App Modernization at Scale

App Modernization at Scale Highlights from a Government Roundtable, hosted by ATARC in partnership with Red Hat, March 2023

In a recent roundtable discussion hosted by the Advanced Technology Academic Research Center in partnership with Red Hat, federal experts discussed challenges and opportunities with modernizing applications at scale.

Agencies represented at the roundtable shared various strategies to achieve application modernization at scale, including adaptive maintenance, APIs and event-driven software development solutions. Discussion also focused on how app modernization impacts both the customer and developer experience. Participants also discussed the effort necessary to achieve application modernization at scale in the current environment, and some of the challenges that are preventing this from happening.

Strategies for Successful App Modernization

"Technology is obsolete every minute. The minute we reach even halfway through any kind of modernization approach, we already have a new system, new technologies, new IT adoptions, new variations. Even though we are modernizing, we need to keep in mind what's coming next."

Adaptive Maintenance

Some agencies are working towards an adaptive maintenance model, where agencies architect systems to allow for incremental improvement over time without a need to wholesale replace the underlying software. Instead of waiting five years to retire and replace the system with entirely new software, agencies can modernize components of the system along the way. Today, this is often accomplished by employing microservices, which breaks software down into smaller, more manageable parts. This approach allows enterprises to keep up to date with new requirements or making smaller, granular changes to the system rather than large-scale changes.

APIs for Modernizing Applications Application Programming Interfaces (APIs) are an important component of modernizing systems, particularly with data management and data usability. Without an API strategy, agencies may not be able to share, integrate, aggregate or orchestrate data with other applications or data sets. If applications are built without considering an API-centric approach, they will find it difficult to build fast integrations with new applications.

Agencies want to understand how data will be collected, how it can be used, and whether the data is compatible with machine learning or other initiatives. Agencies should also consider the long-term maintenance of an API and what the impact of the API will be at different levels of implementation.

Event-Driven Software Development

Some agencies are employing event-driven architectures as a solution, using event-streaming platforms such as Kafka. Instead of relying on a direct API connection between applications to pull data, agencies can utilize another service to subscribe to a data stream and extract the desired data. This allows agencies and developers to rethink their software architecture to decouple data streams.

Considerations for App Modernization at Scale

Agencies typically modernize applications to address a security concern or replace end of life systems. But another reason to modernize applications is to improve the customer experience (CX).

Some agencies even use CX as a primary success metric for a given system. Other agencies are focusing efforts on moving from product-based outcomes to performance-based outcomes. This allows agencies to consider the total impact of the application on user experience.

However, CX encompasses more than just the end customer. It also includes other stakeholders who may be involved in the system one way or another. The collective stakeholder experience, including those in the field, may not be captured in a single success metric. That's why it's important for agencies to consult all other stakeholders, especially those out in the field, about their experience and concerns with the system instead of using a metric to judge application effectiveness.

> "If people feel like they're being left behind, that means you didn't include them in a solution... You give people opportunities to be contributors if you give them opportunities to build skills. If they understand where they've been in the organization, you're going to do a lot better when you go through any sort of organizational change."

Developer Experience

One of the lesser discussed challenges with the modernization process is the developer experience. Many IT and DevSecOps teams inherit systems that have been built over a 20 or 25 year period by many different people and teams. The systems may have been transferred between units, built with different pipelines, programming languages, and branching strategies. Essentially, each system is unique.

Yet, there is an assumption and explicit expectation among agencies that IT contractors can easily move resources from one project to another. In reality, when an application developer moves from one team to another, even within the same organization, they are sometimes walking into a completely different world. They may encounter a completely different set software development pipeline with different tools.

Agencies and contracting partners should consider using consistent development platforms, or software factories, so that they can improve team agility and allow for resources to move around more easily. This will enable agencies to onboard, train, and retain software developers.

To ensure consistency, agencies must require better documentation from contractors who work on applications. Often, contractors will do the work and leave. Unless agency employees are shadowing the contractors or proper documentation is left, agencies are left unaware of how something was implemented, or whether there are dependencies.

> "The modernization strategy has to include the wellbeing of the development teams - not just the existing teams, but the ones that will eventually replace them through contract renewals and changes over time."

Challenges with Funding

Every single person in technology is working on application modernization. The challenges on the government side are two fold: although agencies build systems, they are funded for steady-state operations. Agencies may receive additional funding to build a system in order to put out a new set of capabilities, but then the focus shifts back to maintenance.

It's rare for agencies to have the resources to build out applications that fit into an IT lifecycle. The challenges arise with funding sources, and how to manage the operational functions with modernization. Agencies find themselves with severely outdated systems, which makes modernization much harder.

Because agencies are typically funded for static processes, modernization should be planned and considered from a funding approach at all levels: application, implementation and contractual. There are many unknowns when agencies start to modernize. When a project is partially funded, agencies will be forced to create band-aid solutions, which can create greater issues.

From an implementation perspective, agencies need to ensure they have the right resources and a transition plan for each system. Many people are retiring, which is causing issues when a person leading a modernization initiative is expected to leave the agency in short order. One participant suggests that employees on the verge of retirement may not have the same drive for modernization as a newcomer might.

Feasibility of Scaling App Modernization

Ultimately, agencies need to give developers the tools they need to be more efficient. Software factories are a tool that removes the challenge of platform and infrastructure development by providing agencies common tooling and reference architectures. With better tooling and standards for deploying code to production, application modernization is more achievable for many agencies.

Having enterprise architecture as a leading indicator for application modernization builds consistency and prevents teams from developing bespoke pipelines or branching strategies. An effective modernization strategy provides guiding principles that enable developers to build a consistent foundation, so modernization is more achievable and scalable.