



**Application
Modernization Roadmap:
Prepare and Implement
for Seamless Transition**

Xebia

App modernization is a hot topic, but did you know that only 21% of efforts are successful? So, what makes it so hard to pull it off? And how can you make sure you're preparing for success? Dive in and learn how to prepare for modernization and implement it successfully.

It's simple. If you are a company that has developed a piece of software, you will need to update it at some point and continue to do so for as long as it is actively used. First, because tech changes. Fast. Second, your users' needs change — maybe even faster! And third, cybercriminals find new ways to exploit software every day, which can pose a real risk for your company and your users. So, it makes sense to start your app-modernization journey before things become complex and slow, and you risk losing market share or becoming a security breach headline.

But, as the 21% above may have given away, application modernization can be challenging. Do you know who built what part of the application and when? Why were specific choices made? What application modernization strategy best suits your needs? And what impact will modernizing one piece of the software have on the system as a whole?

Unfortunately, vFunction's 2022 research reports many modernization efforts fail due to a lack of skills, alignment, resources, and tools or a corporate culture that does not support the change. However, with the right strategy and an intricate understanding of what it takes to pull it off, it can propel your application to new heights.

Dive into this e-book to learn how to set yourself up for success! We discuss the initial hurdles that can disrupt your modernization journey and how to overcome them. We've also included a cheatsheet so you can get started on this endeavour on the right foot.



What is Application Modernization?

Application modernization is the process of upgrading and transforming existing software applications to adapt them to modern-day technologies, practices, and user expectations. It involves revamping outdated or legacy systems to improve performance, scalability, security, and user experience. There are several strategies through which you can achieve application modernization – often referred to as the “6 Rs”. They can range from a ‘lift and shift’ (also referred to as rehosting) to rebuilding or even completely replacing the app. The purpose of application modernization is to enable organizations to remain competitive, deliver value faster, and respond to changing market demands more effectively.

Application modernization involves so much more than simply updating software and continuing business as usual.

For instance, the Deutsche Bahn (Germany's national railway) replaced its legacy system with an easy-to-use cloud application, including smart data use and IoT — a move that unlocked a huge potential but also impacted the whole organization. Another example is the U.S. Internal Revenue Service (IRS), which moved its 80's e-filing system to the cloud, making it accessible to any taxpayer from any device in a secure and scalable way — imagine what impact that had on every single part of the company!

Application modernization is known to spark a vast transformation that involves changing your company's structure, processes, culture, and skillset to fit the evolving landscape and the principles underlying the modernization effort. Sounds challenging? This e-book is here to help.



Index

- Challenge 1: Cultural Shift
- Challenge 2: Skills Gap
- Challenge 3: Resistance to Change
- Challenge 4: Unclear Priorities
- Challenge 5: Integration with Legacy Systems
- Your Cheat sheet to App Modernization

Challenge 1: Cultural Shift

App modernization is often a complex project that significantly influences the organization and the people in it. This first chapter will focus on the cultural shift required to make sure the way your people work (together) is in line with the new application.

1.1. What's Culture Got to Do with It?

Let's start this chapter with an example of how app modernization can impact your **company culture**. *

“Meet company X, whose software had remained unchanged for years. It operated top-down - decisions were centralized, and change was met with hesitation. Employees often felt that their ideas went unheard, and there was a sense of stagnation in the air. Then, a transformational decision was made - it was time to modernize one of the company's more prominent, and outdated, applications, embracing new technologies and methodologies. As the journey began, rigid processes started to take a backseat, and employees were encouraged to actively participate. Teams collaborated more openly, sharing insights and brainstorming innovative solutions. The culture shifted towards adaptability and change readiness. Employees became excited about the possibilities of the modernized software, which promised efficiency and improved customer experiences. The organization invested in training and skill development, nurturing

** What is the definition of company culture? Company culture can be described as all the values, beliefs, attitudes, behaviors, and norms that shape a work environment. It affects how people interact, work together, and make decisions and is crucial for identity, morale, and productivity.*

a culture of continuous learning. Finally, the modernization effort had another remarkable effect - it placed the user at the center of their operations. Employees were motivated to understand and meet customer needs, resulting in improved products and services. Of course, the journey was not without challenges. Some employees resisted change, fearing job role shifts or uncertainty. Effective change management and clear communication were crucial to addressing these concerns. Over time, the organization successfully aligned its evolving culture with its strategic objectives. X's decision to modernize software transformed its technological landscape and company culture. X became a place where creativity thrived, innovation was celebrated, and employees felt empowered to impact the organization's future."

We admit this story is a simplified version of reality, but we decided to share it anyway. Why? Because it effectively demonstrates that successful modernization requires changes beyond simply adopting a new tech stack. It is necessary to provide the right environment and sufficient support to take full advantage of the benefits of modernization grants. Discounting these factors will guarantee failure.

"Modernizing applications involves changes in processes, tools, and workflows. Employees may need to adapt to new technologies and ways of working. If managed poorly, this can lead to resistance and frustration, potentially affecting morale and job satisfaction."

Building Engineering Cultures

At Xebia, we take culture and its impact on the organization very seriously. We are committed to helping you establish the right ecosystem, where IT facilitates organizational progress rather than hindering it. Building an engineering culture has benefits beyond supporting modernization efforts – it helps retain top talent, establish a strong foundation for innovation, improve collaboration, and more.

To assist you in this endeavor, we have put together essential information on Building an Engineering Culture as well as common misconceptions that can lead you astray. Find them here:

- [Building Engineering Cultures](#)
- [Six Misconceptions About Engineering Culture](#)

1.2. Culture Eats Strategy for Breakfast

Do you remember the famous saying, “Culture eats strategy for breakfast”? It aptly describes companies trying to pursue a strategy its people aren’t ready for yet. Sure, modernization strategies promise technological advancement and a competitive edge, but they will most likely fall flat if not embraced by the workforce.

