SOFTWARE MODERNIZATION: SOLUTIONS, STRATEGIES, AND EXAMPLES.



Digital transformation requires being proactive. And modernizing software and applications is a critical part of any successful digital transformation. After all, legacy software applications and technologies limit your organization's ability to enable a digital-first user experience and business operations.

This sentiment rings true across the enterprise IT industry. According to a research survey conducted across 800 senior IT decision makers in global enterprises, 80% of the respondents believe that failing to modernize applications and IT infrastructure will negatively affect the long-term growth of their business.

And organizations that successfully transition from legacy to modernized infrastructure technologies should expect a 14% annual increase in revenue. These organization are also better poised to take advantage of the next phase of technology evolution. Frontier technologies—everything from artificial intelligence and blockchain to gene editing and nanotechnologies—is expected to reach $3.2 trillion by the year 2025, according to a recent UN research report.

In this article, we look at what application and software modernization means. We'll explore the opportunities and the risks, and then we discuss actionable approaches and best practices to follow.

Let's get started!

What is modernization in software development?

Application and software modernization is just that—modernizing apps and software.

In practice, it is the transitioning from existing and/or traditional software functionalities to a context that is compliant with up-to-date IT landscape potential. The most obvious example is moving an enterprise IT environment from a traditional on-premises data center model powered by mainframes to a cloud environment rich with microservices and containers.

Cloud vs mainframe

Of course, this isn't to say that every company needs to move everything to the cloud.

The mainframe, for example, continues to play a significant role for many companies. Though many, many companies in every industries are moving workloads and data storage to the cloud, not every single workload or data set must go to the cloud.

Today, thousands of companies at the global scale have critical core business processes based on corporate software that dates back some 30 or even 60 years. Most of the companies will continue to keep critical processes on the mainframe. In fact, the 2025 Mainframe Survey Report, which polled more than 1,000 IT professionals and directors underscores this:

- 90% of respondents see the mainframe as a long-term platform for growth
- 67% of extra-large shops have more than 50% of their data on the mainframe

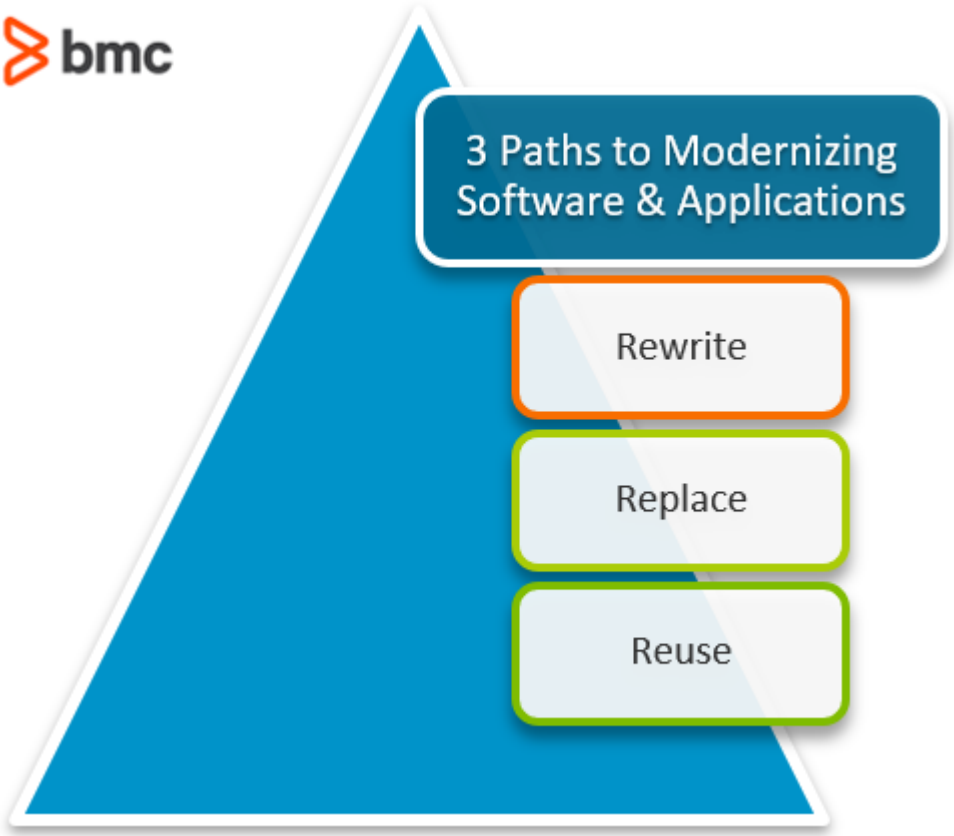
(Read about mainframe modernization.)

