altexsoft
software r&d engineering

WHITEPAPER

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

Dining at a fancy restaurant, you want to spend some quality time, enjoying tasty food and drinks. When choosing the latter, 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 Therefore, using outdated and underperforming software to manage it is analogous to running Windows XP on your new ZenBook.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. What is a Legacy System?

Do you remember the last time you used a pager? Probably, 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 pagers remain a mainstay communication device in healthcare. It's fair to say that some countries like Japan and the UK are finally shutting the services down. Nevertheless, three-quarters of US organizations continue to support at least one type of pager. [1]

Aside from being outdated, pager technology is a huge source of expense. 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 healthcare 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As defined by Gartner, **legacy application** is *"an information system that may be based on outdated technologies, but is critical to day-to-day operations."*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. They use legacy applications running on an obsolete mainframe for core business operations such as high-volume data processing. Fine-tuned over the years, these systems are adapted to deliver specific functionality. [2] Why bother about modernizing them then?

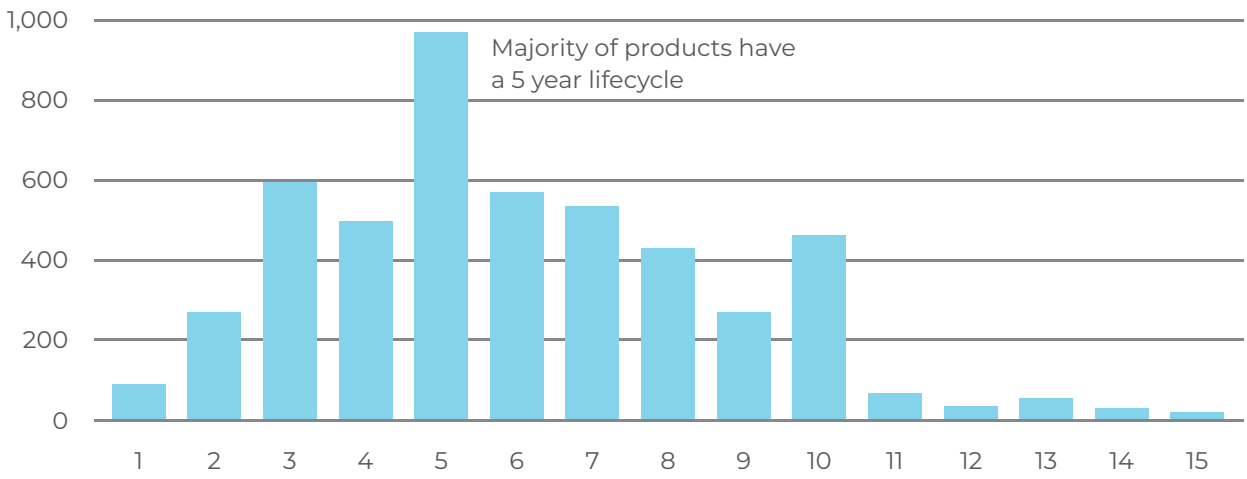
Recently, the US Government Accountability Office (GAO) identified the ten most critical federal legacy systems in need of modernization, some of which date back to the 1970s. However, the full list included 65 systems submitted. Many of them depend on antiquated programming languages like COBOL, have hardware or software support issues, and operate with security vulnerabilities.[3]

Agency	System name	Age of system, in years	Age of oldest hardware, in years	System criticality (according to agency)	Security risk (accorfing to agency)
Department of Defence	System 1	14	3	Moderately high	Moderate
Department of Education	System 2	46	3	High	High
Department of Health aand Human Services	System 3	50	Unknown	High	High
Department of Homeland Security	System 4	8 – 11	11	High	High
Department of the Interior	System 5	18	18	High	Moderately high
Department of the Treasury	System 6	51	4	High	Moderately low
Department of Transportation	System 7	35	7	High	Moderately high
Office of Personnel Management	System 8	34	14	High	Moderately low
Small Business Administration	System 9	17	10	High	Moderately high
Social Secutity Administration	System 10	45	5	High	Moderate

The 10 Most Critical Federal Legacy Systems in Need of Modernization

Legacy systems require modernization. Otherwise, they can be exposed to crashes anytime. That's what happened on Tax Day 2018. Facing technical problems, the Internal Revenue Service couldn't process electronically-filed tax returns. Although the IRS did not specify what went wrong, the fact that many of their IT systems were outdated at that time – two of them being nearly six decades old – might have contributed to the computer glitch. [4]

Flexera in their Product EOL/EOS 2018 Report found that the majority of products have a five-year lifecycle.



Lifecycle span: release date though EOL/EOS date, Source: Flexera

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.

That said, among the CIOs surveyed by Logicalis, more than half have to dedicate from 40 to 60 percent of their time to managing legacy IT instead of shifting towards strategic activities. So, we can conclude that legacy technology a significant barrier to digital transformation.

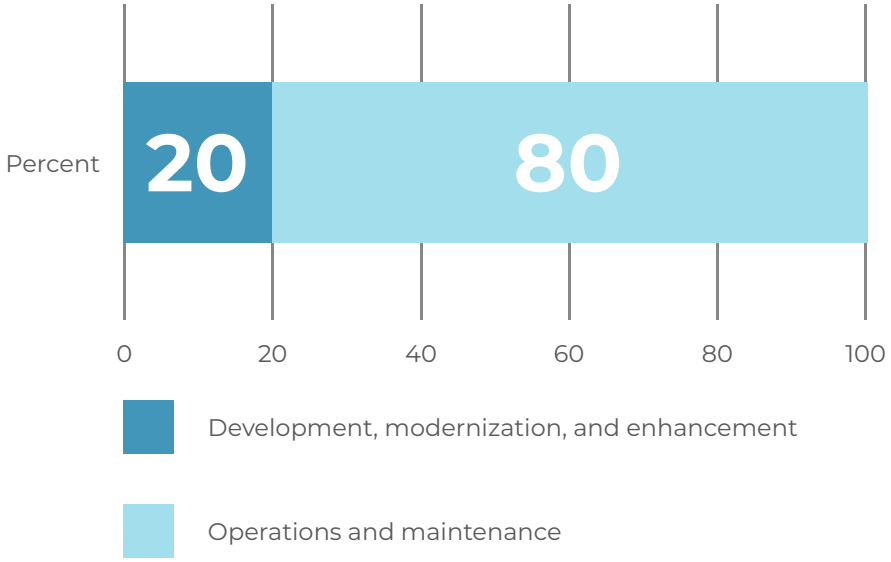
2. The Hidden Costs of Legacy Software

Many companies continue using outdated systems, regardless of the age or quality of the 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.

