

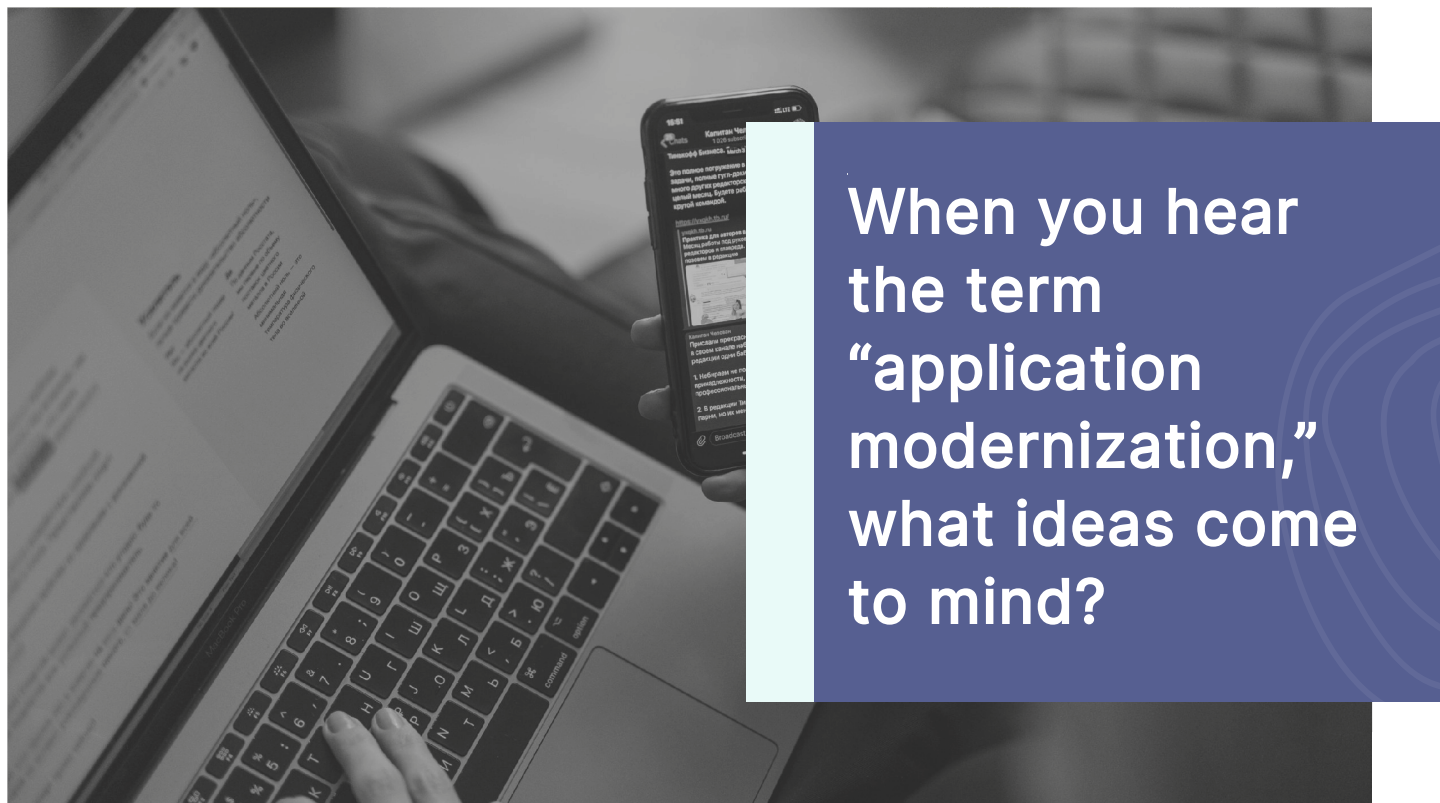


SOLUTION  MACHINE

JUNE 2023

Guide to Application Modernization

Guide to Application Modernization



For some, it might be keeping applications updated with their latest releases, and others might think of supplementing legacy applications with custom software to fill capability gaps. And some dream of getting rid of legacy applications altogether, replacing them with something that performs better, something that can scale more easily, or something that simply costs less to use.

