



altexsoft
software r&d engineering

WHITEPAPER

Legacy System Modernization: **How to Transform the Enterprise for Digital Future**

1. The State of Software

1.1 What is Legacy System?

1.2 The Hidden Costs of Legacy Software

1.2.1 Maintenance and Support

1.2.2 Integration and Compliance

1.2.3 Security

1.2.4 Lost Business Opportunities

1.2.5 Organizational Agility and Efficiency

2. Preparing for the Digital Future

2.1 The Business Case for Software Modernization

2.2 Considering the Risks

3. Legacy Software Modernization Best Practices

3.1 Legacy System Assessment Framework

3.2 Legacy Modernization Approaches

3.2.1 Migration & Enhancements

3.2.2 Correction & Growth

3.2.3 Complete Software Reengineering

4. Checklist for Successful Application Modernization

Conclusion

References

1. The State of Software

Dining at a fancy restaurant, you want to spend some quality time, enjoying tasty food and drinks. When choosing the latter, the chances are, you will prefer a glass of good wine, the older the better. For that matter, we all know old wine and old friends are the best.

The problem is, unlike wine and friends, software doesn't get better with age.

Just think about it. There is a good reason your new computer runs Windows 10 instead of Windows XP.

Obviously, your current computer is more powerful and more capable than the one you owned 10 years ago. Similarly, the business you run is not the same as it was when you started it. Therefore, using outdated and underperforming software to manage it is analogous to running Windows XP on your new ZenBook.

1.1 What is Legacy System?

Do you remember the last time you used a pager? Probably, it was in the late '90s. But the technology is not as dead as you might have thought. In fact, your own life might depend on it as 90 percent of hospitals surveyed by HIMSS Analytics "rely on pagers as a cost of doing business."^[1]

Aside from being outdated, pager technology is a huge source of expense, costing a hospital on average around \$180,000 per year.^[1] So why is it still so widely used?

The answer is simple: Some systems are just

hard to replace - especially the ones that handle vital business processes within an organization.

Pagers in health care are not the only example of such a phenomenon. Commonly referred to as a "legacy" system/technology, it is relatively widespread in a number of other industries, including banking, finance, insurance, and transportation.

As [defined by Gartner](#), **legacy application** is *"an information system that may be based on outdated technologies, but is critical to day-to-day operations."*

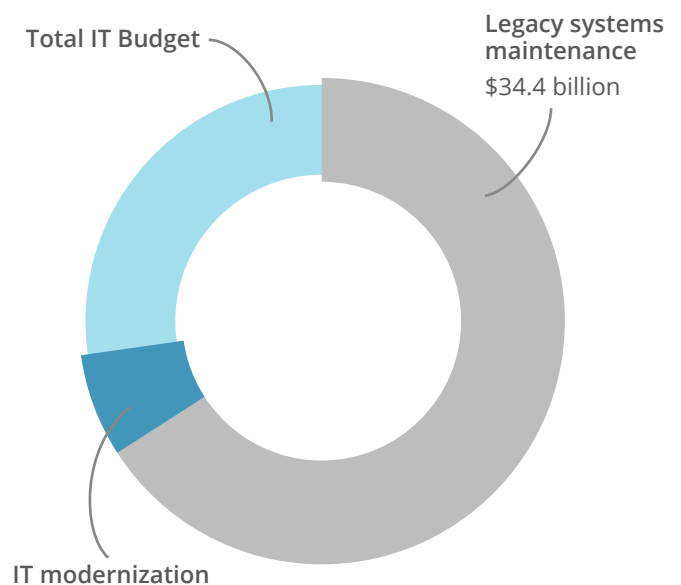
A number of examples of such legacy systems can be found across some major federal organizations. For example, a system, used by the Department of Defense to operate the US nuclear forces, including intercontinental ballistic missiles, nuclear-capable bombers, and tanker support aircraft, runs on an IBM Series/1 Computer (a 1970s computing system) and uses 8-inch floppy disks.^[2]

However, a legacy system is not always defined by its age. It might be due to the lack of support or its inability to meet the needs of a business or organization that a system is considered to be legacy. Such software is usually difficult (or impossible) to maintain, support, improve, or integrate with the new systems due to its architecture, underlying technology, or design.

