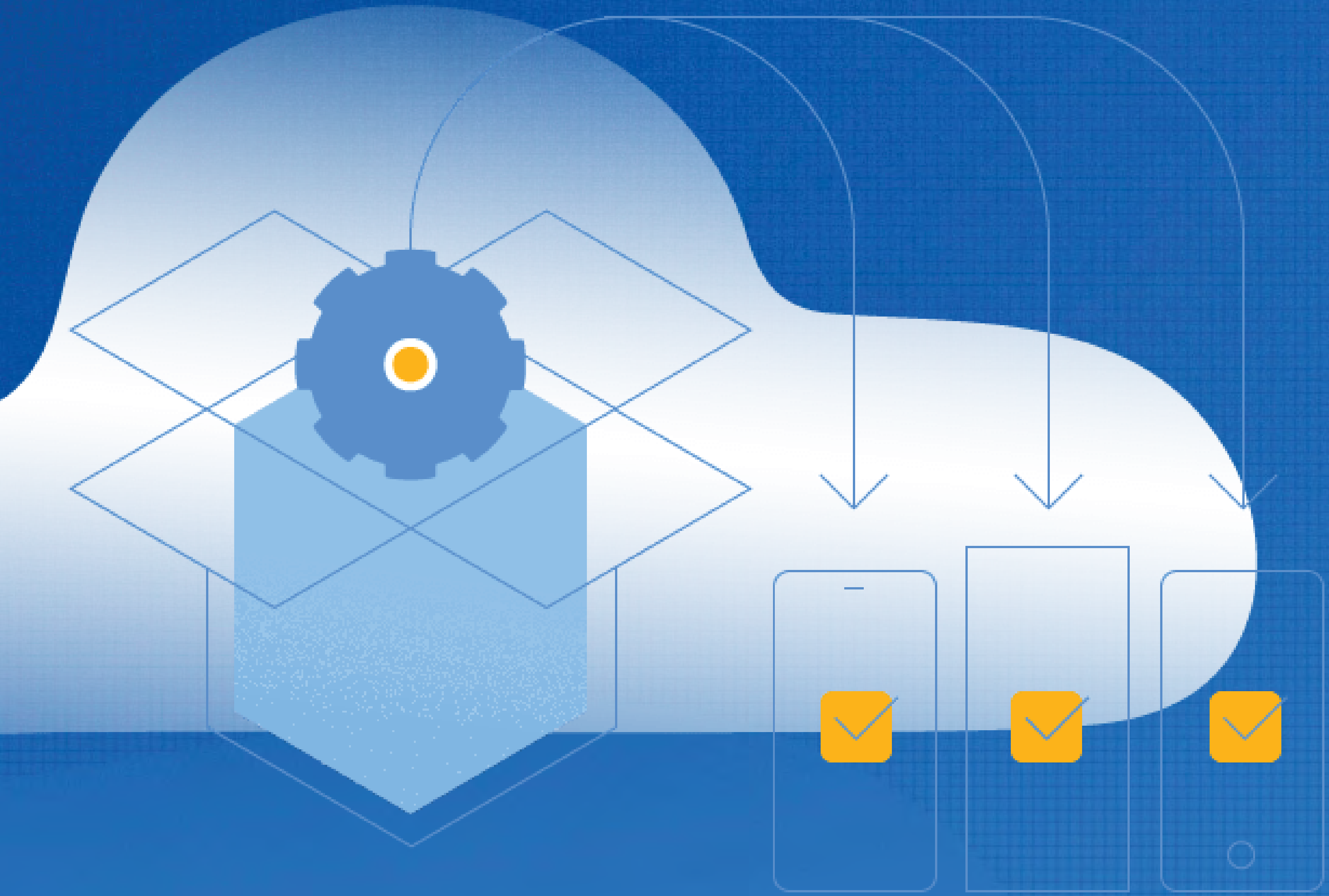
Progress®



The Cost of an Enterprise Mobility Strategy

And Why Every Enterprise Needs to Invest in Mobile

WHITEPAPER

© 2019 Progress. All Rights Reserved.

Introduction

You'd be surprised how many companies don't have a mobility strategy this late in the game. And yes, it is late in the game as enterprises were defining their mobile strategies and building mobile apps long before Apple released the iPhone in 2007. There are a lot of questions you'll need to answer as you set your enterprise mobile strategy; the two most important being: "What's the business goal for this strategy?" and "What's this strategy going to cost?"

The first is the most critical; if you don't know what you plan to accomplish with your mobile strategy, what's the point of even having one? Most organizations have a business goal tightly tied to some aspect of revenue; whether it's directly increasing revenue in the form of more sales opportunities, faster time to close, etc., or indirectly increasing revenue as happier and more engaged customers buy more products and services.

The second question drives what you'll be able to accomplish with your strategy; measuring the potential business impact helps management determine how much money to spend. As you go to management with your strategy, it helps to understand how big of a breadbox you're asking for - what investment must you make to deliver on your business goals related to mobile and mobile customers.

This eBook assumes you have the business case for your enterprise mobility strategy all figured out, that you already know what mobile devices you'll allow into your corporate environment and how you'll secure and manage them. Here we'll focus on the hardware and software you'll need to implement your strategy plus the human costs required to deliver mobile apps into your environment. In an earlier eBook, we laid out possible approaches for building mobile apps; in this publication, we'll cover the financial aspect of the choices you make.

© 2019 Progress. All Rights Reserved.

