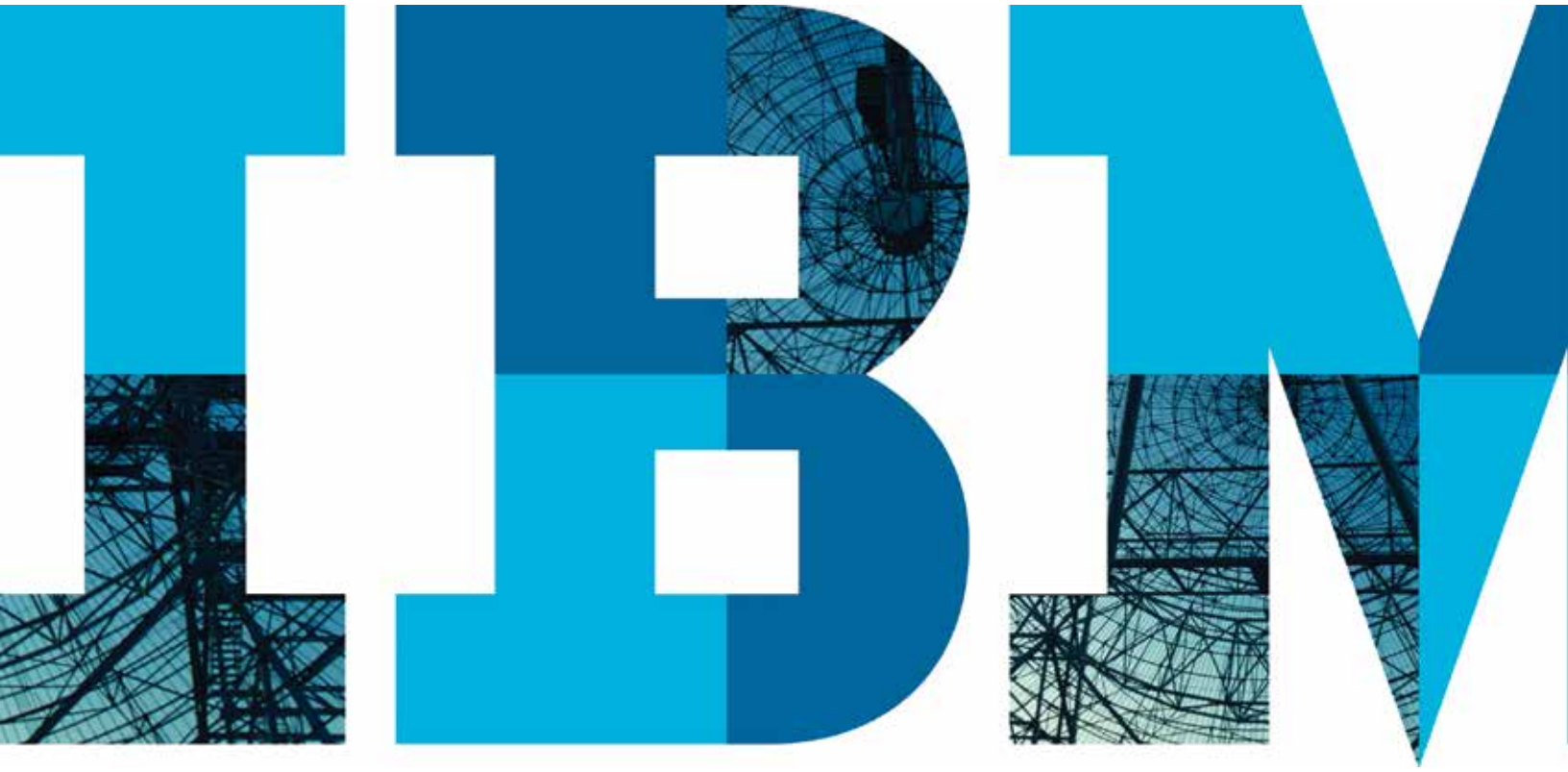
# A mobile app development primer

*A guide for enterprise teams working on mobile application projects*



IBM

## Executive summary

Industries of all kinds have begun to realize that the target audiences for their business applications (i.e., customers, employees or business partners) have shifted in massive numbers from using traditional personal computers, such as desktops and laptops, to using mobile devices, such as smart phones and tablets, for everything from shopping to working. This crucial shift in user behavior has motivated enterprises to develop mobile channels for their existing business applications, and to plan for new kinds of applications that can meet audience needs and expectations.

