ZUCI SYSTEMS
PERFECTION. ALWAYS.

Application Modernization:

Assessment, Methodology, Types, Advantages, and Challenges.

Business applications are crucial to the success of any company in the modern era since they power critical operations and aid in client interaction. Many of these programs have evolved into their current state to serve needs better and boost efficiency.

Technology that is trending today may be obsolete tomorrow; this cycle never ends. Despite how challenging it may be, businesses must keep up with the pace of these shifts. Regarding technology, a company must have a clear and results-oriented application modernization strategy. So, let's get to know it.

TABLE OF CONTENT

What is Application Modernization?

Application Modernization aims to upgrade inefficient, hard-to-maintain, or out-of-date corporate software. “**Application Modernization**" improves the effectiveness, scalability, and cost-effectiveness of an existing software system or suite of applications. Modernizing an application can involve various activities, such as refactoring code, migrating to a new platform or architecture, adopting new technologies, adding new features or functionality, improving performance, and enhancing security.

With a CAGR of 16.7 percent between 2022 and 2027, the market for application modernization services is projected to expand from a current value of USD 15.2 billion.*

***Source: MarketsandMarkets**

Why Is Application Modernization Necessary?

Companies may experience difficulties if they continue using outdated programs. For one thing, it'll make it harder for people to work together because your company's legacy apps can't talk to the cutting-edge ones you're using.

Moreover, legacy apps must improve security, interoperability, and workflow sharing. Organizations will be set back by their prolonged penetration.

The task of scalability will be complex, as most modern technologies will not be capable of supporting older systems. Thus, updating older programs is the only option.

Many factors call for updating legacy applications:

Legacy applications: Legacy programs are nevertheless used by many companies today, even though they were built years ago and cannot handle the complexities of modern operations. In addition to needing the scalability and adaptability required by current programs, these may be difficult to update, manage, and integrate with other systems.

Technological advancements: Applications that were cutting-edge a few years ago may need to be updated due to the rapidity with which technology evolves. Adopting advanced tools and services like ***cloud computing, machine learning, and artificial intelligence*** can be facilitated by updating and modernizing legacy programs.

Changing business needs: Businesses must update their software as their needs evolve. It is necessary to update apps developed initially for use on-premises so they can be deployed in the cloud or to update desktop applications for mobile devices.

Cost efficiency: Because of the potential need for specialized personnel or hardware, keeping older apps up and running could cost more than doing so with a newer one. Businesses can save money on upkeep and boost productivity by updating their software to the latest version.

Security: Legacy applications may be vulnerable to security threats, as they may lack modern security features and not receive updates or patches. Organizations can improve their security posture by modernizing applications and protecting against cyber threats.

Modernization Approaches For Applications

Strategies that can be used to upgrade or modernize an existing piece of software. Rehosting, refactoring, re-platforming, rearchitecting, and rewriting the program are all examples of such strategies.

Each method has benefits and drawbacks; ultimately, the decision will be influenced by things like the existing state of the application, the needs of the business, and the availability of resources. The purpose of application modernization is to enhance the usability, speed, scalability, and safety of older programs so that they can function in today's advanced technological settings.

There are several types of application modernization approaches, including:

Rehosting
This method entails relocating an application without changing its source code from its present infrastructure to a new environment. Moving on-premises apps to the cloud via rehosting is expected due to potential financial and operational benefits.

Refactoring
During a refactoring process, minor modifications are made to the code of an application to enhance its efficiency, usability, or maintainability. This method could improve the quality of the application as a whole while lowering technical debt.

Replatforming
"Re-platforming" refers to moving a program to a new platform while keeping its source code intact. This method can enhance performance and scalability with minimal impact on the underlying application.

Rearchitecting
During re-architecture, large-scale alterations to an application's design are made to take advantage of the most recent developments and industry standards. While this strategy has the potential to enhance scalability, flexibility, and maintainability, it also necessitates a substantial commitment of time and resources. To optimize an existing application's codebase for scalability and reliability on a **cloud-migration,** developers make changes to or add new features to the application's source code.

Rebuilding
When an application is rebuilt, it is developed from scratch, utilizing up-to-date tools and infrastructure. This strategy can yield the most significant enhancements in functionality, performance, and scalability, but it is also the most time-consuming and expensive to implement.

Replace
It entails removing an old program and switching to one hosted and maintained in the cloud (SaaS). As part of this approach, a service provider handles the app's infrastructure, middleware, application software, and data under the terms of a service agreement that guarantees the apps and its data's availability and security. The application's current status, the business's needs, and the availability of resources are all things to consider while deciding on an approach. The best strategy for a company depends on careful study and analysis of its demands.