In 2019, the US Federal government spent 80 percent of the IT budget on Operations and Maintenance. This spending mainly included aging legacy systems, which posed efficiency, cybersecurity, and mission risk issues. To put that into context, only 20 percent of the IT funding was assigned to Development, Modernization, and Enhancement.[4]

Chart 16-5. IT Spending by DME and O&M



The percent of the governmentwide IT funding going toward DME and O&M (2018)

Returning to the top ten GAO legacy systems, their operation and maintenance cost taxpayers about $337 million every year. One of them, the 14-year-old Air Force "System 1" responsible for aircraft wartime readiness, has been recently completely updated. As a result, the initially anticipated increase in maintenance (from $21.8 million in 2018 to turned into $34 million annual savings.

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.**

$ **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.

2.1 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 the codebase 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 the 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, 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 a 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.

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 – navigation for drivers and journey visualization for customers. Indeed, why reinvent the wheel, when you can use the existing, tried and true solution at a fraction of the cost?

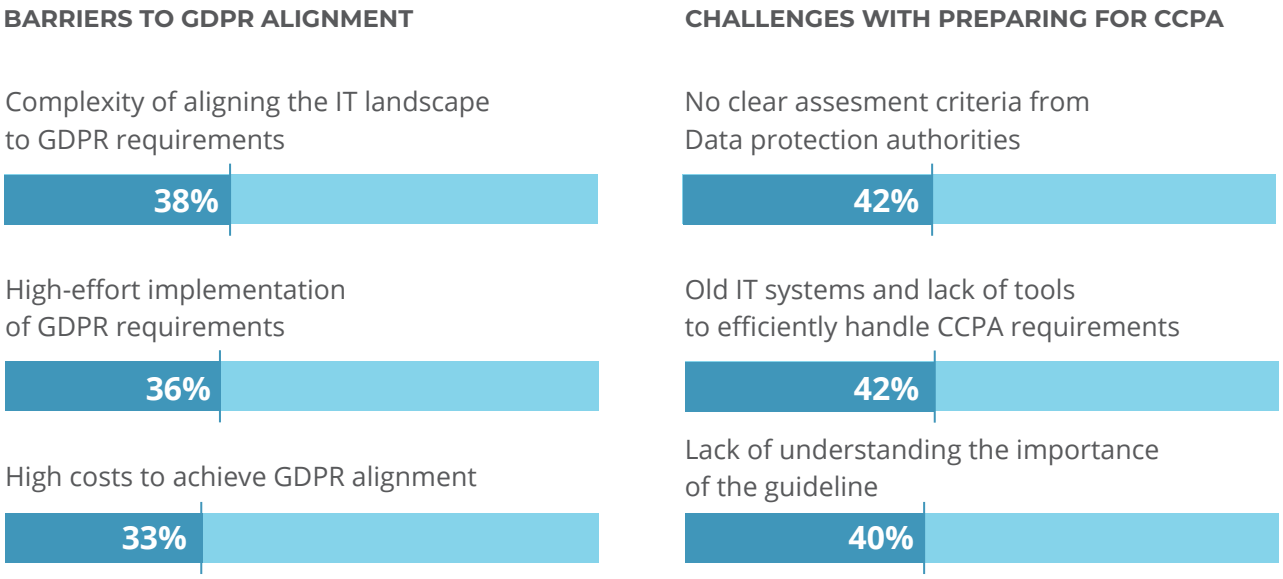
Modern technologies are integration-ready by default. API vendors typically provide support for most of the programming languages and frameworks out of the box. Yet, obsolete or rare technologies typically lack compatibility.

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 sectors, such as politics or law.

It's been over a year since the General Data Protection Regulation (GDPR) became effective, but organizations are still struggling to comply with it. And in the US, California has similar state legislation becoming effective January 1, 2020 - California Consumer Privacy Act (CCPA), which only adds to the frustration.

On studying this question, the Capgemini Research Institute found that legacy IT is one of the major (42%) challenges organizations face while preparing for the CCPA. [6]



Legacy IT - a major headache for the GDPR and CCPA, Source: Capgemini Research Institute, Data Privacy

2.3 Security

In light of increasing security breaches and compromises, almost 70% of the 2019 HIMSS Cybersecurity Survey respondents are still exploiting some legacy systems. The survey concludes that the continued use of legacy systems *"raises grave concerns regarding the vulnerability of the healthcare ecosystem."* But this can be applied to any other industry.

Indeed, legacy systems are usually less resistant to cyberattacks, 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.