As with many major changes in the information technology industry, the first years see frantic activity to meet demand and create market presence without considering more strategic issues, such as application development costs, maintainability, quality and security. As the mobile application market matures and the initial rush to market settles, these more comprehensive software development issues become a priority for those responsible for longer-term planning and economics.

This primer focuses on a comprehensive approach for developing mobile business applications, covering best practices for collaborative software lifecycle management combined with newer requirements unique to the creation of mobile applications. The intent is to provide value for all roles involved in mobile enterprise application development projects: architects planning for mobile projects, development teams making implementation decisions, project managers detailing essential activities, test organizations addressing new applications, and executives needing to understand how these new mobile apps fit with existing enterprise applications and development processes. It all starts with understanding the characteristics of successful mobile apps.

## What makes for a really good mobile app?

It might not be what you think. In an IBM®-commissioned study of global consumers and business leaders, Forrester Consulting examined the characteristics and business impact of good and bad mobile apps. One of the noteworthy findings indicates there's a disconnect between mobile app development teams, and consumers and employees (i.e. users) about the qualities of a good app; this disconnect typically results in development over-spend and app under-performance[1].


Good Apps, Bad Apps: The ROI of creating exceptional mobile moments

Read the full Forrester study here.
**https://ibm.biz/CostOfBadAppReport**

What makes for a really good mobile app? The Forrester study identified three key characteristics:

- A good app serves an immediate, transactional purpose such as, checking the weather, conducting a financial transaction, looking at a map. **Context matters**.
- A good app works quickly, all of the time, and doesn't crash. **Performance rules**.
  - Users identified crashing as the most significant issue leading to a poor experience, followed by low battery life, while more typical "experience" elements such as the user interface (UI) or interaction conventions were less significant in survey results.
- A good app is secure without hindering response time, stability and performance. **Security *and* performance rule**.
  - Even though security was not rated to be of high importance to mobile users, it can be very costly and damaging to a company. Security therefore needs to be embedded without hindering response time, stability and performance.

There are downsides to not getting mobile apps right. According to the Forrester study:

- **Consumers react decisively to a bad experience *across touch points*, including:**
  - 65% would not purchase products from the company that owns/issued/sold the software.
  - 63% would tell their friend about their negative experience/impression.
  - Consumers do not seem to like getting help for resolving application issues … and their patience level is low, with 50% uninstalling the app if they have problems.
  - Generally, they will be less willing to interact with the company on its website or via social media.
- **Employees lose productivity:**
  - They get stuck and can't continue doing their work when the app doesn't work, or they spend additional time working around the app's poor performance to get their job done.
  - They do not adopt the app as a productivity tool in the first place, yielding the enterprise no productivity gains.

The good news? If you get it right, your business can benefit. According to the Forrester study:

- **Consumers respond well to a good app:**
  - 58% would purchase products from the company that owns/issued/sold the software
  - 72% would tell their friend about their positive experience/impression
  - A good app leads to more customer engagement, including using the app more frequently, recommending the app to a friend, purchasing products/services from the company, and visiting the company's social media sites.
- **Employees also respond well to a good app:**
  - With a successful app, the average increase in employee productivity was 38%, partner productivity increased an average of 40%, and the cost of a process or project was reduced an average of 36%

Good mobile apps can increase wallet share, customer loyalty, employee productivity, brand value or reputation. They are context-aware. They perform quickly. They go beyond the UI to integrate usability with security and connect back-end systems, optimized for mobile, for enhanced functionality.

## What should you consider and why?

The creation of mobile applications involves unique requirements and challenges that need to be considered before you start your mobile app development project.

### 1. Form factors and user input technology

The first and most obvious unique aspect of mobile applications is that the form factor for display and user interaction is significantly different from prior forms of software. Smart phones usually provide only a four-inch area in which to display the application content and offer lower screen resolution pixel density compared to personal computer (PC) displays, which are trending toward greater display sizes and number of screen pixels. Even tablet devices have generally lower display sizes than PCs, especially when compared to the large flat-screen displays in use for newer desktop PCs.

A smaller form factor means that the amount of data displayed to the end user, and layout of that data, needs to be different for these applications than for apps expected to run on PC devices. Significantly less data can be displayed at one time and therefore it must be exactly the "right" data, most relevant to what the user needs at that point in the application.

