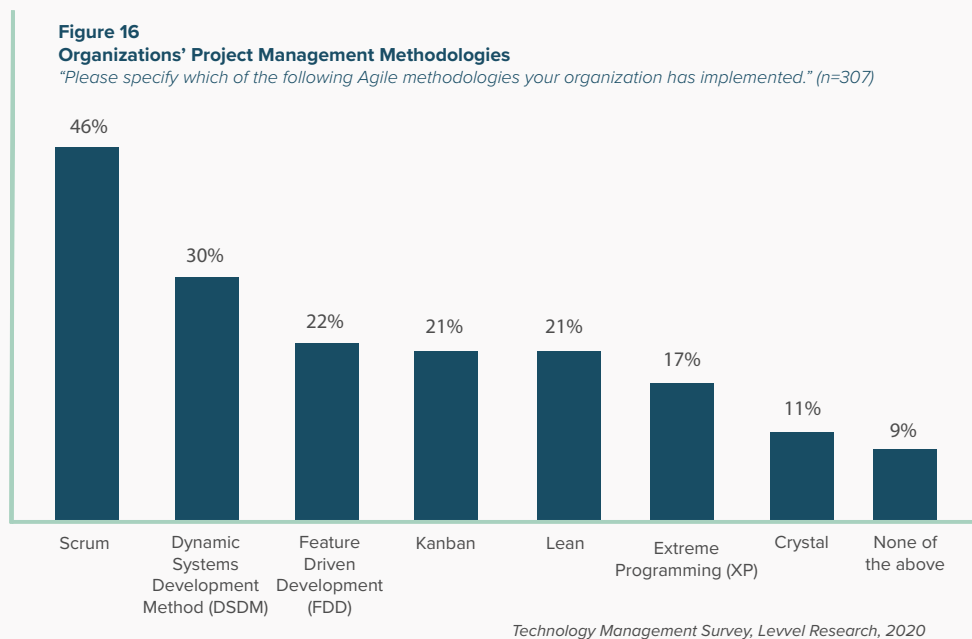
Delivery Project Management

Leveraging the right delivery project management methodologies is critical to improving cross-team integration and building a more modern culture. There are two primary project management approaches that technology teams use in software development, Waterfall and Agile, and which of these companies use is a good indicator of whether they are modern or legacy dependent.

³ "Ensure Long-term Code Quality and Prevent Future Issues with Unit Testing", Ian Duckwoth, 2018, Level.io



Figure 16 shows the Agile methodologies that respondents use, with Scrum being the most common. Over 10% of respondents are likely using traditional project management styles like Waterfall, which is a common methodology used across many industries and processes. However, although many technology teams still use it, industry standards no longer deem Waterfall acceptable for the software development process — primarily because it was not designed with software development in mind. Agile principles and methodologies are considered “technology-centric “ project management.



Agile is a framework for managing SDLC processes that encourages organizational flexibility and collaboration, and is another important part of enabling a DevOps process. Agile methodologies, like Scrum, involve numerous short and iterative stages so as to identify and address issues as the project progresses rather than when it is over. This also allows a reduction in the cost of change, due to the frequency with which the work is checked as having business value. This is key for technology and product teams, as software development involves many pieces of interconnecting and dependent code and interopating systems and applications, as well as numerous stakeholders with different roles and goals. Because of the continuous review, transparency, and quick correction, Agile methodologies lead to higher quality code and more responsiveness to issues with in-production or released software. While Scrum is an effective tool, modern organizations typically implement several practices from different Agile methodologies so as to enable a versatile SDLC process that accounts for different teams, product types, and business objectives.



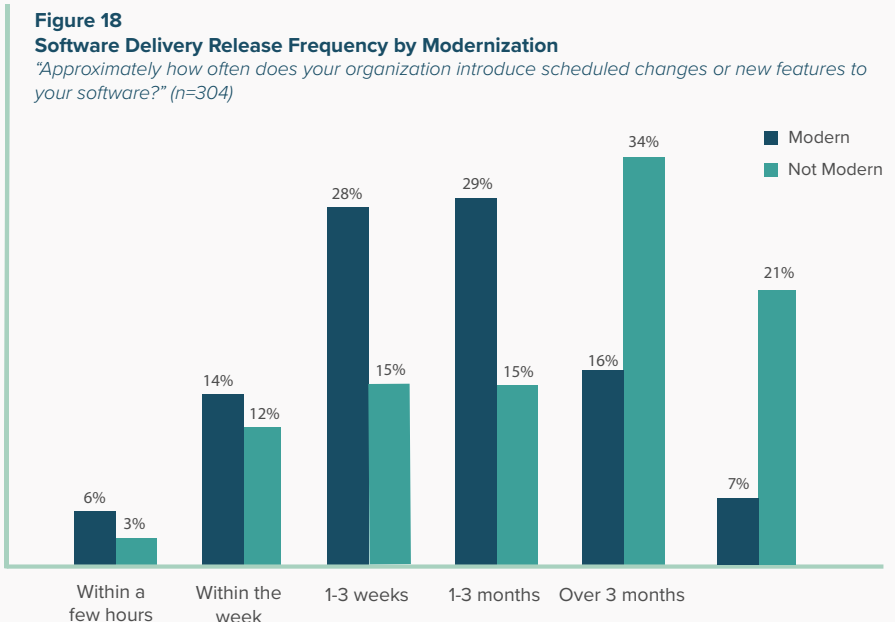
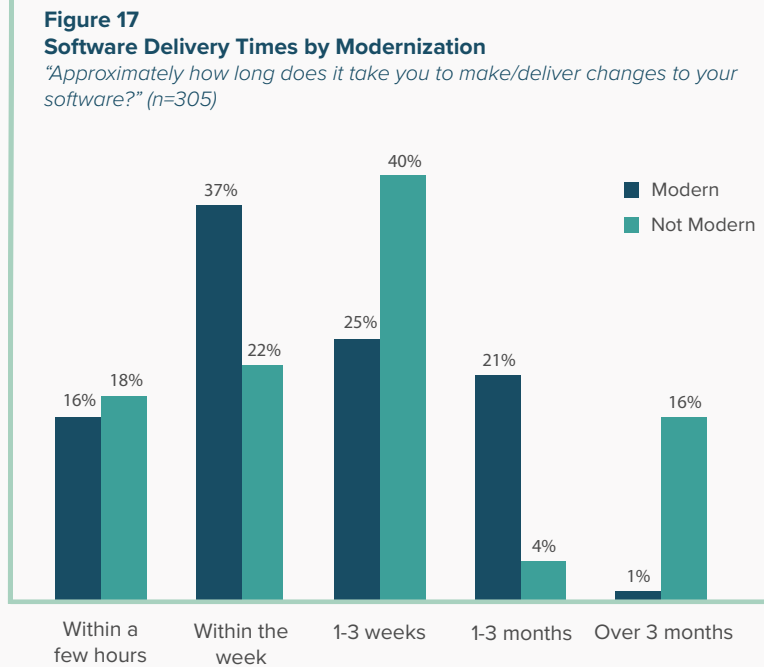
Software Release Delivery Times & Scheduling

The effect that legacy dependence has on product delivery can be seen in a company’s software delivery times. Modern company delivery times are skewed toward efficiency, whereas less-modern companies were significantly more likely to report that delivery takes at least three months more than it does at modern companies.