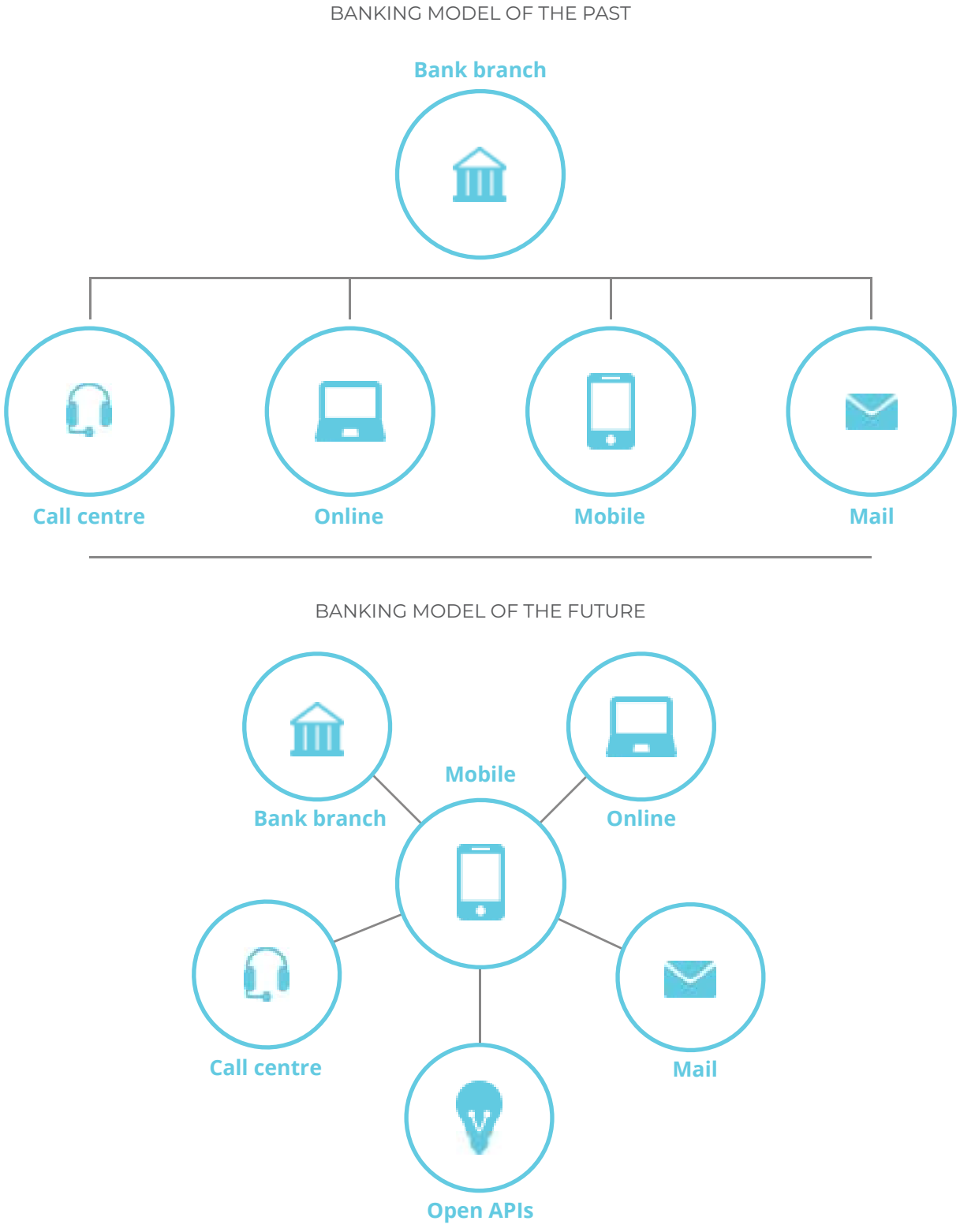
> *"Supporting a legacy operating system in your enterprise is as much about risk management as it is about traditional IT service management."* [13]
>
> — Vijay Samtani, Chief Information Security Officer at Cambridge University

In the worst case, this might resemble a leaky bucket, where you get a new breach as soon as the previous one is fixed.

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.

Currently, digital channels are increasingly driving growth in deposits and consumer lending, as evidenced by Citigroup and mobile banks like German N26 and American SoFi. Thus, a competitor's convenient website or a lightweight mobile application can lure away some of your customers. That's what is happening in mortg age and personal loan markets right now. Nonbank digital lending providers like Quicken Loans have captured a large market share. [22]



Mobile banking as a new banking model, Source: Deloitte

Among the digital transformation drivers in finance are business growth and competitiveness. But the primary motivators are greater efficiency and cost reduction. [22] The latter was proven back in 2013: Javelin Strategy & Research found that on average, processing a mobile banking transaction costs 42 times less than doing it offline at a physical bank. At that time, the prices were 10 cents and $4.25 respectively.

Such overhead is another significant element that adds to the cost of missed opportunity. While others are swiftly adopting digital and remote as their main channel for interactions saving on efficient and low-cost transaction processing, you might be lagging behind using outdated POS terminals and manual assistance to do the same job.

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. Meanwhile, IT modernization is grounded on agility, along with digital M&A (mergers and acquisitions) and cooperation with digital startups, as identified by the Harvard Business Review research. Agile methodologies can speed up the provision of IT services by 30 to 50 percent and they are particularly suitable for transformations with a great deal of uncertainty.[24] Thus, it's more effective to initiate modernization with the focus on continuous delivery integrating product development with IT operations. Such an approach is commonly referred to as DevOps.

Although moving from cumbersome legacy IT to agile and digitally-enabled IT can be complex and challenging, it is necessary to meet modern IT demands. Partially due to their inability to be updated and modified, legacy systems can hold back innovation, resulting in significant losses. Moreover, outdated 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. Initially done by humans, reengineering was later delegated to machines to make process changes constant and driven by the predictive capabilities of machine-learning algorithms. The process involves redesigning a system in response to changes in regulations or other environmental factors; migrating across platforms, databases, and operating systems, etc.

Based on another 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."[28]

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.

Academic research "The Fall of the Labor Share and the Rise of Superstar Firms" suggests "that the industries becoming more concentrated are those with faster technological progress." If you still doubt whether it is necessary to modernize your outmoded system, read our article that addresses thirteen signs identifying the need for digital transformation.

3. Preparing for the Digital Future

Despite the problems and risks related to outdated software, some companies are still lacking legacy-modernization initiatives. Most of them would only consider reengineering the current solution in case of an emergency, such as a complete system outage. Still, these days we can see a positive shift towards innovation. When in 2018 only 5 percent of CIOs considered themselves "digital innovators," 2019 shows that already 32 percent of them claimed to play a leading role in organization-wide innovation.

