MOBILEQ™

The Freedom to Do Business

Debunking Mobile Application Myths

The Real Issue Is User Experience

Summary

Whether it's a small business making its order-entry system available to sales people in the field, a major financial institution making account data available to customers, or a telecommunications service provider offering personalized online services, every company needs to tackle the challenge of mobilizing its applications.

Mobile application technology is new and not widely understood. The marketplace is awash with mythology that keeps most companies from investing in strategies that will be viable for the long term.

This paper identifies and debunks the key myths of mobile application development, explains the critical issues, and provides a framework for selecting appropriate products and development strategies.

The stakes are enormous: successful deployment of mobile information systems is widely regarded as a key competitive advantage. However, as some early adopters have already learned, the key to success is a focus on user experience and constant updates to respond to evolving user needs. A solution that doesn't place an emphasis on usability will simply not be used.

Table of Contents

Mobile Applications

As information and communication technology continue to evolve and emerge they become increasingly personal and increasingly pervasive. The history of technology reflects an ongoing struggle to develop products and services that more closely reflect the unique needs of different individuals. This personalization of technology is now giving rise to two key trends that go hand-in-hand:

- Mobile Applications
- User Experience Development

Mobile applications can be used whenever and wherever necessary, using whatever device may be at hand.

For applications to adapt to the needs of their users, they must be instantly available and always at hand. Traditional application development was focused on the devices—typically personal computers and desktop computer terminals—that were used to run them. This meant that these applications were accessible only when users were sitting in front of the device in question. On the other hand, people by their very nature are always on the move. Few people remain fixed to their desks throughout a workday, and everyone's day involves moving between home, work, and other places. The growing diversity of computing devices and networking technology allows applications to be delivered anywhere. Mobile applications, or "applications anywhere," can be used by an individual whenever and wherever necessary using whatever device may be at hand.

One of the most important differentiators of technology products is the user experience they offer. This experience applies to both hardware and software. The shape and size of a device, the ease with which it can be held and used, the number, location, and type of its inputs, the media types it supports for output, and its responsiveness are all factors that determine if a particular device will be useful to a given individual. Similarly, the user experience associated with an application is determined by such things as how information is presented, how commands are entered, how intuitive and easy to learn it is, and how quickly it performs its tasks. User experience development involves tailoring both hardware and software products to the requirements of individual users. Technology has not reached a level where we are able to deploy significant computing and communications power in inexpensive devices that surround people throughout their daily lives. Vendors are learning that to be successful in producing both new devices and applications, they must invest heavily in providing the best possible user experience; this is fundamental to having their product both accepted and used.

User experience platforms for mobile applications are development tools and corresponding deployment servers that allow organizations to mobilize their applications. They include tools for designing, integrating, managing, and evolving the user experience for mobile applications along with software for deploying, securing, and administering the result.

MYTH # 1: Wireless = Mobile

The terms mobile and wireless have been used interchangeably in the popular press to describe applications operating on wireless devices. But in reality, there are important differences between the two terms.

Mobile applications are available to a user regardless of where he or she travels. Mobile applications adapt to work on different devices at different times and in different places. Wireless applications, on the other hand, are developed strictly for use with wireless devices, and are therefore a subset of mobile applications.

The key difference between mobile and wireless applications, then, is that mobile applications are designed with the many changing needs of the mobile user in mind, whereas wireless application development focuses narrowly on a wireless network and some combination of wireless devices.

Defining Mobility

Mobility refers not to the device, but to the individual who is mobile. In this context, it refers to the ability for an individual to access an application from a variety of locations. Mobility revolves around people being mobile.

Mobility is about supporting the computing needs of people regardless of where they are or what device they are using.

Deploying wireless devices that can be carried everywhere an individual goes is just one approach to mobility. However, an equally important aspect of mobility involves deploying an application on a variety of devices so that users have the technology at their fingertips regardless of where they travel.

The conventional paradigm for information technology has reflected the corresponding assumption that information and applications are only used while the user is seated at a desk. Mobility is about supporting the computing needs of people regardless of where they are. Throughout a person's day he or she interacts with an increasing number of computing devices from home and office PCs, to wireless pagers, mobile telephone, and laptop computers, and eventually to TV set-top boxes, dashboard-based car computers, and interactive kitchen appliances. At any time they might want to access personal or corporate data and applications. Mobility is about allowing people to use any application on whatever device they have at hand.