Per **Figure 17**, over half of modern companies could deliver within the week or sooner, while the delivery time of less-modern companies was longer.

A factor that influences delivery times is when software releases are actually scheduled.

Per **Figure 18**, modern companies are able to perform scheduled changes more frequently on average than less modern companies. This implies a more agile, scalable, and automated SLDC management process that enables faster code releases. Alternatively, companies that struggle to modernize tend to make scheduled changes less frequently than more modern companies.

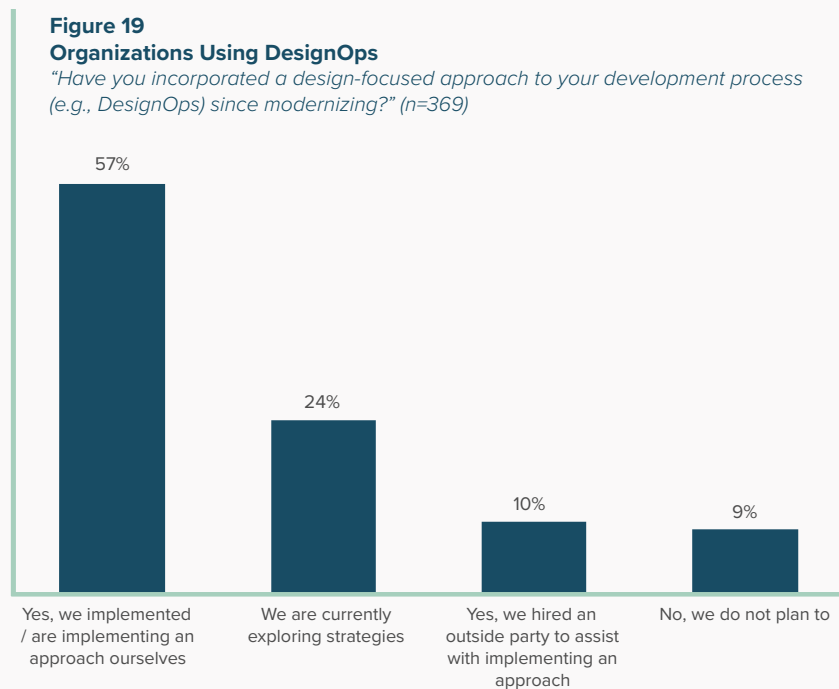


Technology Management Survey, Level Research, 2020



Design-Focused Delivery

Another indicator of modernization lies in how software design processes are structured at an organization, specifically when technology teams have adopted a “design-focused approach.” This is broadly associated with system maturity and innovative practices, and a majority of respondents reported that they were exploring or have already implemented a design-focused approach in their technology processes (**Figure 19**).



Technology Management Survey, Level Research, 2020

This practice, officially called DesignOps, arose under the umbrella of DevOps. Like DevOps, DesignOps is designed to streamline the connection between developers and other internal teams, as well as to remove antiquated and inefficient operational processes and bottlenecks⁴. And, while DevOps speaks to the relationship between development and IT operations, DesignOps empowers designers to work better and smarter with their colleagues. In relation to the design of IT, DesignOps entails creating product and development teams that understand different design disciplines and even possess different design specializations, such as visual design, interactive design, and so on; implementing processes like rapid iteration, prototyping, and usability testing; and leveraging the right tools, such as design systems—like style guide documentation and component libraries—as well as modern software that enables scalable design strategies.

⁴ “Intro to DesignOps”, Devin Smith, Level.io, 2018

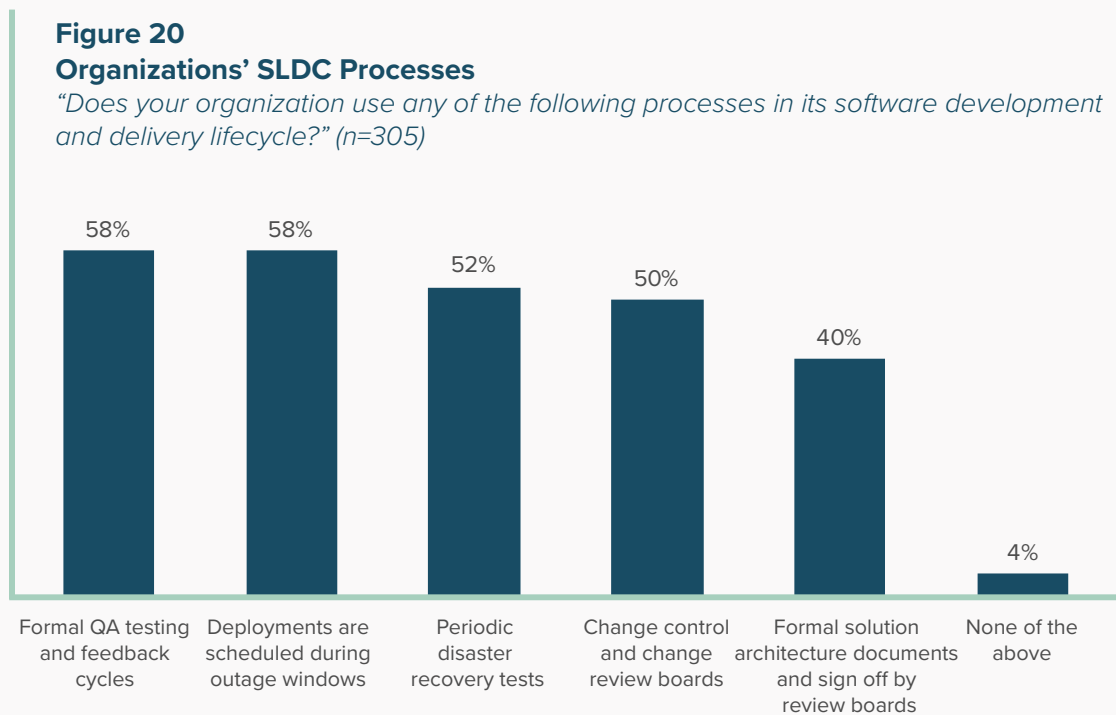


SDLC Processes

Antiquated methods of software development have followed many technology groups into the 21st century, thereby hindering their ability to adapt to an increasingly digital, globalized, and competitive business environment. There are several common, established processes and procedures that work against the objective of delivering features and critical changes faster.

The ways in which organizations structure their SDLC process has a significant influence on product development. For example, even if a technology team uses CI/CD, fully automated testing, and Agile methodologies, outdated methods of releasing the end product deducts from the efficiencies gained by those tools. Furthermore, the use of outdated methods is often associated with a desire for more control and quality, but their incompatibility with today’s tools and best practices creates the opposite effect.

Figure 20 shows the established processes and procedures that work against the objective of quickly delivering features and critical changes. Most organizations use one or more of these processes; only 4% use none of them.



Technology Management Survey, Level Research, 2020



Most of these methodologies are related to controlling changes to software, and many organizations leverage formal QA testing and feedback cycles, which require extensive testing on applications due to unknown changes to the business-critical system that the application is coupled with. Other common processes include code change control and change review boards,

[Modernization] has greatly reduced turn around time for getting fixes and enhancements to our clients and increased visibility into the quality of our software.



which involve a multi-step and heavily involved process to remediate the risk that code changes can have on an organization's other systems—risk that is heightened for monolithic, legacy systems. In this situation, even minor changes have the potential to affect a great deal of business logic, so it's important to ensure that a change is self-isolated.

