The Role of Packaged Apps in Mobile Enterprise Strategy

a white paper

written by

KEVIN BENEDICT

Founder, Netcentric Strategies

sponsored by

appsFreedom

About the Author

Kevin Benedict, Founder

Netcentric Strategies, www.netcentric-strategies.com

Kevin Benedict is an independent mobility analyst with 22 years of experience in enterprise applications, including executive roles at MobileDataforce and Crossgate. Kevin is a SAP Mentor and a frequent speaker at SAP industry events.

Table of Contents

I. Executive Overview

As smartphones and tablets have become ubiquitous both at work and home, enterprise mobility has been elevated to occupy a central role in corporate communications and IT strategy. Demand for mobile solutions can now be seen throughout the enterprise, from sales and field services to supply chain logistics and operations management.

According to the "Enterprise Mobility Survey 2011," more than 90 percent of companies expect to implement mobile enterprise applications in 2012. Of those, 42 percent plan to implement six or more applications, while 23 percent will implement 11 or more.

Unfortunately, there is no single solution that will enable a company to meet the demand for such a diverse array of user needs. In some cases, a custom built native application may be the optimal solution, while other situations may call for a web-based application. Mobile packaged applications, which serve as cost effective alternatives to SDKs, MEAPs, and other custom software development tools, are also an important class of mobile solutions because they enable companies to cost effectively deploy multiple high impact mobile solutions in a short period of time without the ongoing cost of development.

In order to meet demand for mobile applications in the enterprise, data and experience collected over the past several years indicates that organizations will need to employ a variety of different approaches to building and deploying their apps.

[1] "Enterprise Mobility Survey 2011," Netcentric Strategies, 2011

90 percent of companies expect to implement mobile enterprise applications in 2012. Of those, 42 percent plan to implement six or more applications, while 23 percent will implement 11 or more.[1]

II. Current Trends in Enterprise Mobility

For several years running, the number of mobile devices and operating systems has increased dramatically. Only five short years ago, many companies satisfied their mobility needs with nothing more than a BlackBerry Enterprise Solution (BES) and a BlackBerry. Today, the BlackBerry has been eclipsed by iPhone and Android, new devices are introduced daily, and tablets are poised to radically change the way that we work at home and on the job. The end result is that companies are faced with deploying many apps for many device types, disparate mobile operating systems, and various user groups including employees, business partners and consumers.

Not only have smartphones and tablets become popular, but mobile applications are so ubiquitous that in June 2011, the amount of time people spent using mobile apps overtook the use of desktop and mobile web combined. At that time, the average user spent 81 minutes per day using mobile applications as compared to 74 minutes using desktop and mobile web apps.[2] This user behavior is driving a radical paradigm shift within IT departments that further elevates mobile technology to a critical role in facilitating worker efficiency.

Although this data suggests that a mobile strategy is more important than ever, most companies report that they simply don't have one. Even though 80 percent of companies report that enterprise mobility is "very important" or "critical" to the future success of their companies,[3] survey results from *InsiderResearch* indicate that only 33 percent report having an enterprise-wide mobile strategy, and 52 percent report little to no oversight of mobile strategy.[4]

Given the demand being placed on IT to supply a diverse array of mobile applications combined with the complexity of mobility platforms and software development tools, a comprehensive mobile strategy is critical. According to the *InsiderResearch* survey, companies are building and executing on mobile strategies for the following reasons:

[2] http://blog.flurry.com/bid/63907/Mobile-Apps-Put-the-Web-in-Their-Rear-view-Mirror "Mobile Apps Put the Web in Their Rear-view Mirror," Charles Newark-French, Flurry.com, Jun 20, 2011
[3] "Enterprise Mobility Survey 2011," NetCentric Strategies, October, 2011
[4] "Mobile Outlook 2012," InsiderResearch, February, 2011