Application Modernization: **The Challenges**

Transitioning from the current design to one that offers flexibility, efficiency, and many other benefits can be challenging because of certain people's resistance to change. It takes a lot of work to update applications because of all the obstacles. It is also widely known that not all businesses that embark on the road to application modernization make it to the end.

Technological and human talent gaps

A steep learning curve involves working with the mainframe structure and its parts. Learning about each technology is challenging but figuring out how they all work together is a distinct challenge. The app is functioning without any more input from the developers. Finding new employees to work on the transferred app may be necessary. Because of this, expenses and complexity rise. Also, there are problems such as missing or obsolete documentation and patched source code.

Not having sufficient financial resources

In many instances, money is a significant obstacle. Updating an application would require a significant initial investment, which few businesses can muster.

Traditional ecosystem

Technology from several sources is brought together to form an application upgrade. A single application may require multiple protocols, languages, databases, file formats, etc. These elements include the application's ecology, and abruptly altering them could have disastrous effects on the company's operations.

Making sure the information is integrated

The difficulty of this situation is enormous. Whether attempting a complete infrastructure migration or the "life-and-shift" methodology, there is always the chance that some data will be lost. Another widespread problem is redundant data. Due to the critical nature of data for running a firm, mistakes in this area are not tolerated.

Data Transfer

Much information could be lost in the backend of an old application. Terabytes or petabytes of data may have been exchanged during business transactions. As a result, moving information from an older to a newer program has become difficult. Data Migrations are more sensitive and losing them could lead to mistakes if they happen. To overcome these obstacles, firms should implement an application modernization plan that promotes ease of use, productivity, and the ability to scale.

No solid strategy

A lack of strategy is why companies must do more to update their apps. It's not easy to do; only some people can find the apps they need, edit them, agree on a plan, and create a budget.

Modernization Of Applications: **Best Practices**

Code reviews, feature extensions, and software migrations are all systematic methods of **Application Modernization.** Let's go deeper into the reasoning behind these procedures and how they work together to provide a comprehensive approach to application modernization.

Dissecting the complexities

Always begin by segmenting monolithic apps into smaller pieces before continuing. Concentrate on one thing at a time. Tools such as cloud APIs and containers make this possible.

Identify and isolate the application from the infrastructure

Unshackling your IT and software systems is a prerequisite before starting any legacy transformation. This allows for a continuous application running despite any changes to the underlying infrastructure. Database, resources, and network services are all part of the infrastructure that needs to be protected here.

Establish Measurable Goals

Use this to gauge how well you've done. Having a risk assessment model for each method of application modernisation or migration is also helpful. Save yourself time and trouble in the future by writing down every step of the route.

Split apart a program into its parts

Versions, deployment, and the application ecosystem all contribute to the whole that is an application. Find out what they are all called and put them in that folder. Because of this document, releasing new software versions is straightforward. It takes little work to release a new version. It helps with the migrating, testing, and planning processes.

Establish Measurable Goals

Use this to gauge how well you've done. Having a risk assessment model for each method of application modernisation or migration is also helpful. Save yourself time and trouble in the future by writing down every step of the route.

The Process of Moving Software

"Software migration" refers to switching between different computer operating systems. The term "IT infrastructure migration" describes moving anything from a single program to an entire central server.

Analysis of Source Code

There is an increased danger of code inconsistencies, errors, bugs, and vulnerabilities as more features and modules are introduced to your software system, such as a new payment platform or an audience analytics tool. It's also possible that your system will get increasingly slow when more modules and plugins are added, and their compatibility layer is activated.

As the nature of work and how people use their computers evolve, so must the software packages that support them. The code will need regular updates and tweaks to improve system efficiency and enable smoother integration with other programs. In addition to enhancing software quality, code review has many other benefits, such as promoting information sharing, aiding in developer skill development, encouraging teamwork, cutting down on expenses, and more. To automate processes like risk assessment, fraud detection, and policy customization, current platforms must account for rising technologies like blockchain and AI.

Increased Capability

Sometimes the best way to enhance your existing software is to add new features or technical abilities. You can achieve this by adjusting the components of your system as they are or by adding new ones. You may extend the functionality of your CMS by adding features like automated social media posting, augmented reality and virtual reality lenses to the eCommerce site, and user interaction tools for any streaming app.

While adding new features may seem simple, it is one of the trickiest ways to modernize an application if done incorrectly. It can, however, give your aging programs a fresh lease on life, provided you go about it correctly. To improve upon pre-existing features, one must comprehensively investigate the underlying source code.