However, conducting these testing cycles, reviews, and change management steps in the

traditional, manual way is not conducive to rapid software development. Alternatively, the modern approach is to implement software and project management methodologies that streamline this process. This includes setting up systems that allow developers to make isolated changes easily and without affecting other critical applications, which can then automatically track and control these changes.

Formal architecture review boards are also an inhibitor to productivity because they involve extensive reviews of documents and lengthy processes. While they are likely to remain a fixture in the SDLC process for most organizations aiming to ensure consistent architecture, the review process itself can still be made more efficient. For instance, modern organizations embrace design iterations that encourage self-evaluation and practical feedback, thereby allowing architecture to be changed based on what is learned while building the application.

Additionally, many companies schedule code releases only during outage windows, which are not always planned on a regular basis. Consequently, this restricts the ability of the development teams to react to issues swiftly, and hampers a company's own agility in the market when releasing products to consumers. The modern approach to this is to implement Continuous Delivery so that code is implemented as soon as it is tested and ready, and when it best serves the organization's product strategy.



Interdepartmental Collaboration

Collaboration in SDLC management— particularly interdepartmental collaboration—is another crucial indicator of modernization. Unfortunately, less-modern companies suffer from a lack of interdepartmental/global collaboration and communication, as well as not having cross-functional teams—each with their own specialization of productivity and tasks (e.g., Product, SDET, Design, DevOps, Engineering)— work together on a daily basis. In this setting, there is greater risk in software development and technology management, such as the risk of making detrimental changes in code, and the process of ascertaining technical requirements and constraints may be delayed. In contrast, modern companies leverage constant collaboration, which continues to move objectives forward and eliminates cycles of review and development changes. This also allows everyone to see changes together and provide frequent feedback.

Talent Management

Talent management and staffing—the cultural aspects of legacy modernization—are also noteworthy signs of modernity. Specifically, there are two staffing challenges that come from legacy dependence: expertise and quantity. That’s because staff who are knowledgeable enough to maintain legacy systems and outdated technology stacks are increasingly hard to come by. Plus, their retention becomes even more difficult when relying on legacy systems, as employees want to keep their skills relevant and continue to be challenged.

Legacy dependence stifles innovation as well, and makes it more difficult to attract talent that has experience with more modern and innovative ways of implementing technology. Along the same lines, legacy systems also tend to require more hands to maintain and run, and could cost companies a premium because of the relative scarcity of candidates fluent in outdated tech stacks. The most commonly reported example of this is organizations’ difficulty in finding COBOL-proficient talent. As industry veterans begin to retire, new graduates in technology fields are not learning COBOL as part of their curriculum, leaving a growing hole in the job market.

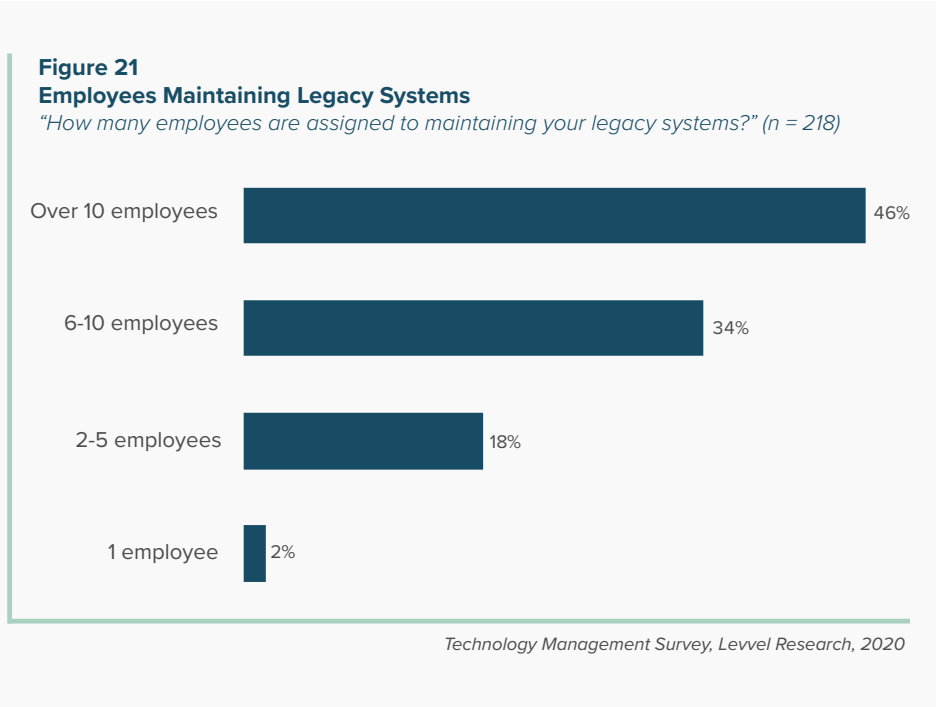


“ We're having trouble maintaining a talent pool knowledgeable of our existing technologies.”



The general costs of staffing employees to maintain legacy systems—in addition to the premium rates required for specialized skill sets in some cases—are significant. **Figure 21** indicates that the greatest portion of organizations typically assign over 10 employees to managing legacy systems;

at an average rate of \$35 an hour, this can be a substantial annual expense. On the other hand, modern systems are designed to require less maintenance and involvement from IT teams as a result of better vendor support and more automation, and by replacing these systems, organizations can reallocate their employees to higher value activity, such as product development.



Cultural Transformation

Some companies face challenges related to cultural change and the acceptance of new technology, which greatly inhibits their ability to optimize their process or deliver products efficiently. Specifically, managers and/or leaders who don’t have the right mindset on technology or don’t support improvement initiatives fail to leverage industry best practices. A lack of openness and communication between technology teams and other business units can also lead to silos in SDLC processes and cause problems down the line. For example, teams that don’t involve legal, security, compliance, and risk early on in the design and initial gathering phases may be forced to revisit crucial elements later in the process—when changes are more costly.