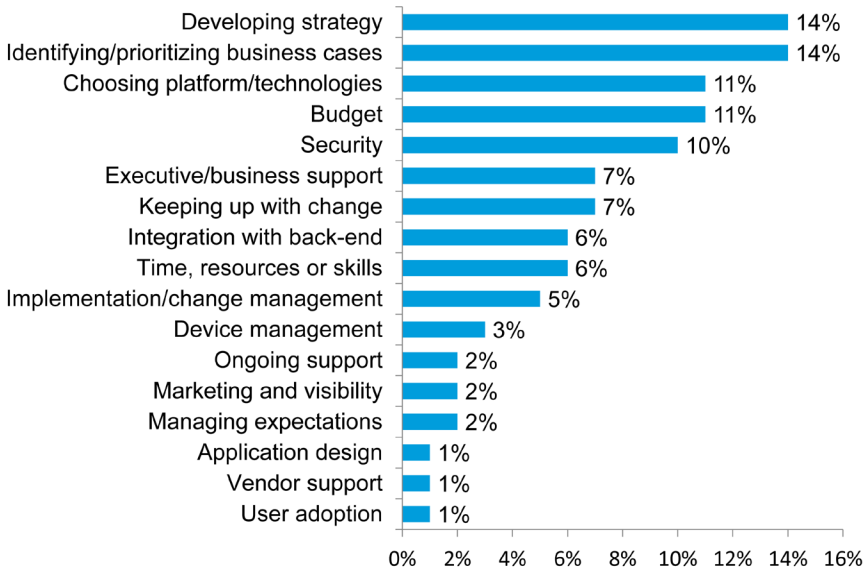
Mobile applications are so ubiquitous that in June 2011, the amount of time people spent using mobile apps overtook the use of desktop and mobile web combined.

So if companies are faced with complex requirements that necessitate the use of complex solutions, why aren't more organizations utilizing a mobile strategy? The simple answer is that developing a mobile strategy takes time, thought, and resources – three things that most companies are short on.

The following survey results from *InsiderResearch's* Mobile Outlook 2012 report reveals a more detailed list of what companies consider the primary barriers to developing a mobile strategy, the first two of which are:

1) Putting pen to paper and actually writing down the strategy, and
2) Figuring out who needs certain apps.

Following the issues of "developing strategy" and "identifying/prioritizing business cases," the next biggest challenge is choosing the right platforms/ technologies. This speaks to the complexity of the mobile solutions available today. Many solutions, such as MEAPs, require extensive analysis in order to ensure that they contain the features and functionality necessary for fulfilling the stated business and IT requirements.



Only 33 percent of companies report having an enterprise-wide mobile strategy.[4]

III. Mobile Application Types

There are many development tools and platforms available on the market, and there are four primary mobile application types available today: web-based (non-HTML5), HTML5, hybrid, and native. Web-based (both HTML4 and HTML5) applications run through the device's browser.

Hybrids utilize a combination of web-based (HTML5) and proprietary native "containers", while pure native apps are written in the native language of each device operating system. There are distinct advantages and concerns associated with each approach:

Application Type	Benefits	Concerns
Native Applications	Superior UI functionality and user experience Integration with native OS optimizes use of native phone features and hardware Ability to store data locally enables offline use More security and authentication options available than with web-based apps	A separate application must be developed and maintained for each mobile operating system Updating applications can be a complex process, especially when dealing with multiple operating systems Development costs are generally higher, particularly with cross-platform deployments
Web-based Applications (non-HTML5)	Leveraging the browser simplifies cross-platform deployment Upgrade and version updates are easy and changes are immediate Development costs are generally lower than native Lost, stolen, or compromised devices pose less of a security threat because data cannot be stored offline	Applications require an active network connection at all times in order to run Since data is not stored locally, offline data storage and functionality is not possible Since the application is not integrated with the OS, the UI capabilities are less robust than native apps Fewer security options are available than with native applications
Hybrid	Combines the benefits of HTML5 with the power of a native application	Requires additional development, support, and maintenance costs and usually requires vendor software licenses and maintenance fees
HTML5 Applications	Simplified cross-platform development, taking about half the time of a native application development Easier upgrading and version updating in real time Development costs are generally lower than native Offline functionality and data storage	Limited UI capabilities compared with native applications Slower performance than applications programmed in native code Fewer security options are available than native applications Not as robust data storage capabilities when compared to other specialized mobile database options

Since mobility is a complex equation, a well thought-out mobile strategy breaks down the big issues into discrete problems that can be addressed one at a time.

These advantages and limitations mean that most organizations will have to utilize a combination of different application types. In each case, the best technology for the job depends largely on the use case and user requirements. This is where a mobile strategy can help an organization lay out the various pieces of their comprehensive plan to quickly and easily meet the demands of various subsets of their user population. Since mobility is a complex equation, a well thought-out mobile strategy breaks down the big issues into discrete problems that can be addressed one at a time. Doing so enables a company to identify smaller, quick-win scenarios that can produce a great ROI in a short period of time.

IV. Development Approaches

Three approaches to mobile application development are most often used in the enterprise today:

- SDKs and programming tools to custom code applications
- Mobile enterprise application platforms (MEAPs) with programming tools and templates
- Packaged applications

Each type of system has its advantages and disadvantages, and organizations that are publishing multiple applications will likely use a combination of different approaches based on application and user requirements.