“Application modernization” is all of these, and more. Every organization looking to modernize has its own motivations for doing so and its own notion of what “modernization” looks like. In this guide, we discuss some of the motivations for and advantages of application modernization, some best practices for planning and executing a modernization project, and some pitfalls to avoid.

Why Modernize?

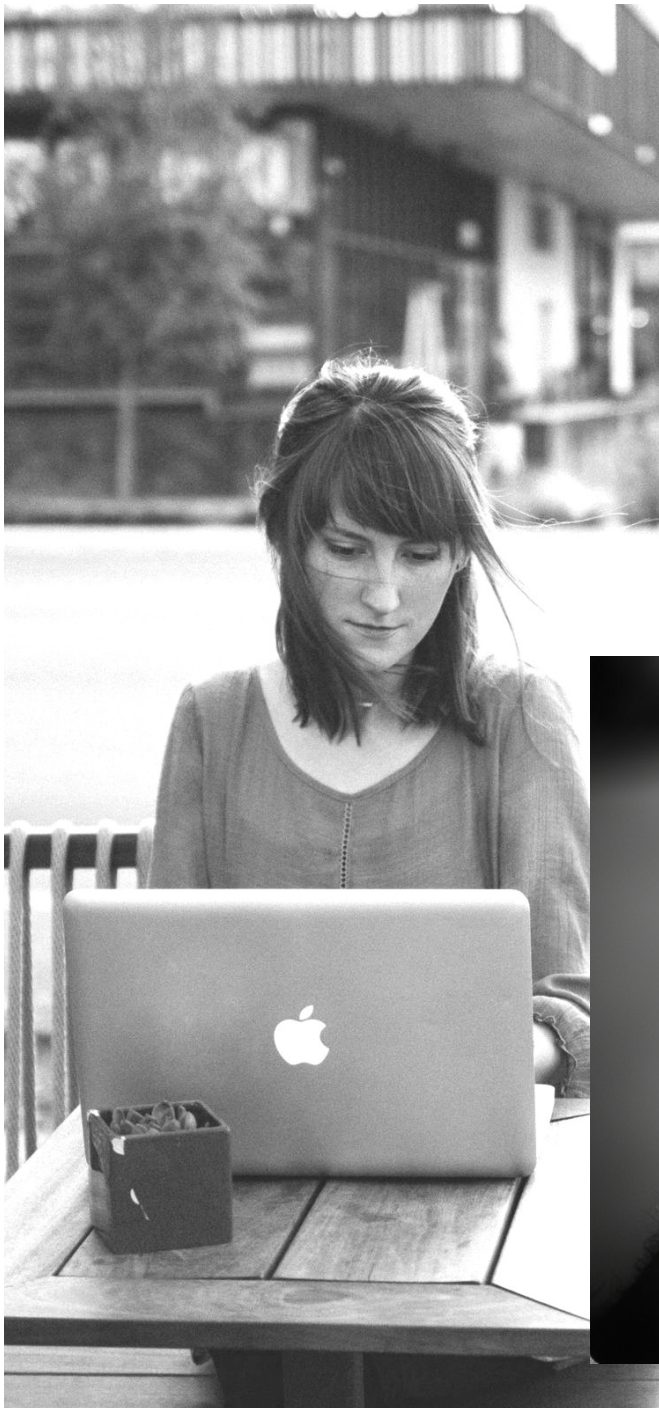
“Don’t fix it if it ain’t broke” is a good guideline to follow for physical devices. If everything is in working order, it’s often best to leave it alone and not tinker with it. Software, however, is a little different.

Unlike hardware, which can wear out and break down over time, older software that has had most or all of its bugs worked out does not often “break” on its own. As long as whatever computer it’s running on can still support it, you can keep using a legacy application indefinitely.



Why Modernize?