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.

3.1 The Business Case for Software Modernization

So, why modernize legacy systems? 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.

3.2 Considering the challenges and 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.

Challenges that derive from legacy modernization include the following:

- Personnel is usually unwilling to adjust to management changes. Motivation, training, and coaching will press them in that direction but will entail additional risk and cost.
- If there are multiple legacy systems within one corporation, their modernization should be articulated and prioritized in a corporate program that considers the required effort and time window for each system individually. To the contrary, simultaneous modernization may lead to a catastrophic impact that is not easily absorbed.
- Initially tailored for the specific platform functionalities the app ran on, legacy code should be handled with extra care, even if some pieces of it can appear to be no longer relevant and in need of replacement. For the same reason, it is important to make sure while migrating that the underlying software will comply with the new data interchange rules and requirements dictated by client applications and support resources.
- Having to deal with countless lines of code that only address a given corporate process can be a real headache, especially if there is a skills shortage.

Besides the challenges, there are multiple risks 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.[29]

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.

3.3 Estimating system modernization costs

The most expensive way to modernize is to not modernize at all. But, as in any software development project, you will need an estimation of your efforts, which is usually done using specific methodologies and tools.

COCOMO (Constructive Cost Model).
This methodology was created back in the 1980s and it uses a simple calculation: Man - Months = K1* (Thousands of lines of code) K2, where K1 and K2 are the constant values chosen in regard to the team size, their experience, and the complexity of the system. There are two models in this methodology: COCOMO I is used for estimating maintenance and COCOMO II calculates maintenance, migration, and reengineering efforts.

FPA (Function Point Analysis).
Another classic model, FPA, uses functional requirements to assess the functionality delivered to the user, which is manifested in UFPs or unadjusted functions points. These points are counted and evaluated in different project KPIs like performance or quality. FPA has inspired such popular frameworks as ESTIMACS and SPQR/20 [25], each considering different factors in the function point measurement. While SPQR focuses on estimating the complexity of the algorithms, code, and data structures, ESTIMACS considers the business side of a project, like staff or hours of effort.

The Putnam model.
This methodology allows for estimating the time and effort needed to finish the project if you know its size. Size is usually measured in lines of code. This is easily the simplest estimation method as it doesn't require any hard-to-get data any company can gather information about the time, effort (man hours) and size of a previous project. The model also allows for correcting the schedule easily correct the schedule if size changes or estimating the growth of effort when the delivery date moves closer.[26]

3.4 Checklist for successful application modernization

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 approach that would be the fastest to deliver value.
Based on the assessment conducted in the first phase, choose the modernization approach that best fits your needs and will help you deliver results fast. Aside from the modernization approaches, consider existing products you can use instead. There is no need to reinvent the wheel if there is a SaaS solution available at a fraction of the 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 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 makes your software easier 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 without a hitch, 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.

3.5 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.

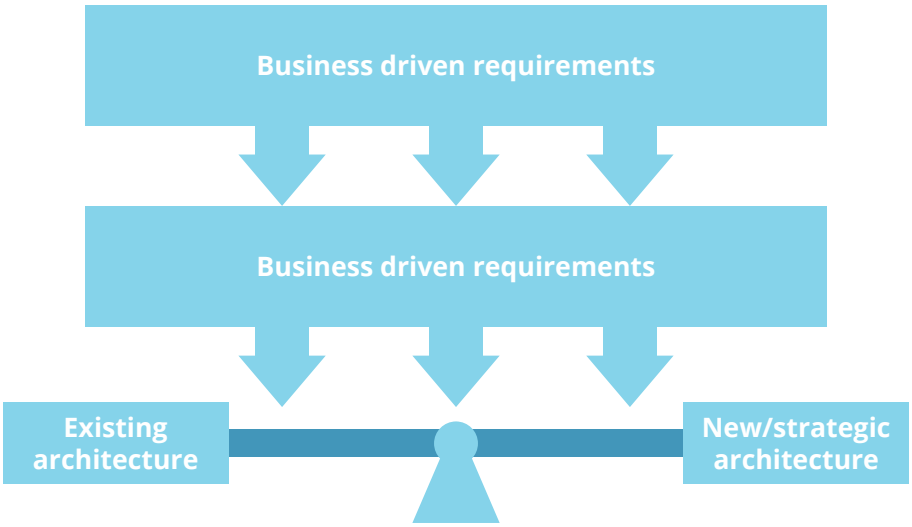
4. Legacy Software Modernization Best Practices

After conducting a thorough assessment of the legacy framework, it's time to decide on the modernization practices to apply.

4.1 Legacy enterprise systems modernization and replacement strategies

A well-chosen strategy is an efficient pattern for managing the legacy system modernization process. Here are the most popular legacy modernization strategies to consider while planning your technology transformation.

Architecture-Driven Modernization (ADM) is a coordinated strategy of understanding and advancing existing software assets like its functionality, components, or requirements. The main advantage of ADM is that it approaches modernization from an analysis- and design-based perspective, rather than source-to-source migrations. The main use of architecture-driven modernization comes in the forms of platform and language independence and interoperability. ADM will enable projects with aging software to become more agile.[17]



Bridging between the existing and new architecture through ADM, Source: International Journal of Scientific & Engineering Research

SABA is a high-level framework for planning organizational and technical issues during legacy system evolution and migration. SABA's iterative method assists in making decisions among different modernization options, starting from discarding the old software completely, to freezing it, or outsourcing. The framework tackles various legacy system problems while also providing a means to analyze the future implications of software choices.

Reverse Engineering Model is a strategy where the legacy applications are gradually migrated to the new environment, but legacy data migration is the last phase of the migration process. To access legacy data, the applications in the new environment use a Reverse Gateway. Reverse Engineering Model is a good choice for high-cost, long projects that may be undermined by the technology pace.