Have a reliable safety record

In the current cyber environment, security is the primary concern; as such, it should be the primary focus while modernizing legacy applications. Over the application's whole lifecycle, it is crucial to employ proper application security standards such as rigorous testing, authentication protocols, API protection, etc.

Adapt a modular perspective

What we meant by using a modular approach was to break an app down into its constituent parts and then work on those parts separately before putting them all together and running the whole thing in a virtualization or simulated environment to see how it would perform in the real world. For a trouble-free journey into app modernization, this is essential. It neutralizes your experience's dependence on the process infrastructure.

You might also be interested in our latest blog: Click here

Methods For **Updating Software**

Phase 1: Meticulous Preparation

The first step is careful preparation, during which businesses think through their ultimate objectives and the best ways to get there. Identifying which app needs updating, determining the best strategy, estimating how much time would be required, and so on is a time-consuming procedure. Determining who will be involved in the process and setting a budget for it are also part of this step.

Phase 2: Tailoring the Plan to Meet Specific Demands

This phase initiates the iterative process of modernization. Iterative approaches are favored because they adjust both strategy and scope painlessly.

Phase 3: Simplify the process

Even after a project has been successfully implemented, work continues. The cloud-first methodology and its status are then monitored. Keep a close eye on the equipment and how it's being used so you can make quick, effective adjustments if necessary.

Benefits Of **App Modernization**

Keeping your apps up to date helps you maintain agility and competitiveness and frees up capital for investing in game-changing innovations like AI, ML, and the cloud. Besides this, it's a great way to increase consumer involvement and attract new followers. **Application modernization can provide several benefits for enterprises, including the following:**

Improved efficiency: Modernizing applications can help enterprises streamline their processes and automate manual tasks, improving efficiency and productivity.

Reduced costs: Moving to the cloud can reduce costs that would have otherwise gone toward setting up and maintaining an in-house data centre. Cloud migration can also lessen the burden of maintaining your organization's legacy systems. You'll only ever pay for the storage space you use in the cloud database. Organizations can save money on the upkeep of aging infrastructure, human involvement, and system downtime by updating their application infrastructure.

Increased agility and scalability: Modernized applications are typically more flexible and scalable, enabling enterprises to adapt quickly to changing business needs and market conditions.

Enhanced customer experience: If you introduce new features and services your clients want, you'll have an advantage over the other businesses. Businesses can enhance their customers' experiences by delivering quicker, more responsive, and more individualized services via modernized software. Modifying a program makes it simpler to introduce new features and services, which in turn boosts both user satisfaction and revenue. More leads will be generated in the long run, and there will be a greater probability of client referrals.

Improved security and compliance: Modernized applications are typically more secure and compliant with current regulatory requirements than legacy systems.

Competitive advantage: The needs of today's clients are often beyond the capabilities of antiquated legacy systems. As a result, they may decide to switch to another software that offers more enticing features. To keep up with market shifts, client demands, and technological developments, businesses might benefit from modernizing their apps. If your competitors are still debating whether or not to implement a specific modernization, jumping on board early will provide you with a distinct advantage. You can also use it to embrace new technologies and platforms, giving you a leg up on the competition. In addition, unlike with most legacy apps, you have access to real-time data analytics compiled worldwide.

Businesses can benefit from application modernization in several ways, including enhanced operations, lower costs, increased competitiveness, and an enhanced ability to satisfy client requests. While developing and launching an application modernization strategy, businesses should consider their unique needs and priorities.

The Modernization Of **Apps Using A Variety Of Tools**

Bringing your apps up to date is challenging. Sometimes it feels like you're going nowhere and all by yourself. But, with the help of appropriate application modernization technologies, this can be avoided.

Thankfully, there are a lot of them. Have a peek at the top tools. Companies can choose from a variety of application modernization tools.

Some of the more frequent examples include:

The Containerization Toolkit

Containers are compact, portable environments that may operate effectively. Containerization tools such as Docker and Kubernetes are used to construct and manage containers. With the help of containerization tools, businesses may create increasingly modular and portable software applications, which is essential for the modernization of the workforce.

Low-code software frameworks

Low-code development platforms like Microsoft Power Applications and OutSystems allow businesses to rapidly develop and release apps without requiring in-house developers with extensive coding knowledge. Businesses can benefit from low-code development platforms since they speed up creating and deploying new apps.

Development and Operations Tools

DevOps tools like Jenkins, GitLab, and GitHub help businesses simplify business software creation and execution procedures. Businesses can benefit from using DevOps tools to upgrade their systems because they increase speed and agility while decreasing the likelihood of mistakes or outages.