Software modernization strategy

When considering whether to modernize software and app, you can choose from three options:

- **Rewrite**. Migrate the coding to a more recent development language. This option has significant potential for underestimating both inherent costs and required time frame—so it often turns into a never-ending story.
- **Replace**. Throw away the old system, replacing it with a new one. This option usually involves having to build an entirely new application ecosystems with semi off-the-shelf components that mimic or mirror some functionalities of your existing systems. This usually increases complexity without achieving 100% functionality.
- **Reuse**. Find a way to turn a legacy system into something more portable, allowing remote sites and users to access it. If applicable, this option allows a step-by-step approach where change is tested and fine-tuned.

Prior to deciding the best route for your modernization project, you first need to understand the pros and cons, with a clear picture about current status and future goals.



Benefits of modernizing software

Having old software supporting business operations often means increasing your risk, increasing your costs, and severely decreasing your business agility.

So, let's look at the many benefits of choosing to modernize your apps and software.

Minimizing obsolescence

- **Losing control**. Having a part of one's core business dependent on some piece of code developed in some "old school" programming language isn't good for business—you need people today who can service that code even if it has been in use for decades.
- **No support**. Having any type of critical business processes running on hardware and/or operating systems that are no longer supported by their manufacturer or there is even no longer a manufacturer to resort to; well, hardly the position any business manager is eager to be in.
- **The missing component**. The support infrastructure no longer exists. Some older software was developed having its backup process and policies dependent on either the backup hardware or the logic inherent to some specific backup product or solution. Most likely such backup product has already been discontinued which potentially renders restore activities unfeasible.

Reducing cost

- **Choosing the lesser problem**. If a legacy system is still around, it's probably playing a critical role towards corporate core business. In most cases, replacing it comes with significant cost, both from an investment perspective and an operational perspective. (Consider the harm that stoppages could do if that system is offline.) Still, doing nothing, which may save money in the

short term, could increase your risk of unable to work at all for the foreseeable future, particularly if the hardware or operating system are no longer supported by a manufacturer.
- **OpEx vs CapEx**. Having old systems running may imply investing in discontinued components stock such as backup tapes, hard disk drives, RAM memories or resorting to high-cost contractors that leverage their niche knowledge.

Increasing agility

- **Responsiveness**. As more burden is placed on digital systems to support customers and business critical processes, some legacy systems may be causing a bottleneck and slowing over application responsiveness. It's important to examine if it's an input/output issue or a processing issue that's causing the issue.
- **Business Cycle**. Being competitive in today's market requires the ability to promptly address an unforeseen demand or release new features quickly. Some legacy and traditional systems may not able to cope with the required flexibility and speed to deploy new code quickly and efficiently.

Integration

- IT has been growing in terms of platforms served (mobile and social networks) and complexity in terms of IT landscape components—cloud, hybrid and on-premise data centers. Agile "plugin" integrability is becoming a competitive edge in business terms, which is often held back by legacy systems.

Minimizing non-alignment

- **The law**. In some cases, legislation or market regulation may imply the need of migrating an existing software so that it may comply with newly posed requirements. GDPR and related privacy legislation, for example, could force you to a newer platform that has the privacy you need.
- **The Market**. Competition is another common compelling reason for modernizing software. Upstart competitors are typically not weighed down by technical debt, older systems, and processes, making them nimbler by nature. When competition gets ahead in the market by having more effective business tools, the time has come to move forward quickly or perish.