Leaders in small- and medium-sized businesses cite numerous reasons behind their efforts to modernize their applications. In general, most of these reasons fall into one of three categories: **Cost control**, **productivity**, and **competitive advantage**. Let’s look at each of these in detail.



Cost Control

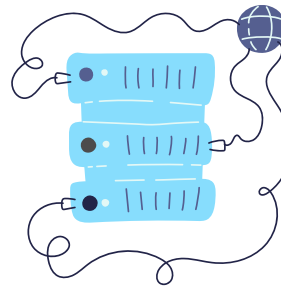
Even for a legacy application for which you own a perpetual license with no annual maintenance charges, the total cost of ownership can be high. For example:



If it can run only on an operating system that is no longer supported by its developer, you might pay extra to keep the operating system running.



If the hardware architecture is outdated, you might pay extra to call on a shrinking pool of experts who can help you when it has issues.



If the hardware is running in your own on-premise data center, you are paying for the electricity to keep it running and cooled. Cloud services are often more cost effective.



If a disaster befalls your facility, the cost to find and deploy replacement hardware and restore the application and data might be more than you can afford.

Productivity

When you continue using legacy applications year after year—in particular, applications that are not easy to update with new capabilities—you are leaving potential productivity gains on the table.

Your business is (hopefully) not standing still: It's growing and evolving; old business processes are changing and new ones are being added. You are introducing new products and services and expanding into new markets. Software that can't adapt to the changing business environment is not helping your team's productivity.

As well, it's important to consider business continuity and disaster recovery. A disruption to your IT systems, whether from physical damage, cyberattacks, or even an extended power outage, can do irreparable harm to your productivity. Modernizing your applications and IT infrastructure can help your business continue and recover from a disaster.



Employee Experience

A common metaphor for being stuck with outdated technology is that of the buggy-whip manufacturer. As automobiles became more popular in the early 20th century, makers of horse-drawn carriages and accessories (such as buggy whips) found themselves with declining demand. Employees who were skilled in making buggy whips found it difficult to transition to other lines of work.

In a similar way, from an IT perspective, your employees don't want to be stuck using outdated applications. In general, they want to learn new things and acquire new skills to keep themselves marketable and grow in their careers. Modernization can help the employees feel like they are progressing and give them opportunities to learn new, up-to-date skills. Happier employees are more likely to stick around rather than seek greener (or at least more modern) pastures, and having the newest technology is a good way to attract new talent as well.

Competitive Advantage

Customers don't (or shouldn't) make buying decisions on the basis of price alone. By and large, customers want to do business with partners who understand their pain points and can provide compelling solutions at a reasonable price.

Moreover, they appreciate a partner with whom it is easy to do business. Bottom line: You're much more likely to acquire and keep a customer if you can provide the best customer experience.

Modern applications provide a wealth of advantages in this area. Using software to make it easy for customers to evaluate their options among your products and services, place orders, learn the status of those orders, and communicate with your customer-facing teams gives you a major advantage over your competitors who don't measure up.