The App Store Economy Spoiled Your Users

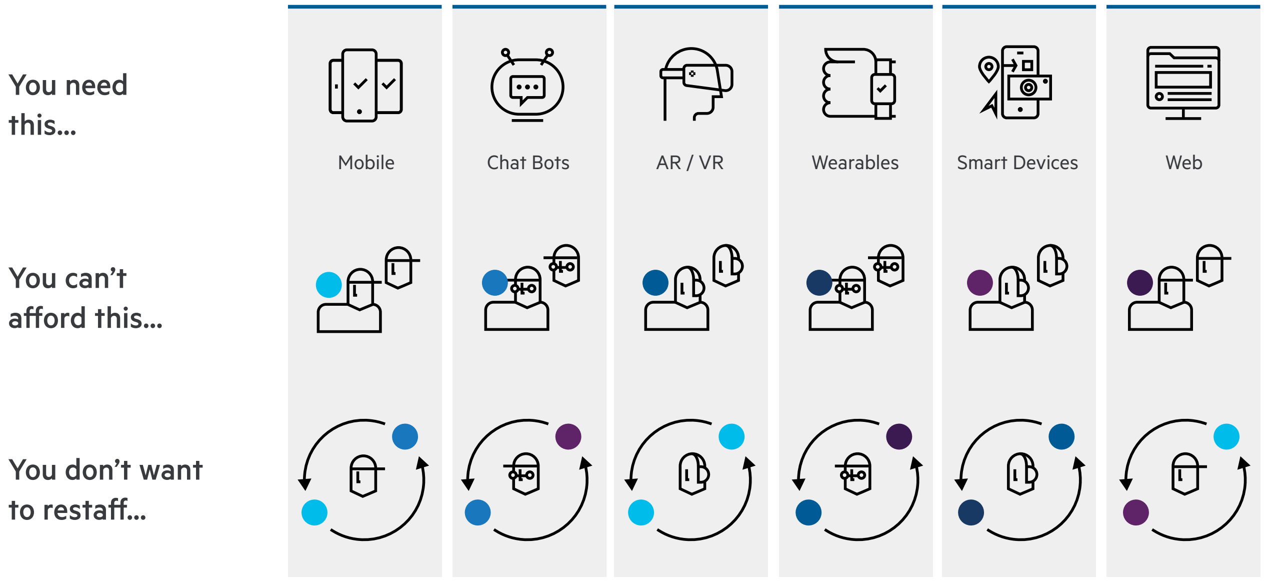
One of the biggest problems you'll face with your mobile strategy is that your user base is used to interacting with well-made, engaging apps they pick from an app store. When they encounter an app that isn't up to snuff, they abandon it and go looking for something better. Consumer app developers spoiled your user base with shiny, fancy apps that get regular updates with cool new features. This sets the enterprise mobile app bar higher than you probably want.

We once encountered a company servicing the construction industry and what we heard from users was "if it isn't beautiful, nobody will use it." Beauty was apparently more important than function for that user base.

How do you make mobile apps for this crowd?

Your user base is a captive audience, they have[1] to use the apps you provide them, they don't have any other choice. Unfortunately, delivering subpar apps won't work, even though employees must use your apps. If they're not happy, they're going to let a lot of people know. Also, if employees can't use the app effectively, they will find other, and usually more expensive ways to get what they need.

Users Demand Consumer-Grade Apps



© 2019 Progress. All Rights Reserved.

To ensure success of your app, be prepared to deliver frequent updates. Don't kill yourself trying to deliver the monthly or weekly updates of some popular consumer apps, but do plan to deliver app updates to keep the app fresh, on at least a quarterly basis.

Publish your roadmap, perhaps displaying it at the bottom of a "What's New?" page you show the first time the user launches the app after an update.

Provide an easy way for users to give you feedback on your apps. You know they're opinionated, so put that feedback to good use and get as much information as you can about their experience with the app. This also means implementing analytics so you can see how and when the app is used as well as what isn't being used or isn't working. Don't forget about the venerable Shake to Share feedback option[2] .

Finally, keep your eye out for new ways to help your users. Use AI and bots to deliver a better experience. Implementing AI capabilities in your infrastructure and mobile apps enables you to deliver data the user needs before the user needs it, dramatically reducing the time to get answers and increasing productivity simultaneously. Use bots instead of humans to answer frequently asked or common questions. Automate responses to "How do I?" questions as well as "What's the status of?" questions using bots.

[1] Well, they don't really ***have*** to, they could quit if they wanted to avoid using your app.

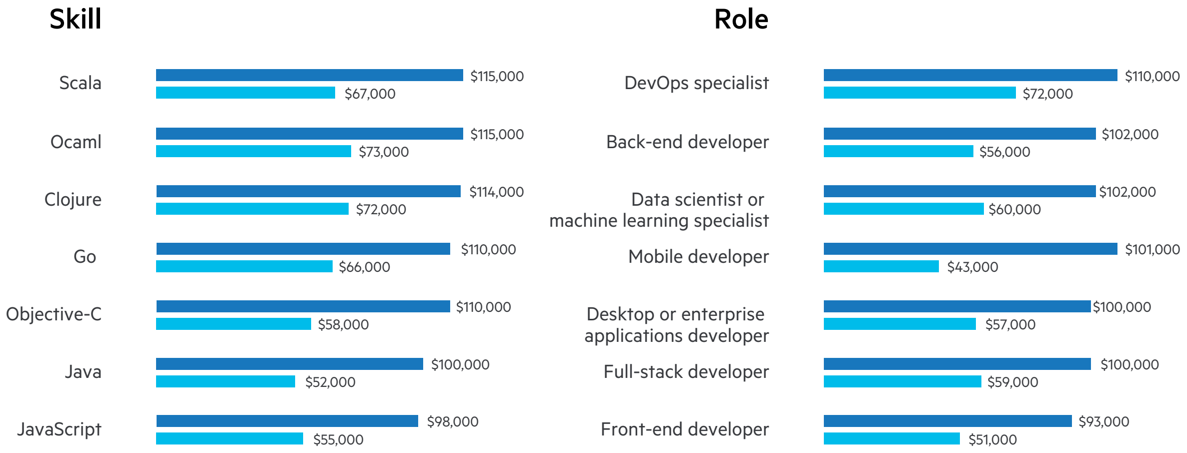
[2] When the user shakes the phone while the app is in the foreground, take a screenshot of the app at that moment, then let the user include that, along with their comments, in an email to your suggestions@ email account.

© 2019 Progress. All Rights Reserved.