Aligning your organizational culture with your application modernization strategy and goals is paramount for success. To succeed in today’s dynamic digital ecosystem, you must cultivate a culture that not only embraces change but thrives on it — **a culture of adaptability** that encourages people to experiment, learn, and adapt to new technologies and methodologies.

According to the [PWC 2023 CEO](#) survey, 40% of global CEOs believe their company will only be economically viable in ten years if they learn to adapt to change. When your culture supports change, it becomes a driving force behind your modernization efforts, ensuring that your teams are eager to embrace the necessary transformations.



1.3. Get Your Culture to Work in Your Favor

“You should want your company culture to foster innovation, risk-taking, cross-functional collaboration, and continuous learning - the critical components of achieving modernization success.”

While no one can change company culture overnight, there are some things you can do to move towards a culture of adaptability. These are four pointers we recommend you focus on (first):

- 01. Make innovation a priority:** One of the reasons that modernization efforts fail is that they are not prioritized at C-level-where risk aversion, short-term oriented behavior, cost efficiency, and reluctance to change are common. So, first and foremost, it is important to encourage innovation, long-term thinking, and (calculated) risk-taking among leaders.
- 02. Take risks, place bets:** encourage innovation by moving to a culture of “placing bets” - first introduced by Eric Ries, author of “The Lean Startup” as a conscious decision to invest resources, time, and effort into developing a new product or feature with the expectation that it will bring value to customers and the company. It’s a concept that has been adopted and adjusted by many product leaders and organizations seeking to foster a culture of adaptability, encouraging continuous learning and creativity.

“At Xebia, we are a great fan of creating opportunities for our clients and employees to safely explore new ideas, experiment with emerging technologies, and challenge the status quo without fear of failure. For example, we organize Innovation Days and internal hackathons to inspire people to develop smart solutions and technological breakthroughs.”

- 03. Facilitate cross-functional collaboration:** Organizations increasingly opt for cross-collaboration because it enhances and speeds up processes. For example, setting up value-stream teams removes any dependencies on other teams to get work into production. We recommend enabling holistic problem-solving and creating clear channels for communication and information sharing across skills and roles to foster understanding and alignment.
- 04. Adopt a learning mindset:** An Agile mindset views its values and principles as a guiding framework, not a rigid prescription. Regardless, its influence spreads through all aspects of the organization, from our approach to challenges to cultivating innovation, collaboration, and holistic growth. So, how does this work in practice? Encourage an iterative process, break initiatives into smaller outcome-focused deliverables for quick feedback, and prioritize customer-centricity. Remember, for Agile to succeed, the whole organization must embrace this new way of working, from leaders to managers to engineers to legal, marketing, and support teams!

1.4. Pre- and Post-Modernization Culture: Spot the Differences!

By now, we know that parts of your culture will need to change to be successful at modernizing your application. But just how big a difference is there between a pre- and post-modernization work culture? Read on, it's illustrated below:

A traditional (or pre-modernization) work culture:

- **Stuck in silos:** different teams work in isolation. There is limited collaboration between development, operations, and IT; as a result, handoffs add delays in getting work done.
- **One responsibility:** employees have specific (narrow) roles and responsibilities. For example, developers focus on coding, IT manages infrastructure, and operations ensures stability.
- **Staying in the comfort zone:** employees are accustomed to on-premises infrastructure, plus the technologies they've been working with for years.
- **Lengthy processes:** deploying and scaling are done manually, which is time-consuming, but the team is used to it being that way. In addition, dedicated teams handle the provisioning and maintenance of hardware, and change boards might exist to (manually) approve any changes made.
- **Limited individual empowerment:** rigid hierarchies, bureaucratic processes, and centralized decision-making stifle creativity and initiative. Employees may lack autonomy, hindering their ability to make meaningful contributions or innovate.

Versus

A modern work culture focused on outcomes:

- **Bye-bye silos:** teams work more collaboratively, breaking down silos. Developers, operations, and IT gather to optimize all aspects of their work, including customer needs, cloud resources, and application performance.
- **On the lookout for Pi- and comb-shaped profiles:** in this new set-up, it is super valuable when employees are not an expert in just one specific area but are good at a combination of things, like security and development or engineering and project management.
Read more: <https://www.linkedin.com/pulse/which-letter-shaped-future-employees-leaders-esin-akay/>
- **What's next?** A cultural shift has taken place that makes people more open to new technologies and ways of working. Employees are keen to experiment with innovative tools and techniques and explore the opportunities the cloud platform offers.
- **Customer value is #1:** with operational tasks streamlined, teams can focus more on delivering customer value through rapid feature development and improvements.
- **Empowered individuals:** people feel empowered to make decisions, collaborate freely, and take initiative. In this setting, creativity and innovation are encouraged, and hierarchies are more flexible, fostering a sense of ownership and impact among employees.

4 Key Takeaways:

- App modernization involves more than technology. It requires a cultural shift in line with the new technology landscape and modernization strategy.
- Expect changes in your daily work, like new tools to reflect new ways of working, iterative development, cross-functional teams, and a focus on continuous learning and improvement.
- Support the culture shift by prioritizing change, encouraging innovation and learning from experiments and failures, facilitating cross-functional collaboration, and adopting an Agile mindset to help overcome resistance.
- Unlike traditional cultures, a modern, outcome-focused culture shatters silos, values multi-skilled individuals, embraces new technology, prioritizes customer value, and empowers individuals.

Challenge 2: Skills Gap

The success of your modernization initiative is reliant on several factors. In addition to the culture, another important factor to address is your company's skillset. What skills do your team master today, and what will they need to learn to make the modernization effort successful?