Another obvious physical difference for mobile applications is that the mechanisms for user input are different. Mobile devices have pioneered the use of non-keyboard "gestures" as an effective and popular method of user input. Touch, swipe, and pinch gestures must be planned for and supported in a satisfying mobile application user experience. These tactile user input mechanisms have proven to be so popular that they are now being retrofitted into traditional desktop PC systems such as the Apple "Lion" OS X release and Windows 8 "Metro" OS. In addition to tactile user input, mobile devices are a natural target for voice-based user input. In fact, the traditional keyboard typing form of user input is probably the least effective and least popular mechanism for input delivery from mobile application users.

Besides input directly from the user, mobile devices have the capability to receive input from other sources, such as geo-location input from the GPS component of the device and image information from the camera typically built into the device. These unique forms of input must be considered during mobile application design and development. They offer new and valuable mechanisms to make mobile apps more powerful and useful than applications with a more limited array of input possibilities.

### 2. Usability and user interaction design

Several factors motivate the need for more attention to usability and user interaction design for mobile applications. The first is the difference in form factors and user input methods. It is much more difficult and time consuming to plan how to display only the data that is precisely necessary than it is to simply display all possible data and let users visually sift through it for what they want. An analogy for the written word is that it is harder to write a concise abstract than it is to write an entire paper. The mobile app designer has to consider the screen real estate. When an application needs to present a broader scope of data, with multiple layers of detail, it is usually better to use a progressive discovery approach that allows the user to "drill down" into incrementally greater levels of detail that is focused on fewer specific items.

The rich variety of input methods available on mobile devices also is a motivation for early design work to identify and use more efficient ways to deliver input data–rather than the simple "just type it in a form" design that is a default for traditional web and PC applications. Extensive keyboard typing for mobile apps must be avoided in order to reduce user frustration, particularly with drastically smaller touch keyboards and lack of traditional typing feedback. Identifying non-keyboard ways in which information can be gathered and delivered to the mobile app is a significant design challenge.

There is yet another more subtle motivation for extra attention to the design effort for mobile applications. The way in which end users interact with mobile devices and the applications running on them is different from how they interact with stationary PCs and even laptops. End users of a mobile device are typically holding the device in their hand while also interacting with the surrounding reality of their physical situation. These application users typically cannot concentrate intently on the mobile app for very long before they need to switch their attention to their physical surroundings. The interaction model for users of mobile apps is short, interrupted, and "bursty," meaning that they need to very quickly complete the application task before switching attention.

All of these factors drive the need for more investment in user-centric design for mobile applications very early in the development project. Ideally, these usability considerations and design aspects should be codified in the requirements for the mobile application and then linked to the later stage development deliverables, along with the tests that validate that the user interaction and "consumeability" of the app is as satisfying as possible.

## 3. Choice of implementation technology – native, hybrid or web?

There is a spectrum of implementation choices for mobile applications in the market. There is no one perfect answer for the choice of implementation for a mobile application, and all of the choices across the spectrum have their advantages and disadvantages. Therefore, the challenge for mobile development teams is to understand the trade-offs between the technologies and make a choice based on the specific application requirements.

The choice of implementation technology for a mobile project will have an impact on other decisions related to the application's development. It may limit the choices for development tools. The implementation choice will likely have an impact on the team roles and structure. It may have an impact on how the application is tested and verified, and how it is distributed and delivered to the user. So, the choice of implementation approach for a mobile application is a crucial, early-stage decision to be made very carefully.

### Native application implementation

A "native" implementation means that you are writing the application using the programming language and programmatic interfaces exposed by the mobile operating system of a specific type of device. For example, a native implementation for an iPhone will be written using the Objective-C or Swift language and the iOS operating system Application Programming Interfaces (APIs) that Apple supplies and supports.

Native application implementation has the advantage of offering the highest fidelity with the mobile device. Since the APIs used are at a low level and are specific to the device for which the application is dedicated, the application can take full advantage of every feature and service exposed by that device.

Native implementations of mobile apps are completely non-portable to any other mobile operating system. A native Apple iOS app must be totally rewritten if it is to run on an Android device. That makes this choice a very costly way of implementing a mobile business application.