1.2 The Hidden Costs of Legacy Software

Many companies continue using outdated systems, regardless of its age or the quality of underlying technologies. The software has been working just fine for decades and is still able to cope with most of its tasks, they argue. Indeed, why fix it if it ain't broke?

Actually, there are quite a lot of reasons to "fix" your legacy systems. The real cost of running such software is the major one among them. According to some estimates, about 71 percent of the IT budget slated for US federal civilian agencies will be spent on maintaining legacy systems next year. This equals more than \$34 billion. To put that into context, only \$3.1 billion are assigned for the IT modernization in 2017.^[3]



IT Cost Breakdown – the US Federal Civilian Agencies Budget (2017)

Yet, these numbers represent just the tip of the iceberg. The hidden costs not stated in any

budget are even bigger. Namely, there are several **sources of legacy software expenditure**.

1.2.1 Maintenance and Support

The costs of legacy system maintenance operations include the following:

- **Updates and changes**

Legacy systems are typically quite large in terms of code base as well as functionality. Taking into account their monolithic nature, you cannot just change or replace one system module. A small update might result in multiple conflicts across the system. Thus, any change or update to the legacy system requires time and effort, neither of which come cheap. Additionally, legacy systems usually have vast amounts of documentation as well as a number of undocumented features. So, there is always a certain risk involved when interfering with the source code.

- **Infrastructure**

Just like software itself, the underlying infrastructure becomes harder and more expensive to maintain as it ages. Legacy systems

often require a specific technical environment, including hardware. Thus, the infrastructure maintenance spending remains high, as compared to modern cloud-based solutions. Legacy data represents another significant infrastructure issue. Being scattered across several databases and storage resources, it is hard to reorganize for increased storage space optimization. Gathering and systematizing legacy data manually to further transfer it to the new database is a time- and cost-intensive task.

- **Staff training**

Depending on obsolete technologies, the legacy system support and maintenance requires a specific set of skills and expertise. While the developers who have built the software might retire or switch to other technologies, it becomes increasingly harder to find and retain the right talent. Dedicated staff training might be an even bigger source of expense.

1.2.2 Integration and Compliance

Modern software platforms often rely on third-party APIs to access a few capabilities, such as geolocation, user authentication, data sharing, and transactions. For example, Uber relies on the data provided through the Google Maps API for its core functionality - an ability to find and track cars nearby.

Indeed, why reinvent the wheel, when you can use the existing, tried and true solution at a fraction of the cost?

Modern technologies are integration-ready by default. API vendors typically provide support for most of the programming languages and frameworks out of the box. Yet, obsolete or rare technologies typically lack the compatibility.

Connecting a legacy software to a third-party tool or service often requires a significant amount of custom code. And there is still a chance that the final integration won't work as well as intended or that it will work at all.

Another aspect of legacy systems that comes at a high cost is compliance. This is especially true for heavily-regulated industries, such as finance or health care. Similar to the introduction of the HIPAA omnibus rule back in 2013, when many providers were unprepared to change their systems in order to comply with the new regulations, any new standards set by the government could lead to millions in penalties, if you fail to adapt to them.

1.2.3 Security

According to the Better Data, Better Decisions (BDNA) State of the Enterprise Report, out-of-date software and hardware assets represent *"one of the core vulnerabilities that typically plagues almost all corporate and government networks"*.^[5] Indeed, legacy systems are usually

less resistant to cyber attacks, harmful programs and malware, which is only logical. If the software solution had been around for years, the attackers most likely had enough time to get familiar with the code and find its vulnerabilities.

Another reason for this is that outdated software might no longer be supported by the vendor. This means that no patches are provided and no one keeps the system compliant with the latest security requirements.

Even if your system is custom-built and you have the resources to maintain it, adding more patches means additional investment in security. In the worst case, this might resemble

a leaky bucket, where you get a new breach as soon as the previous one is fixed.

