UBM Tech

October 2014

Mobile App Development: Moving From Good to Great

To build a great mobile app, you need the right skills, tools, and support. That's why many companies are turning to integrated and collaborative mobile application development environments that let them meet business needs and exceed customer expectations.

INSIDE

SPONSORED BY IBM

1409294

Mobile App Development Today

Any mobile developer can build a good mobile app, but good often is not enough to increase wallet share, customer loyalty, employee productivity, brand value, or reputation. Good needs to move to great. To build a *great* mobile app, you need the right skills, tools, and support.

A great mobile app goes beyond the UI, providing a robust end-to-end solution. It integrates usability and security and connects with backend systems, optimized for mobile, for enhanced functionality. Thinking end-to-end, and beyond just one app, is critical. After all, users expect to have access to all the data and services around them. And don't forget these key intangibles: elegance, dependability, and usability. Combine all this with tools to support continuous integration and testing across multiple device types and platforms, and you're on your way to building great mobile apps.

App quality is critical, because mobile is no longer considered an

"add-on" to applications. Users demand functional, engaging, and robust mobile apps to do what they need, wherever they are, whenever they want. In fact, mobile is increasingly becoming the only planned platform for new application development.

Why? Mobile provides a unique opportunity to address user feedback through continuous development, so much so that in the mobile world, rapid development and deployment is a user expectation.

With so much at stake, how do you deliver quality, security, engagement, end-to-end services integration, dependability, and total flexibility in terms of platform support in an efficient and cost-effective manner?

Simply throwing together disparate open-source components quickly reveals its limitations. This includes its primary focus on UI development with lack of support for integration, continuous development and testing, analytics, management, and more. Let's explore some best practices and suggestions on how to achieve greatness.

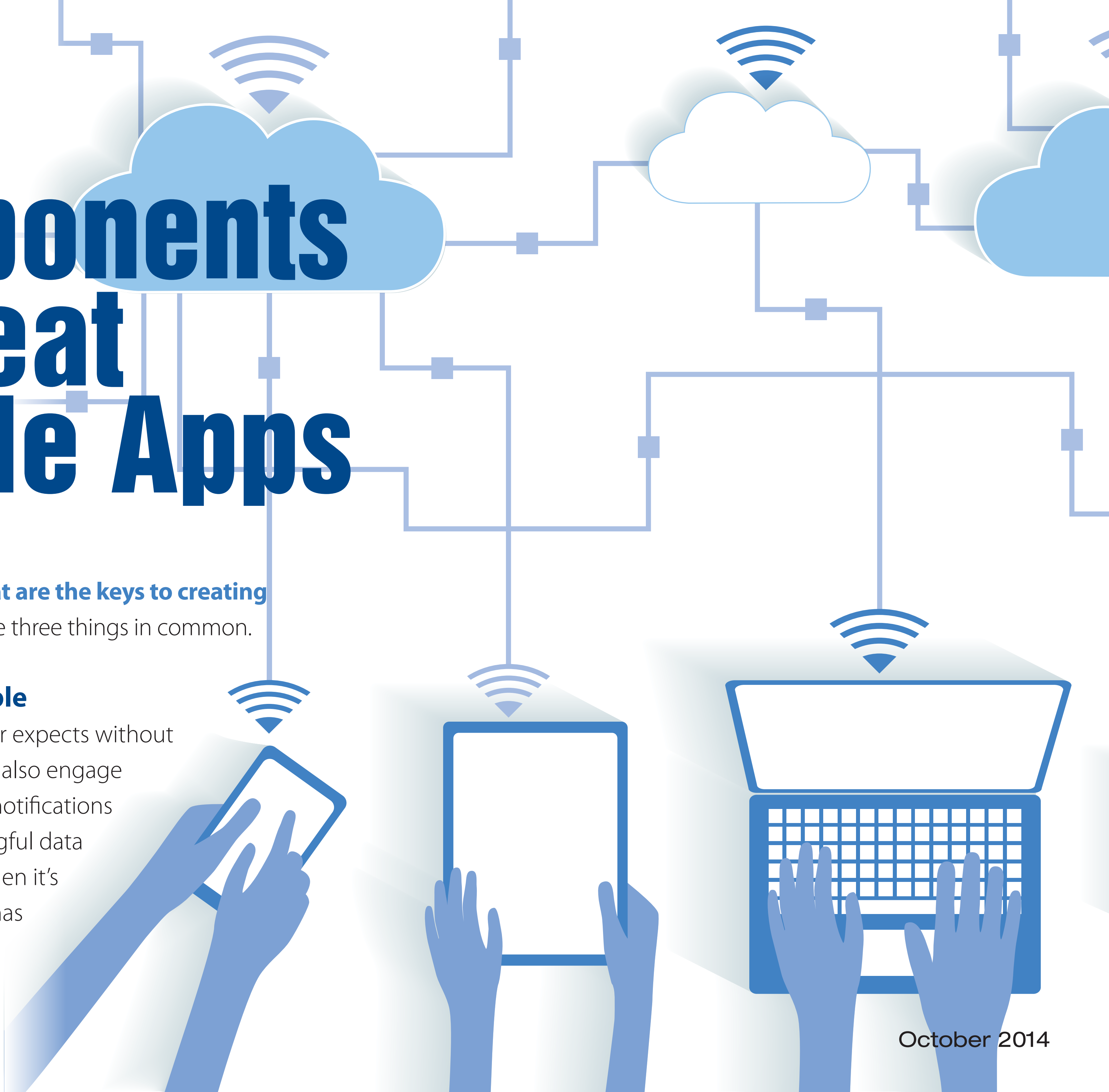
Any mobile developer can build a good mobile app, but good often is not enough to increase wallet share, customer loyalty, employee productivity, brand value, or reputation.