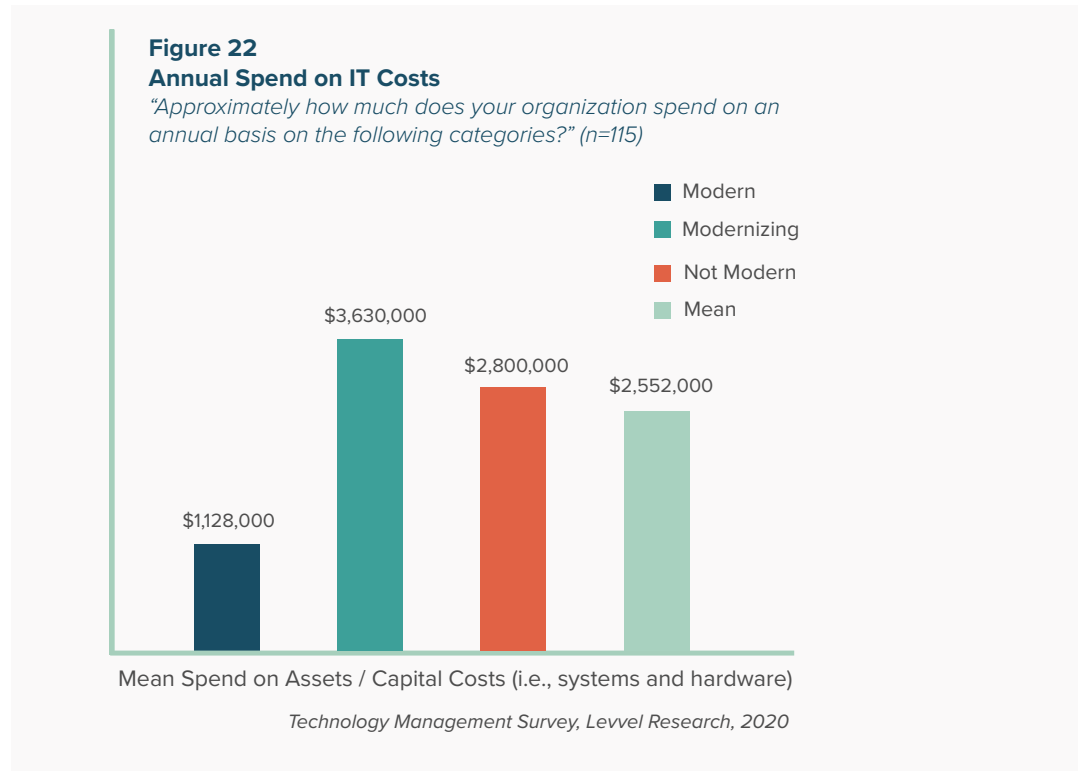
Therefore, proper management of systems and managerial buy-in for modernization are critical to ensure the development of a reliable environment for sustainable, accelerated product delivery. The cultural shift associated with a successful legacy modernization initiative is also critical to the meaningful reduction of technical debt—and not just in the days and weeks following the installation of new technology. Rather, a culture of modernization must be performed continuously in order to be successful.

Costs

High costs are a symptom that every business group understands intimately. Per **Figure 22**, companies that are in the midst of their modernization or struggling to modernize spend⁵ more on their systems and hardware annually, on average. The reason for this is the maintenance of legacy systems and the manual processes associated with a lack of modernization.

Companies that are currently working toward modernization have the highest mean spend. This is likely due to the fact that legacy modernization initiatives are costly in the short run, and thereby inflate their annual spend as they strive for the comparatively low mean spend of modern companies. However, the

majority of modern companies spend, on average, less than half of the overall mean on their systems, which indicates that modernization reduces costs in the long run.



Diagnosing Legacy Dependence

Once an organization understands the elements of legacy dependence, it's easy to see the effect that modernization could have on their business success. What's more difficult, though, is assessing exactly where legacy systems and processes are creating barriers to that success. In many cases, when companies lack a full and accurate picture of their legacy dependence, they begin modernization initiatives and encounter complexities that cause delays. Or, they complete the initiatives only to end up with disappointing results. This is because they identified and targeted only part of the problem.

⁵ Costs were calculated with the mean spend across five categories: labor (employees dedicated to managing cloud tech or on premise systems), assets/capital costs (systems/hardware), new product development, vendor costs, and licensing/support

When diagnosing why they're unable to deliver software to market faster, many organizations simply focus on the symptoms without properly identifying the causes. For example, if development teams are unable to deploy new code quickly enough, they may interpret this as a lack of testing automation. While a lack of testing automation may indeed be a factor of legacy dependence, more often than not, it's only a contributor and not the root cause. In many cases, the root cause is that the organization's applications are outdated and, therefore, not even designed for automated testing. It's important to consider the holistic relationship of legacy systems and processes in order to determine the true causes of product development and delivery failure.

Several common symptoms may indicate an area where legacy systems restrain a business. In particular, maintenance of and dependence on legacy systems and processes is often the true cause of many process failures in software development and delivery. **Table 1** illustrates the connections between a company's legacy dependence (causes) and its technology issues (symptoms).

Table 1

SYMPTOMS	SURFACE CAUSES	FOUNDATIONAL CAUSES
Slow code release schedules	Lack of CI/CD automation tools	Processes, infrastructure, and culture not designed to support SDLC automation tools
Unplanned outages	Human error missed during review and testing	Lack of testing automation and use of best practices in the SDLC process
Talent management challenges	Talent is expensive and hard to come by	Systems' technology stack is built from outdated code, for which skilled talent pools are decreasing
Security breaches	Not using proper security protocols or lack of dedicated security personnel	Dependence on systems not designed for modern cyber threats

Understanding these signs of inefficiency is key to sustainable legacy modernization, although addressing the symptoms alone is not enough. Like any illness, treating the symptoms may help the patient feel better, but it won't ensure that the issue doesn't lead to further complications. The next sections identify proper technology applications and solutions to legacy dependence.



The Road to Legacy Modernization

Legacy Modernization Initiatives

When organizations decide to modernize their technology environment, very few address all areas at once or use the same practices. Instead, the path varies by company and depends on their current state, constraints, business characteristics, and priorities. In general, organizations take a phased approach to launching initiatives in different legacy modernization areas and use different modernization practices in those projects accordingly.

Data shows that organizations typically begin initiatives in a different order than they complete initiatives, as illustrated in **Table 2**. For example, many companies begin with cloud migration or removing legacy platforms in the early stages of their modernization efforts, but they are more likely to fully implement design-focused delivery processes first. This is because being fully modernized in certain areas, like cloud migration, often takes much longer than modernizing other areas, such as implementing DesignOps.

Table 2

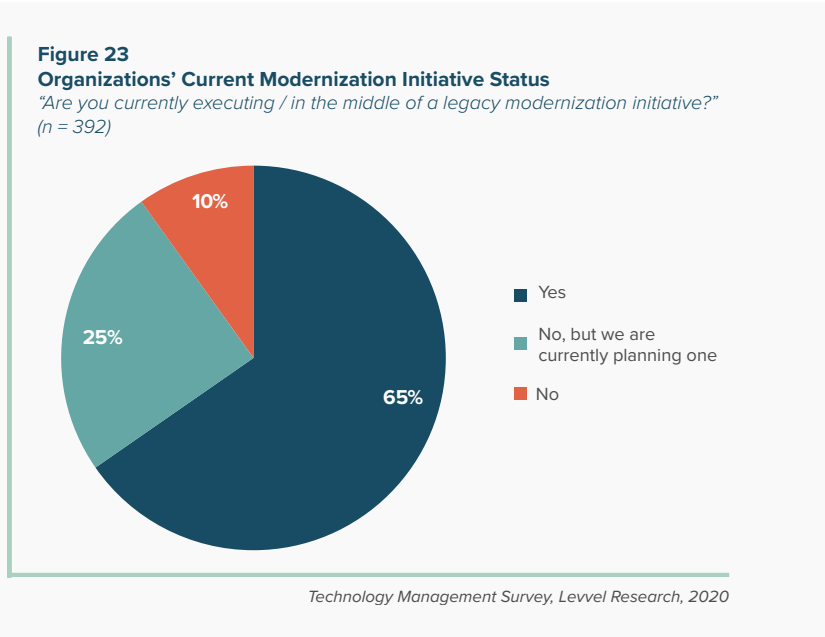
ORDER	COMPANIES BEGIN INITIATIVES	COMPANIES COMPLETE INITIATIVES
1st	DesignOps Integration	DesignOps Integration
2nd	Platform Modernization	SDLC Automation
3rd	Cloud Migration	Agile Methodologies
4th	SDLC Automation	Platform Modernization
5th	Agile Methodologies	Cloud Migration
6th	SDLC Management Optimization	SDLC Management Optimization



In addition to ease of execution, the speed at which companies reach a fully modern state in these categories also depends on the nature of the projects—some initiatives go hand in hand. For example, data shows that it’s very common for organizations to implement SDLC automation and Agile methodologies simultaneously, but, although they’re often pursued at the same time, companies tend to achieve SDLC automation first. That’s because implementing SDLC automation is a more technical project, while Agile modernization is dependent on broader factors, including process and workload restructuring and cultural change.

