mobidev mobidev.biz contact@mobidev.biz +1 916 243 0946 (USA/Canada)

Andrii Sulymka, JavaScript Team Leader at MobiDev

Choosing the right modernization approach for legacy software involves several critical steps to ensure the process is efficient, cost-effective, and aligns with business goals. Here's a concise checklist to help you decide.

Healthcare Software Modernization Checklist

☐ **Define Objectives:**	Clearly outline the goals of modernization. These could include improving performance, enhancing security, reducing maintenance costs, or adding new features. Ensure these objectives align with the broader business strategy.
☐ **Engage Stakeholders:**	Involve all relevant stakeholders, including business leaders, developers, IT staff, and end-users. Gather their input to understand their needs and expectations. This helps in prioritizing features and identifying potential risks.
☐ **Assess the Current State:**	Begin with a thorough assessment of the existing system. Identify outdated components, dependencies, and pain points. Evaluate the architecture, code quality, and technological stack. Understand the business processes the software supports and the criticality of each function.
☐ **Evaluate Modernization Approaches:**	

Rehosting (Lift and Shift):	Move the legacy application to new hardware or cloud-based hosting. The core application remains unchanged. This is great when the legacy system is stable and functional, but the underlying infrastructure is outdated. This helps your organization take advan tage of the improved scalability, reliability, and cost-efficiency of modern hosting platforms. Pay special attention to the compatibility of the new hosting environment. Compatibility problems can cause performance issues if the application was not designed for modern hosting architecture.
Software Refactoring:	Modify the code to improve its structure without changing its external behavior. This approach enhances maintainability and scalability.
Application Replatforming:	Migrate to a new platform or technology stack while making necessary modifications. It balances between cost and improvement.
Rebuilding:	Rewrite the application from scratch using modern technologies and architectures. This is the most comprehensive approach but also the most resource-intensive. Although it can help your organization take advantage of the latest software technologies, there is significant cost and risk involved when rebuilding a mission-critical legacy system.
Replacing:	Replace the legacy system with a new off-the-shelf solution. This can be quicker but may require significant changes to business processes.
Encapsulating:	Wrap your app in a new interface called a wrapper. This approach provides more modern capabilities and integration without modifying the core legacy system. It's best to use this approach when the legacy application is especially important and cannot be easily replaced or upgraded. This also allows your organization to quickly add new features and functionality to the existing system. However, maintaining the wrapper can be difficult as the legacy system evolves. Seamless data flow between the legacy system and new interfaces can also be a challenge.

☐ **Perform a Cost-Benefit Analysis:**	Analyze the costs, benefits, and risks associated with each approach. Consider factors like time, budget, technical feasibility, and potential disruptions to the business.
☐ **Develop a Tech Strategy:**	Create a detailed plan that includes timelines, milestones, resource allocation, and risk management strategies. Ensure there is a clear path from the current state to the desired future state.
☐ **Implement and Iterate:**	Execute the modernization plan in phases, starting with less critical components to minimize risk. Continuously monitor progress, gather feedback, and make adjustments as necessary.

By systematically assessing the current state, defining clear objectives, and carefully selecting an approach that balances cost and benefit, you can effectively modernize your legacy software and align it with your business needs.

MobiDev can help you with:

- Current healthcare system assessment by providing software audit
- Modernization approach selection taking into consideration business goals, technical limitations, budget and time restrictions, formed into a Tech strategy with timelines, milestones, resource allocation, and risk management strategies
- Allocation of a dedicated development team of healthcare software modernization experts or in-house team augmentation with Middle & Senior level software engineers