Top 3 Components of Great Mobile Apps

If any developer can build an app, what are the keys to creating apps that stand out? Great apps have these three things in common.

1 They're engaging and reliable

Great mobile apps do what the user expects without latency and without crashing. They also engage the user in new ways. For instance, push notifications enhance engagement by driving meaningful data to the app and its user at the moment when it's most relevant (for example, when a user has reached a retail location).



Push notifications make this straightforward for mobile, while the web world continues to struggle to provide this capability. Location services and beaconing provide for intelligent engagement, context awareness, and the timely delivery of data and services.

Tip **Use a single API with server-side** *components to push data to apps across native platforms or to handle location-based contextual awareness and implement related services.*

Tools that support analytics provide insight into how, when, and why your apps are used, and they can help predict additional user needs. This capability allows you to manage apps actively, offer new services to users, optimize user workflows, optimize backend systems, identify performance issues, and develop related apps to extend your engagement with users.

Tip **To achieve reliability, look for** *tools that offer secure, guaranteed messaging services to deliver data end-to-end regardless of platform. Cross-platform app diagnostics and logging tools can aid in the remote monitoring of applications in the event problems arise or response times fail to meet expectations.*

2 They integrate with the cloud

Integration is key to fostering the level of functionality and interaction users expect from their mobile apps. Traditionally, it's been a challenge to integrate third-party libraries, components, and even code into mobile apps, but tools and frameworks make this easier by offering these things built-in.

Libraries alone can't deliver the full spectrum of integration requirements; mobile apps need to integrate with services and data to offer true value. To do so, mobile apps must connect to services that were built long before mobile was in existence. As a result, the use of HTTP, JMS, and even database connectivity is required.

One strategy is to decouple integration logic from mobile app logic. With this approach, integration logic is encapsulated on the server, isolating a mobile app from changes that occur in the data and service layers. One set of integration services can then be used by mobile apps running across a number of different platforms.

This allows server-side transformation to optimize the data delivered to mobile apps, including data payload size reduction, and the aggregation of data from multiple services into a single network call. As a result, costly round trips and the impact of unreliable mobile networks are reduced. Incorporating cloud-based services also helps to achieve the elasticity required in the mobile world.

Tools that help you achieve all these optimizations not only improve perceived app performance and response times; they also help you achieve zero-code integration, or close to it, across mobile app platforms.

Tip **Use zero-configuration tools to** *configure and optimize mobile-app services automatically without building new systems.*

3 They foster trust

Secure mobile apps are covered end-to-end. This involves securing the device, the app, the user, data, and backend systems from unauthorized use should the device be stolen or compromised.

Considerations include user authentication, device authentication, and more. For example, developers of great apps need a way to make sure the app hasn't been compromised, and that the device is trusted (not jailbroken or stolen) and free of malware. Tools are critical for this detection and for providing device provisioning with ongoing management.