2.1. Skills in High Demand

Looking at what's trending, what users are requesting, and the technology that people in SaaS companies are using, today's modernization efforts often call for skills in the following areas:

- **Cloud computing:** modernizing legacy usually involves migrating from on-prem to the cloud. Employees must learn about cloud platforms, deployment models, and services.
- **DevOps:** your team needs to know DevOps, continuous integration, delivery, deployment, and tools like Docker, Kubernetes, and CI/CD pipelines.
- **Microservices architecture:** transitioning from monolithic legacy systems to microservices architecture demands experience with service decomposition, API design, and containerization.
- **Modern frameworks:** mastering modern frameworks like Angular, React, Flutter, and others allows employees to fully leverage the latest technologies and best practices resulting in enhanced performance, scalability, future-proof strategy, and developer productivity.
- **Version control and collaboration:** proficiency in tools that enable synchronous communication and collaboration, distributed version control, visual boards, task management, and road mapping is vital for modern and efficient software development. Think Azure DevOps, GitHub, Slack, and so on.
- **Testing/test automation:** testing is crucial to ensure consistent performance and reliability of the new changes. That's why a dependable testing environment and its consequent automation will help streamline the process and solidify

a foundation to ensure you deliver quality in all future changes.

- **Security best practices:** as hackers become to adept, cybersecurity threats evolve. That's why your team needs to know security best practices, including encryption, authentication, secure coding, and be able to integrate these earlier in the process. Having updated knowledge on this topic will help minimize vulnerabilities in the system and safeguard your organization and customers firmly from privacy and security threats.
- **Data analytics and visualization:** want to benefit from data? Your teams must be proficient in understanding what data would be helpful for them to act on, as well as how to interpret insights. Consider customer feedback and analytics, code quality, DORA metrics, test coverage, service uptime, error rates, and more.
- **Agile ways of working:** your team will need to build new skills in communication, teamwork, and cross-functional collaboration, as well as how to work iteratively and incrementally by focusing on outcomes. This approach fosters a work environment that readily accommodates ongoing adjustments and pivots in response to fresh insights and customer feedback.
- **Developer efficiency:** the team should embrace Generative AI-based assistants to help them with their work. For example, Developers could use tools like GitHub Copilot to help them understand, write, test, and improve code. This makes them more efficient. To do this, the organization must implement such a tool and educate the users.

As you can see, to keep your tech up to date, your team's skills must be up to date first. But here's the catch: finding the right talent is not easy. Many organizations are caught between a rock and a hard place. Finding developers who are able and willing to deal with old code and outdated systems is tough. Sometimes, it's even harder than finding skilled individuals to work on innovative, complex projects. It's all about uncovering hidden talents in your existing crew, offering upskilling and re-skilling programs, and finding those who can lead, inspire, and contribute. Investing in improving team skills will also feed into the culture of learning discussed earlier while contributing significantly to employee motivation.

State of the CIO Survey 2023:

“As IT departments look to fill critical skill gaps, they are focusing on areas related to modernization and transformation, with technology integration/implementation (42%), IT cloud architecture (40%), and risk/security management (36%) most in demand. Manufacturing companies (50%) and financial services companies (51%) are most in need of integration/implementation skills, while risk/security management remains a greater priority for education (54%) and healthcare (50%). IT leaders anticipate difficulty finding qualified experts, especially in advanced areas such as blockchain (22%), cybersecurity (22%), and data science/analytics (22%). Cutting-edge competencies like AI/ML and robotic process automation are also expected to be difficult to fill when staffing key projects.”

2.2. So, How Do You Bridge the Skills Gap?

Training, upskilling, re-skilling, managed services, or even far-reaching automation can all serve as a solution. But many roads lead to Rome. Let's observe one - let's look at how Dutch software company Wigo4it overcame its skills gap.

Setting the scene

In the quest to futureproof its software on Azure and transition to SaaS, Wigo4it faced a significant challenge: an IT skills gap in new technologies and modern methodologies.

How Wigo4it overcame its skills gap

- 01. Prioritize Learning and development:** Wigo4it invested in tailored training programs: online sessions, workshops, and training courses to upskill teams in Agile, DevOps, and Microsoft cloud technologies.
- 02. Collaborate with technology partners:** Wigo4it collaborated with partners such as Xebia and Pluralsight to offer targeted training and familiarize employees with advanced technologies, closing the knowledge gap. However, apart from upskilling efforts, the success of the initiative largely depended on the willingness of individuals within the teams to take charge, provide ongoing support and encouragement, and foster a collaborative environment. By having leaders in crucial roles, the team could establish a shared vision, facilitate seamless collaboration and communication, and ultimately achieve successful implementation
- 03. Encourage knowledge sharing:** a culture of knowledge sharing was fostered through lunch sessions, workshops, and internal knowledge-sharing platforms. Collaborative practices like pair programming, refinement sessions, Azure meetups, and innovation focused “Funovation Days” were introduced to make learning more engaging.
- 04. Embrace Agile and cross functional teams:** Agile methodologies and cross-functional teams enabled employees from diverse backgrounds to share their expertise and collaborate on innovative product initiatives.

Results

- **More operational efficiency and innovation:** Wigo4it empowered its workforce to build and run applications on Azure, leading to increased innovation and operational efficiency gains.
- **Trailblazers attract talent:** Wigo4it is now a more attractive employer in the public sector, drawing talented individuals seeking challenging and satisfying experiences.
- **An enhanced employment experience:** a culture of initiative, confidence, problem-solving, and low turnover emerged, empowering teams to take ownership of their work.

By prioritizing digital skill development, investing in employee education, and fostering a culture of continuous learning, Wigo4it successfully bridged the IT skills gap and positioned itself for success in the realm of SaaS.