### Web applications

Newer smart phones and tablets come with advanced web browsers pre-installed, and it is very feasible to implement a mobile business application that is a standard web application, plus special style sheets to accommodate the mobile form factor and approximate the mobile device "look and feel." Mobile applications implemented using this approach support the widest variety of mobile devices, since web browser support for JavaScript and HTML5 is fairly consistent. There are several commercial and open source libraries of Web 2.0 widgets that help with this approach. The web programming model for mobile application implementation also has an advantage for enterprises that already have developers trained in the languages and techniques for web application development.

The disadvantage of pure web application implementation is that such apps have no access to functions and features that run directly on the mobile device, such as the camera and contact list. However, if your mobile application does not depend on local services running on the device, the pure web application approach could be sufficient. As the HTML5 specification matures and becomes more widely supported by mobile web browsers, many of the services local to mobile devices will become exposed for pure web applications through that W3C programming standard.

Another consideration that differs between web applications and native applications is the manner in which the application is distributed and made available on the device. Native applications must be downloaded and installed from some kind of "App Store," such as the publicly accessible Apple App Store or the Google Play Store. The app store distribution mechanism has the advantage of allowing the mobile app to easily be located using search algorithms. Enterprises sometimes appreciate the market visibility and user feedback that mobile app stores facilitate.

The downside of app stores, especially the public ones, is that they sit directly between the enterprise and its intended target audience. All mobile application updates must go through the app store, and it can be difficult to remotely control and manage the mobile app that has been delivered through app store mechanisms. Web applications aren't distributed through an app store. The end user simply enters the web address for the application into the mobile web browser, and the application is delivered over the Internet. Updating the mobile web application is as easy as updating the server or servers that host the app. The next time any user accesses the web site, the new version of the mobile app is downloaded to the device.

### Hybrid mobile application implementation

Hybrid mobile application implementation is a form of compromise between pure native implementation and pure web implementation.

You write the mobile apps using industry standard web programming languages and techniques such as HTML5 and JavaScript, but you package the app into a natively installable format that is distributed through the app store mechanism.

Hybrid apps are linked to additional native libraries that allow the app to have access to native device features from the single application code base. Because the bulk of a hybrid application is implemented using technology not unique to any single device, most of the code for the application is portable and reusable across many different mobile operating systems. However, small segments of native code also can be integrated with the hybrid app. So the developer can decide how much of the application implementation is a shared, common code base and how much is device-specific customization.

You can also choose how much of the code to package as a "native" installable app delivered through the app store and how much to download over the network. The first elements of the app to be displayed can be packaged for installation directly on the device, so they load quickly when the user launches the app. Other, more dynamic elements can be structured as web pages that are managed on a server and always provide the latest version of the application when accessed.

For the average mobile business application, many industry analysts have a strong conviction that the economics of code reuse and flexible application development will favor the compromise hybrid approach over the long term.

### How IBM supports the developer:

**IBM empowers developers working on the mobile app design, development, integration, and test with application lifecycle management tooling integrated with the IBM MobileFirst Platform Studio IDE, open-source Eclipse, or Apple Xcode. Developers working on services supporting the mobile app, either in the cloud or enterprise systems-of-record will find collaborative development support integrated in all of the Eclipse-based IBM Rational® IDEs to refactor or extend existing systems supporting these platforms – for example z (RDz), Power® (RDPower), and WebSphere® (Rational® Application Developer or Rational Software Architect).**

## Watch an Introduction to IBM's MobileFirst Platform Foundation


**https://ibm.biz/MFPOverview**

IBM's MobileFirst Platform Foundation provides an open and comprehensive platform to not only build, but test, run, manage and continuously improve mobile applications.

## Are you leveraging best practices?

The following four sections cover what should be addressed and adopted when implementing an enterprise-class mobile app–from both a software development and an integration architecture perspective.

### 1. Adopt an agile development approach

Mobile development projects typically have extremely aggressive time lines because of the strong business motivations to quickly deliver mobile applications into the market. Inception to delivery periods of a few months are common. The pressure to quickly deliver mobile apps results in the adoption of agile development methods for most successful mobile projects

There are several important elements of an agile development mobile project. Let's take a close look at each of these agile development practices.