A company’s size and age also plays a role in where it chooses to focus. For instance, younger,

smaller companies achieve full cloud migration and legacy systems replacement sooner than larger, older companies. This is likely due to the differences in scale of the systems that need to be updated, as well as the technology debt between each group. Larger companies are more likely to prioritize improving software delivery first and, consequently, achieve full testing automation and Agile adoption earlier than smaller companies.



It’s worth noting that all companies surveyed were modernizing in some way; the majority of respondents said they were in the middle of an initiative, whereas one-quarter were planning one (Figure 23). As for those respondents who had not yet begun a legacy modernization initiative, 70% were planning to do so in the next year.

Drivers to Modernize

Drivers to modernize typically come from both internal and external forces and issues. External examples include major events like the COVID-19 pandemic, which can create urgency to invest in technology that brings more durability to market disruptions, or new or adjusted regulations that bring interest in systems designed specifically for compliance and security. It can also come from the danger of becoming market obsolete, inspiring a company to update their product development processes and become more competitive. Internal examples include high costs,

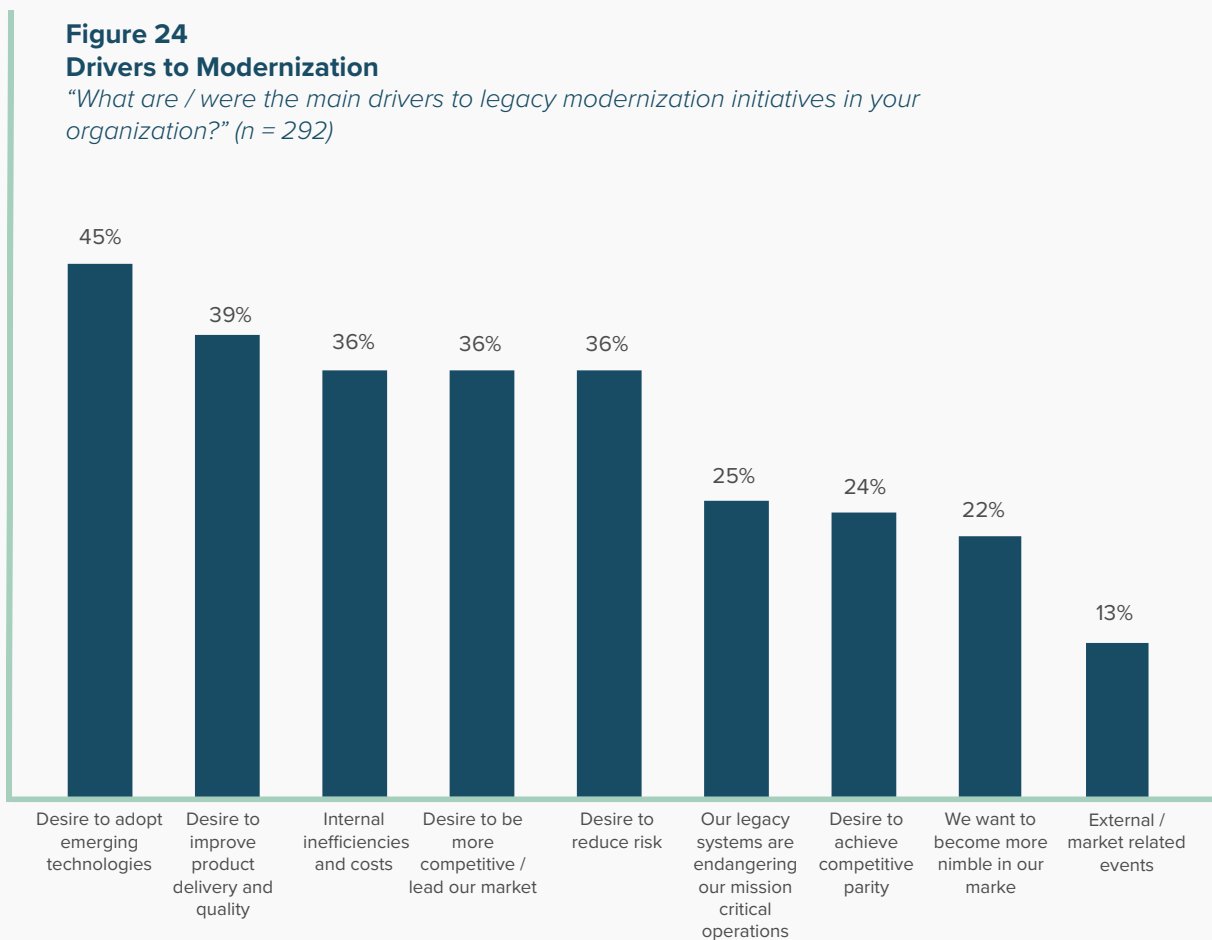


difficulty sourcing and retaining talent, and of course, the many inefficiencies (symptoms) associated with legacy dependence. While originating in different places, these factors are closely related and share a common solution — legacy modernization.

Figure 24 shows that survey respondents in leadership positions (management and executive roles) report drivers that include many of the above scenarios, and that respondents are particularly focused on maintaining competitiveness and lowering risk. The majority of respondents see the value in modernizing for market agility and creating stronger products.

Figure 24
Drivers to Modernization

“What are / were the main drivers to legacy modernization initiatives in your organization?” (n = 292)