Check out the complete case study here:

<https://customers.xebia.com/xebia-help-wigo4it-transit-to-azure-and-devops>

4 Key takeaways:

- Modernizing an application involves upgrading skills in many areas, including Cloud computing, DevOps, GenAI, frameworks, version control, testing, security, data analytics, lean-agile working methods, and more.
- Finding skilled professionals for outdated systems is as challenging as finding people trained in new technologies. Not having the correct skills or training significantly contributes to the failure of app modernization efforts.
- To support modernization, leaders must prioritize investing in employee skills such as technology integration/implementation, cloud architecture, and risk/security/cost management.
- IT leaders should expect challenges finding experts, especially in advanced and emerging technologies and areas.

Challenge 3: Resistance to Change

Another reason app modernization is tricky is people's natural need for autonomy and tendency to avoid perceived risk.

“ App Modernization goes beyond technology changes. It affects people directly, and getting your team involved early and ensuring that they have a say in the decisions that are made along the way is a crucial part of the journey. Organizations that push force changes on teams, without buy-in, risk resistance to change and a lack of buy-in during implementation and ongoing adoption”

- Esteban Garcia, Managing Director - Xebia Microsoft Services US

3.1. Reasons People Resist Change

App modernization is a big change. And like all change, it is often met with resistance. Research shows that app modernization involves as many as [97% of people expecting resistance from someone or some part of the organization](#). Do you foresee any objections? It's important to understand and address them as early as possible. Below, you'll find the most common reasons people struggle to adapt. We have presented them as a checklist to help you prepare your team and, hopefully, reduce the reluctance to change!

Checklist 'Reduce Resistance to Change'

- **The why:** Have you clearly communicated the reason for the change?
- **Incentives Incentives:** Are current incentives and metrics aligned with the change, or do they oppose it?
- **Job security:** Have you addressed possible concerns about job security and performance?
- **Workload:** Will employees be expected to do more? Can they deal, or are they at capacity?
- **Success:** Do employees believe they can be successful in the new situation?
- **Peer pressure:** Is peer pressure leading to resistance?
- **Status quoquo:** Is there a strong attachment to the current state due to past success or comfort?
- **Fear:** Have you addressed fears, uncertainty, and possible challenges?
- **Comfort zone:** Do you have a plan to support employees as they step out of their comfort zones? Are your employees aware of such plans?
- **Experiences:** Have previous attempts to change been unsuccessful? Did you learn a lesson from these past experiences, and will you do things differently this time?

“On the tech side, the biggest fears associated with modernizing are large-scale change and losing your job, while on the business side, people mostly fear the financial implications and a poor ROI.”

- *Why App Modernization Projects Fail, 2022 Research Report by vFunction*

3.2. From Resisting Change to Embracing a New Future

There are many different reasons for people to question change. And although a checklist can provide a framework to address (potential) challenges, we believe it's important to dive deeper into a selection of change management techniques in an app modernization context.

- **Clear communication:** Communicate the reasons behind the modernization effort and the benefits it will bring to the organization. Address concerns and involve employees in decision-making.

“One of the reasons modernization efforts fails is because executives do not accurately set expectations. We advise you to set achievable goals with your teams instead of setting deadlines for them. Foster open dialogue, manage priorities, and provide feedback regularly to ensure everyone is (and stays) aligned with the bigger picture.”

- **Collaborative decision-making:** involve people in decisions that impact them to ensure they feel heard.

“For a large retail chain considering app modernization, a collaborative EventStorming session engaged the team in understanding the issues (the why). After the session, we focused on Value Stream Mapping to distinguish the high-level challenges and what a positive future state would look like. Employees were involved throughout the process, and the proposed solutions agreed upon

at the end of these activities had their full buy-in.”

- **Pilots and proof points:** start with small initiatives to demonstrate the positive impact of modernization, alleviate skepticism, and build momentum.

“Hired to build an API that would serve as a template for future APIs; we met the company’s development team — keen to collaborate, but also used to working with legacy tech and outdated languages. So, we decided to use Clean Architecture principles and share our ‘way of working’ with them. The team-built parts of the new API while we were available for questions. The project acted as a learning opportunity for the client team and focused on the “how” and “why” behind technology choices, allowing them to better manage future projects by themselves. In addition to learning new skills that would change their development forever, the API offered immediate value by removing a huge dependency between teams.”

- **Empower change advocates:** identify change champions within your organization who can advocate for the benefits of modernization, mentor, and guide peers through the process.

“We highly recommend establishing Communities of Practice (CoPs) - structured groups where professionals with shared missions or interests (e.g., Engineering Managers, Product Managers, Architects) share knowledge and insights. For example, a manager dealing with the complexity of migrating a specific application to the cloud can present a problem to this community. Here, colleagues can share lessons learned from their own migrations, like pitfalls in containerization or best practices for API integration. By openly discussing these topics, the community helps leaders avoid common mistakes and apply proven strategies, increasing the organization’s overall ability and buy-in to modernize applications.”

- **Listen to people:** implement feedback mechanisms for employees to voice their concerns and suggestions, making them feel valued and engaged in the transition.

“For one of our customers, we enabled a Miro board as a place to voice concerns, opinions, and ideas. Hundreds of people joined and gave feedback, which the facilitators then consolidated into themes and presented to leadership. Action items were created based on the input given. Because there was also a lot of positive feedback, the board also gave managers a good idea of what the teams appreciated.”

- **Allocate space:** it’s important to make space for change to happen. Don’t just assume that people will “fit it in” to their schedule. The bigger the change, the more time people will need.

“One executive took four months off, canceled 80% of client work, and relinquished his operational duties to give the company space to reinvent itself while he was in a mountain cabin rethinking the future of the business. While this might be extreme for most, the outcome was remarkable!”