Risks of modernizing

Legacy modernization may seem like an easy and logical decision when dealing with systems that are multiple decades old—modernize and become more productive. But, we all know the adage "If it ain't broke, don't fix it." Gartner Research warns that moving away from legacy systems such as the IBM mainframe could end up costing more and pose a risk to quality.

Before leaping into a modernization effort that touches every single company process, move with caution. Gartner advises organizations considering a modernization project to start with these steps:

- **Focus on business needs and capabilities, not perception.** Phrases like "fragile" or "legacy"—or even Gartner's preferred term "traditional"—can carry connotations and bias decision makers towards modernization when it isn't necessary or even beneficial.

- **Audit existing platforms and processes** with a focus on misalignments or gaps between business requirements and what the platforms deliver. Start with what you've got, not with the assumption that change is necessary.
- **Consider total cost of ownership,** including cost of transitioning and dependencies over a multiple year period. By focusing on business needs and cost of ownership, organizations can correctly prioritize projects that will drive the most business value while minimizing risk.

Considerations for the cost of such a move need to go beyond current CapEx for the traditional system versus expected OpEx of a new system. Businesses need to also look at things like:

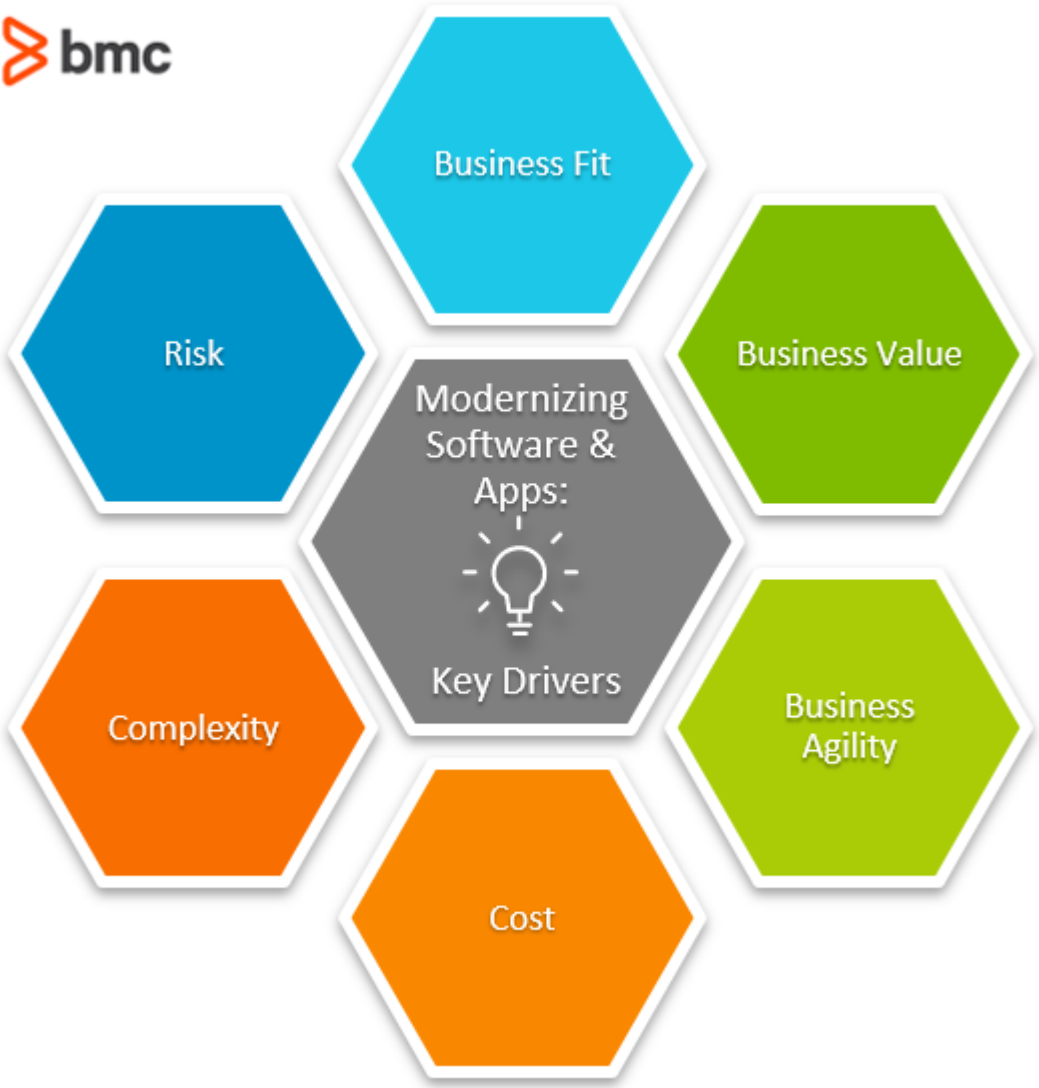
- Change order costs
- The cost of running multiple systems simultaneously during any transition
- Training costs
- Any new security exposure