! Tip **Look for mobile tools that** *include the ability to perform contextual risk assessments, remotely wipe data, disable an app or the entire device, and prevent login by user, device, or an entire application's user base.*

For authentication, the framework should plug into SSO, LDAP, and other third-party security systems without exposing security holes (e.g., requiring you to move LDAP servers outside a trusted zone for mobile).

! Tip **Use an intermediate server** *to provide authentication and authorization services across platforms, and to be firewall friendly.*

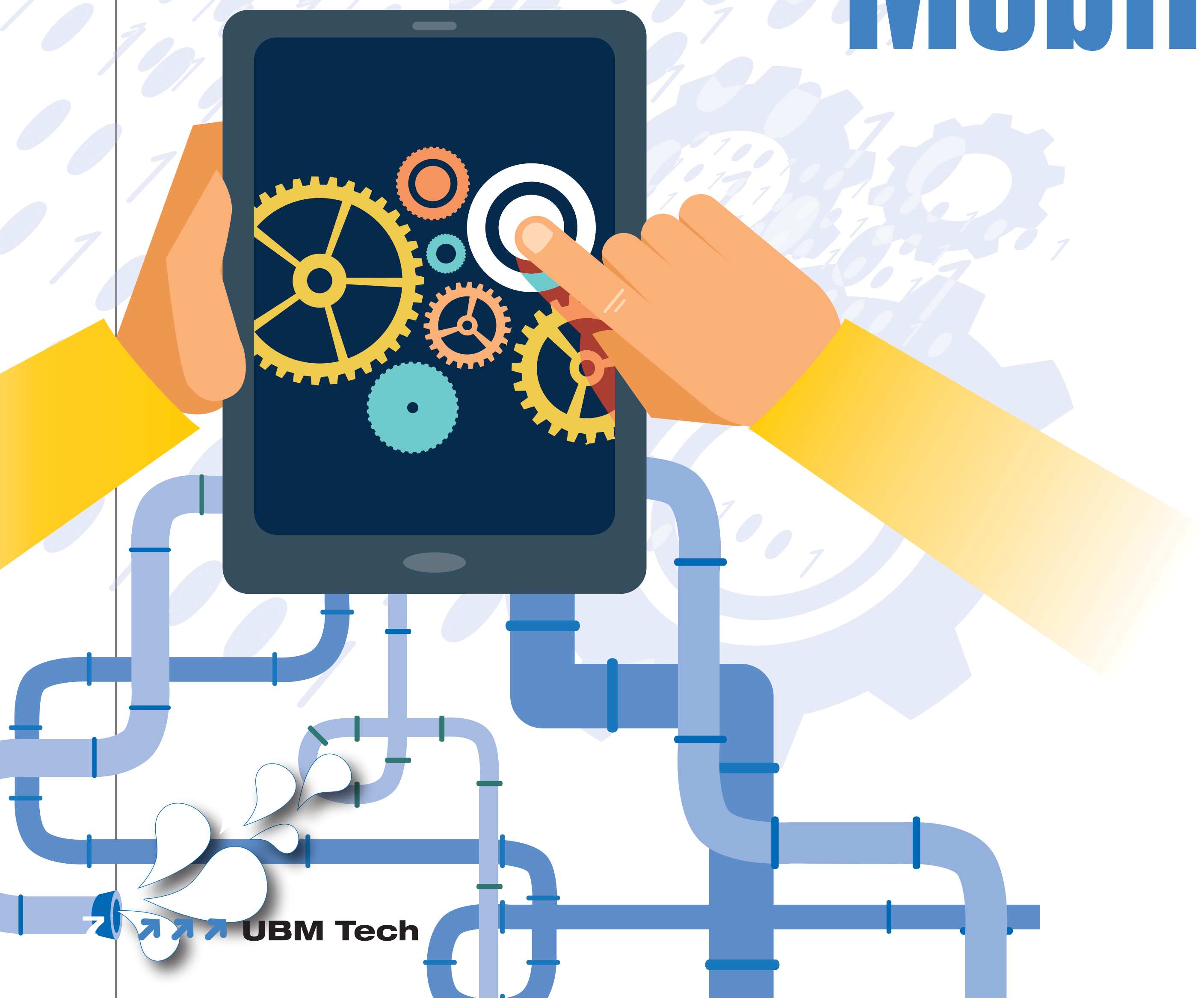
Great mobile app tools provide certificate-based security and multi-factor authentication where needed, leveraging both device and server enforcement.