3.3. Examples of Companies Resisting Change: What Happened?

BlackBerry: BlackBerry resisted the shift toward touchscreen smartphones and app ecosystems, instead sticking to physical keyboard-based devices. This decision led to a decline in market share, eventually pushing the company out of the smartphone market.

Yahoo: without a clear purpose, Yahoo struggled to keep up with Google’s search innovation and grasp opportunities like purchasing Facebook. The company’s reluctance to adapt and innovate led to a significant decline in relevance and market share.

The above examples are drastic. Of course, not all resistance to change equals the end of your business. However, in today’s digital world, companies are moving fast, and there are always startups about to enter the scene with one mission: disruption! Be prepared to continually evolve your business proactively, not merely respond to change or wait for a new competitor to appear.

4 Key Takeaways:

- Involve employees in decision-making and allow them to put their stamp on the change. If people feel they have a say and are a part of the process, they are more likely to embrace it.
- Factors contributing to resistance include the fear of an increased workload, job security, financial implications, or stepping outside their comfort zone.
- To effectively address resistance, choose transparency, clear communication, collaborative decision-making, pilot projects that demonstrate benefits, and introduce employee feedback mechanisms.
- The examples of BlackBerry, Nokia, Yahoo, Myspace, and many others show that when a market seems to be moving in one direction, it’s good to explore what (unique) way you can move with it to avoid losing your market position altogether.

Nokia: slow to embrace touchscreens and app stores and sticking with traditional mobile phone designs and old operating systems for years resulted in a dramatic decline in market share, eventually leading to the sale of Nokia’s mobile phone business to Microsoft.

MySpace: one of the early social networking giants missed a few critical opportunities: it failed to address user privacy concerns adequately and struggled to provide a seamless user experience. As a result, it lost its leading position in the social media space.

Challenge 4: Unclear Priorities

The previous chapter touched on the importance of shared understanding for the success of modernization. Your team needs to know and understand why it is important, what goals you are trying to achieve, and what challenges you may encounter along the way. If everyone is not on the same page, you run the risk of people going in different directions, leading to chaos.

This chapter will discuss another huge challenge organizations face when merging technology and innovation: unclear priorities fueled by conflicting ambitions and a profound divide in interests.

4.1. Different Interests

Having clear priorities that align with your goals is critical - your chosen strategy must fit the desired outcomes. For example, if you want your app to be lightning-fast and include advanced data-driven features, you won't achieve that with just lift-and-shift (rehost). In that case, you need to choose a strategy that improves the speed of your app and supports data integration.

So, first and foremost, answer the following questions: what is your goal? What strategy have you chosen? Does everyone agree? With that, can we conclude that we are all on the same page and pursuing the same priorities? It sounds simple, but business and technical professionals, for example, have very different interests and profiles. A CEO is often more concerned with finance and ROI, while an engineer focuses on the benefits of a particular technical design. Without proper alignment, you're likely to sail two different courses.

“Different roles have different drivers, different perspectives on business strategy, a different understanding of the issue, and a different way of looking at impact and desired business outcome.”

4.2. Challenges in Prioritization

Different roles have different perspectives and priorities. It makes sense. But what else can cause priorities to be so far apart?

- Unclear business strategy: if it's not clear what your company stands for and what you want to achieve, your foundation is already shaky. It's hard to explain how modernizing an app helps to achieve your goal without a goal.
- Changing market conditions: market conditions, customer expectations, and technology are all changing rapidly. This can make it challenging to determine which aspects of app modernization should be prioritized.
- A lack of a disciplined approach: you need the correct data and key performance indicators (KPIs) to measure impact and expected return on investment. Otherwise, you risk relying on subjective assessments rather than data-based insights, resulting in incorrect prioritization.
- Limited resources: limited budgets, talents, and time can force organizations to make difficult trade-offs. As a result, it may be challenging to prioritize modernization initiatives on an ongoing basis.
- Poor communication: ineffective communication and a lack of shared understanding and language among stakeholders can lead to misalignments about the goals, benefits, and priorities of app modernization.



4.3. The 7 Steps to Clarity

Follow these seven steps to ensure that priorities and goals are clear before and during your modernization effort:

01. Preparation**Preparation:** understand what you want to migrate, how you want to migrate, how customers use your current application, and what is more important to do now versus later. Don't fall into the trap of rushing without understanding what is needed.

"We worked with a U.S. life insurance company that had been through some research and training themselves and, as a result, believed they needed containers. We listened to their current challenges and end goals and then determined that an API would be most beneficial to them. We built this with their team successfully."

02. Active advocacy: Leaders play an important role in fostering confidence in both the goal and the process. They must provide active sponsorship, clear communication, and support throughout the modernization process to increase the likelihood of successful outcomes.

"One of the most important things executives can do to prevent efforts from failing is recognize that app modernization is a strategic initiative, and it's your job to allocate staff and resources and support your teams so they can be successful."

03. Business case: by working with your teams to build a strong business case, you'll gain insight into key priorities and what's most important to focus on first.

"Evaluate your application portfolio and assess architectural complexity, technical debt, security posture, and practices that no longer meet customer or employee needs. Determine the driving force behind the modernization project, whether it is a cloud mandate, growth strategy, or the need to retain customers. Quantify expected business outcomes and use this data to set priorities."

04. Resources: getting the funding, talent, and technology you need for a successful outcome can be challenging, especially in times of a skills gap! Linking the work required to strategic business outcomes will help strengthen your case.

"Close to 50% of executives and technical professionals find securing the budget and staff needed for change the most challenging step in modernization."