The emphasis is to look at the issue from a business perspective and ask questions that relate to problems such as:

- Competitive ability
- Current backlog of change requests
- Friction in the business workflow
- Where failure occurs

As Gartner summarizes:

"The objective is to determine whether the traditional platform is helping the business or hindering it from meeting its goals."



Gartner model for application modernization

The first step to transition from legacy to modern applications is to identify, evaluate, and mitigate the risk-to-reward ratio of your IT modernization initiatives. Legacy technologies are typically characterized by two factors:

- They're in use to serve some critical purpose.
- They usually exist because the modernization process faces steep obstacles.

In this context, research firm Gartner presents a guideline for evaluating legacy applications while reducing the risks associated with your modernization project.

Legacy software modernization

Successful digital transformation projects are driven by the business case. For example, legacy technologies may:

- Prevent organizations from competing in a digital-first business market.
- Expose undue security risks that can be easily mitigated by using modern technology infrastructure.

It's important to find opportunities and risk-aversion in the new technologies and operating models

in the modern era. So, when debating whether to modernize, consider these key drivers:

- **Business fit.** Consider alignment with organizational vision, market needs, and the competitive strength delivered with your current IT environment. See how application modernization fits with your goals.
- **Business value.** By investing in new technologies, what added value are you delivering, in the short and the long term?
- **Business agility.** In the age of software, business agility should be a key driving factor for evaluating existing technologies. Embracing business agility can be challenging, but it's increasingly a business imperative.
- **Cost.** Consider the OpEx, CapEx and the total cost of ownership (TCO) of your digital transformation and application modernization initiatives.
- **Complexity.** Migrating to new technologies can become exponentially difficult, adding to the cost of transition.
- **Risk.** Compare the risks and opportunities associated with new technology investments. These risks can include technical challenges, cost, and soft factors such as cultural change issues and end-user acceptance of new technologies.

Evaluating software modernization solutions

Once you've identified any legacy technologies as possible candidates for modernizing and digital transformation, you'll want to follow a stringent process that maximizes innovation and future proofing. Use these steps:

1. **Encapsulate.** Capture the functions and data of the legacy application and deliver the encapsulated service as an external API connection.
2. **Rehost.** Migrate the application from present mainframe data center servers to cloud infrastructure.
3. **Replatform.** Migrate the software code to a new operating platform. This will require changes at the code level (interface and functionality) as well as the software architecture.
4. **Refactor.** Optimize existing code for improved functionality on new platforms and cloud infrastructure.
5. **Rearchitect.** Thoroughly change the software architecture to introduce new features and functionality.
6. **Rebuild.** Restart the project from scratch, replicating the functions and features.
7. **Replace.** Eliminate the legacy technology and invest in an alternate solution.