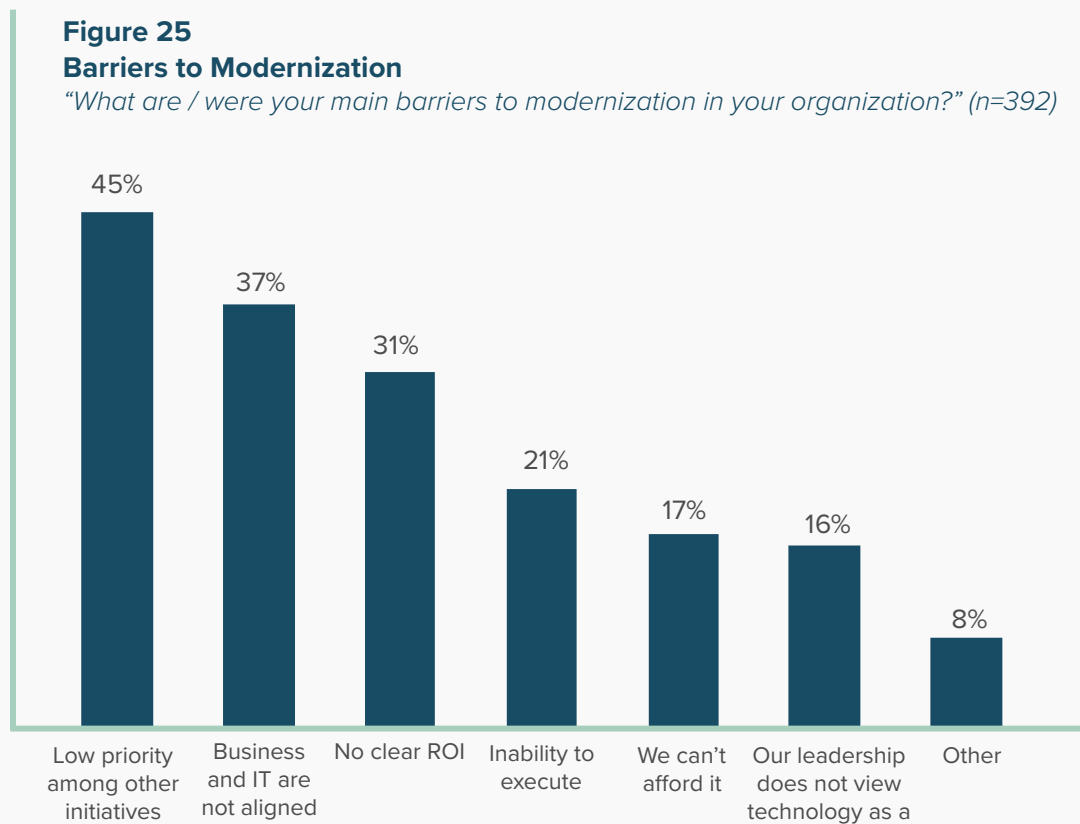


Technology Management Survey, Level Research, 2020



Barriers to Modernization

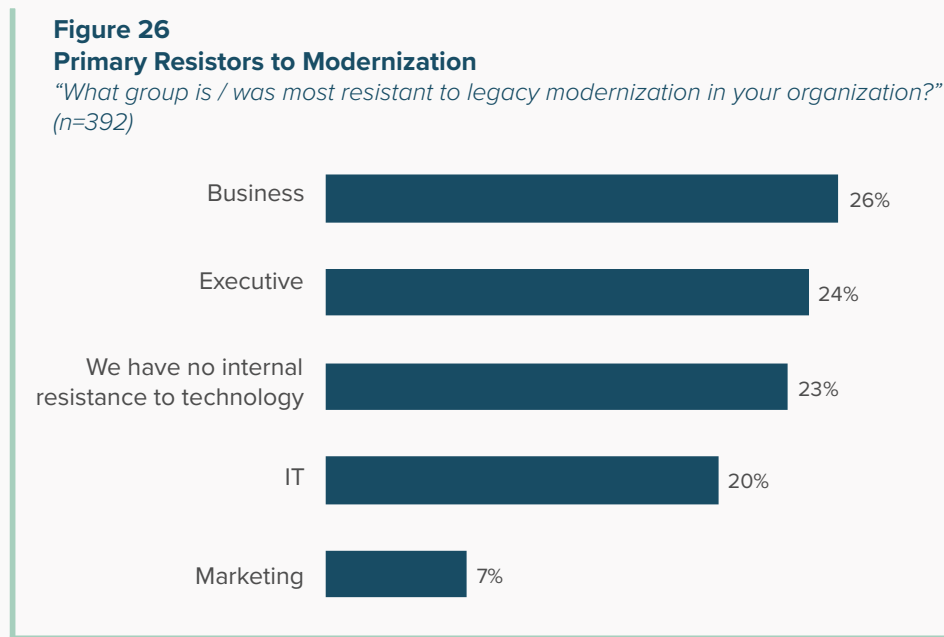
Despite the impetus to modernize, significant barriers and obstacles remain that prevent companies from realizing their modernization potential. Namely, the top-reported barrier to modernization cited by respondents in leadership roles was a “low priority among other initiatives” (**Figure 25**).



Low prioritization is largely due to lack of buy-in from the key decision makers, and in many cases business and executive roles can be the primary resisters against modernization. Unclear ROI is another common barrier, and one of the best ways to increase the priority of legacy modernization is to solve the ROI issue first, revealing the cost benefit of the initiative. This can be done by drawing a clear connection between legacy modernization and the symptoms they produce, such as more outages, slower software delivery times, and higher costs.



A lack of shared sentiment towards the initiative can also be an issue— misalignment between business and IT was the second top barrier. **Figure 26** shows that a substantial segment (20%) of respondents reported IT as being the most resistant to modernization. Interestingly, those whose business-critical functions ran primarily on legacy systems were almost twice as likely as those who were migrating/have migrated to report resistance from IT as the group that prevents modernization.



Technology Management Survey, Level Research, 2020

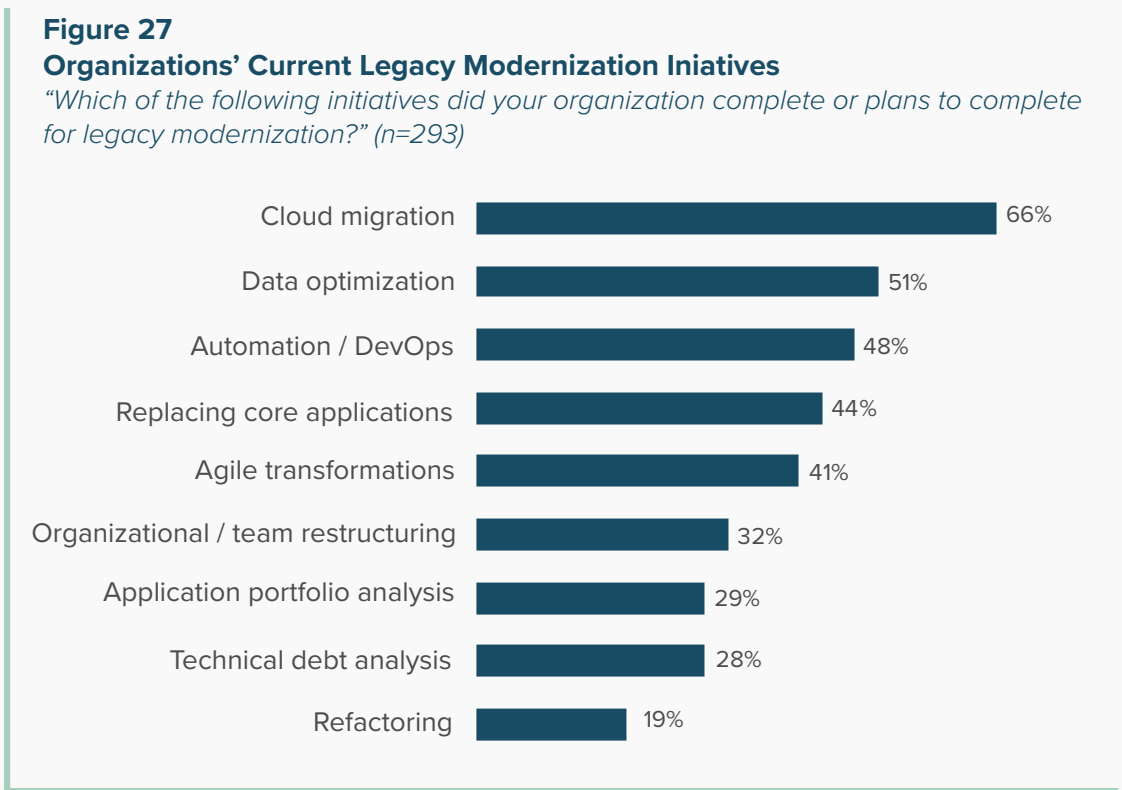
Resistance from IT can seem counter-intuitive, as modernization greatly benefits technology professionals and processes. However, there are many factors that can create a sense of resistance, such as a lack of trust between IT and other groups, or a lack of resources for the work required. It can also come from IT's distaste for initiatives that are externally driven rather than those originating from IT, based on their own goals and challenges. Obstruction to modernization by IT groups may also be prevalent because the tenured employees have experience with the long-standing systems, and may be wary of newer technology stacks.