05. Long-term perspective: given that the average modernization project in large organizations takes 16 months to 2 years to complete, ongoing management support is necessary to ensure that efforts do not stagnate or derail as circumstances change.

06. Skills development: evaluate the skills within your team and identify any gaps that need to be addressed. Consider upskilling your team to prevent a two-class system, where existing employees only focus on old technology and newcomers get to play with the new tools.

07. Tools: employees often demand specific tools to speed up release times and improve work effectiveness and efficiency. Make sure your team has access to what they need.

"In one company, the expense of a virtual whiteboard was questioned until people realized it could save them time and money due to the reduced number of business trips made purely for easy collaboration purposes. The cost of the tool seemed negligible in comparison."

4 Key Takeaways:

- Different organizational roles often have conflicting priorities; aligning on common goals and strategies is crucial.
- An unclear business strategy, changing market conditions, a lack of data-driven insights, limited resources, and poor communication can lead to ambiguous priorities.
- Leaders must provide active sponsorship, clear communication, and support throughout the modernization process to increase the likelihood of successful outcomes.
- To address the challenge of unclear priorities, follow seven steps: preparation, active sponsorship, building a business case, securing resources, maintaining a long-term perspective, skills development, and providing necessary tools for the team.



Challenge 5: Integrating Legacy Systems

Many companies now find themselves at a crossroads in updating their software and systems. They understand that modernizing applications is critical to staying competitive in today's business landscape, but as we've read, it comes with challenges. The final hurdle we will highlight in this e-book before moving on to the implementation cheatsheet is integrating modern technology with legacy systems.

5.1. Bridge the Divide

Legacy systems, by their very nature, are built on outdated technologies and programming languages, which often makes them difficult to integrate with modern architectures and frameworks. This 'technological heterogeneity' poses quite a challenge — but it's not the only one. Let's look at what else you might need to prepare for!

5.2. The 7 Challenges of Legacy Systems

Below, we have listed seven characteristics of legacy software that can make it hard(er) to integrate the software with a modernized application.

01. Data

Legacy systems often store data in formats and structures that differ from those in their modern counterparts. During migration, this can lead to data integrity, mapping, compatibility, testing, and governance challenges. Ensure that data fits seamlessly into the new application's scheme without loss or corruption through careful planning and following a meticulous process.

02. APIs

Today, APIs (Application Programming Interfaces) are the lifeblood of seamless integration. Unfortunately, legacy systems may lack standardized APIs or offer limited integration capabilities. Retrofitting with appropriate APIs can be time-consuming and often requires complex reverse engineering.

03. Scalability

Unlike legacy, modern applications quickly scale with growing workloads and user demands. You may need to re-engineer large portions of the legacy codebase to ensure that your applications can meet performance requirements without becoming sluggish.

04. Security

Older technologies tend to have weaker security measures, making them attractive cyberattack targets. Integrating legacy with modern cloud-based environments calls for excellent planning to keep sensitive data safe and vulnerabilities identified as early as possible.

05. Documentation

Over the years, documentation for legacy systems may have become outdated or lost, complicating modernization efforts because understanding and reverse-engineering legacy systems now becomes a Herculean task.

06. Users

Legacy systems were developed for past requirements, not the present. It is likely that user needs have changed, so it is often advisable not to just "lift and shift" but reconsider what is most essential to modernize and what can be left behind.

07. Dependency

Many legacy systems rely on third-party software that may no longer be supported or compatible with current technologies, further complicating integration, creating security risks, and requiring creative workarounds.

5.3. Integration Is Not Impossible

Reading the above, you might feel disheartened. But there's light at the end of the tunnel. Integrating modern technology with legacy systems is challenging but not impossible if you embark on the journey well-prepared. There are several ways to deal with legacy systems. Depending on your organization's specific needs, the nature of the systems, and the desired outcomes, here are five possible ways to go about this challenge!

○ Event-Driven Architecture (EDA)

Event-Driven Architecture (EDA) is a software design paradigm that enables systems to react to and process events and state changes. This approach is characterized by the production, detection, consumption, and reaction to events. Events are varied; they can range from user actions (like button clicks and form submissions) to system events (receiving an API request) to error events (connection failure, unavailable files) and more. In an EDA, system components operate asynchronously, often through event producers (which generate events) and event consumers (which process those events). The architecture is highly adaptable and scalable, making it ideal for environments where systems must respond dynamically to complex, unpredictable conditions, such as real-time data processing, user interface design, or distributed systems.

○ Data virtualization and Migration

Since existing data may be structured in a way that is not compatible with the new technology/architecture, data transfer from the legacy system to the new infrastructure is often an unavoidable part of app modernization.

Data migration is the process of moving, transforming, and integrating relevant data into the updated system to ensure the modernized application functions properly and leverages the full potential of the updated infrastructure. Data migration should be done in a way that enables the legacy app to run side-by-side

with the modernized version for a while to verify the behavior and performance of the updated system with production data. This allows the new application to be taken into production, with great confidence. This does require the migration process to be fully automated and repeatable.

On the other hand, data virtualization involves technology that allows organizations to access and manipulate data without needing to move or replicate it physically. It is also a way to access and use data from differing sources, including legacy systems, as if in a single, unified database.

Data virtualization and migration both ensure that the new system has access to relevant information. However, by providing a virtual layer that abstracts underlying sources, data virtualization will give you real-time data access and transformation capabilities while simplifying integration and minimizing the need for extensive data migration. It can serve as a precursory step before complete migration by reducing complexity, minimizing downtime during the migration itself, and saving costs.

“Let's say you're a bank that wants to provide customers with a modern banking app. However, your legacy mainframe systems store customer account data in non-standard formats. To avoid the complexity of data migration, you can opt to implement a data virtualization solution, which will allow your new app to access and manipulate customer account data stored in the legacy systems as if in a modern, unified database without needing a full-scale data migration project.”