People Are the Most Expensive Resource

We must be up front about the most important topic in this publication: the most expensive resources in your enterprise mobile strategy are your human resources. The systems and tooling you need to build, deploy and manage your mobile apps plus manage your employee's mobile devices incur some cost, but that's nothing close to what you'll spend on the people making apps.

Not only are these human resources—developers, designers, quality assurance engineers, project managers, and more—going to consume the majority of your budget, they're likely also going to consume the most acquisition time. Although there are lot of skilled resources in the market, many organizations are having a hard time finding available or appropriately skilled resources for their projects. There are so many companies out there competing for skilled resources that you may have to settle for less skilled resources in order to fill the positions.



"What Languages Are Associated with the Highest Salaries Worldwide?" https://insights.stackoverflow.com/survey/2018

Another option for you is to implement solutions that enable your organization to deliver mobile apps more efficiently. Let's talk a bit about the costs of mobile development first, then we'll wrap up offering alternative approaches that will save you time and money.

© 2019 Progress. All Rights Reserved.

Mobile Apps Are Different Than Desktop Apps

If you think about the traditional desktop app, whether browser-based or native, these apps are usually feature-complete before they're released to the target audience. For this reason, desktop app developers require a complete understanding of the features the app will eventually have. You may deliver a substantial subset of the app features at the start, and add less used features over time, but there's no such thing as minimal viable product (MVP) when it comes to enterprise desktop apps. These are the apps you're running your business on, so that means long development cycles, app stability and infrequent releases.

Mobile apps are different:

- Mobile developers typically start with a minimal viable product (MVP) approach, getting one, or a few features into the hands of mobile users to whet their appetites before building more. If a feature a user needs infrequently isn't in the mobile app, they'll just switch to the desktop to get that part of their job done.

- Mobile app users are more demanding, expecting that the app be beautiful and use as many of the phone's cool features as appropriate.

- Mobile app users expect more frequent updates to their apps, feeling that the app is useless if it's not updated frequently.

- Performance is much more important for mobile apps; developers building mobile apps must consider a smaller processor and memory footprint plus frequently unreliable wireless networks.

Mobile Brings Support Issues

Organizations generally will support one, or perhaps two, desktop operating systems and have a stable catalog of supported devices. You're probably running one, possibly two, desktop OS versions in your organization.

With mobile, chances are you're supporting two operating systems. iOS delivers a consistent experience across devices and forces users to stay up on OS releases, so your audience is relatively stable. There are, however, many iOS device models in the market and the number continues to grow. The Android operating system

© 2019 Progress. All Rights Reserved.

on the other hand is the wild west. With hundreds or thousands of device manufacturers in the market and each one adding special features to differentiate themselves, your users could be running on any device and using an operating system version that hasn't been updated in years. This makes testing and supporting mobile apps a nightmare.

Mobile Offers Many Approaches

Mobile application development platform (MADP) vendors offer tools that simplify desktop app development, enabling developers to easily deliver desktop and mobile apps at the same time across multiple target platforms (mobile OS). These tools dramatically reduce time to market and the cost of building apps for your business.

For desktop applications, organizations typically deliver as many apps as they can via the browser, delivering a single app that runs anywhere. For more sophisticated apps, or for apps running on proprietary hardware, they'll build native apps for the desktop OS, which is becoming less common.

Mobile development is different across different mobile OS, so you need a different set of tools and skills for each target platform. The same is true for desktop apps, but enterprises usually pick a single target platform for their native apps and use browser apps for everything else. With mobile, that's not possible.

For mobile apps, you have the following options:

- Web applications: Leverages the same development skills you use for your desktop web apps, resulting in some economies of scale. Here you'll build and deploy an app for each target platform. Projected cost: Low.

- Native applications: Built using the native tools and languages for the target mobile device. The most expensive option as each target platform requires a different set of skills and you must accommodate users carrying older devices. Projected cost: High.

- JavaScript-driven Native applications: Coded using JavaScript or its variants running in a native app delivering a native user interface (UI). NativeScript is a popular choice in this realm. Projected cost: Medium

- Adjacent Native applications: Code your app using a popular language, not one of the native languages for your device's OS, then compile down into a native app that runs on each target platform. Xamarin is a popular choice in this realm. Projected cost: Medium.

© 2019 Progress. All Rights Reserved.

- Mobile Application Development Platforms (MADP): A suite of developer tools and backend services that enable developers and non-developers to build cross-platform mobile apps using a point and click, drag and drop interface. Developers build a single app that can run on desktop and mobile devices, automatically accommodating differences between the platforms. Progress Kinvey is a popular choice in this realm. Projected cost: Low.

These options are described in much more technical detail in The New Mobile Development Landscape.

Approach	Development Language	Complexity	Cross-platform	Code Reusability	Cost	Quality / User Experience
Web	HTML, CSS, & JavaScript	Low	Yes	All	Low	Low
Native	Java, Kotlin, Objective-C, Swift	High	No	None	High	High
JavaScript-Driven Native	JavaScript, TypeScript	Medium	Yes	Some to Most	Medium	High
Adjacent Native	C#	High	Yes	Most	Medium	High
MADP	Proprietary	Low	Yes	All	Low	Depends