#### Full lifecycle for the project

The lifecycle of a software development project generally follows a similar pattern, regardless of the type of software being created. It starts with the business decision, based on some analysis, to invest in the mobile app. Requirements for the app are captured and elaborated. These requirements are further decomposed into user stories and feature work items, which are assembled into a backlog of work for the iterations and releases to be completed for delivery of the app. Team members acting in various roles are assigned the work items and use various tools to complete the work and deliver whatever that work result consists of into the project. The resulting app is tested and certified to deliver the requirements. The exact process and lifecycle followed for software projects at a particular company usually are tailored to that specific enterprise's goals and policies. For instance, a bank or other regulated enterprise may have more involved and stringent development processes than a mobile game development company.

Mobile app development is generally characterized by small teams, use of existing infrastructure, and highly user-interactive applications. Agile methods and continuous testing principles are ideally suited for such a scenario. Though the specific requirements for a mobile app are likely different from some other software development, the tools and processes for gathering, elaborating and communicating those requirements are the same. The need to link the requirements to the code changes that deliver the implementation of those requirements also is the same for mobile apps as it is for other software. In other words, the flow of the project and the need for integration and traceability across the project is the same for mobile and other software development projects.

#### Integration of multiple tools

There are very few, if any, software development projects that can be delivered using one single development tool. Most projects involve a wide range of tools from different vendors, designed to meet the needs of a specific role or task in the overall lifecycle of the project.

For example, an individual developer of code for a mobile app may find that a simple code construction tool, matched to the mobile platform on which the application is to execute, will suffice for his or her needs. However, that tool is missing features that facilitate the collaboration and coordination needed when an agile development team is involved in creating the application.

By integrating the individual developer's code construction tool with a compatible collaborative team development platform, agile teams can achieve improved efficiencies and quality.

#### Need for collaboration across the development team

Mobile apps are typically created by a small team with varying skills and expertise. A typical team may consist of a couple of programmers of the fundamental business logic and web services, a couple of user interface developers, a user experience designer, a few testers, and a team leader or manager.

Mobile app projects supported across different mobile operating systems require code sharing and reuse. One developer may specialize in Android and another in iOS skills. Clear understanding of the work that team members, including designers, developers, testers, and line of business, are expected to perform, and when they need to deliver it, is essential. Project requirements, timelines, plans and so on are shared in that case whereas only source code, tests, and builds may differ.

**Integrated change management, software version control**

All code changes associated with a particular work item need to be tied together into a specific "change set" or "code push"–the list of changed source code files that are delivered in one shot so that the full code change can be tracked as a unit. Ideally, this process of assembling a change set should be as unobtrusive and seamless as possible, so it does not cause interruption of the developers' concentration on the logic they are creating.

The processes for source code version control and merge/rollback also need to be automatic and intuitive. Any time a developer has to switch their working context in order to perform some kind of task, it represents a point of interruption and a potential speed bump in the development process.

**Need for traceability across the project**

The typical agile team approach to software development is to define multiple short iterations in which a small set of application enhancements are to be implemented and validated. A typical agile iteration is from two to four weeks long. The team leader can work with the team to map work items from a backlog list into the specific iterations and assign the work items to individual developers.

As the developers pick up the work items and begin to make progress on them, their effort needs to be automatically recorded and made available to the team leader to track and view. Everyone on the team needs to be able to see how the iteration is progressing and the status of the work items planned for that iteration.

When the testers on the team start the functional testing of the mobile application, they need to open work items in the shared development project for defects uncovered during the course of testing. If the test case that failed is linked to a particular change set or feature item in the project plan, then the information about the code that was changed, and is likely to be associated with the test case failure, can automatically be entered into the defect data. Furthermore, if the change set is linked to the original requirement that motivated the code change, there is traceability throughout the whole project lifecycle, from the original requirement to the test case that verified that the requirement was delivered in the application.

This kind of "whole project view" and end-to-end traceability is extremely important for any kind of software development project, but especially relevant to mobile app development teams working on tight schedules and employing agile development methods. These kinds of lean development teams cannot afford to spend time tracking down details about whether and when a particular requirement was verified and delivered.

## 2. Plan for continuous software delivery of mobile apps–the importance of DevOps in mobile development

Providing an overall high quality mobile application experience is highly dependent on a continuous software delivery approach that goes beyond just the code construction. It's not sufficient anymore to outsource or build the mobile app once and then let it stagnate–mobile apps require not only an initial operational capability, but also an ongoing, agile evolution as user and business needs evolve, competitive apps are released, user interface innovations occur, the underlying platform adds more features, and other information sources become available for integration including Big Data, social collaboration, systems of record, and managed APIs.