SDKs and Programming Tools

Native development tools like the iOS, Blackberry, and Android SDKs are commonly used to build custom mobile applications. Using SDKs and other programming tools is often the first approach that developers think of when evaluating a mobile project or piloting an app for a small subset of users. Custom coding is attractive because it gives the developer complete control over UI elements and security features. Although it's a good approach for companies looking to get their feet wet with minimal upfront investment, it lacks several key elements for enterprise deployment such as scalability, app management, and user management. Although SDKs and related programming tools are appealing to ambitious programmers who like to get their hands dirty, they carry significant liabilities associated with the development and maintenance of multiple code bases and can quickly become a time consuming and expensive drain on resources.

Since mobility is a complex equation, a well thought-out mobile strategy breaks down the big issues into discrete problems that can be addressed one at a time. Doing so enables a company to identify smaller, quick-win scenarios that can produce a great ROI in a short period of time.

Mobile Enterprise Application Platforms

A MEAP is a sophisticated platform and development solution that often enables programmers and even IT administrators to build one application with a WYSIWYG IDE and deploy it across multiple operating systems. MEAPs offer many benefits including scalability, app and user management, and configurable security features, and many work hand-in-hand with mobile device management (MDM) solutions in order to provide additional security and control. Most importantly, since MEAP providers are responsible for ensuring continued support for new devices and operating systems, they transfer the compliance burden from the development team to the vendor. The major stumbling block with MEAPs, however, is the price. They often come with licensing, support services, and training costs that usually start in the hundred-thousand-dollar range and go up from there, making them a significant investment for enterprise customers and by and large too expensive for small to mid-sized businesses.

Also, selecting, deploying, and managing a platform is a complex, time consuming approach. When an organization needs to buy a platform, they typically have to undergo a long evaluation process to ensure the solution will meet their requirements, followed by in-depth training. As a platform will eventually touch all business applications, IT needs to make sure there are no hidden pitfalls that will cause problems later on. Implementing and developing on a platform also takes time and money, typically involving specialized skill-sets that may not be available in-house. So it may take a while, typically months, before the first mobile app is rolled out, by then, the organization has already committed to the mobile platform whether the apps are successful or not.

Packaged applications are designed to mobilize specific systems and process, they do so very efficiently while their integration and deployment mechanisms provide a rapid speed to market.

Packaged Applications

A packaged application is a pre-built mobile app that is provided and maintained by a third party provider. The app may be native, HTML5, hybrid, or even web-based and is generally built to run across all major operating systems and device types.

Many packaged apps come pre-integrated with core back-office solutions and provide user and app management capabilities similar to features found in MEAP systems. Often, packaged apps will work hand in hand with MDM solutions, providing the scalability and security features required for an enterprise.

Packaged apps can usually be installed on premise or hosted in the cloud. One of the important benefits to consider is that many packaged applications are pre-integrated with core back-office business solutions. This enables rapid and efficient integration that can potentially save significant time and money.

Packaged applications combine many of the benefits of custom coded apps and MEAPs, while delivering those benefits at an affordable price. Perhaps the most important advantage to packaged applications is that the vendor shoulders the responsibility for maintenance, upgrades, and support. This alleviates the burden on IT departments to continually support new operating systems and devices, freeing up resources to focus on other mission critical projects and core competencies.

Most packaged applications can be tailored to the needs of the customer, though they don't offer the same level of customization as an app that is built from the ground up. However, reputable vendors keep pace with new devices and OS upgrades and offer a suite of specialized applications that run across all major device platforms. Since packaged applications are designed to mobilize specific systems and process, they do so very efficiently while their integration and deployment mechanisms provide a rapid speed to market. In comparison to many MEAPs, packaged apps are often available at low price points.

Security

Companies that mobilize enterprise data systems need enterprise-grade security protections. Single sign-on, full data encryption, 2-factor authentication, and a secure tunneling framework are essential requirements for securing data and devices. In order to meet corporate security requirements and protect against lost or stolen devices, many companies choose real-time applications that prevent data from being stored locally on-device. Further protections include IT admin and control over app settings and permissions, role-based access, and dual-password protected data access.

Critical Considerations

When choosing development tools, it is important to anticipate the effects of the following two circumstances on budget and developer resources:

1) When an app needs to be deployed to more than one device type and/or operating system, which necessitates the development and maintenance of multiple code bases.
2) When new device models and operating system versions are released into the market, requiring developers to upgrade their apps in order to ensure compliance with new device hardware and software.