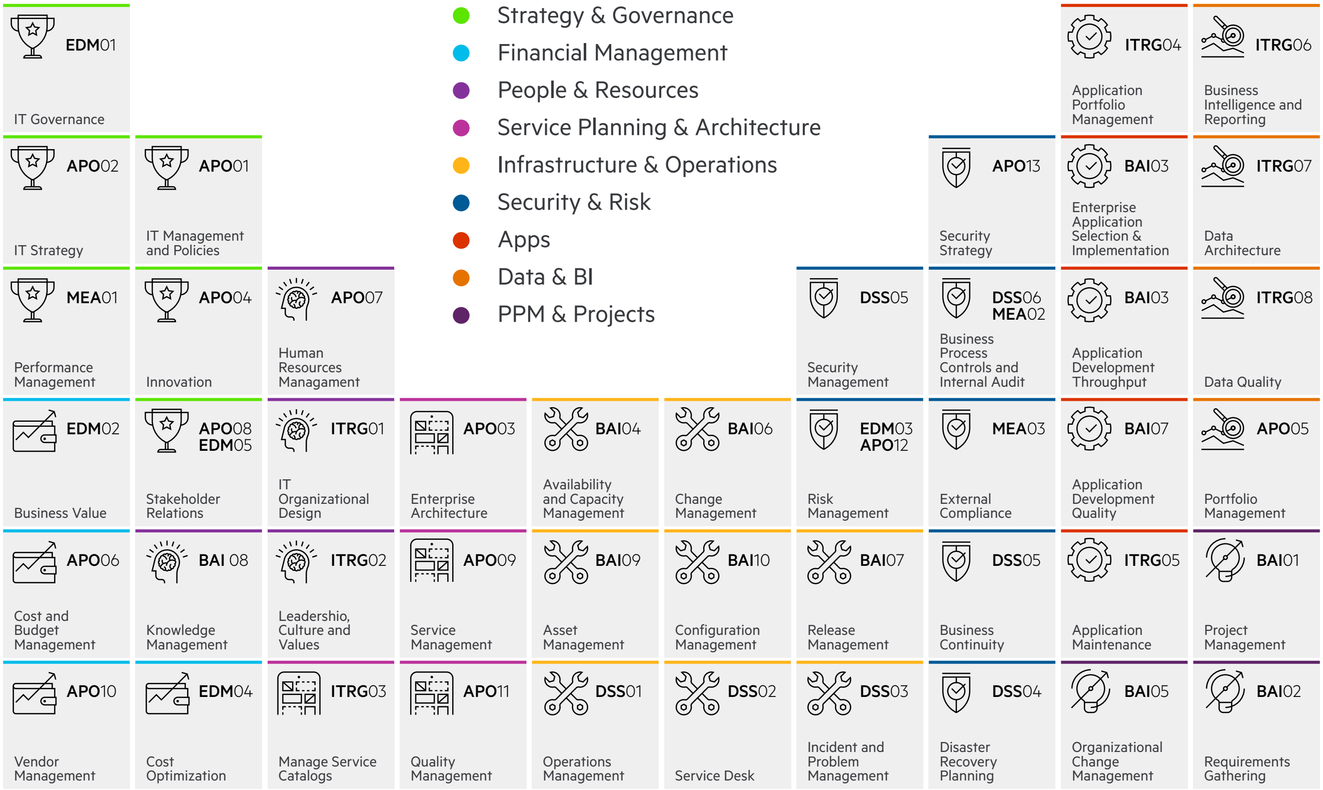
© 2019 Progress. All Rights Reserved.

The Cost of Mobile Development

Developing mobile apps for your business can be expensive, but it doesn't necessarily have to be. The choices you make in crafting your enterprise mobility strategy affect both your short-term and long-term costs. Choosing the right strategy can save you money; even making initial choices that require an up-front investment can save you money over time. The right strategy involves smart choices, the right amount of governance, and using fewer people to manage your mobile environment and deliver capabilities to your mobile users.

Picking a Governance Approach

Many organizations, and teams within organizations, take a willy-nilly approach to building mobile apps, assembling a team and sets about building the app. While this enables teams to build and deploy the apps they need, it reduces the organization's ability to achieve the cost and quality efficiencies it gains through application of a little governance.



- ● Strategy & Governance
- ● Financial Management
- ● People & Resources
- ● Service Planning & Architecture
- ● Infrastructure & Operations
- ● Security & Risk
- ● Apps
- ● Data & BI
- ● PPM & Projects

EDM01 IT Governance								ITRG04 Application Portfolio Management	ITRG06 Business Intelligence and Reporting
APO02 IT Strategy	APO01 IT Management and Policies						APO13 Security Strategy	BAI03 Enterprise Application Selection & Implementation	ITRG07 Data Architecture
MEA01 Performance Management	APO04 Innovation	APO07 Human Resources Managament				DSS05 Security Management	DSS06 MEA02 Business Process Controls and Internal Audit	BAI03 Application Development Throughput	ITRG08 Data Quality
EDM02 Business Value	APO08 EDM05 Stakeholder Relations	ITRG01 IT Organizational Design	APO03 Enterprise Architecture	BAI04 Availability and Capacity Management	BAI06 Change Management	EDM03 APO12 Risk Management	MEA03 External Compliance	BAI07 Application Development Quality	APO05 Portfolio Management
APO06 Cost and Budget Management	BAI 08 Knowledge Management	ITRG02 Leadershio, Culture and Values	APO09 Service Management	BAI09 Asset Management	BAI10 Configuration Management	BAI07 Release Management	DSS05 Business Continuity	ITRG05 Application Maintenance	BAI01 Project Management
APO10 Vendor Management	EDM04 Cost Optimization	ITRG03 Manage Service Catalogs	APO11 Quality Management	DSS01 Operations Management	DSS02 Service Desk	DSS03 Incident and Problem Management	DSS04 Disaster Recovery Planning	BAI05 Organizational Change Management	BAI02 Requirements Gathering

© 2019 Progress. All Rights Reserved.

Other organizations mandate a corporate structure to mobile app development with IT controlling the process and providing all the tools and services. This delivers cost efficiencies, enabling IT to negotiate better pricing for human resources like coders and designers, as well as tooling. Ultimately, this creates a bottleneck in IT as different departments lobby for resources; unless your company is all-in on your mobile strategy, there's no way IT is ever going to have enough resources to keep up with customer demand. This means you must prioritize projects, and some important projects will never see the light of day.

What happens next? Those divisions and departments that can't get resources from IT will simply hire their own developers, pick their own tools, and get to work building mobile apps in the shadows. In this scenario, IT has done nothing except turn business units against them.

Parts of your organization simply want access to the tools they need to effectively run their part of the business. If they need a mobile app, then they should have access to the resources they need to build it. That means a hybrid approach, where business units create project teams and deliver mobile apps with IT's help and governance.