Visaggio's Value-based Decision Model (VDM) selects the most suitable software renewal process based on technical and economic metrics.[16] It is applied when economic returns or quality of a legacy system are lower than expected and helps decide on a better way to revitalize the system.

DevOps Contribution allows for speeding up the legacy modernization processes by swiftly deploying new software releases with a low degree of bug or errors while still complying with the target operational IT environment.

The **Renaissanc** model proposes a two-stage modernization: first, building a strong basis for the system evolution, often with the help of reengineering, and second, applying continuous improvement for the rest of the system's life. This method supports the notion that the system should never need another modernization as it will be changed iteratively.

WMU (Warrants, Maintenance, Upgrade) is a customer-centric method that chooses maintenance strategies based on customer satisfaction. For this, a lot of information should be collected and analyzed, like a customer satisfaction index (Net Promoter Score, for example), implementation quality (number of customer complaints), the volatility of the market (like the number of competitors), customer expectations, and more.

4.2 Legacy modernization methods

The system evolves influenced by different development methods – from adding a line of code to complete reimplementation. There are two methods for dealing with the legacy problem that involve major structural changes: revolutionary (big-bang) and evolutionary (band-aid). Yet, both have benefits as well as drawbacks.

The revolutionary method revolves around developing and carrying out a legacy system replacement strategy. Its implementation 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.

It is no surprise that many companies stick to modernizing their platforms rather than replacing them. However, synchronizing the operations of new digital and legacy IT teams poses a number of challenges:

- compatibility issues
- communications involving legacy systems can consume more network bandwidth than their modern counterparts, owing to the serial nature of their output.
- maintaining security on legacy systems can be difficult, since users cannot expect automatic protection from new threats.

The evolutionary method 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 method, where you focus on solving the problems instead of removing the factors that cause them.

> *"The technology is always getting faster, and automation simpler. We need to keep up with that and the business use cases and requirements that may emerge. In this case, we replaced every platform and phased out traditional software— although in the beginning we kept the legacy technology running until we had fully replaced the required functionality"*
>
> — Leon Bedaux, head of digital IT, KPN[8]

Nevertheless, there are many examples of successfully integrated mainframe systems. Liverpool Victoria is one of them. This insurance company's car business suffered from a website that mapped directly on to mainframe data. By separating the mainframe processes from the front end, the company introduced more user-friendly pages and made it easier to add features in the future.

Both revolutionary and evolutionary methods serve as a basis for the approaches to legacy modernization that will be described further.

4.3 Approaches to legacy system modernization

According to International Data Corporation, 65 percent of organizations will aggressively modernize legacy systems with extensive new technology platform investments through 2023. To be more precise, IDC predicts that digital transformation spending will grow from today's 36 percent to over 53 percent of all information and communications technology investment by 2023. The efficiency of the invested funds will heavily rely on the approach an enterprise chooses to follow.

Stefan van der Zijden, senior director analyst at Gartner, says: "*If you're faced with a legacy challenge, the best approach depends on the problem you're trying to solve. Replacement isn't the only option. The key is to understand if your problem is caused by technology, architecture or functionality of the application, and how each modernization approach improves those aspects*".

Below we suggest three modernization approaches to help you pick the one that can deal best with your current legacy challenges. While both Migration & Enhancements and Correction & Growth are drawn on the evolutionary method, the third approach called Complete Software Reengineering takes a revolutionary turn.

In addition, Cognizant presents an alternative classification of approaches primarily based on the level of new technology applied to a legacy system to modernize it. The classification includes the following approaches: total transformation, gradual replacement, the duct tape approach, improve existing, and no system change. The latter is the case when a company decides to delay transformation and observe the industry while checking whether its systems are agile enough to manage the growth. Later, we'll also reference Cognizant approaches to expand the vision on legacy modernisation best practices.

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 change requires 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

Correction & Growth. If the product technology stack is relatively modern and does not represent a threat to 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 updated, you can add more features on top of it. These might be third-party integrations or custom-built modules.

Following Cognizant classification this approach can be divided into two: *improve existing and the duct tape approach.* While the former presupposes a few minor additions, the latter offers comparatively big returns. Oftentimes the *duct tape approach* entails building a new application and then synching it with the legacy one to bridge the gap in functionality.

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

Complete Software Reengineering. Considered the most extreme approach, 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.

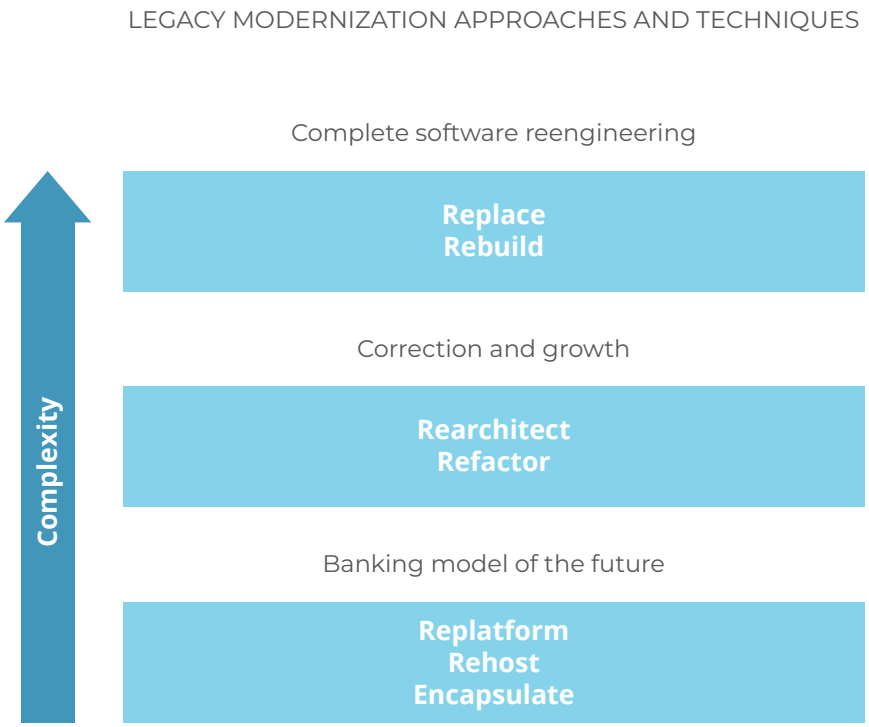
Depending on the functionality analysis and prioritization, the new product might 100% match the previous version in terms of functionality, or lack some features that are no longer required/used.
Cognizant gives a detailed description of the reengineering process separating gradual replacement and total transformation.