Hosting in the cloud

Amazon Web Services (AWS), Google Cloud Platform, and Microsoft Azure are a few cloud platforms that help businesses operate their apps and store their data online. Incorporating cloud-based platforms into an organization's application modernization strategy can boost scalability, reduce operational expenses, and increase customization options.

Resources for managing APIs

Tools for managing application programming interfaces (APIs) are available in business settings; examples include **Apigee** and **MuleSoft**. APIs are used to facilitate communication between disparate software programs and services. API management tools can help enterprises modernize their applications by promoting enhanced integration and conversation between systems.

Microservices

The ability of microservices to decompose extensive monolithic programs into more minor, independent services that can be independently created, deployed, and scaled has made them a crucial technology for application modernization. The benefits of this method include higher flexibility, greater scalability, and enhanced resilience to errors. The ability to divide massive apps into smaller, more manageable chunks is a significant perk of using microservices. As a result, developers can more easily focus on fixing bugs in some regions of an app without affecting other areas. It also facilitates the testing and deploying of modifications to individual components without requiring the deployment of the complete program. Developers may iterate more rapidly and effectively in response to shifting market conditions with microservices since they allow them to work on individual services separately. Examples include Istio, Kubernetes, and Apache Kafka.

Automated and Orchestrated processes

The ability of development, operations, and security teams to sustainably manage the scalability of today's apps increasingly depends on adopting automation as a principle and a technology. Software development orchestration automates container-related operations such as deployment, scaling, and networking. Examples include Ansible, Puppet, Chef, and Jenkins.

The use of these technologies can aid businesses in keeping up with the fast-paced, ever-evolving digital landscape by modernizing their apps and infrastructure. Companies must consider their unique requirements and objectives when choosing and deploying these technologies.

Real-world Instances Of Firms That Upgraded Their Apps And Saw Considerable Benefits

Lufthansa

German carrier Lufthansa updated their smartphone app to enhance the customer service they provide. They utilized cutting-edge tools like AI and machine learning to make trip planning more accessible and provide more customized suggestions for each individual. There was an increase in bookings made via mobile as a result.

Capital One

Capital One, a financial services provider, adopted microservices and containerization to bring their aging systems up to speed with the cloud. This resulted in annual savings, a faster time to market, and more scalability.

vodafone

Vodafone, a telecoms corporation, used microservices and containerization to bring its antiquated billing system into the cloud. As a result, they cut their annual expenses while reducing their time to market for new items.

nab

Using microservices and application programming interfaces (APIs), the National Australia Bank (NAB) migrated its ageing mainframe software to the cloud. This in savings, a shorter time to market for new features, and more agility.

UNDER ARMOUR

The sportswear manufacturer Under Armour upgraded their online storefront to handle more traffic and perform better. Because they migrated to a cloud-based architecture that uses containerization and automation, they cut their time to market for new goods in half while simultaneously increasing their online sales.

Cost savings, enhanced performance and scalability, a better customer experience, and enhanced agility are ways application modernization can benefit businesses of all sizes and sectors.

Zuci: The Strategic App Modernization Partner

App modernization aims to enhance the efficiency, scalability, and potential value of older apps by bringing them up to date or altering them. The Zuci Systems team takes a novel approach to app modernization, making them a leading provider of technological solutions.

There are three pillars to Zuci Systems' strategy for modernizing applications:

The Agile Transition:
We employ an agile technique to bring legacy programs up to date. This means they do their work in short iterations and regularly consult with their clients for input. This method guarantees the client's expectations will be met and the upgraded application delivered on schedule and under budget.

Automation and Updating:
Automation tools are used by us to bring legacy programs up to date. By taking this route, they can cut down on the amount of work and time needed for modernization, lowering costs and boosting the updated application's quality.

Development for the Cloud:
We create cloud-native apps optimized for use with cloud services. Because of their superior scalability, reliability, and cost-effectiveness, cloud-native apps are ideally suited for updating older systems.

We aim to position our clients at the forefront of the digital revolution by facilitating quick cloud adoption and helping them build a business prepared for the years ahead through intelligent automation, specialized talents, and global delivery capabilities. At Zuci, we'll assist you in formulating a strategy that makes the most of your current software while also preparing you to make the most of future technological developments.

ZUCI SYSTEMS
PERFECTION. ALWAYS.

USA & Canada

Texas
+1 (469) 320-1156

Toronto
+1 (343) 309-5455

Europe

Belgium & Geneva
+32 477411912

India

Chennai
+9 (44) 69178900

✉ sales@zucisystems.com

www.zucisystems.com