In this model, IT leverages the company's overall buying power to get the best price for both resources and tools for continuous integration, security, code scanning, test automation and more, and shares them across the company. IT guides mobile development teams through the process of building their mobile apps in a way that most effectively:

- Protects the company's brand.
- Protects the company's data from misuse or loss, especially when an employee leaves the company.
- Ensures that apps comply with industry and governmental regulations.
- Ensures that apps comply with internal company security guidelines.
- Ensures that employees, partners and customers realize a consistent experience with all company apps.

This lightweight governance approach forms a more collaborative effort between IT and business units. Business units allocate budget for developers, designers and quality assurance, then build all the apps they need/want within the structure provided by IT. With this approach, business units are enabled to deliver the mobile apps they need to achieve their business goals.

© 2019 Progress. All Rights Reserved.

Delivering Shared Tooling

The greatest value from IT comes from shared tooling acquired by IT and leveraged by the whole company for greatest budget impact. They assess the market and internal business needs, then select the tools teams will use to more effectively deliver mobile apps. Some examples of tools in this category are listed in the following table.

Product	Description
Development Platforms	Mobile Application Development platforms (MADP) deliver a simpler and more reliable way to deliver apps, both desktop and mobile for a variety of target platforms. They can dramatically reduce the cost of application development; we share some examples later in this document.
Test Automation	You can hire quality assurance (QA) engineers to test your applications manually, but that's a lot of expensive fingers poking at glass. Instead, invest in a test automation suite and train your developers to deliver functional and unit tests with every check-in. With that in place, use your QA team to look for edge cases and perform exploratory testing of your app.
Mobile Device Lab	Developers will often use the devices they have lying around to test new features for their apps. Likely, your app's target audience probably isn't well represented in that catalog of devices. Invest in an on-premises or cloud-based device lab, delivering a modern and robust catalog of devices ready to be used for functional testing.
Continuous Integration (CI)	Developers build modern apps as a collaborative effort, contributing code to a single repository. Many teams schedule periodic integration tests, validating that recent code changes haven't broken anything. Instead, invest in a continuous integration tool that builds and tests your application with every code check-in. This gives you peace of mind knowing exactly when the app broke, who broke it, and how to more quickly identify and fix faulty code.
Continuous Deployment (CD)	Getting mobile apps into the hands of testers, beta users, or your app's target audience shouldn't be a manual process. Invest in continuous deployment tools to streamline app deployments to the company or the world.
Code Scanning Solutions	Developers pull code from a variety of locations: open source and commercial libraries, web sites, colleagues, etc., and there's often very little known about the quality of the code. Scanners identify licenses, look for malware and scan for faults and errors that expose security holes in your apps.

Delivering these tools as shared resources for your development teams to use helps deliver a more efficient development environment for your whole company plus drives down overall cost through consolidated purchases and volume pricing.

© 2019 Progress. All Rights Reserved.

Delivering a Consistent Experience

You can let development teams build the apps they want with limited style guidance and hope they deliver usable apps requiring limited training for effective use. Alternatively, you can build a style guide outlining common UI elements and styling, then encourage all developers to use it.

Delivering a consistent look and feel across multiple apps reduces the overhead required to learn each new app. Implementing company styles in an app might increase the initial cost to build the app, but will reduce the overall support costs for the organization. This is a worthwhile cost tradeoff that pays for itself with each subsequent app added to the portfolio.

When your company is acquired or rebrands itself, simply update the central UI library used by your apps and everyone updates its UI automatically with the next release.

Implementing a Security Culture

You can assume developers across your company know how to build secure apps, but we all know what happens when we assume anything - great disappointment sprinkled with surprise when our expectations are unexpectedly not met. You must implement policies and procedures that ensure a secure profile for any app released into employee, partner, or customer hands.

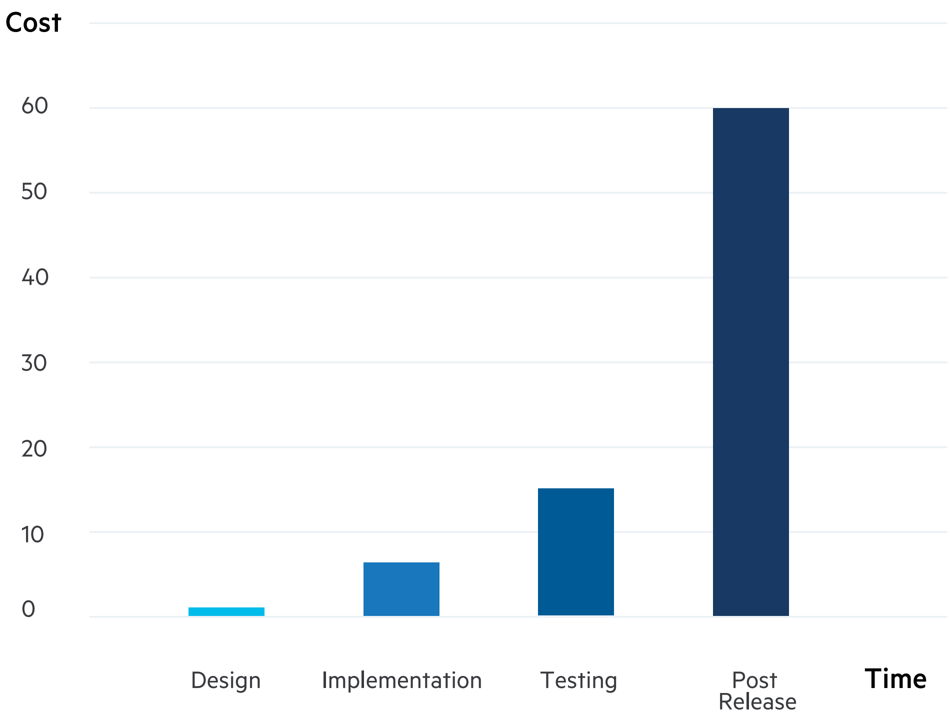
Research by IBM indicates that it's 60 times more expensive to fix security issues in production than at design. Hopefully it's clear that you'll want, at all costs, to implement a security-focused culture and implement policies and procedures specifically designed to reduce security risks in the apps you deliver. To reduce your overall costs for delivering secure apps:

- Encourage your development teams to work closely with the global security team.
- Invest in regular security training for your entire development organization. No exceptions.
- Foster a growth mindset in developers so they won't be afraid to fail.
- Trust but verify - implement security scanners that scan source code for vulnerabilities.
- Mitigate security risk by transferring secure code responsibility to a third-party.