Explore BMC range of products to help with your modernization journey

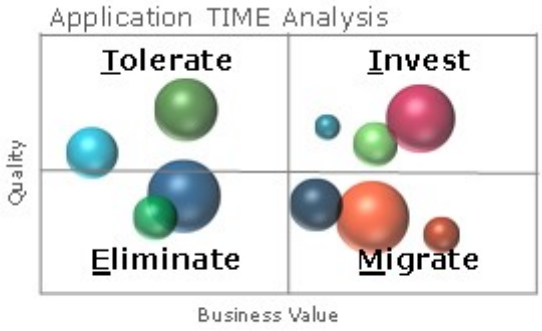
Gartner's TIME framework

Once you've identified and collected information on a variety of candidate legacy technologies, it's time to prioritize.

Gartner offers the Tolerate-Invest-Migrate-Eliminate (TIME) framework that helps organizations develop a clear change strategy and observe options from a larger organizational perspective:

- **Tolerate.** Re-engineer.
- **Invest.** Innovate and evolve.

- **Migrate.** Modernize.
- **Eliminate.** Replace and consolidate.



(Source)

By using this simple 2x2 matrix, organizations can evaluate how well existing and new technologies can perform in terms of business fit, functionality, architecture, and adoption.

- **Business Value** represents the importance of a technology to achieve business goals.
- **Quality** refers to the technical integrity of the technology, with high quality indicating the ease of change of the legacy technology.

Popular approaches to software modernization

Of course, there are other approaches you can take when it comes to modernizing. In fact, several approaches and methodologies have been developed:

- **Architecture Driven Modernization (ADM)** resorts to support infrastructure to mitigate the complexity of modernization. Virtualization is one great example.
- **The SABA Framework** entails planning ahead for both the organizational and the technical impacts. This is a best practice that can minimized some serious headaches for enterprises.
- **Reverse Engineering Model** represents high costs and a very long project that may be undermined by the pace of technology.
- **Visaggio's Decision Model (VDM)** is a decision model that aims to reach the suitable software renewal processes at a component-level based for each case combining both technological and economic perspectives.
- **Economic Model to Software Rewriting and Replacement Times (SRRT)**. Similar to the above mentioned VDM model.
- **DevOps contribution**. DevOps focus is to allow swift deployment of new software releases with an absolute minimum degree of bug or errors in total compliance with target operational IT environment. This, by itself, represents a major enabler factor to speed up Legacy Modernization processes.

Best practices for modernizing apps/software

Whether you follow a prescribed approach or are DIYing it, these best practices ring true for any modernization effort.

- **Identify and map all IT systems and inherent software applications** within your corporate landscape, regarding role, interactions (both amongst apps and towards human users), their

support platforms, resources, and ageing status within the current standards.

- **Consider the business needs, not just IT's needs.** You must look at the business as well as IT. If your company still considers IT "something annoying that we must have," you might have a hard time getting the unequivocal sponsorship of the board of management—which you need for any major modernization effort.
- **Proceed with a dual perspective.** On one hand, compare your existing IT systems and software applications age and technological status versus current standards. On the other hand, align these with the business' roadmap.
- **Cross-check process knowledge.** Often, key users know how processes work and they likely understand the various steps involved in a process. But remember what they might not know: the logic behind the system, or how it does or doesn't affect other business processes.

The key takeaway: modernizing software and applications is a necessary activity for most businesses today. This does not mean that every single business process or tool is replaced with a shiny, cloud-first option. Instead, it means choosing the practices that will benefit the most from this concerted effort.

Related reading

- BMC Business of IT Blog
- The Chief Transformation Officer Role Explained
- Business vs IT vs Digital Transformation: Strategy Across 3 Critical Domains
- Application Mapping: Concepts & Best Practices for the Enterprise
- 4 Common Pitfalls on the Path to IT Modernization
- Application Performance Management in DevOps