Gradual replacement follows the evolutionary pattern while modernizing the entire system one piece at a time Contrarily, total transformation is a complete rebuild from scratch using new technology or a third-party package as a foundation layer.

Case Study: AltexSoft & Fareboom: Co-Building Innovative Travel and Booking Solution to Outperform the Competition

4.4 Legacy system modernization techniques

Legacy modernization approaches have specific techniques applied to update obsolete technologies.



Legacy modernization approaches and the techniques they use, graded in terms of their complexity

Encapsulation is a technique for reusing legacy software components. While leaving the code in its current environment, encapsulation connects it to the new presentation and accesses layers via an API. That helps leverage the application and extend its features and value.

Encapsulation is based on the technology of wrapping that provides a new interface to a legacy component making it easily accessible to other software components. The little changes to the code minimize the risks. Consequently, encapsulation is among the fastest and most economical solutions. It is a good option when a legacy system has a high business value and good quality code. However, encapsulation will not solve problems already present, such as difficulties with maintenance and upgrading, since its main concern is the interface and not the internals of the legacy system.

Rehosting means moving a mainframe application unchanged to other physical, virtual, or cloud infrastructure. This technique is of the lowest cost and risk. While re-engineering projects can take years, rehosting is faster and keeps the underlying business logic intact meaning zero negative impact on the enterprise. As a result, the system operates in exactly the same way.

In terms of modernization, it makes sense migrating to modern open systems, like multi-tiered, SQL-based x86 environment, or the cloud.

Migrating to x86 architecture systems result in lower purchase costs and have reduced space, power and cooling requirements.

Cloud migration offers flexibility over on-site hardware in terms of resource scaling to match user demand. Aside from a cheaper upfront investment and no hardware outlays, operating through the cloud allows for more reliable data security, greater stability, and continuous updates. Cloud migration can be executed not only using rehosting, but also replatforming and refactoring techniques.

With the rehosting technique, an application is forklifted to the cloud as is, without any code modification. While offering a less resource-intensive migration process, rehosting doesn't generally make use of cloud-native features as do replatforming and refactoring techniques.

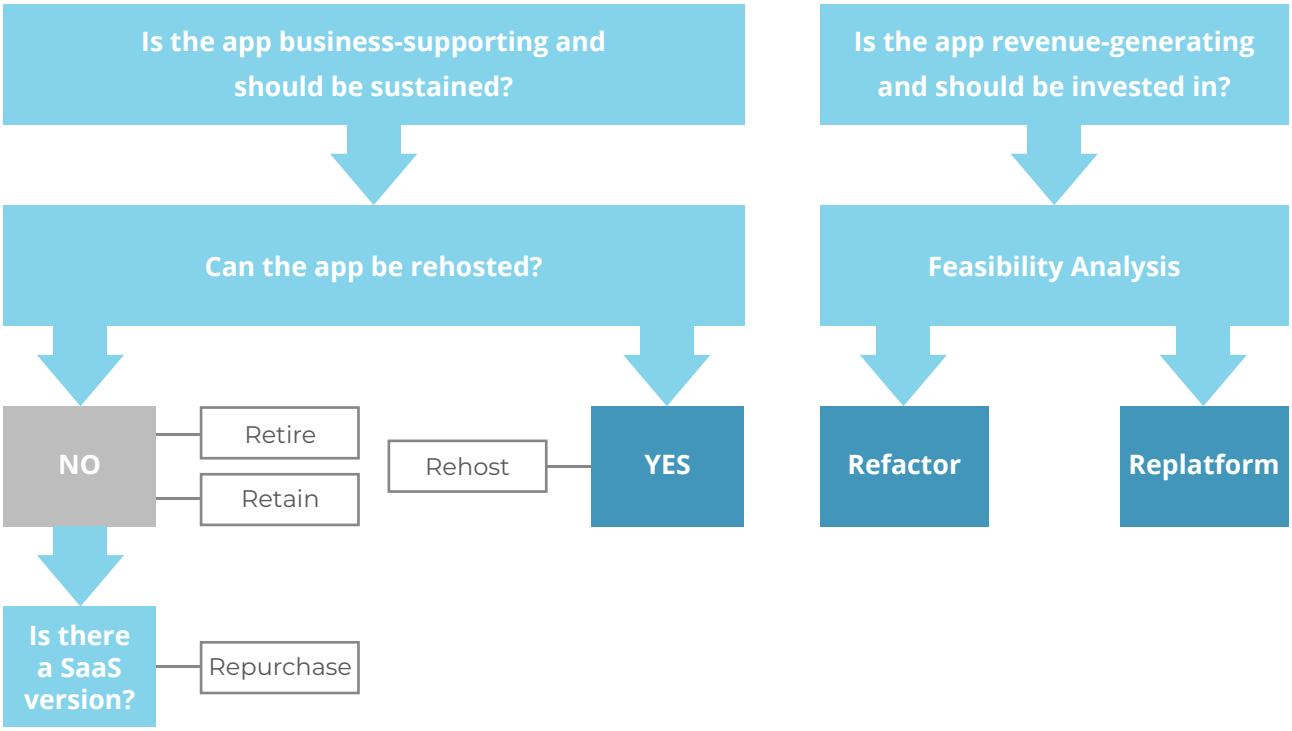
Replatform migrations include a bit of up-versioning to adjust the code to a new platform while preserving existing functionality. The minimal changes like using a managed database

offering or adding auto-scaling, a feature that automatically adds or removes compute resources, can help return the basic profit of cloud infrastructure. And that's perfectly fine because not all applications need the full benefits of being cloud-native. In terms of cloud migration, replatforming allows for base cloud functionality and cost optimization, without resource commitments required for refactoring.