“Every piece of hardware and version of software that is not maintained regularly could be a gateway for unauthorized access.”^[5]

— BDNA State of the Enterprise Report (Q2 2016)

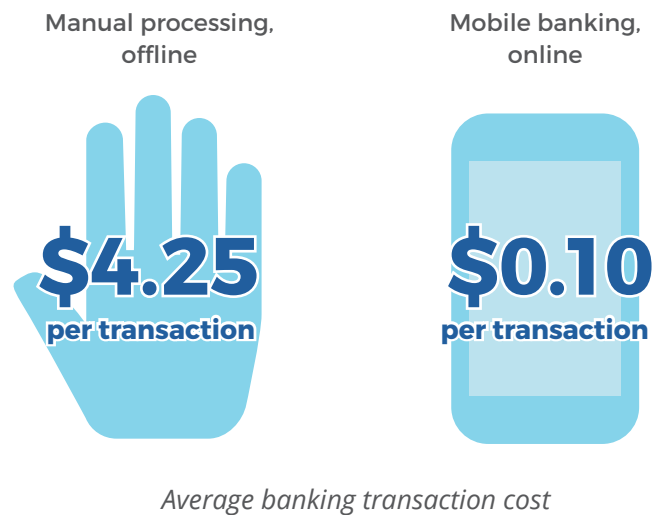
1.2.4 Lost Business Opportunities

By investing in legacy software support and maintenance, you leave less room for innovations. Instead of adopting new technologies and business models, you are stuck with your old software, letting new opportunities in your industry go unnoticed. This leaves your competitors more openings to outperform you and take over your market share.

A study found that mobile banking solutions might result in 72 percent higher revenue for banks.^[4] Thus, a lightweight mobile application

provided by a competing bank can lure away some of your customers. In this case, the cost of a missed opportunity equals the revenue your competitor will get from newly acquired customers.

Additionally, Javelin Strategy & Research found that an average cost per mobile banking transaction equals just 10 cents to process, while processing an offline transaction at a physical bank costs \$4.25.^[6]



Such overhead is another significant element that adds to the cost of missed opportunity. While mobile banking providers save more on efficient

and low-cost ways to process transactions, you might still be using outdated POS terminals and manual assistance to do the same job.

1.2.5 Organizational Agility and Efficiency

Talking about business opportunities, timing proves to be crucial. How fast can you respond to the market challenges? Will it take weeks to adopt new technologies and solutions? Or rather several months? The truth is, in most cases, businesses bound to legacy systems lack organizational agility to adapt to the upcoming challenges.

Indeed, 73 percent of the business leaders surveyed by IDG and Dell list business agility as a critical or very important objective of IT modernization.^[7]

Partially due to their inability to be updated and modified, legacy systems can hold back innovation, resulting in significant losses.

Moreover, outdated, cumbersome software is less efficient, which has a negative impact on the employee productivity.

Machine-reengineering, the [business processes automation with the help of machine learning](#), is a great example of how innovation can optimize business performance. Based on Harvard Business Review research, more than a third of the early adopters *“saw gains in bottom-line performance using machine-reengineering to slash 15% to 70% of costs from certain processes. At the*

same time, some saw a tenfold improvement in workforce effectiveness or value creation.”^[8]

For example, the [resource](#) lists a number of successful cases of processes reengineering. Namely, a financial services provider implemented a biometrics solution based on voice recognition. The company was able to eliminate a four-step authentication process by using customers’ voices as passwords. This resulted in 50 percent more efficient call routing, the improvement achieved largely due to the use of advanced technology.

\$ Maintenance & Support

Maintenance cost can exceed the original development budget in just 5 years after the product release.

\$ Integration & Compliance

Failure to comply with industry regulations could potentially lead to millions in penalties.

\$ Security

The estimated average cost of a data breach is \$4 million, according to IBM.

\$ Lost Business Opportunities

The lack of innovation can impact your business’ long-term profitability and competitiveness.

\$ Organizational Agility and Efficiency

The average efficiency of some processes can increase up to 50% through automation and modernization.

2. Preparing for the Digital Future

Despite the problems and risks related to the outdated software, a majority of the companies surveyed by the University of Surrey are still lacking legacy-modernization initiatives. The research shows that most of them would only consider reengineering the current solution in case of an emergency, such as a complete system outage.^[9]

“Although many core systems have become efficient and stable over the many decades of their existence [...], they are now seen as a key obstacle in the rapid evolution to a fully integrated digital business.”^[9]

— University of Surrey report

To bridge the gap between the current offerings and customer expectations, companies need to rethink their business models, making them digital-ready. Yet, the legacy software is only one aspect of the problem. Sometimes, a far bigger issue is the mindset that comes with it. That is why proving a business case for software modernization is the first challenge to be faced by the initiating party.