The reality of adding more agile development capabilities to your team is that it's very similar to adding functional capabilities to your application: you sprint your way there and add more capabilities as you progress, evolving the team's capabilities over the life span of the project. Attempting to add all of these capabilities at once however, would be too disruptive for any team as new technologies often involve a learning curve. A **DevOps Assessment Workshop** can provide guidance for your actual implementation of capabilities that match your business needs and financial justification.

Even though there are unique aspects to mobile application development, many of the roles and tasks involved in the overall development lifecycle are the same as for enterprise-class development of other kinds of software (so-called Systems of Record). In the paper "Measuring Agility and Architectural Integrity,"[2] Walker Royce describes the key techniques and practices for effectively delivering software using agile and continuous-test principles. The software delivery practices in Walker's paper are a perfect fit for mobile development projects.

DevOps is an enterprise capability for continuous software delivery of applications, including mobile applications. An Institute of Business Value study reveals that 86% of companies believe software delivery is important or critical, yet only 25% leverage software delivery effectively today, and nearly 70% of those that do, outperform their competitors who don't leverage this capability. DevOps for mobile focuses on four key capabilities:

- *Collaborative development* brings business, development, and quality assurance teams together so they can continuously adjust the project's business goals based on customer feedback. They stay closely aligned throughout the project lifecycle to gain faster delivery of customer value.
- *Continuous testing and quality management* is a critical part of the handoff between development and IT operations. Successful Agile teams using DevOps practices fully integrate testing into their software development lifecycle. Continuous testing provides project teams with continuous feedback and enables them to test earlier, with greater coverage removing traditional testing bottlenecks such as unavailable test environments.
- *Continuous release and deployment* practices help to drive down costs and reduce waste by eliminating errors and rework with an accelerated time to market using automated, transparent, repeatable deployment processes.
- *Continuous monitoring, customer feedback and optimization* practices help enterprise businesses manage complex and interdependent applications, providing faster diagnosis of problems with reduction in both outage costs and risk to the business as well as achieve higher service levels by providing visibility to the end user experience, transaction path analyses, diagnostic capabilities and predictive analytics.

### How IBM delivers DevOps capabilities to enterprise mobile:

IBM DevOps provide organizations the ability to innovate faster by accelerating software delivery across the enterprise, shorten feedback loops by making delivery process fast and efficient, and create a culture of continuous improvement and trust.

*Collaborative Development:* Collaboration between agile team members, working across the multiple tiers of a mobile app are supported through IBM DevOps Services for Bluemix and IBM Rational Team Concert.

*Continuous Test and Quality Management:* The IBM MobileFirst Platform provides automated functional testing as well as manual testing capabilities to support rapid delivery of mobile apps in response to feedback. End-to-end testing of the mobile app is also supported by IBM software quality assurance offerings.

*Continuous Release and Deployment:* Builds, tests, and deployments can all be automated in workflows which enable rapid delivery of quality mobile apps and integrated cloud and enterprise service components. IBM UrbanCode Deploy provides these capabilities and integrates with many open source tools that are found in the market today. It also adds additional value with integration to the IBM MobileFirst Platform to streamline the deployment of mobile apps into the MobileFirst Application Center.

*Continuous monitoring, customer feedback and optimization:* IBM MobileFirst Quality Assurance, part of the IBM MobileFirst Platform, delivers rapid insight into user experience through in-app feedback and crash reporting, as well as analytics applied to the public app store ratings and reviews. IBM® Tealeaf® CX Mobile enables companies to apply Tealeaf's powerful customer experience management solutions to their mobile websites, native applications and hybrid applications, including support for HTML5 and responsive web design (RWD).


https://ibm.biz/MQA4BluemixStepByStep

Watch a demonstration of the IBM MobileFirst Quality Assurance capabilities. Discover the power of user-centric testing and rapid feedback to deliver high quality mobile apps.

### 3. Integrate secure mobile development practices in each stage of the development lifecycle

Mobile application design and development should be performed with a security conscience, and testing teams need to automate their security-specific unit tests early and often during the development lifecycle. This process will help avoid discovery of critical security issues later in the development process and avoid delay of the mobile app to the market.