In the first circumstance, an organization might build an application for one type of device, like the iPhone. If the app is successful, there will be demand to recreate it for the BlackBerry and Android. And with the rise of tablets, developers are finding they need to rebuild their apps in ways that take advantage of the larger screen size. So what started off as a simple iPhone app has now turned into an iPhone - iPad - BlackBerry - Playbook - Android - Android tablet app.

And what was one simple code base has grown into six distinct code bases written in four different programming languages. Creating and maintaining this is a complicated and resource-intensive process that companies need to plan for by putting the right tools and strategies in place from the beginning.

The second set of circumstances is often overlooked due to the fast rate at which mobile operating systems are evolving and new devices are being introduced onto the market. Each month, dozens of new smartphones and tablets are released with varying screen sizes, resolutions, and new hardware features. Additionally, Apple, BlackBerry, Android, and even Windows are innovating their operating systems at an extremely fast pace with several major upgrade announcements made each year. Organizations that build and maintain their own applications carry the burden of maintaining compliance with all of these new devices and OS upgrades – a burden that falls squarely on the shoulders of the development team.

It is important to remember that in most cases, developing the application is just the first step. The real work comes after the initial development in the form of ongoing support and maintenance.

Four development models provide enterprises with different options and key points to consider before developing their enterprise mobility strategy.

V. Overview of Development Models

The following implementation models take into account a variety of common needs, including the number of apps being developed, who they are being developed for, and the devices they will run on:

Model #1	**Developing one app for employees to run on one kind of company issued device**
Development Options	In-house utilizing SDK's or free development tools Outsourcing to a third party Web-based version optimized for employee devices Packaged application that meets end user requirements
Key Decision Points	Do we have the internal resources to build, maintain, and update this application? Can we host the application internally? What security parameters are necessary? How do we protect company data when devices are lost or stolen? Can BES and/or Exchange Server provide the MDM functionality necessary? How are we going to provide access to backend systems? Is there an existing packaged application available that is offered by a software provider (e.g. Salesforce.com Mobile, Oracle Sales Assistant, SharePoint Mobile, etc.) and if so, does it provide the necessary functionality and security? Ongoing support – which business budget will support future upgrades and maintenance?
Model #2	**Developing multiple apps for employees to run on one kind of company issued device**
Development Options	Custom coding with available IDEs and SDKs, using internal staff augmented by external mobility experts Outsourcing to a third party while utilizing internal resources to mitigate extensive upgrade and maintenance costs When possible, web-based solutions that can be developed and maintained in-house When possible, packaged application that meets end user requirements
Key Decision Points	Do we have the internal resources to build and then maintain and update multiple applications? Can we host the applications internally? What security parameters are necessary for each app? How do we protect company data in the event of a lost or stolen device? Can our BES and/or Exchange Server provide the MDM functionality necessary? How are we going to provide access to backend systems? Is there one or more packaged application(s) that we can use to meet some of our needs? Ongoing support – which business budget will support future upgrades and maintenance?

Developing the application is just the first step. The real work comes after the initial development in the form of ongoing support and maintenance.

Model #3	**Developing one cross platform app for employees to run on a wide variety of devices**
Development Options	Custom coding web-based or native applications with available SDK's or free development tools is an option, but companies should be aware there will be a significant amount of future work associated with maintaining the code base(s). The availability of quality development resources and ongoing budgets are critical factors that will determine project success. Outsourcing development to a third party mobility expert is also an option, however, companies should be aware there are significant long term costs to maintaining multiple code bases and should only contract with third party developers that are sure to be available in the future to update the apps for new operating systems and device types. Use packaged applications to mitigate cost and shift development, upgrade, and maintenance cost off of the internal development team.
Key Decision Points	Do we have the internal resources to build and then maintain multiple code bases? Will we be able to ensure compatibility with new operating systems, upgrades, and device types as they are released onto the market and adopted by end users? Can we host the application internally? What security parameters are necessary? How do we protect company data when devices are lost or stolen? Can a BES and/or Exchange Server provide the MDM functionality necessary? How are we going to provide access to backend systems? Is there an existing mobile application provided by a software vendor (e.g. Salesforce.com Mobile, Oracle Sales Assistant, SharePoint Mobile, etc.) and if so, does it provide the necessary functionality and security? Ongoing support – which business budget will support future upgrades and maintenance?
Model #4	**Developing multiple customized cross platform apps for employees or customers to run on a wide variety of devices**
Development Options	A cross-platform development tool (MEAP) A long-term and strategic development partner is essential if development resources are not available. It is important to require your third party development partner to use a MEAP and IDE of your choice so you can edit, update and control future development efforts. When possible, web-based solutions that can be developed and maintained in-house Use packaged applications whenever possible to mitigate cost and shift development, upgrade, and maintenance cost off of the internal development team.
Key Decision Points	If we have the internal resources, how are we going to build and then maintain multiple code bases? What tools are necessary to do this effectively? How do we ensure compatibility with new operating systems, upgrades, and device types as they are released into the market and adopted by end users? Can we host the applications internally? What security parameters are necessary? How do we protect company data in the event of a lost or stolen device? What MDM solutions are available that fit our needs and provide the required functionality? What is the cost/benefit of purchasing an MDM solution? How are we going to provide access to backend systems? Is there one or more packaged application(s) that we can use to meet some of our needs? Ongoing support – which business budget will support future upgrades and maintenance?