○ Microservices

Microservices enable the gradual integration of legacy software with modern systems by breaking down monolithic applications into smaller, manageable components. This approach allows legacy and modern services to coexist, communicate, and evolve independently, simplifying integration and promoting system flexibility.

○ Containerization

Containerization allows legacy software to be encapsulated in isolated containers, providing a consistent and portable environment. This simplifies integration with modern software by ensuring compatibility, efficient deployment, and scalability while minimizing conflicts and dependencies between the two.

○ Incremental approach

Refrain from being tempted to go for a big-bang approach when integrating systems. Instead, consider a phased approach. Start with incorporating specific modules or functionalities, testing them thoroughly, and then gradually expanding the integration to cover the entire system. This can help identify and address integration issues in a controlled manner.

4 Key Takeaways:

- Familiarize yourself with the term “technological heterogeneity” and understand that this is why you cannot assume that, after modernization, your apps will still interact in the same way.
- Be aware of the seven features of legacy software that may complicate modernization efforts.
- Your integration approach should align with your strategic goals, budget, and timelines. Each method offers its advantages and challenges, and you should carefully consider your unique circumstances when selecting the most suitable integration strategy.
- Integration with legacy systems is challenging, but with the right strategies and a well-thought-out plan, you can bridge the gap between the old and the new to remain competitive and innovative in a rapidly changing technology landscape.



Your Cheat sheet to App Modernization

Modernization is a fact of life. Various factors both internal and external, constantly drive the need for software evolution and so, it cannot be ignored. These factors can range from the necessity to improve upon outdated or deteriorating software, streamline complex architecture compounded over the years, scale to accommodate a larger user base, reduce dependencies on obsolete systems, or simply stay compliant to updated laws and regulations. Modernizations enable companies to pivot, scale, and foster resilience, allowing businesses to navigate disruptions and seize opportunities confidently.

Deep-dive into your why:

As we mentioned above, there are many reasons to modernize. However, when embarking on this journey, the first thing to understand is whether your reason for modernization is a reactive one or a proactive one.

- Reactive: Do you have a problem now that is hindering your software functionality?
- Proactive: Are you anticipating a problem in the future? Or perhaps you have an ambition to improve upon your existing infrastructure but not sure how to respond to it.

Clearly discerning why you need to change will not only allow assess the urgency and necessities of the modernization but will also lay the foundation for your modernization strategy.

Xebia offers preconfigured and tailormade assessments to dive deeper into your goals and you can build a solid foundation for your modernization strategy.

Ensure you've thought about the necessary factors:

One cannot simply jump into a modernization process. Depending on the scale of your project, it can be intricate and complex. Here is what we recommend factoring in before starting:

Your (desired) business impact: Drill down on exactly what needs to be changed. While the assessment, as mentioned earlier, can help you better decipher your needs, it is important to be specific about the desired changes (speed, usability, AI inclusion)

Security implications: Data security should be a key priority in ensuring your software and business remain healthy in the long run. Understand how your modernization process impacts your current system.

Costs: Software modernization will often require a budget to ensure not only that the tech is in place (new technology/adoption costs) but also that the team is able to work with it (upskilling/re-skilling costs). Not everything has to be at a detriment to the organization though. It is also crucial to factor in how it will impact the costs of the system in the long run. For example, can the optimization help save money in the long run because of the switch?

Development: We've reiterated a number of times the importance of maintaining a happy and empowered team. This is a big transition that will require change, so, ensuring your team feels supported with updated tools, upskilling opportunities, and automation possibilities will help you create a great Dev experience.

Run: Fully account for how the changes impact performance, scalability, reliability, stability, observability, and other factors that are crucial for the business.

Understand your options

There is no one-size-fits-all approach to app modernization. Depending on your needs, it can either be a small adjustment or a complete overhaul of the existing infrastructure.

Here is a quick rundown of some common approaches to modernization.

Rehost (or Lift-and-Shift): Move to the Cloud

Quickly move your application to the cloud without major modifications. This is a flexible option that offers scalability while minimizing disruption.

Re-platform: Enhance Compatibility Re-platforming entails moving an application to the cloud while making necessary modifications to optimize it for the chosen platform, striking a balance between rapid migration and leveraging cloud benefits.

Refactor: Restructure for efficiency Reimagine your application's internal code to improve maintainability, performance, and scalability, achieving long-term efficiency gains. This is often achieved by reorganizing code modules or optimizing algorithms.

Rearchitect: Build for the future

This is a holistic approach where you redesign your application from the ground up, adopting modern architectural patterns like microservices and containerization for future growth.

Replace: Adopt new solutions Strategically adopt new solutions when your existing application no longer aligns with business objectives or technology trends.

Rebuild: Start Anew

Start fresh by identifying a new off-the-shelf solution or developing a custom application that better meets your organization's needs.



Build a strategy

Assess your application landscape: Conduct a thorough analysis of your legacy applications, identifying their functionalities, dependencies, and technical debt. This assessment will lay the groundwork for you to start figuring out the right approach for your modernization plans.

Define clear objectives and goals: Set specific and measurable goals for your modernization effort, such as enhancing user experience, improving system performance, reducing maintenance costs, or increasing scalability. These objectives will guide your modernization roadmap. Ensure that you factor in (desired) business impact, security implications, costs, development, and performance factors as discussed earlier.

Choose the appropriate modernization approach: Select the right approach based on the complexity of your applications, urgency of objectives, and available resources.

Allocate budget and resources: Plan and allocate the necessary funds and resources for the modernization project. Consider both short-term and long-term costs, including technology upgrades, hiring skilled professionals, and training and development initiatives.