To meet the quality of services for a secured mobile app workload, IBM has developed a comprehensive portfolio of end-to-end mobile security solutions that address four key mobile security challenges and can integrate with the mobile back-end service security and its quality of services:

- *Device security* is challenging because of the diverse set of devices spanning from corporate-owned assets to employee-owned devices (BYOD).
- *Content security* focuses on the information stored on the device that might be corporate-owned or confidential data that needs additional protection.
- *Application security* looks at what is needed to integrate security by design into the application development approach, from concept through delivery.
- *Transaction security* ensures that a mobile transaction receives end-to-end security, for every transaction, starting with user authentication, identity federation, fine-granular authorization, to logging and auditing.

## What IBM offers to support mobile security

*Device Security*: With Fiberlink® MaaS360® Enterprise Mobility Management, IBM delivers a solution to enroll, provision and configure devices and enables unique identification of mobile devices and location, locking and wiping remotely if lost or stolen.

*Content Security:* With Fiberlink MaaS360, IBM provides solutions to securely control corporate mail, calendar and contacts as well as intranet sites and attachments to prevent data leakage.

*Application Security:* IBM MobileFirst Platform Application Scanning is optimized to scan the mobile app source code for vulnerabilities and can be integrated early into the process. IBM AppScan delivers other components to help assess existing web sites and runtime web services for vulnerabilities. The IBM MobileFirst Platform Foundation is a Mobile Enterprise Application Platform (MEAP) provides functionality for safeguarding mobile apps at the device, application and network layer. Furthermore it forms a central interface to govern the mobile app portfolio across all mobile ecosystems.

*Transaction security:* IBM Security Access Manager for Mobile provides mobile access security protection in a modular package. The layers of transactional defense can be established by adding run-time protection to secure applications against hacking attacks and malware exploits.

Security Intelligence: The security intelligence layer represents the unification of the security concepts for mobile workloads end-to-end. IBM QRadar® Security Intelligence provides a unified architecture for integrating mobile security intelligence, log management, anomaly detection and vulnerability management.

**IBM MobileFirst Application Scanning**


https://ibm.biz/AppScanMobile

Identify mobile application security vulnerabilities and maximize the effectiveness of security reports that are generated by the service, permitting you to remediate security vulnerabilities in your mobile applications quickly and efficiently.

### 4. Focus on mobile data management and integration

Not long ago, a thousand daily users of an application was a lot, while ten thousand was an extreme case. Today, nearly 3 billion people are connected to the internet and the amount of time they spend online–about 35 billion hours a month in 2014–is steadily growing, creating an explosion in the number of concurrent users.

It's not uncommon for apps to have millions of different users a day and be required to support global users 24 hours a day, 365 days a year. With the rapidly growing population of online users, it is not only important to support concurrent users but paramount to dynamically respond to rapid fluctuations in the user base. These fluctuations can be caused by a variety of reasons, including an app going viral overnight causing a spike in users from a few to a million overnight. In addition, the growth of internet connected devices (the Internet of Things) will further drive scalability requirements.

What does this mean? Databases need to be scalable to dynamically handle increasing volumes and velocity of data. They also need to be flexible to process varying data formats generated from a variety of sources.

With relational technologies, many application developers find it difficult, or even impossible, to get the dynamic scalability and level of scale they need while also maintaining the performance that users demand.

**Enter NoSql Technology**

A "NoSql" or "Not Only Sql" database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

The benefits of NoSql are:

- The schema-less structure and horizontal scaling capability make it ideal for dynamically handling large data volumes and a variety of data formats.
- Traditional database applications take a long time due to schema rigidity–generally meaning inflexibility to data structure changes–and mismatch between programming language and the row-column model of data. Using the simplicity of NoSQL, customers are able to increase coding velocity for fast and iterative application construction.