The average lifecycle of a mobile consumer application is less than 30 days. While mobile enterprise applications generally last longer, their lifespan is significantly curtailed by the rapid evolution of devices and their operating systems. Updating applications is a key component of mobile application development that is often overlooked, and purchasing packaged applications can provide several strategic advantages. Deploying targeted high-value packaged apps in a relatively short timeframe allows an organization to recognize ROI very quickly. Shifting the development, maintenance, and upgrade burden to venders saves money and resources while reserving the ability to scale services as demand increases.

VI. The Value of Packaged Apps

Incremental Solutions with Rapid Implementation: as demonstrated by the development models above, packaged applications can be utilized alongside other approaches to achieve the overarching goals of a comprehensive mobile strategy. Utilizing an incremental approach to mobility allows an organization to identify use cases where packaged applications can meet user requirements. Since packaged applications leverage out-of-the-box integration, the implementation process is streamlined, resulting in a very fast time to market.

Small User Groups: packaged applications are also particularly relevant when targeted implementations are required for a small user group. Since applications are generally priced on a per-user basis, deploying to a small number of users is cost effective and easy on resources, enabling an IT organization to provide solutions for user groups that may otherwise go overlooked due to their small size.

Extensions to Core Systems: due to their specialized nature, packaged applications are often a very effective way of extending CRM, ERP, and other enterprise systems.

Freeing Internal Resources: in situations where IT resources are constrained, packaged applications offer a means of providing mobile applications to key user groups without burdening developer resources or diverting personnel away from their areas of core competency and mission critical projects. Since the vendor shoulders the responsibility of developing, deploying, and maintaining the applications, customers don't have to worry about implementation models, development platforms, internal developer expertise, etc. Packaged applications can generally be installed on-premise or in the cloud depending upon the security needs of the company and the IT architecture they prefer.

Rapid Implementations: when a quick ROI is needed, packaged applications offer a very fast speed to market, rapid deployment, and immediate ROI.

Scalable Solutions: packaged applications are often a good fit for small deployments, and most offer a cost effective way to scale services to additional users. Companies opting to build custom apps for a few users may not realize the complexity of scaling it to the entire organization, particularly to multiple device types and operating systems. Maintaining multiple apps and multiple code bases on different platforms is a high-cost burden that is often overlooked. Packaged applications offer a very effective way to side step those costs.

Pre-built and integrated packaged apps offer speed to market advantages, and vendor-supported updates, support, and maintenance.

VII. Conclusion

Packaged applications are an alternative to SDKs, MEAPs, and other custom software development tools because they enable companies to cost effectively deploy multiple high-impact mobile solutions in a short period of time without the ongoing cost of development. They play an important role in enterprise mobility and should be utilized as a cost effective alternative to custom building with SDKs, programming tools, and MEAPs. By shifting the development, support, maintenance, and compliance burden to the vendor, the strategic use of packaged applications frees up development resources for other mission critical projects that require custom development. Many packaged applications support multiple device types and operating systems, which simplifies the task of supporting user-owned devices associated with the BYOD movement.

In summary, many pre-built and integrated packaged apps offer speed to market advantages, and vendor-supported updates, support, and maintenance. These benefits are often available with subscription based pricing that allows IT departments to accurately predict budgetary requirements, which results in a reduced risk profile and lower total cost of ownership.

appsFreedom™
Driving Business Productivity

1.855.APPSFREEDOM
(1.855.277.7373)

info@appsFreedom.com
www.appsFreedom.com

appsFreedom @appsFreedom