Implement and monitor progress: Execute your modernization plan according to the defined timeline and budget, working with trusted partners or in-house teams. Monitor progress against key milestones and evaluate results against your original goals and ROI. Iterate and repeat the process for other applications.

Anticipate the hurdles

We've gone into length about the challenges you'll face during your app modernization process in this e-book. From cultural shifts to skills gaps, resistance to change, unclear priorities, and integration complexities, when you embark on this journey, your leadership skills will be put to the test! However, you and your team can overcome any hurdle with careful planning, strong commitment, open communication, change management, and the right technical strategies.

So, how do you set your business up for success? Ensure you have a clear vision aligned with business goals, realistic timelines, adequate resources, and the willingness to evolve technology and culture in tandem. Modernization must have active sponsorship from the top down, with leaders continuously conveying its importance and providing ongoing support.

Cross-functional collaboration, upskilling employees, incentivizing change advocates, and implementing feedback loops are critical for success. Last but not least, be sure to have some tech strategies up your sleeve to smoothen integration between legacy and your sparkly new app!

Execute!

Once a strategy is in place, we can finally begin the actual work. We have compiled some practical suggestions for you below. These tips will serve as accelerators into your modernization adventure.

- 01. Embrace AI Assistance:** Embracing AI assistance, such as integrating GitHub Copilot, can help streamline the process of code changes, code generation, bug fixes, improving overall development efficiency. The use of gen AI in app modernization not only accelerates the development cycle but has the potential to enhance the quality and accuracy of the code, ultimately leading to improved operational effectiveness.
- 02. Establish a Cloud Landing Zone:** Create secure and scalable cloud environments using best practices for networking, security, and governance. Cloud providers often provide extensive information on best practices and guidelines to help you get started. This will help you create a well-architected cloud foundation for your modern applications.
- 03. Prioritize Loose Coupling:** Design applications with loosely coupled components to minimize dependencies and enable independent development, testing, and deployment. Utilize technologies such as APIs, message queues, and event-driven architecture to facilitate seamless communication and integration between application components.

- 04.** Implement monitoring mechanisms: The journey does not conclude upon modernization; it requires continuous maintenance. Implement monitoring, logging, and alerting mechanisms to ensure the smooth operation of your modernized application. Utilize tools like Prometheus, Grafana, Azure Application Insights and AWS CloudWatch to monitor performance, troubleshoot issues, and optimize resource utilization.
- 05.** Prioritize Security and Compliance: It is crucial to keep up with security and compliance measures to ensure the long-term health of your system. By implementing robust identity and access management (IAM) controls, encryption, and compliance monitoring, you can safeguard your applications and its data. Leveraging cloud-native security services and following best practices will also help protect your modernized application.
- 06.** Automate reasonably: Implement CI/CD pipelines to automate the building, testing, and deployment of your modernized applications. Utilize tools such as GitHub Actions to enable rapid and reliable delivery of new features and updates. Make sure to separate deployment of the new app from release; this way, taking a new version into production becomes a human decision.
- 07.** Pay attention to User Experience: User experience is a critical factor in attracting and retaining users. Legacy apps are often difficult to understand or access for end-users. Additionally, it is often very difficult to measure how satisfied the customers are with the product. Leverage modern frontend frameworks, responsive design principles, and accessibility standards to enhance the user experience of your modernized applications. Rely on techniques like A/B testing to get feedback on how the users use your system.

Conclusion

Application modernization is a strategic necessity for organizations to remain competitive, meet evolving customer demands, and safeguard the company, its data, and its customers. Simply put, modernization is a fact.

If you fail to update, your software will quickly deteriorate, with all associated consequences. Yet, as we have explored, application modernization is also a complex undertaking with potential pitfalls. However, they are not entirely impossible to overcome. When well-prepared, application modernization holds the power to transform organizations and smartly position them for innovation and resilience to market shifts.

Trouble Getting Started?

App modernization is a complex endeavor. Regularly reviewing your tech stack and adjusting it according to your long-term strategy, customer demands, and market conditions is crucial in driving success. However, given the variations in operational needs for different organizations (based on needs, industries, taxonomy, and business KPIs), best practices are simply a starting point. It is crucial to tailor these approaches to the organization's unique needs.

If you need guidance in this endeavour, Xebia is here to help. We house seasoned experts who can tailor your modernization to your unique needs and ensure you get the results you want.

So, if you are determined to ensure your modernization is a success, but are unsure which path is best for your business, we're happy to help.

Contact us today. www.xebia.com



Sources and Recommended Reading

Sources:

[vFunction, 2022, Why app modernization efforts fail](#)

[Foundry's State of the CIO 2023 Study](#)

[PwC's 26th Annual Global CEO Survey](#)

[Deutsche Bahn Migrates Its Entire IT to the Cloud](#)

[The IRS is building its own online tax filing system. Tax-prep companies aren't happy](#)

[State of DevOps 2023](#)

Recommended Reading (per topic)

<https://www.goodreads.com/book/show/57196550-the-programmer-s-brain>

<https://www.manning.com/books/re-engineering-legacy-software>

<https://www.amazon.com/Beyond-Legacy-Code-Practices-Software/dp/1680500791>

<https://leanpub.com/legacycode>

[<https://www.amazon.com/Working-Effectively-Legacy-Michael-Feathers/dp/0131177052>](https://www.oreilly.com/library/view/software-design-x-rays/9781680505795/<u>https://www.oreilly.com/library/view/software-design-x-rays/9781680505795/</u></p></div><div data-bbox=)

[25](https://www.linkedin.com/pulse/which-letter-shaped-future-employees-leaders-esin-akay/<u>https://www.linkedin.com/pulse/which-letter-shaped-future-employees-leaders-esin-akay/</u></p></div><div data-bbox=)