Code refactoring presupposes restructuring and optimizing existing code without changing its external behavior. Refactoring an application component allows for solving technology problems and improving the component's features and structure. To learn more about code refactoring, visit our article on refactoring best practices.

By re-coding some portion of an existing application organizations are able to take full advantage of cloud-native features and maximize operational cost efficiency in the cloud.

Meanwhile, there are certain technologies that are not runnable in the cloud. In this case you should look for a SaaS alternative.



Cloud migration path, Source: Flux7

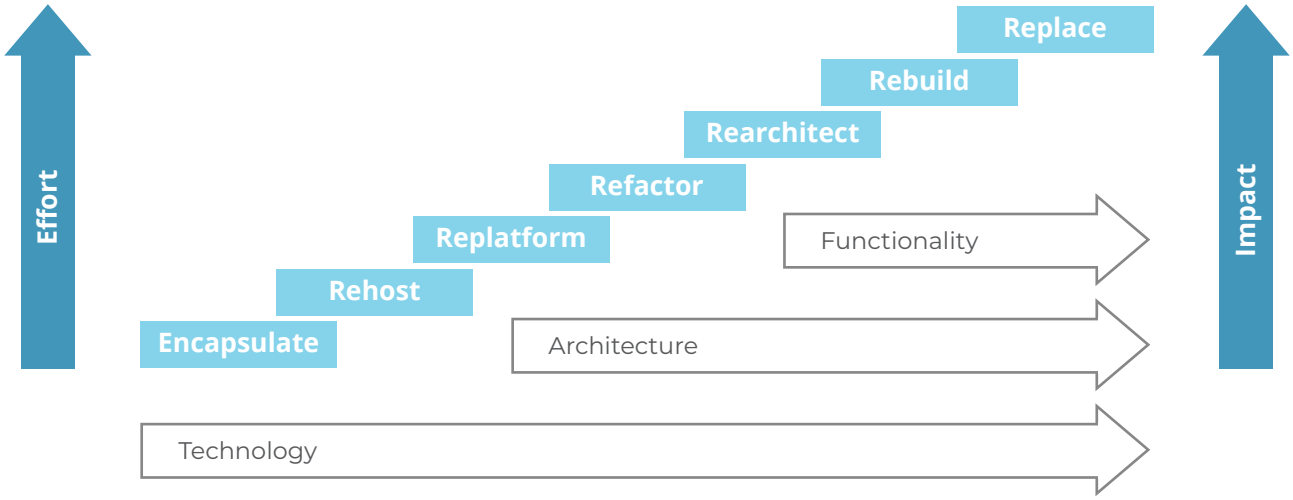
Rearchitecting means shifting to a new application architecture while altering the code to fully exploit the new and better capabilities of the platform. This technique has medium cost and risk, but also medium results.

Accenture identifies one of the latest tech trends as technology-driven partnerships that will allow companies to expand and scale into new environments and ecosystems. Enterprises can increase their partnerships by improving integration capabilities. This requires first moving on from an internal legacy system and rearchitecting themselves, since internal transformations and adoption of new technologies are crucial for making use of the tech-based partnerships. To do this, it is sensible to invest in microservices architecture, which enables rapid integration with many new partners through agility.

According to IDC Worldwide Digital Transformation 2019 Predictions, "By 2022, 90 percent of all new apps will feature microservices architectures that improve the ability to design, debug, update, and leverage third-party code." IDC claims that microservices architecture will lead to development of "hyperagile apps" that are highly modular distributed, continuously updated, and leveraging cloud-native technologies such as containers and serverless computing.

Rebuilding (Redesign) rewrites the application components from scratch while preserving their scope and specifications. At the same time, redesigning your application opens the door to new features, functionality, and processes that leverage the capabilities of modern technology and third-party platforms.

Replacing. Sometimes it is better to entirely replace the app with a different tool rather than invest in its modernization. While the reuse of existing legacy business logic is not possible in this case, some level of reengineering or customization of packages and rewriting business logic may be involved in this process. To sum up, modernization techniques impact different aspects of the legacy system. That said, encapsulation, rehosting, and replatforming affect the technology platform. Refactoring and rearchitecting can solve problems in the technology and architecture domains. Rebuilding and replacing allow for changing and adding functions and features, among other things.



Modernization techniques arranged according to the effort they take and the impact they bring to the application components, Source: Gartner

Conclusion

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

IDC predicts that digital transformation will attain macroeconomic scale over the next three to four years, changing the way enterprises operate and reshaping the global economy. According to their research, "More than half the global economy turns digital by 2023 requiring new species of enterprise to compete and thrive." [14]

To live up to the demands of the new digital transformation economy, organizations have to cease relying on outdated software and modernize their core technologies. Enterprises will benefit only when they stop seeing modernization as a one-time project and embrace it as a cycle.