The use of on-device virtualization provides sandboxing for security and other BYOD issues.

! Tip **To secure data and allow for** *safe, effective, offline app usage, look for tools to encrypt on-device data storage.*

Security starts with a secure coding environment. The use of a continuous integration process with additional tools to detect potential security holes, data leaks, SQL injection risks, and other high-risk patterns in code during the development lifecycle goes a long way toward runtime security. ■

How to Build Great Mobile Applications



To begin, you need a strategy built on:

- Your own unique domain knowledge
- Your business data
- Analytics and workflow
- Tools and processes for overall efficiency, effectiveness, and satisfaction

Your strategy should include four focus areas:

1. **Reach and engage:** app usability and contextual awareness
2. **Integrate and extend:** deep integration of both systems and data
3. **Secure and optimize:** security, robustness, and efficiency
4. **Innovate and expand:** the cloud, backend services, and predictive analytics

Underlying each focus area is an agile development process anchored by DevOps capabilities from ideation through delivery.

UBM Tech

October 2014

In the mobile world, if an app fails to meet expectations, doesn't evolve quickly enough, performs poorly, or fails in some way, users may delete it and move on to another one. Knowing about potential issues early in the development process requires a user-feedback loop — and it is critical to success.

Continuous Delivery

Apps grow more productive with continuous use and improvement. An iterative process of exploration of user interaction, transactions, and collected data will optimize and enhance an app's efficiency. The agile development methodology encourages rapid iterations through the development cycle, eliciting end-user feedback at every step, and making early corrections based on this feedback.

This process demands collaboration with all stakeholders, including product management, development, QA, IT operations staff, and users. Additionally, due to the iterative nature of this approach, integrated and automated tools that support the ongoing activities of design, build, test, and deploy are needed.

User Involvement

Although some may find the build-and-deploy-often approach disconcerting, mobile users have come to expect it. It's critical to ensure that user feedback (either directly or via analytics) is driving change.

The first step is to create a feedback-driven process for continuous improvement and optimization using a secure online tool to gather and collect feedback and provide two-way collaboration. This involves sharing development milestones such as agile planning, the resulting stories, the backlog and in-process items, status of tasks, and the results of testing and deployment.

Continuous Testing

The testing process and platform you choose should be flexible enough to enable any mobile development approach (native, web, and hybrid), integrating the tools you choose without forcing you down a prescribed path.

The use of automation enables a continuous testing process, and should include the testing of functionality and app-specific characteristics such as user workflow, security risks, data optimization, and backend interaction.

Mobile apps introduce new challenges, such as varying device platforms and sizes, gesture-based interactions, the need for extreme scalability, ultra-low response times, internationalization, overall usability, and considerations for resource-constrained environments. it's best to use simulators for the many variations and permutations of devices, screen sizes, platforms, and product families.

Testing tools need to support all this on the device side, but they should also support virtualization of backend services. With an agile and DevOps-driven cycle, mobile app functionality may need to be tested before server rollouts are complete.

A Closer Look at DevOps

DevOps comes from the combined terms "development" and "operations." Driven by agile development principles, demand for rapid deployments, the use of virtualization and the cloud, and the maturity of continuous integration tools, DevOps stresses collaboration between development and operations staff, helping them work together more closely to achieve common goals. This collaboration has become more important as teams deploy and release software more often.

Traditionally, developer and operations roles have been at odds, as development disrupts the production environments that the operations team tries to keep stable.