**Modern
Applications
Provide a
Wealth of
Advantages**

Motivations: A Summary

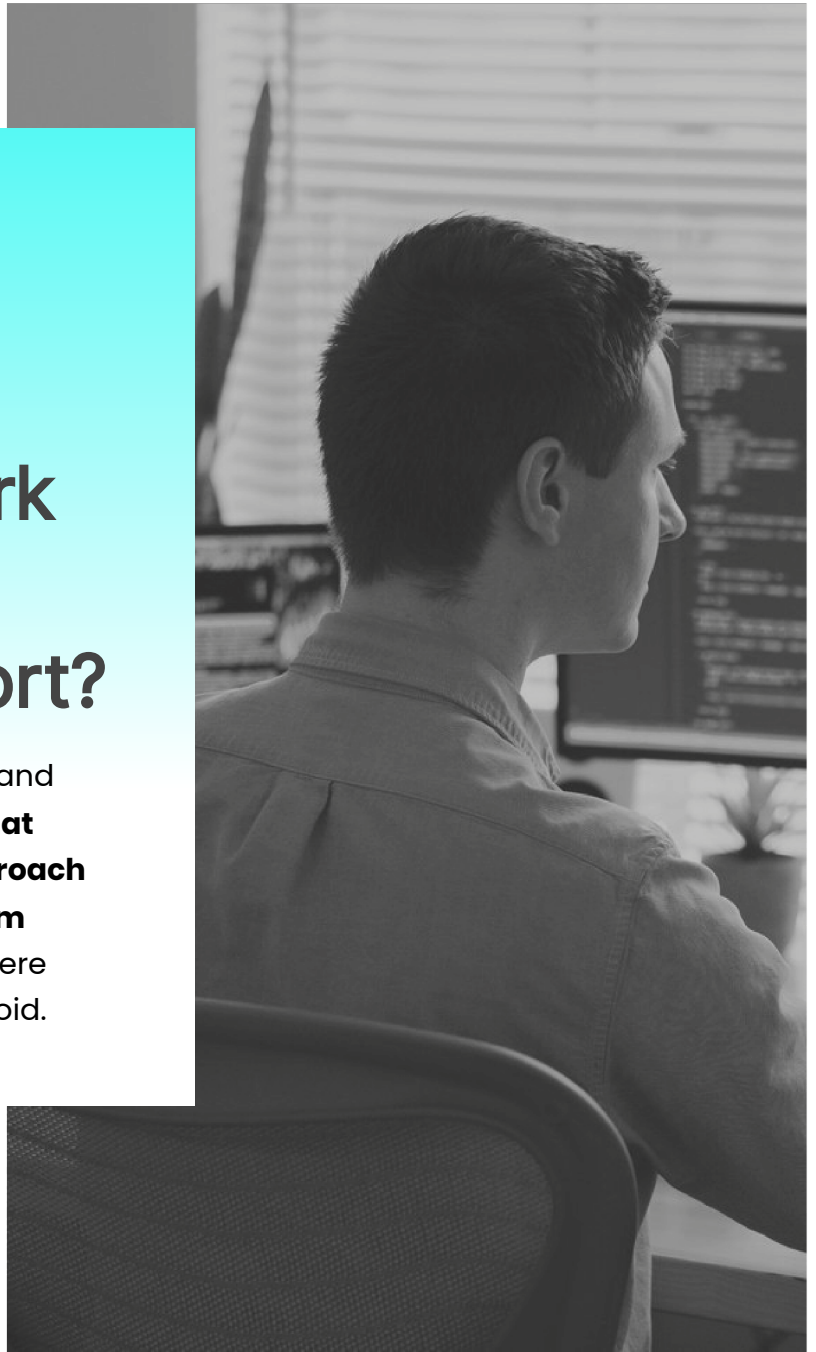
Overall, businesses that modernize their applications and IT infrastructure can experience the following advantages, among others:

- ▶ **Faster go-to-market**
- ▶ **Enhanced security**
- ▶ **Increased productivity**
- ▶ **Cost reduction**
- ▶ **Revenue growth**
- ▶ **Happier employees**
- ▶ **Optimized customer experiences**
- ▶ **Greater efficiency and scalability**
- ▶ **Improved competitiveness and agility**

Modernization: Best Practices and Pitfalls to Avoid

How do you embark on an application modernization effort?

It's not a matter of just buying something and implementing it. **It's a complex process that takes careful consideration, and the approach you use will vary according to the problem you're trying to solve.** With that in mind, here are some best practices and pitfalls to avoid.



Best Practices

Understand the current pain points and risks:

Document everything that's "wrong" with the current state, making sure to include both the business and IT perspectives. What capabilities are you missing? Are there IT infrastructure risks associated with keeping the current system running? Is the current state too complex, either from a business or IT standpoint? Are there regulatory requirements that the current system can't meet?

This step will help you define your business transformation opportunities.

Understand and optimize your business processes:

Without reference to any software, you should take a long, hard look at your business processes. Are they as efficient as they can be? Do they have unneeded complexities? Are they all even necessary?