2.1 The Business Case for Software Modernization

The following benefits prove that legacy-system modernization is a vital part of the overall business digitization.

**Competitive Advantage**

Operating in an industry dominated by decades-old technology behemoths, you have a chance to outperform the competitors simply by offering a modern and lightweight solution.

**Happier Clients/ Employees**

User experience and design standards have evolved significantly over the last several years. By introducing modern, sleek UI and user-centric intuitive experiences, you can improve your customer satisfaction and employee performance, thus, increasing your revenue.

**Future-Ready Business**

By replacing your legacy system with a modern solution, you make sure your business is ready to evolve and expand, keeping up with technology advances. Plus, keeping up with the latest technology trends creates an internal culture of business agility and innovation in your company.

**Unlocking Big Data Opportunities**

Outdated storage solutions prevent you from accessing and making use of your data. Database migration and optimization is required to successfully tap the big data opportunities.

**Better Performance and Reliability**

Legacy IT faults are one of the most common reasons for [delays in the air transportation industry](#), as well as in many others. Thus, systems modernization might help you reduce the outage risks and cut the related losses.

Yet, despite all the benefits, the resistance to modernization is often well-grounded.

2.2 Considering the Risks

Two major arguments are typically used when talking about a software modernization initiative. Those are the time and cost involved. Indeed, a solution that took a team of developers years to implement cannot be re-created in a week, even if you hire twice as many developers to handle the task. Thus, in some cases software reengineering cost might exceed the initial investments.

Yet, there are more pitfalls to avoid. Some of them have been described by a group of Carnegie Mellon University researchers back in 1999. The report “Why Reengineering Projects Fail” lists the following reasons for legacy reengineering effort failure:

- 1. The organization inadvertently adopts a flawed or incomplete reengineering strategy.*
- 2. The organization makes inappropriate use of outside consultants and outside contractors.*

- 3. The workforce is tied to old technologies with inadequate training programs.*
- 4. The organization does not have its legacy system under control.*
- 5. There is too little elicitation and validation of requirements.*
- 6. Software architecture is not a primary reengineering consideration.*
- 7. There is no notion of a separate and distinct “reengineering process.”*
- 8. There is inadequate planning or inadequate resolve to follow the plans.*
- 9. Management lacks long-term commitment.*
- 10. Management predetermines technical decisions.*^[10]

Therefore, successful software reengineering requires a solid modernization strategy and great attention to detail. In this regard, we can share some of the best practices and approaches we have developed at AltexSoft.

3. Legacy Software Modernization Best Practices

There are two approaches to dealing with the legacy problem: revolutionary (big-bang) and evolutionary. Yet, both have benefits as well as drawbacks.

The revolutionary method requires shutting down the old system and building a new one from scratch. The approach might be considered extreme, but sometimes it is better to retire the system completely to avoid some serious damage, such as security breaches, lost data, system downtime. Or it can be applied in a case when the original product cannot solve

the existing business problems anymore, so it makes no sense to reengineer or port it to the new technologies.

The evolutionary approach presupposes a systematic, step-by-step software modernization process. It is usually less painful: It does not disrupt the major business processes and implies significantly lower risks for the company. Yet, it often turns into a band-aid approach, where you focus on solving the problems instead of removing the factors that cause them.

3.1 Legacy System Assessment Framework

Often dealing with legacy systems, we at AltexSoft have developed our own approach to choosing an appropriate way to modernize business-critical software. Namely, we take several steps first to assess the existing solution.

Technologies Analysis

The first step in our plan is to **identify and analyze the technology stack** of the existing product. Thus, we know if the programming language or frameworks used are still relevant and supported by the vendors. If the product relies completely on outdated technologies, the chances are we would need to completely rewrite it in the process of modernization.

Architecture Audit

In case the tech stack (or some parts of it) is still relevant, it is necessary to **conduct an architecture audit**. This will help you define the system elements which are functioning well and focus on the ones that need modernization. Plus, you will be able to see how different parts of the system interrelate, so that your future changes won't affect the whole product.