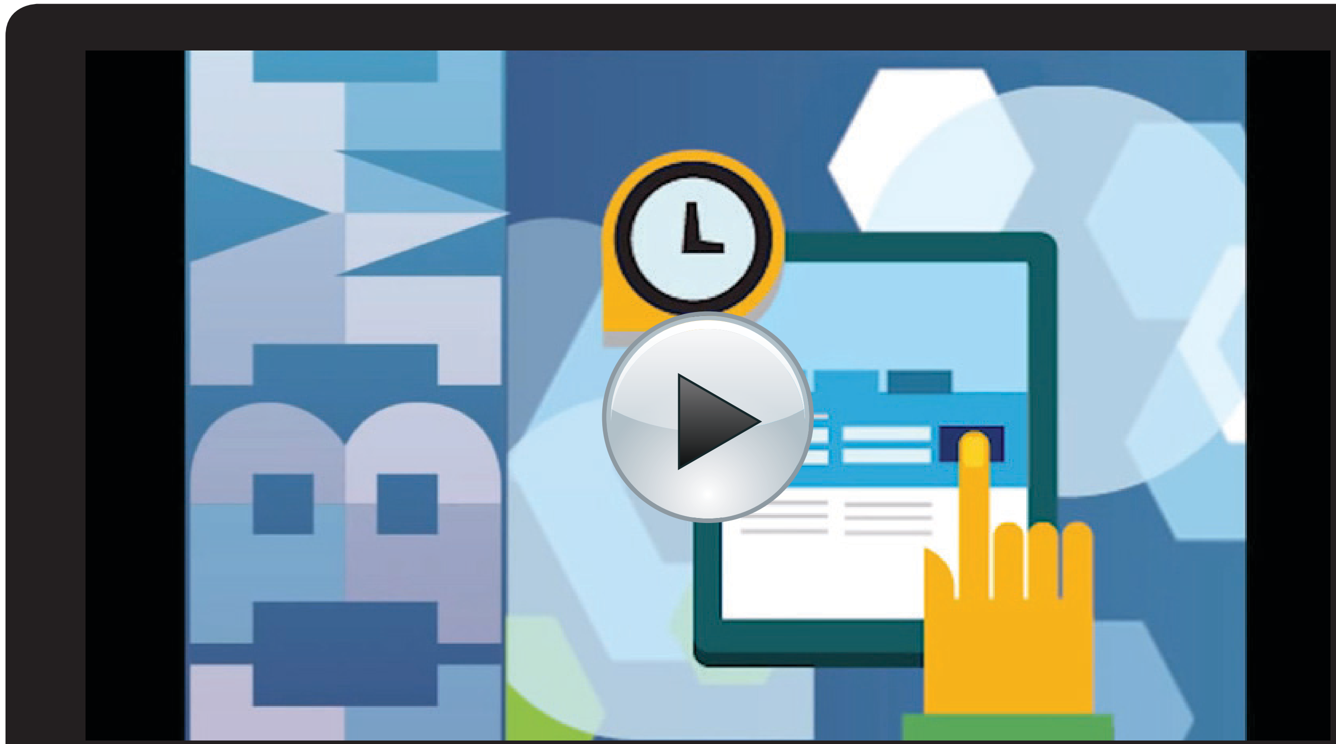
DevOps unifies their goals and builds their interdependence in achieving successful rapid deployments. Developers gain greater control over QA and production environments, and operations teams become involved in development and testing efforts early on.

To save time, reduce effort, and remove errors, development tools are supportiing DevOps practices, including Collaborative Development, Continuous Testing, Continuous Release and Deployment, Continuous Monitoring, and Continuous Customer Feedback and Optimization, providing support for full lifecycle development, management, and deployment from the idea through delivery.

Virtualization helps with this, and tools should help, too. However, mobile increases the need for QA systems to monitor a production environment, because performance issues or crashes experienced by real users can be detrimental.

Continuous Release and Deployment

In DevOps, IT operations teams collaborate with developers and users in the assembly and deployment of each release. Employing a traceable process that tracks all assets from inception through QA and into end users' hands is critical to success. Continuous release and deployment provides a continuous delivery pipeline that automates deployments to test and production environments. It reduces the amount of manual labor, resource wait-time, and rework by means of push-button deployments that allow higher frequency of releases, reduced errors, and end-to-end transparency for compliance.



↑ Mobile DevOps in action

Simplifying Support

A mobile framework that supports a push model for app updates, instead of relying on users to update manually, offers advantages. Without push support, mobile DevOps requires additional cross-system support, as old versions of apps may exist alongside new ones.

A framework that supports cross-platform development can eliminate the need to support multiple native build-and-deploy tools.■

Native, Web, or Hybrid Apps?

The first step in building a great mobile app is determining the appropriate development approach that best supports the app, including which platforms to support natively. For instance, you might choose a subset of platforms on which to build natively, while less popular platforms can be addressed with mobile web apps or a hybrid approach.

Next, determine screen sizes and related devices to support, such as the iPhone 4S, the iPhone 5, and Android devices with 4.7" and 5.5" screens (e.g., the Samsung Galaxy S3 through S5). Next, think beyond code to consider how to handle push notifications, app logs and error reporting, data security, backend communications, and provisioning support across platforms. Choose a mobile framework that supports push notifications across device types and offers other cross-platform features such as remote log retrieval and unified server communications.