"Change is now the norm. Just as we set a course based on our understanding of the technology landscape, that landscape changes. CIOs must accept that change is constant and work out how to get on the front foot - to shape change rather than being governed by it." [11]
— Mark Rogers, Chief Executive Officer, Logicalis Group

Taking advantage of third-party expertise might be of great help.We at AltexSoft 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. 9th annual Spok survey: Mobile Strategies in Healthcare – http://cloud.spok.com/EB-AMER-2019-Mobility-Strategy-Survey.pdf
2. Starting your modernization journey – https://www2.deloitte.com/us/en/pages/technology/articles/legacy-infrastructure-modernization.html
3. Nextgov: 10 Government Legacy Systems Cost Taxpayers $337 Million Every Year – https://www.nextgov.com/it-modernization/2019/06/10-government-legacy-systems-cost-taxpayers-337-million-every-year/157682/
4. The Washington Post: IRS to delay tax deadline by one day after technology collapse – https://www.washingtonpost.com/business/economy/irs-electronic-filing-system-breaks-down-hours-before-tax-deadline/2018/04/17/4c05ecae-4255-11e8-ad8f-27a8c409298b_story.html?noredirect=on&utm_term=.b4b8b62be966
5. Overview of Federal Information Technology 2018 – https://www.whitehouse.gov/wp-content/uploads/2018/02/ap_16_it-fy2019.pdf
6. Championing Data Protection and Privacy. A Source of Competitive Advantage in the Digital Century – https://www.capgemini.com/gb-en/wp-content/uploads/sites/3/2019/09/Report-%E2%80%93-GDPR.pdf
7. 2018 State of Cyber Resilience – https://www.accenture.com/t00010101T000000Z__w__/fr-fr/_acnmedia/PDF-84/Accenture-Security-State-of-Cyber-Resilience-2018.pdf
8. Modernizing IT for digital reinvention – https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Modernizing%20IT%20for%20digital%20reinvention/Modernizing-IT-for-digital-reinvention-Collection-July-2018.ashx
9. Accenture Technology Vision 2018 – https://www.accenture.com/t20180227T215953Z__w__/us-en/_acnmedia/Accenture/next-gen-7/tech-vision-2018/pdf/Accenture-TechVision-2018-Tech-Trends-Report.pdfChoose the Right Approach to Modernize Your Legacy Systems – https://www.gartner.com/doc/reprints?id=1-57V77J8&ct=180719&st=sb
10. Approaches and techniques for legacy software modernization – https://www.researchgate.net/publication/267181092_Approaches_and_techniques_for_legacy_software_modernization?enrichId=rgreq-00c4de99f6125c2c176b04d310f68f41-XXX&enrichSource=Y292ZXJQYWdlOzI2NzE4MTA5MjtBUzozNjIzOTMxODkyNzM2MDBAMTQ2MzQxMjY3NzE5OA%3D%3D&el=1_x_3&_esc=publicationCoverPdf
11. Logicalis Global CIO Survey 2017–2018 – http://www.us.logicalis.com/globalassets/united-states/downloads/cio-reports/2017-cio-survey-report.pdf
12. Choose the Right Approach to Modernize Your Legacy Systems – https://www.gartner.com/doc/reprints?id=1-57V77J8&ct=180719&st=sb
13. Encapsulation of legacy software: A technique for reusing legacy software components – https://www.researchgate.net/publication/220300651_Encapsulation_of_legacy_software_A_technique_for_reusing_legacy_software_componentsA Review on Architecture Driven Modernization – https://pdfs.semanticscholar.org/b412/62876eb791ebc47a76c5626bad04b224cc7b.pdf
14. IDC FutureScape: Worldwide IT Industry 2020 Predictions – https://www.idc.com/getdoc.jsp?containerId=US45599219
15. Legacy systems continue to have a place in the enterprise – https://www.computerweekly.com/feature/Legacy-systems-continue-to-have-a-place-in-the-enterprise
16. New Trends in Software Methodologies, Tools and Techniques – https://books.google.com.ua/books?id=oN3YBAAAQBAJ&dq=Visaggio%E2%80%99s+Decision+Model+(VDM)&source=gbs_navlinks_s
17. A Review on Architecture Driven Modernization – https://pdfs.semanticscholar.org/b412/62876eb791ebc47a76c5626bad04b224cc7b.pdf
18. Legacy Enterprise Systems Modernization: Five Ways of Responding to Market Forces – https://www.cognizant.com/whitepapers/legacy-enterprise-systems-modernization-five-ways-of-responding-to-market-forces-codex1377.pdf

19. Mobile Banking Adoption: Where Is the Revenue for Financial Institutions – https://www.fiserv.com/resources/Mobile-Adoption-White-Paper-January-2016.pdf
20. Javelin Identifies $1.5 B in Mobile Banking Cost Savings by Leveraging Omnichannel Approach – https://www.javelinstrategy.com/press-release/javelin-identifies-15-b-mobile-banking-cost-savings-leveraging-omnichannel-approach
21. 2018 Banking Outlook Accelerating the transformation – https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/gx-fsi-dcfs-2018-banking-outlook.pdf
22. Deloitte Insights: 2020 banking and capital markets outlook – https://documents.deloitte.com/insights/2020bankingoutlook
23. Fujitsu's Global Digital Transformation Survey Keport, 2018 – https://www.fujitsu.com/downloads/GLOBAL/vision/2018/download-center/FTSV2018_Survey_EN-1.pdf
24. Twenty-five years of digitization: Ten insights into how to play it right – https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/twenty-five%20years%20of%20digitization%20ten%20insights%20into%20how%20to%20play%20it%20right/mgi-briefing-note-twenty-five-years-of-digitization-may-2019.ashx
25. Software Estimation, Enterprise-Wide – https://www.ibm.com/developerworks/rational/library/jun07/temnenco/index.html
26. The Comparison of the Software Cost Estimating Methods – https://www.computing.dcu.ie/~renaat/ca421/LWu1.html
27. Renaissance: A Method to Support Software System Evolution – http://www.cse.dmu.ac.uk/COMPSAC/wimpe/secretpath/authors/author.93/paper/paper.93.pdf
28. Companies Are Reimagining Business Processes with Algorithms – https://hbr.org/2016/02/companies-are-reimagining-business-processes-with-algorithm
29. Why Reengineering Projects Fail – https://www.cs.cmu.edu/~aldrich/courses/654-sp05/readings/Bergey99.pdf

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,

Suite 300, Carlsbad, CA 92011

+1 (877) 777-9097

Global HQ

11/13 Hromadyanska Str.,

Kharkiv, Ukraine 61057

+38 (057) 714-1537