Code Review

Legacy software usually has an excessive codebase, requiring regular reviews and refactoring. If not treated properly, the software tends to “rot.” This might lead to more design flaws and conflicts as you try to introduce new features or update some parts of the system. That is why, as a part of any modernization or changes, we typically **conduct a complete code review**, assessing the quality and “updateability” of the system’s source code.

UI/UX Review

The same principle applies to the UI and UX design. A **thorough design review** is required to understand which parts of the system interface need a “facelift.”

Performance Testing

Performance testing aims at uncovering further potential issues with the legacy systems. Poor performance or major flaws can serve as a reason for a complete system reengineering as well as selective improvements.

Current Requirements and Opportunities for Future Growth

While **considering the current business needs and requirements** articulated by the client, we also focus on opportunities for the **future growth**. Thus, we help you make an informed decision by providing a well-grounded and unbiased opinion on the software modernization options.

Considering the results of the above-listed evaluation stages, we can offer **three possible ways to modernize legacy software**.

3.2 Legacy Modernization Approaches

3.2.1 Migration & Enhancements

This is one of the most popular approaches to application modernization and the easiest way to make sure your product will keep serving your needs for years to come. It presupposes the system migration (typically re-hosting, using cloud solutions) and some minor enhancements. This includes UI/UX updates, performance optimization, and database migration.

Yet, this method has a number of limitations. Namely, the core business logic and architecture mostly remain unchanged, as this type of changes require a more invasive approach.

Case Study: [AltexSoft Improves an Innovative Business Evaluation Tool by Enabling Automated and Reliable Data Collection and Analysis in the Cloud](#)

3.2.2 Correction & Growth

If the product technology stack is relatively modern and does not represent a threat for future product growth, modernization can involve some minor enhancements/corrections. This might be architecture optimization or code refactoring, UX updates or performance optimization without significant changes in product business logic.

As soon as the product is up to date, you can add more features on top of it. These might be third-party integrations or custom-built modules.

Case Study: [AltexSoft helps Merlot Aero advance airline management by enhancing the legacy system and building up new features for its transportation SaaS product](#)

3.2.3 Complete Software Reengineering

Considered the most extreme method, features extraction relies on your business strategy and growth outlook. This means, in order to reengineer the product, you need to identify the features that are still crucial to your business and the ones that are no longer used or required. After that, the required features are prioritized and modified if needed.

Taking the legacy system as a basis, the team creates an up-to-date product with matching

capabilities, but better performance, look and feel, modern technologies and scalable architecture. Depending on the functionality analysis and prioritization, the new product might 100% match the previous version in terms of functionality, or lack some features that are no longer required/used.

Case Study: [AltexSoft & Fareboom: Co-Building Innovative Travel and Booking Solution to Outperform the Competition](#)

4. Checklist for Successful Application Modernization

Aside from adopting the best-suited strategy for your technology modernization initiative, there is more you can do to make sure the process runs smoothly. Here is our **checklist of 7 things to consider for a successful software modernization project**:

1. Assess the current state of legacy systems.

Legacy software does not always fall under “old” or “outdated” definitions. There are more aspects to assess when identifying the legacy. That is why you need to assess all systems in place to uncover the current and potential issues it can bring up in the near future. The assessment should be systematic and detailed: Study all aspects of your technology, from code and architecture to visual look and feel, taking into account your future business plans for product growth.

2. Select the modernization method that would be the fastest to deliver value.

Based on the assessment conducted at the first phase, choose the modernization approach that best fits your needs and will help you deliver results fast. Aside from the three approaches

we have listed above, consider existing products you can use instead. There is no need to reinvent the wheel if there is an SaaS solution available at a fraction of cost. Yet, if your system solves rather specific tasks or you want to be able to build more features on top of it, [custom product development services](#) might be right for you. In this case, adopting [agile software development practices](#) can help you speed up the process and deliver value fast.