Native mobile development toolsets are efficient, result in high-performing apps, and allow access to the platform's official app store. However, with each native environment comes the need to maintain separate code bases and skillsets. Additionally, native toolsets often lack tools for backend integration, continuous development, security, and testing.

Mobile web applications hold the promise of a single app code base that runs across multiple devices. However, challenges include the lack of device feature support and issues around varying screen sizes.

Most organizations strive to find a balance when it comes to multi-platform support (iOS, Android, Windows Mobile, and so on) on a per-app basis.

Tips for Pure Web Apps

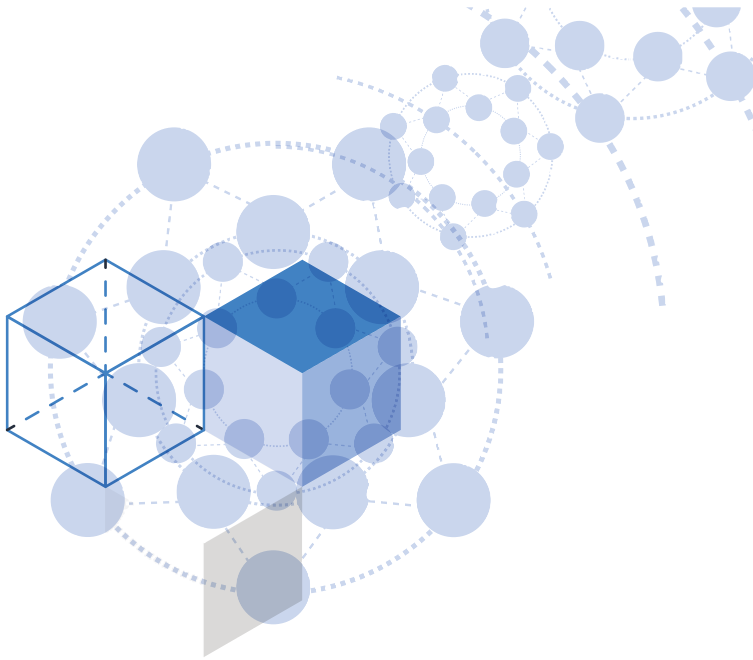
The mobile web with dynamic user interaction via Ajax or HTML5 is one strategy for building device-agnostic mobile apps. Often, mobile web apps maintain the look and feel of native apps and can even offer native feature integration (for example, access to device contacts).

To be effective, aim to maintain a single code base across devices using a standard such as HTML5.

Strive to build single-page apps with common UI anchors, which are similar in appearance and behavior to native apps, to navigate between pages. . Some frameworks offer a unified API with limited access to mobile features across devices types, such as using the camera, making phone calls, and sending SMS messages.

Tips for Hybrid Apps

Though mobile web apps solve many multi-device issues, they're not always ideal. A hybrid mobile app is part native and part web, leveraging device features natively while maintaining a common HTML5 codebase.



Mobile tools can help build hybrid apps and offer additional multi-platform support. For instance, advanced view composition makes it possible to combine native code with web controls and components to balance the user experience while supporting many devices and screen sizes at once.

With this approach, the toolset needs to offer the ability to coordinate user activity and state, along with data, properly across devices and platforms.■

Mobility Pros and Cons: Native, Web, or Hybrid?

The debate about building native, mobile web, or hybrid apps can be endless. Each approach has its own pros and cons:

Native Mobile Apps

Pros:

- Top performance
- Full device feature integration
- Platform app store availability
- Full gesture support
- Offline access

Cons:

- Specific skillset required (expensive to develop)
- Specific toolset needed
- App store approval required
- Multiple code bases (one per platform)

Mobile Web Apps

Pros:

- Similar UI across devices
- Single cross-platform codebase
- Updates immediately available
- More general skillset (cheaper to develop)
- URL access

Cons:

- Browser visible on screen
- Lack of device integration
- No device-specific features
- Mediocre performance

Hybrid Mobile Apps

Pros:

- Cross-platform codebase
- Integrates with device-specific features
- Improved device integration
- App updates immediately available

Cons:

- Requires a mix of skillsets
- Requires app store approval
- User "pull" model for installation
- Per-device browser behavior differences

Integrated Development Platform Advantages

In a successful mobile-first development strategy, frameworks and tools should never get in the way. They should enable developers, managers, and operations, and they should help automate tedious and repetitive tasks. In a nutshell, they should remove complexity while providing enterprise functionality (including integration, security, reliability, and analytics) out of the box .