The last one might be a bit of a surprise, but when you build mobile applications using a third-party MADP, you get several security features right out of the box.

© 2019 Progress. All Rights Reserved.

Fixing security bugs at design time costs 1/60th what it costs to fix the same bug with a patch after release



Most vendors automatically encrypt data at rest, and during transmission, so that's something you won't have to worry about. With a platform, you're essentially passing responsibility for your app security footprint to the platform vendor. Since these platforms deliver application source code generated through the visual app builder, there's limited opportunity to deliver malware or security faults into your apps.

Consolidating your mobile app development to a single platform vendor rather than to multiple development teams distributed throughout your organization automatically creates a more secure app environment - provided your platform vendor is up to snuff. Make sure your platform vendor scanned all their code and validated the security of their backend connectors and other features, you'll reduce your security risk when you select the right platform.

Fixed Costs in Your Enterprise Mobility Strategy

As you've seen already in this document, there are a lot of moving parts that affect the cost of your enterprise mobile strategy. These are the things you should implement no matter what approach you take, and some things you'll pay more or less for depending on some choices you make for your environment. The fixed cost items listed in the following sections are software solutions designed to make your environment more secure for you and easier to work with for your users.

© 2019 Progress. All Rights Reserved.

Enterprise Mobility Management (EMM)

Any time you let a computing device connect to your corporate network or access a corporate application, you expose company assets to risk. There are many strategies for protecting your systems, most of which are beyond the scope of this publication.

On the mobile side, you should invest in a software solution that enables your IT department to manage the mobile devices connected to your corporate systems. In general, these enterprise mobility management solutions:

- Manage deployment of security certificates to devices.
- Block access for rooted devices.
- Enforce consistent policies like password required, minimum password length, password history, etc. on the device.
- Restrict actions performed against corporate data, like blocking copy and paste, for example.
- Remote wipe of corporate data when an employee leaves the company or loses an authorized device.

Your mobile users will also benefit from implementation of a system for managing deployment of mobile applications and certificates to mobile devices connected to your systems. Mobile device vendors offer solutions for this, but they only address deploying apps to their own platforms. Most EMM vendors offer a mobile application management (MAM) solution you can use to manage app deployments across multiple mobile device platforms.

Enterprise Directory and Single Sign-On

As you add mobile apps to your enterprise app catalog, your users will soon find themselves logging into multiple applications, multiple times throughout the day. With small keyboards and the lack of fingerprint scanners on many device models, logging into an application multiple times a day quickly becomes frustrating for users. Consider investing in a Single Sign-On (SSO) solution for your environment that enables mobile users to login once and share their credentials across multiple apps.

© 2019 Progress. All Rights Reserved.

Variable Costs in Your Enterprise Mobility Strategy

This is where this gets interesting; where most of your costs come into play. If you think about the enterprise apps you build, they typically consist of a couple of system components:

- The server infrastructure or backend to run your app. This typically consists of a database or databases coupled with server-side logic interacting with the database.

- A client application to access the application and application data; in general, that's a web or desktop application.

Assuming your employees and partners will want to use the app while they're on the road, you'll probably have to add a mobile-friendly version of the web app as well.

Your employees and partners likely won't be happy with a web browser experience on a mobile device so you'll probably need:

- The server infrastructure to run your app. This typically consists of a database or databases coupled with server-side logic interacting with the database.
- A desktop client application to access the application and application data.
- An Android client application to access the application and application data.
- An iOS client application to access the application and application data.

That adds up to a lot more work. You could assume the app is only used while the employee or partner is away from the office, but you know that's not going to work. What would they use when they're at their desk? The browser version of the app; no need to pull out your mobile phone or tablet when you're already checking and responding to email on your laptop or desktop computers.

Unfortunately, the app backend you built will likely not accommodate both desktop and mobile users. The mobile app's mode of operation is different than the desktop app's. On the desktop, you have almost unlimited processor and network bandwidth, so it's easy to retrieve and manipulate large chunks of data at one time. Mobile apps can't afford to do that, you must only get and display the data you need when you need it, going back to the server to get more when the app user's context changes. That means you're going to have to adjust your backend to accommodate both desktop and mobile users.

Let's look at the resources you're going to need to build this app.

© 2019 Progress. All Rights Reserved.

Role	Description
Project Managers (PM)	One or more project managers to keep these projects on track. For smaller projects, you can likely use one PM for all aspects of the project, but for medium or large projects, you're going to need one PM for each system component—backend and each target platform.
UI Designers	You probably already have a design standard for your desktop apps, so you'll be able to leverage a lot of the existing work in place from apps currently in use. For mobile apps, you'll need an experienced designer to design the user interface for both the Android and iOS versions of the app. It's better to use the same designer across both platforms to help deliver a consistent experience on both.
Developers	Development for each system – server, desktop, mobile – requires a completely different set of skills, so you're going to need more developers than you have to deliver on your enterprise mobile strategy. The good news is that many mobile apps display some web content, so you'll be able to leverage some of your existing web developer's skills in your mobile apps.
Quality Assurance (QA)	Somebody has to test your app; developers and power users can, but developers can always get their code to work and power users probably know too much about how the app should work. The QA department provides agnostic testers to put your app through its paces. If you have a test automation system in place for your mobile apps, you'll employ your QA resources to do exploratory testing, looking for edge cases where your app breaks down.
DevOps	Modern apps are built in teams, with multiple developers regularly checking code into different parts of the project. To protect the project from last minute integration issues, you'll implement a Continuous Integration (CI) pipeline to automate code/build validation at every code check-in. The DevOps team handles configuring and managing the CI pipeline.