3. Rethink the architecture and prioritize for simplicity.

Legacy systems often fail to perform as needed due to their overly complex structure. When modernizing your system, less is more in terms of both architecture and functionality. Start by implementing only the most important features. Consider a microservices architecture approach

to make your product scalable. Additionally, make sure the newly released application will work well with the rest of the tools used in your business by default. If you plan to change any of the tools soon, consider several possible options and keep their requirements in mind when building your application.

4. Choose the technology stack to deliver optimal performance and user experience.

When reengineering your system, make sure you use a solid and future-ready technology stack. The choice of technologies should completely depend on the product specifics. Consult with your internal IT staff or address a professional tech consultancy. The right tech stack contributes to building a performant, reliable and efficient product. Adopt a solid [quality assurance and testing process](#) to deliver the best results.

5. Document for future system growth.

To avoid the same mistakes that made you reengineer your current solution, introduce (or adopt best practices used by other companies) a set of coding standards and internal processes.

Orderly documented and clean code make your software easy to understand, extend and maintain in the future.

6. Create a separate support and retirement schedule for your legacy system.

Even if you have a brand-new system running like a clockwork, you will still need your legacy software, just in case. So, don't kill it all at once. Document and archive your solutions so you can easily access and refer to them when needed. Therefore, you need to support your legacy system for some time and plan for retiring your legacy system only when your new product is up and running.

7. Budget for training and system updates.

Working with the old systems for years, your employees might need some time and guidance to master the new software. So be ready to invest in staff training for better performance and efficiency. Additionally, plan for regular system updates. If you fail to keep your product up to date, you will soon face another modernization challenge.

Conclusion

Regardless of the chosen approach, software modernization is a complex, labor intensive and risky process. Yet, the results seem well-worth the risk.

According to a recent Gartner survey, 45 percent of respondents state that one of the current top 5 IT project priorities is “application modernization of installed on-premises core enterprise applications” and a further 41 percent say that “extending capabilities of core enterprise applications” is a top five priority.^[11]

“The majority of [Worldwide Enterprise Application] spending is going towards modernizing, functionally expanding or substituting long-standing business and office applications with cloud-based software-as-a-service.”^[11]

— Bianca Granetto, research director at Gartner

If your company still relies on outdated software, we can help you make your business future-ready with our end-to-end business digitization services. We handle every aspect of legacy-system modernization: from analyzing the current solution, developing a solid business strategy, prioritizing the features to rebuilding your product from scratch, using latest technologies and architecture solutions.

References

1. <https://www.tigertext.com/press-releases/groundbreaking-research-reveals-hidden-costs-of-pager-usage-in-u-s-hospitals/>
2. <http://www.gao.gov/assets/680/677436.pdf>
3. <http://www.nextgov.com/cio-briefing/2016/02/8-things-look-2017-it-budget/125849/>
4. <https://www.fiserv.com/resources/Mobile-Adoption-White-Paper-January-2016.pdf>
5. <http://www.bdna.com/wp-content/uploads/2016/06/BDNA-State-of-the-Enterprise-Cybersecurity.pdf>
6. <https://www.javelinstrategy.com/press-release/javelin-identifies-15-b-mobile-banking-cost-savings-leveraging-omnichannel-approach>
7. http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/IT_Modernization_MarketPulse_Whitepaper_Dell_v3_Final.pdf
8. <https://hbr.org/2016/02/companies-are-reimagining-business-processes-with-algorithms>
9. <http://www.cbronline.com/news/verticals/cio-agenda/exclusive-software-becomes-the-new-rust-as-cios-face-boardroom-battle-to-escape-legacy-roadblock-4972728>
10. <https://www.cs.cmu.edu/~aldrich/courses/654-sp05/readings/Bergey99.pdf>
11. <http://www.gartner.com/newsroom/id/3119717>

About AltexSoft

AltexSoft is a Technology & Solution Consulting company co-building technology products to help companies accelerate growth. The AltexSoft team achieves this by leveraging their technical, process and domain expertise and access to the best price-for-value Eastern European engineers. Over 100 US-based and 200 worldwide businesses have chosen the company as their Technology Consulting Partner.

US Sales HQ

701 Palomar Airport Road,
Suit 300, Carlsbad, CA 92011
+1 (877) 777-9097

Global HQ

32 Pushkinskaya Str.,
Kharkiv, Ukraine 61057
+38 (057) 714-1537