Organizations that try a do-it-yourself approach with open-source material often find that it requires manual integration of disparate components, tools, and systems. Manually integrating development tools is no easy task. Each comes with a learning curve, specific support requirements, and other challenges.

Open-source tools are a good start, but a solution in which the required components have been preconfigured, integrated, and installed as a single package can accelerate development.

Choosing a vendor-supported commercial framework with tools, backend integration, and full lifecycle development and management support is a partnership with even greater value than the sum of its parts.

IBM MobileFirst Platform: How to develop and deliver great mobile apps quickly

The IBM MobileFirst Platform is an award-winning, integrated, end-to-end platform for developing great mobile apps. Offering support for agile principles and anchored by DevOps capabilities, it supports continuous delivery with automated testing of functionality, user workflows, security risks, data optimization, and backend interaction. Automated deployment of software components supporting the mobile app helps ensure consistency and reliability from the device through the cloud and enterprise systems providing services.

hhgregg grows revenue by 50% with stronger customer engagement

Just as important, it helps coordinate pools of users for manual testing, allowing users to test the entire experience, from download and provisioning to reporting what they see graphically through screen captures. All this is kept in context with the app's usage patterns and collected data for later debugging.

The MobileFirst Platform supports:

- Full app version control
- End-to-end traceability
- Integration with multiple versions of native

development tools, platforms, and SDKs
- Automation for builds, tests, and app store deployment
- Virtualization for server integration
- Separate build/integration areas for different platforms

The solution enables collaboration with users and team members and makes it possible to deliver great cross-platform enterprise apps efficiently and cost effectively.

The Developer View

The IBM MobileFirst Platform offers developers the following advantages:

- **An integrated set of app development** tools and frameworks, including wizard-driven user interface (UI) generators and editors, pre-packaged UI frameworks and components, and the ability to integrate your own components and favorite UI tools.
- **Flexible multi-platform support that** integrates native platform tools with a built-in IDE for cross-platform mobile app development to support multiple mobile platforms, versions, device types, and screen sizes. IBM MobileFirst offers cross-platform push notifications with server support, subscription management, and in-app upgrades across device types. A new command line interface (CLI) enables you to build apps with your preferred IDE and still use the solution's continuous delivery functionality.
- **Remote access to app logs to see how** users are using the app or to debug an issue. Remote logging and information gathering

↑ Introduction to IBM Worklight Foundation works across platforms uniformly.

- **Backend integration through a common** server interface built on standard protocols (such as JMS, SQL, HTTP/S, and REST), aggregating calls and transforming data to be mobile-efficient. Additional services are available across platforms, such as user identity and security, location services, SMS and in-app push notifications, and support for data over 3G/4G cellular connections.
- **Optimization for the expanding set of** Bluemix Mobile Cloud Services – enabling you to build high-quality, enterprise-grade apps faster and more efficiently.
- **Support for services based on public** clouds, private clouds, and on-premises equipment.
- **Enhanced operational analytics to** improve performance management of your apps across platforms.
- **Advanced security that provides** context-based risk assessments and secures interactions at every level, including the device, app, and data.

The IBM MobileFirst Platform allows you to build, run, and manage your apps on an open, integrated platform that is the most advanced for native development and the fastest for hybrid and web formats. The platform was voted the "Best Mobile Development Solution" SIIA CODiE Winner in 2013 and was named the Gartner Magic Quadrant leader for Mobile Application Development in 2013 and 2014.

For native and hybrid solutions, it helps make app store approval and updating less painful, with added support for advanced app and backend versioning, along with emergency rollbacks if needed.■

ABOUT IBM

IBM is the world's largest information technology company, with 80 years of leadership in helping clients innovate. Drawing on a breadth of capabilities and best practices from across IBM and our extensive partner ecosystem, we offer clients within every industry a wide range

of services, solutions, and technologies that can help them improve productivity, respond rapidly to the needs of their business, and reduce development and operations costs. ***www.ibm.com***

ADDITIONAL RESOURCES

Demos on demand: How to develop and deliver top-quality mobile apps

Buyer's guide to mobile application platforms

Build more robust native mobile applications with IBM Worklight

IBM Worklight compared to "do-it-yourself" mobile platforms