When you estimate project costs for a web app plus native mobile apps for Android and iOS based on project sizes of small, medium, and large, the numbers are quite staggering, as shown in the following table.

© 2019 Progress. All Rights Reserved.

Project Size	Duration (months)	Project Managers	Designers	Web Developers	Android Developers	iOS Developers	Back-end Developers	Quality Assurance (QA)	DevOps	Total FTE Months	Cost ($100US /hour)
Small	3.0	1.0	1.0	4.0	4.0	4.0	3.0	2.0	0.5	58.5	$936,000
Medium	6.0	3.0	1.5	7.0	7.0	7.0	4.0	3.0	1.0	201	$3,216,000
Large	12.0	4.0	2.0	10.0	10.0	10.0	5.0	5.0	1.5	570.0	$9,120,000

Unfortunately, the table doesn't tell the complete story. Mobile apps are never done:

- As mobile device platform vendors and manufacturers release new devices and new device OS, you must test your app on popular models and fix issues.
- Users always have "just one more feature" they'll need to make your app perfect. As soon as you deliver that feature, a new one pops up.
- Business requirements change and you'll have to adjust your app to meet new requirements.

So, what does that mean for your budget? Assuming you allocate 15% of the app's initial cost for enhancements and maintenance for years 2 through 5, the numbers are staggering as shown in the following table.

Project Size	Initial Investment	Yearly Maintenance (15% per year)	5-Year Cost
Small	$936,000	$140,400	$1,497,600
Medium	$3,216,000	$482,400	$5,145,600
Large	$9,120,000	$1,368,000	$14,592,000

Testing Your Mobile Apps

Another budget item that many teams ignore is devices for on-device functional testing. For desktop apps, you likely already have a testing environment that covers your desktop or browser platforms, either using dedicated systems or virtual machines (VM). What are you doing for mobile app testing?

Device OS providers, Apple and Google for the most part, provide simulators (Apple) and emulators (Google) you can use to test your apps in a virtual environment. Unfortunately, Apple's simulator is not an exact replica of an iOS device—your app must be compiled into a different format to run on the simulator. Additionally, Google's emulators don't provide any coverage for OS enhancements added by device manufacturers. And finally, the simulators and emulators are slower than physical devices, so any performance tests you do in these environments will not be accurate.

This means you'll have to test on physical devices, but which ones do you get? A quick perusal of the iPhone Wiki indicates that Apple alone released 14 iPad, 4 iPad

© 2019 Progress. All Rights Reserved.

Mini and 21 iPhone devices over the years. To test on a representative sampling of those, how many devices do you need to get? For Android, there are 9 major OS versions (28 total OS versions) running on thousands of device models from multiple manufacturers. Additionally, these devices run a variety of OS versions, so how many device/OS variants do you have to have available for your testing?

Not all of them, fortunately. You can easily drop older devices off your test device catalog. You still must have a large catalog of devices to test your application on. Use analytics from your company web site or device popularity websites to determine which devices are the most popular and start with those.

Should you buy multiple devices and distribute them across your development teams? No. What you need is economy of scale, buying the devices you need, then deploying them through a testing solution that enables them to be shared across the teams in your organization. By sharing devices across multiple teams, you dramatically reduce the company's hardware investment and more likely guarantee that devices are heavily used.

© 2019 Progress. All Rights Reserved.

Getting the Biggest Bang for Your Buck

So far, we've shown that there's a lot of moving parts in your Enterprise Mobile Strategy. At this point, you should have an idea of what things will cost you, or at least where the biggest sinkholes are. The question you're probably asking yourself now is, "How do I cut some of that cost out of my company's implementation of this strategy?" We have a couple of answers for you, neither of which are:

- Don't create mobile apps.
- Deliver subpar apps to your employees, partners and customers.
- For mobile apps, most of your costs are in the resources that produce your apps. To cut costs there, you have two leading options:
  - Increase the value of the work your developers do.
  - Make your developers more efficient in delivering apps.

You may think the two are the same, but they're not.

Increasing the Value of a Developer's Work

Increasing the value of the work your developers do means using the developer's work in more ways than would be possible with other approaches. In more simplified, but technical terms, that means code reusability.

Native apps are great, and that's what most users are used to installing on their phones. Of the other choices available to you, several of them allow you to use the same code base for multiple target platforms.

- Web applications, by their very nature, are cross-platform. When the right choices are made around UI libraries, the code you deliver for your web applications will run quite well on any mobile browser, as long as you stay within the limitations of the web browser.

- JavaScript-driven Native applications (NativeScript, React Native and the like) are mostly cross-platform. Your app's business logic is easily shared between apps for the different platforms, only a portion of your app's UI code is specific to the target platform.

Either of those options enable you to use a developer's work on more than one platform simultaneously, increasing the value of that developer's work.

© 2019 Progress. All Rights Reserved.

As an example of the JavaScript-driven Native option, an application built using NativeScript and Angular code sharing, a feature of the Angular CLI built by the Angular and NativeScript teams that you can read more about here, allows you to reuse 70% of your Angular web code for your NativeScript applications. The app's business logic stays the same between the Angular web app and native mobile apps for Android and iOS, (the NativeScript app). Developers only have to update the UI to deliver the same app on mobile devices.

Making Developers More Efficient

Writing code is a fun, but time-consuming process. Anything you can do to reduce the amount of code a developer must write reduces the overall cost of creating mobile apps. One strategy is to use higher level frameworks and platforms offering out of the box functionality. Generally, mobile applications will need push notification capability, the ability to respond appropriately to network instability, offline data handling and authentication. More specifically, enterprise mobile applications often must access a wide variety of enterprise and legacy systems, centrally manage security and compliance, and interoperate with enterprise authentication providers like Active Directory and CA Siteminder.

Modern platforms have evolved to meet the need. Some approaches deliver high developer productivity but offer little control over the user experience, application functionality, performance, or integrations. Other approaches offer high control over all aspects of application development, allowing for tailored applications, but require a high degree of manual development work. Platforms like Kinvey are emerging that offer high productivity and high control, offering no-code or low-code enterprise functionality development accelerators while providing full control over the application.