A lack of understanding and coordination between groups in external initiatives creates friction and resistance to change. Consequently, achieving buy-in on a cross-team basis through collaboration, openness, and trust is key to a successful modernization.



Legacy Modernization Practices

There are several common approaches to modernizing legacy systems that can be pursued based on business needs. When organizations begin to move toward modernization initiatives, they'll engage in several different smaller initiatives and practices. **Figure 27** shows some of the most common practices that organizations use for legacy modernization.



Technology Management Survey, Level Research, 2020

While many of these initiatives have been touched on earlier in this report, such as cloud migration and Agile transformation, the rest are covered briefly below.

- **Application portfolio analysis** — This is evaluating the organization's current set of applications and deciding on the effective action for each application in the modernization process. This analysis typically involves an application rationalization process in which organizations try to identify which apps they should modernize first according to characteristics like revenue generation, risk level, total cost of ownership (TCO), and mission criticality. App rationalization processes provide guidance on where to start, who to get buy-in and involvement from (e.g., governance for high-risk apps), and the impact on the business



internally and externally. Metrics are important to the application portfolio analysis process, as is patience — the analysis can be seen as a time-consuming and tactical necessity rather than a key component of digital strategy. Organizations should recognize this analysis as a fundamental building block to sustainable modernization.

- **Refactoring** — This is rewriting and restructuring code in accordance with updated technology languages, frameworks, and design best practices. In many cases, organizations can use refactoring to update existing applications rather than replacing them altogether, allowing them to update their code and product without as much impact or cost on operations. Organizations can identify which applications can be refactored during the application portfolio analysis.
- **Technical debt analysis** — This is looking at the total costs of maintaining current operations versus replacing, refactoring, and restructuring them. The type of technical debt analysis depends on the scope in question. Some organizations or teams will prioritize platforms based on the efforts defined during an application rationalization exercise. If a team is focused on a single platform to start, technical subject matter experts can focus on why technical debt appeared. A popular mechanism for this analysis is Martin Fowler’s technical debt quadrant⁶. Once it is understood why this technical debt exists, the teams can begin the work of prioritizing paying down specific types of technical debt based on business goals and needs. Most organizations will focus on more than a single platform and will need to utilize their application rationalization framework to prioritize and program plan parallel efforts to support broader company strategy.

⁶ "Technical Debt Quadrant", Martin Fowler, 2009, martinowler.com



Legacy Modernization Strategies

Modernization initiatives should be addressed with a focus on fixing causes rather than symptoms. Below are strategies for improvement that consider the holistic impact of legacy dependence and how to replace it with sustainable, scalable methods.

Gain buy-in to automate by connecting process to product. One issue that inhibits successful modernization is the lack of alignment between business and IT. Specifically, the business units of many established companies would like to innovate in ways that are expected by consumers used to using the software of tech-based companies. These business units then lean on their IT partners to deliver new features and enhancements, but the IT staff is still delivering software using traditional, established processes and procedures. Business leaders fund IT initiatives that IT teams are unable to deliver on time because they are saddled with legacy processes, procedures, and technology. Therefore, it's important that both teams understand the fundamental changes needed in order to enable IT to achieve the business objectives — namely automating as much the SDLC as possible. Initiatives and momentum alone cannot force innovation through a broken process.

Adopt a product-first approach across all organizational processes. For organizations to truly use technology to improve their business success, they must have the ability to develop and modify their product directly. There are situations where organizations can procure and leverage a third-party solution that still allows flexibility for such needs, but it mostly exists with more modern solutions that are API first. When developing a product in-house, organizations should take a “product-first” approach, which often means installing a designated product team. This gives technology organizations full ownership of the product, and allows them to align the product to the business goals and the technical implementation.

Adopt an Engineering Culture and focus on innovation, open source, and talent. An Engineering Culture allows technology teams flexibility, autonomy, and the ability to impact the business through their creativity. Technologists need cultures that empower them, allow their voices to be heard, and encourage them to express their ideas and implement solutions. Organizations will not be able to grow or attract the top creative minds without creating some tolerance for risk and failure. Some ways that detract from a creative culture are strict rules on which tools and languages teams must use. Teams should be able to support smooth procurement and approval processes for leveraging newer and open source technology; it is also critical that teams have the ability to create proof of concepts. Adopting an Engineering Culture

not only improves the productivity and innovation of current teams, but opens up possibilities for future talent acquisition and management.

Create a clear modernization strategy. Mapping and planning helps companies strategically choose their tech stacks, align with procurement, outline costs, and take other actions that create a comfortable state for getting started. Steps include:

- **Current state evaluation and discovery workshop** in which SMEs, stakeholders, and developers are gathered to assess where to start
- **Enterprise Architecture evaluation and framework building**, especially important for larger organizations, helps to properly measure and map out the impact that initiatives will have on the overall technology architecture at the enterprise, as well as align them with the enterprise’s strategic architecture design goals for the future
- **Cost analysis**, including determining the consolidation of resources that can occur and performing a tech debt analysis
- **Prioritization**, including performing app rationalizations and determining budgets
- **Target state mapping**, including creating a roadmap/list of actions and a 90-day delivery plan

Create modernization plans with a holistic approach. The question is no longer whether organizations have begun to update their technology systems and processes — most have launched and completed at least some type of initiatives. Rather, it’s important to look at how far organizations get, where they get stuck, and why. In many cases, not understanding the holistic nature of their systems and processes can impact the success of an initiative, as the organization may focus on one half of a problem without kicking off another project that improves the second half. For example, the organization could plan to transform a software review process without implementing a tool to enable it, or move an on-premise workload to the cloud without re-architecting it to work properly. These partial initiatives will not lead to the true efficiency gains of modernization, and will lower the ROI of the project. It is important to coordinate parallel efforts that ensure sustainable improvement when building a modernization plan.



That being said, coordinating parallel efforts can be daunting or difficult in complex organizations. If teams and organizations are not aligned, these efforts often fail or stop prematurely. It is important that organizations have a top-down strategy as well as bottom-up buy-in from key technology and business leaders. The overarching strategy should directly correlate to business objectives so that business and technology teams are not working from different playbooks.

Additionally, it takes significant time to cultivate a culture, produce buy-in, and ultimately plan a holistic effort. Combining strategic leadership and tactical program management is a must. Parties should not get discouraged, as there are examples where organizations have done this well, but it is best to set expectations that it is a longer term process — although one that yields significant benefits.



Modernization Assessment

Even armed with the right information regarding the factors and effects of legacy dependence, it’s still difficult to truly ascertain the amount of legacy dependence at one’s own organization. So, to assist in this effort, this report includes a rubric that readers can use to score themselves and determine their own modernization persona. From that starting point, a company gains a more holistic picture of where it stands on its road to modernization, and where to go from there.

Table 3 below is a self-assessment rubric for modernization in 6 technology areas (Table 3 continues on pages 44 and 45).