Take the time to optimize your business processes to eliminate waste, using industry best practices as much as possible, because these are more likely to be supported "out of the box" by new software systems.

Get rid of processes that add no value—among the worst things you can do is spend money to automate a process that you shouldn't be doing in the first place.



Think about the future:

Any new application will need to support not only the current state of your business but also what it will look like in the future. Will you introduce new products or services? Will you expand into new markets? Do you expect to engage in mergers or acquisitions? The solution you choose should be able to scale in any of these directions.

This step will help you define a "future-proof" modernization roadmap that details all the steps in your overall modernization journey.

Keep the end users in the loop:

The end users are the ones who will suffer the most from bad choices, so keep them engaged in the project from start to finish. They should be involved in requirements gathering, prototype evaluation, and user acceptance testing.

Choose the right modernization approach:

Several alternatives are available when considering your modernization approach:



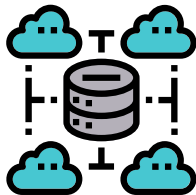
Lift and shift: In this approach, the software is modified or replaced, but hosted in a different environment, such as a local virtual machine or cloud-based service. This approach is useful when the software is good enough (for now) but there is a need to modernize the infrastructure.



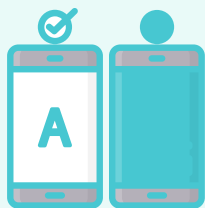
Refactor: Some or all of the software is rewritten to take advantage of modern software frameworks or operating systems. This is a good approach when the software doesn't need many new features but has become difficult to maintain or is performing poorly.



Implement a “strangler pattern”: In this approach, a “façade” is built as an intermediary interface to the legacy system. Functional calls to the system go through the façade. Behind the façade, legacy system modules are replaced one by one over time with new modules, and the façade points to the new modules as they become operational. Eventually, the old modules are “strangled” away, but the façade limits the impact to the outside world.



Replatform: Certain software components (such as a back-end database) are migrated to a new version or environment.



Leverage API integration: This approach takes advantage of application programming interfaces (APIs) to enable software to interface with other applications (on-premise or cloud) and devices, such as laboratory instruments or manufacturing machines.

Your modernization partner should be able to recommend the best approach for your situation.

Pitfalls to Avoid

It's a business project, not an IT project:

IT resources, whether in-house staff or consultants, will help with the technical aspects of your modernization project, but the line-of-business teams and upper management are equally responsible for project success. As the ultimate benefactors of modernization, the business at all levels should be involved throughout the project. Otherwise it is guaranteed to fail.

It's not just about cost:

Choosing an option just because it's the "cheapest" (however you measure that) is a bad idea. You must also consider the benefits and return on investment, in both the short and long term.

Don't "boil the ocean":

No organization, in particular small- and medium-sized businesses, has the money or resources to modernize everything all at once. You still have a business to run and you can't afford to have multiple systems in a state of flux at the same time. Design your modernization roadmap to limit impact to the business at any one time, and arrange the projects in a logical order, considering system dependencies. Draw hard boundaries around each project and avoid "scope creep."

Conclusion

Application modernization is not always a straightforward, simple project. It has to be considered in the context of other applications, the future of the business, and risk factors. Your application modernization partner should have the expertise and experience to help you every step of the way, from concept to delivery.

At Solution Machine, we help clients overcome the most demanding software requirements by reducing complexity, ambiguity, and waste through the seamless delivery of software with the highest business impact. We have extensive experience with application modernization, from solution assessment and architecture design through custom development, system integration, and application migration.

And, it's easy to do business with our friendly and knowledgeable sales staff and consultants.


If you are considering an application modernization strategy but don't know where to start, contact [Solution Machine](#) today. We're here to help you on this journey.

Talking to us about
your project is easy.

 737.757.0967

 www.solutionmachine.com

 team@solutionmachine.com

 1320 Arrow Point Drive,
Suite 501 #TW100
Cedar Park, TX 786133