Speaking with Kinvey customers, the results are dramatic.

For a representative group of customers, pre-Kinvey app release cycles ranged from 6 to 18 months. After implementing Kinvey, each organization significantly reduced the length of their release cycles to about six weeks. The longest cycle post Kinvey was 13 weeks and the shortest was two weeks. For the customer reducing release cycles to two weeks, they had to switch to longer release cycles for the simple reason that end user feedback indicated that two-week releases were too frequent.

Using Kinvey cuts down development and data integration tasks by 75%. One customer spent 60% of their time worrying about their app infrastructure and only 40% addressing user needs. With Kinvey in place, they now focus 10% of their

© 2019 Progress. All Rights Reserved.

efforts on the backend, and 90% of their time focused on new features and what their users need.

To quote one customer: “For something very simple in the past, it took a month’s work for us to deliver it for both the iOS or Android apps, and now it only takes one week with Kinvey, meaning that 75% of development work has gone away. By freeing up our developer’s time, they can now focus on the features our customers want and therefore provide a much more enriched experience.”

Comparing Approaches

So, what does this mean to you in terms of savings? The following table highlights projected project costs for applications built using NativeScript.

Project Size	Duration (months)	Project Managers	Designers	iOS Developers	Android Developers	Web Developers	Back-end Developers	Quality Assurance (QA)	DevOps	Total FTE Months	Cost ($100US /hour)
Small	3.0	1.0	1.0	0.0	0.0	5.0	3.0	2.0	0.5	37.5	$600,000
Medium	6.0	3.0	1.5	0.0	0.0	9.0	4.0	3.0	1.0	129.0	$2,064,000
Large	12.0	4.0	2.0	0.0	0.0	12.0	5.0	5.0	1.5	354.0	$5,664,000

When you expand the cost out to assuming you allocate 15% of the app’s initial cost for enhancements and maintenance for years 2 through 5, the numbers look like the following:

Project Size	Initial Investment	Yearly Maintenance (15% per year)	5-Year Cost
Small	$600,000	$90,000	$960,000
Medium	$2,064,000	$309,600	$3,302,400
Large	$5,664,000	$849,600	$9,062,400

Looking at the Kinvey platform approach, projected project costs for applications built using the platform are:

Project Size	Duration (months)	Project Managers	Designers	iOS Developers	Android Developers	Web Developers	Back-end Developers	Quality Assurance (QA)	DevOps	Total FTE Months	Cost ($100US /hour)
Small	3.0	1.0	1.0	0.0	0.0	5.0	1.0	2.0	0	30	$480,000
Medium	6.0	2.0	1.5	0.0	0.0	9.0	1.5	3.0	0	102	$1,632,000
Large	12.0	3.0	2.0	0.0	0.0	12.0	2	5.0	0	288	$4,608,000

© 2019 Progress. All Rights Reserved.

When you expand the cost out to assuming you allocate 15% of the app's initial cost for enhancements and maintenance for years 2 through 5, the numbers look like the following:

Project Size	Initial Investment	Yearly Maintenance (15% per year)	5-Year Cost
Small	$480,000	$72,000	$768,000
Medium	$1,632,000	$244,800	$2,611,200
Large	$4,608,000	$691,200	$7,372,800

Comparing all three approaches looks like the following:

Project Size	Native Approach		NativeScript Approach		Platform Approach	
Project Size	Implementation Cost	5-Year Cost	Implementation Cost	5-Year Cost	Implementation Cost	5-Year Cost
Small	$936,000	$1,497,600	$600,000	$960,000	$480,000	$768,000
Medium	$3,216,000	$5,145,600	$2,064,000	$2,064,000	$1,632,000	$2,611,200
Large	$9,120,000	$14,592,000	$5,664,000	$9,062,400	$4,608,000	$7,372,800

© 2019 Progress. All Rights Reserved.

Conclusion

About the Author

Dan Wilson

Dan Wilson is the Senior Product Marketing Manager for Mobility technology at Progress. Dan has extensive experience growing technology focused products and services. He got his first taste of fast-moving bleeding edge tech when he joined his first start-up in 1999. An avid participant in technology communities, he contributes to a variety of open-source projects, and presents at numerous developer conferences worldwide. Prior to joining Progress, Dan founded and directed a consulting practice for 10 years.

IT organizations who want to be better organizational partners to other areas of the business are going to have to figure out how to more effectively deliver modern applications across a variety of channels. The IT department that chooses to address these needs by using many different platforms and skill sets will continue to experience issues planning and staffing projects, hiring and training staff, and maintaining software throughout the expected lifecycle. On the other hand, the organization that uses high productivity platforms and broadly available skills will be able to remove bottlenecks in staffing, planning, training, development and maintenance.

The use of high productivity platforms that offer low control will provide some immediate benefit, especially if the skills required are already in house, but these platforms will be only be suitable for certain types of applications. The organization going this route should take care to avoid ending up with another specialty "skill" in house that forces the organization to continue to employ the few employees that know how to maintain apps built with these platforms.

The use of high productivity platforms that also offer high control should be explored. The organization should evaluate how much functionality is available out of the box and the skill sets required. The organization would do well to choose a platform that uses standard skills commonly available to avoid resourcing issues now and in the future. Our experts will help you sift through the various options so you can decide what is best for your organization.

Talk with an Expert

About Progress

Progress (NASDAQ: PRGS) offers the leading platform for developing and deploying strategic business applications. We enable customers and partners to deliver modern, high-impact digital experiences with a fraction of the effort, time and cost. Progress offers powerful tools for easily building adaptive user experiences across any type of device or touchpoint, award-winning machine learning that enables cognitive capabilities to be a part of any application, the flexibility of a serverless cloud to deploy modern apps, business rules, web content management, plus leading data connectivity technology. Over 1,700 independent software vendors, 100,000 enterprise customers, and two million developers rely on Progress to power their applications. Learn about Progress at www.progress.com or +1-800-477-6473.

© 2019 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Progress®

© 2019 Progress. All Rights Reserved.