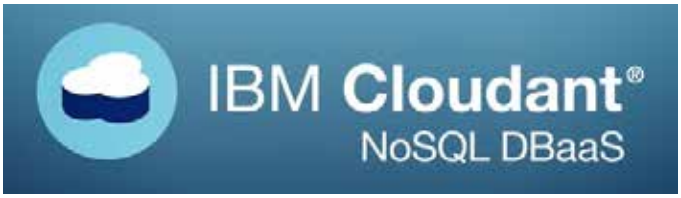
---

### What IBM offers to support data integration

**IBM Cloudant® is a fully managed NoSQL database-as-a-service that takes on the administrative tasks of managing the database, allowing the enterprise to focus on managing the data. These are some of the key benefits:**

- **Manage data anywhere, anytime; on the cloud or on premise, offline on the device or in sync with the backend system**
- **Securely store and retrieve enterprise data**
- **Scale horizontally and geographically**
- **Rapidly develop apps with a simple programming model and rich feature set**
- **Developers can quickly get started with native mobile APIs that support off-line synchronization, advanced queries, secure integration, and analytics.**

**IBM Cloudant can be used with the growing list of services available in IBM Bluemix as well as enterprise services.**




**https://ibm.biz/CloudantNoSQL**

**Watch an overview of IBM Cloudant, NoSQL DBaaS. Focus on your app, not on the database!**

---

## Conclusion

As more and more enterprises in all industries realize the need for mobile versions of their business applications, there is a need for an enterprise-class approach to mobile app development. IBM has established such an approach.

The IBM approach to mobile enterprise application development combines years of experience in the field of general enterprise software development processes with new tools and techniques that are specific to mobile devices and their underlying software foundations. IBM itself has gone through a transformation to employ agile methods across thousands of projects involving tens of thousands of developers.

The IBM approach to mobile application development emphasizes five key themes:

- Simplify the mobile app user experience.
- Integrate first for improved economic governance of the mobile app project.
- Enact ultra-agile methods supported by collaborative, integrated and automated tooling.
- Capture and respond rapidly to user feedback.
- Leverage a mobile app development platform to deliver the highest quality, secure mobile apps, integrated with enterprise and cloud services.

This approach enables mobile specific tools and technology to be used with the same efficiency and rigor as other kinds of enterprise application development.

## For more information

As part of your research into ways to develop and deliver top-quality mobile applications, you may find the following resources helpful:

- **IBM MobileFirst Platform:** Learn more about the long-term advantages of using an integrated environment for mobile app development: **ibm.com**/mobilefirstplatform
- **IBM Bluemix Cloud Services:** Learn how to create apps with greater efficiency using IBM Bluemix Services, including DevOps, Mobile, and Cloud https://www.bluemix.net/
- **For Developers:** Here's your one-stop resource for IBM Mobile Development information and resources: **ibm.com**/developerworks/mobile/
- **The Mobile Frontier Blog:** Keep up on concepts and observations related to the mobile evolution in the computing world. **ibm.com**/developerworks/community/blogs/mobileblog/?lang=en
- **IBM solutions for mobile application development:** please contact your IBM sales representative or IBM Business Partner, or visit the following website: **ibm.com**/software/solutions/mobile-enterprise/

IBM Global Financing can help you acquire the IT solutions that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize an IT financing solution to suit your business goals, enable effective cash management, and improve your total cost of ownership. IBM Global Financing is your smartest choice to fund critical IT investments and propel your business forward. For more information, visit: ibm.com/financing

## About the authors

Leigh Williamson is an IBM Distinguished Engineer who has been working in the Austin, Texas lab since 1988, contributing to IBM's major software projects including OS/2, DB2®, AIX®, OpenDoc, Java, Component Broker, and WebSphere Application Server. He is currently a member of the IBM Rational Software Chief Technology Officer team, influencing the strategic direction for products in the Rational brand and leading the solution definition for mobile application development. Leigh holds a BS degree in Computer Science from Nova University and a Masters degree in Computer Engineering from the University of Texas.

Roger Snook is an IBM Certified Expert IT Specialist innovating in the Washington DC Metro area for over 20 years with both Federal and Commercial clients. Roger is an OMG Certified UML Professional and has been a featured global speaker on DevOps for Mobile, SOA, Design, Rational software topics. Roger holds a BS degree in Computer Science from Rensselaer Polytechnic Institute, Troy, New York.



IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
April 2015



[1] Source: "IBM Mobile App Study," a commissioned study conducted by Forrester Consulting on behalf of IBM, September, 2014. Web. http://www.ibm.com/mobilefirst/us/en/good-apps-bad-apps.html

[2] Walker Royce, "Measuring Agility and Architectural Integrity" whitepaper, www.ijsi.org/ijsi/ch/reader/view_abstract.aspx?file_no=i92

[3] "Five imperatives for application lifecycle management." Carolyn Pampino. 2011. https://jazz.net/library/article/637

Please Recycle

RAW14302-USEN-01