Table 3

DESIGN APPROACH	Modern organizations have implemented a design-focused approach in their software development process (e.g., DesignOps)	
<i>Have you incorporated a design-focused approach in your development process?</i>	Yes, we have fully implemented (5 PT)	Modern: 5PT or more
	We are currently implementing (5 PT)	Modernizing: 2PT or more
	We are exploring strategies (2 PT)	Not Modern: 0-1 PT
	No, we do not plan to (No PT)	TOTAL POINTS: _____
SYSTEM AGE	Modernization entails updating all business-critical applications and workloads that are running on legacy systems (i.e., those over 10 years old)	
<i>Would you say your company is heavily dependent upon legacy systems for business-critical functions?</i>	Yes, our business-critical functions are heavily reliant on legacy systems (No points)	Modern: 5PT or more Modernizing: 1PT or more Not Modern: 0 PT
	We have some reliance on legacy systems, but are currently modernizing (1 PT)	
	Our business-critical systems are primarily running on modern systems (2 PT)	
	Our entire infrastructure is primarily running on modern systems (5 PT)	TOTAL POINTS: _____



CLOUD MIGRATION	Modernization entails migrating key workloads to cloud-native environments, as well as re-architecting the solutions to operate successfully in the cloud (i.e., no “lift and shift” approach)	
<i>Which of the following statements best describes your organization?</i>	We are a fully cloud-native company (5 PT)	Modern: 5 PT or more Modernizing: 3 PT or more Not Modern: 1 PT
	We have embraced or are implementing a hybrid cloud model (3 PT)	
	We are mostly migrated to cloud and are in the process of continued migration (2 PT)	
	We have migrated a few workloads to the cloud, but are hesitant to migrate all of them (1 PT)	
	We would like to be primarily or fully cloud native, but struggle to execute on migration (1 PT)	
<i>As you have migrated your workloads and applications to the cloud, have you optimized (rearchitected) them for the cloud environment?</i>	Yes, we rearchitect all our workloads / applications as we migrate them to the cloud (3 PT)	TOTAL POINTS: _____
	We do for some workloads / on a case-by-case basis (2 PT)	
	No, we typically do not substantially rearchitect our workloads (No PT)	
SDLC AUTOMATION	Modernization entails automating testing, build, delivery, and deployment in the SDLC	
<i>Which of the following tests and processes have you automated?</i>	Build (1 PT)	Modern: 5PT or more Modernizing: 0.5-4 PT Not Modern: 0 PT TOTAL POINTS: _____
	Unit Test (0.5 PT)	
	Integration Test (0.5 PT)	
	SmokeTest (0.5 PT)	
	Regression Test (0.5 PT)	
	Metrics-- Pain point / Bottleneck identification (0.5 PT)	
	Metrics-- Vaue Stream Tracking (0.5 PT)	
	Continuous Delivery -- Release (5 PT)	
	Continuous Delivery -- Deploy (5 PT)	
	None of the above (No PT)	



PROJECT MANAGEMENT METHODOLOGIES	Modernization entails replacing traditional methodologies, like Waterfall, with technology-centric approaches, including Scrum, Agile, and Kanban	
<i>Which of the following Agile methodologies have you implemented?</i>	Scrum (5 PT)	Modern: 5PT or more Modernizing: 1-4 PT or more Not Modern: 0 PT TOTAL POINTS: _____
	Lean (5 PT)	
	Kanban (5 PT)	
	Extreme Programming (1 PT)	
	Crystal (1 PT)	
	Dynamic Systems Development Method (DSDM) (1 PT)	
	Feature Driven Development (FDD) (1 PT)	
	None of the above (No PT)	
SDLC PROCESS OPTIMIZATION	Modernization entails moving away from more traditional SDLC management practices (e.g., only scheduling deployments during outage windows) that do not leverage technology or best practices to optimize reviews, approvals, and deployment processes	
<i>Does your organization use any of the SDLC processes below?</i> <ul style="list-style-type: none"> • Change control and change review boards • Formal QA testing and feedback cycles • Deployments are scheduled during outage windows • Periodic disaster recovery tests • Formal Solution Architecture documents and sign off by the Architecture Review Boards 	We use most of them / 3 or more (No PT)	Modern: 5PT Modernizing: 2PT Not Modern: 0 PT TOTAL POINTS: _____
	We use a few of them / less than 3 (2 PT)	
	We do not use any (5 PT)	



Conclusion

While this report has outlined several indications of what legacy modernization looks like, the clearest marker is this: being able to achieve business goals. Enabling this requires addressing many underlying systems, processes, and principles, but the end result remains the same because business success and organizational growth are tied directly to how a company views and uses technology. While building and executing the challenging process of modernization can be intimidating, it is also a crucial step for businesses hoping to innovate in today's market.

Culturally, not all organizations are equipped to perform such a complex and long-term transformation. It is important that organizations be introspective and understand what roadblocks exist to their own transformation. Strong leadership can help organizations perform such introspective and maintain focus through the inevitable ups and downs their organizations will face. If organizations can cultivate the right culture, can define and carry a strategic vision, and are strong in their program management, large-scale legacy modernization will produce significant results for the business for years to come.



About the Authors

Eric LaForce | Sr. Vice President of Capabilities

Eric LaForce is Level's Senior Vice President of Capabilities and is a seasoned technology executive focused on driving business value through digital transformation. Currently, Eric runs Level's shared services organization where he coaches teams to perform their best work on complex business and technology problems, drives operational excellence, and encourages a strong organizational culture based on Level core values. Eric has experience in building compelling and scalable products, modernizing legacy applications, and driving large-scale digital transformation as a modern technology leader and program manager. He has direct experience in development, building products, solution engineering, program planning, operations, sales, building businesses, and transforming culture.

Stevie Palmateer | Engineering Capability Lead

Stevie Palmateer is Level's Engineering Capability Lead and is responsible for managing the processes and people within the Engineering Capability to ensure quality work is delivered to Level's clients. Stevie also assists in leading research, strategy, and development opportunities for Level's Engineering Capability. Stevie has 10+ years of experience in the design, development, and deployment of software for Fortune 500 companies. She's a full stack application developer and solutions architect who is passionate about bridging the gap between product design and technical strategy. When not working on a technology project, she is actively mentoring and guiding young females to succeed in STEM.

James Boone | Architecture Senior Manager

Jim Boone is an Architecture Senior Manager at Level and has over 25 years of experience in systems engineering, systems management, software design/development, and payment software systems across the power generation, health care, and banking industries. He has helped both large and small clients integrate payment systems into their enterprise architecture, and he has designed and implemented custom payment software solutions.

Anna Barnett | Research Senior Manager

Anna Barnett is a Research Senior Manager for Level Research. She manages Level's team of analysts and all research content delivery, and helps lead research development strategy for the firm's many technology focus areas. Anna has extensive experience conducting and writing market research on a variety of business and technology areas.

Other Contributors

- Chris Madison
- Darren Lasley
- Jeff Levy
- Rachel Hernandez
- Kristen Janick



About Level

You're going to use technology to change the world. We're going to help you create it. Whether you are reinventing your company, creating an industry-changing product, or making existing products even better with new technologies—we exist to make your endeavor a success story.

Our experts help unleash your engineering team's potential. You know that you need to transform your software development lifecycle, and you need to move quickly. We bring seasoned experts to work with you to not only get the processes and tooling right, but to win with the human element of this critical transformation.