Consider the following scenarios:

- Rebecca the Roadwarrior
  Rebecca is a classic "roadwarrior" type. She is a salesperson who is on the road all day everyday. Rebecca's company has deployed a sales force automation (SFA) system to enhance the productivity, effectiveness, and professionalism of its sales force. Rebecca carries both a PDA and a notebook computer on trips and uses a desktop PC on her visits to the home office. She uses the PDA on the street, in taxis, and during meetings. The notebook computer is used in the hotel and on the airplane.

The available bandwidth for connecting to the mobilized SFA varies between locations and between devices. The attributes of Rebecca's devices—such as performance, display size, keyboard (or lack thereof), weight, and power requirements—vary dramatically. The combination of device and networking attributes determine when and how Rebecca uses each of these devices.

- **Tim the Technical Support Consultant**
  Tim the Technical Support Consultant illustrates the popular vision of mobile applications: an individual with a single wireless device.

  Tim is one of several dozen IT consultants employed by a large Wall Street institution to provide onsite support for computer hardware and software at the bank's various Manhattan locations. If an executive or trader is having computer problems, Tim's job is to get the problem resolved as quickly as possible. Employees with problems call the company's help desk, and if the help desk consultant can't resolve the problem over the telephone, a new trouble ticket is created in the help desk database. This ticket triggers a notification message to be sent to the mobile telephones carried by all the roving consultants.

  Tim can access the trouble ticket using the text browser built into his telephone, or he can place a phone call to the system. In text mode, he interacts with the application using buttons on the phone. In voice mode, the system speaks to him using text-to-speech technology (TTS), and he can respond by speaking to the system's automatic speech recognition (ASR) technology. Using the application, Tim can review the details of each trouble ticket, accept or pass on it, and get the exact physical location of the affected computer.

- **Emmet the Executive**
  Emmet, the Purchasing VP for a Silicon Valley-based computer company, illustrates the future of mobile enterprise application use. He is always on the go, traveling between business meetings with small companies around the valley and around the world. He spends as much time in an airplane as he does in meeting rooms, and his weekends are typically spent traveling to or from meetings or catching up on work in a hotel room. As he's rarely at the office, he relies completely on the company's mobilized virtual assistant application, which frees him from the need to travel with a heavy laptop computer.

  On this particular occasion in Paris, his day starts at 6:45am—the time scheduled for his wake-up in his online calendar. The virtual assistant calls his hotel at the prescribed time and reads him the emails that have arrived overnight.

  After his first meeting of the day, he drops by the company's offices in Paris and uses a PC to check his email and any changes to his calendar through the virtual assistant application. During his lunch meeting, he receives a text message on his GSM mobile phone indicating that one of the company's suppliers has been unable to deliver some needed components and a new source must be found. Using the browser built-into his telephone, he searches the supplier database back at the office for alternate suppliers. After finding a prospect, he pulls up the contact information and uses it to trigger a phone call. The supplier is interested and indicates that they can move quickly. They agree on a price, quantity, and schedule, and Emmet promises to send a purchase order.

Emmet's next stop is the Airport. While waiting for the next flight to London, he uses a kiosk in the airport to access the purchasing system, creates the purchase order, and routes it for approval. Once approved by the COO it will be automatically faxed to the supplier.

That night in London, Emmet uses the set-top box in his hotel room to review his schedule for the following day and sets his wake-up call time.

Mobility solutions must support the mobile computing needs of the enterprise, its partners, and its customers.

Mobility is about supporting the widely varying needs of many different kinds of users as they move from place to place and from task to task throughout the day. However, mobility is also about supporting the mobile computing needs of the enterprise as a whole, as well as the enterprise's partners and customers.

Consider the following scenario:

- Silicon Chips

  The Silicon Chip Company has just received information from a supplier that they will not be able to provide a key component of Silicon's Hyperion chip set for another ten days. The result of this delay is that Hyperion chips will not be shipped for the next two weeks; fortunately, Silicon has extra supply of the Tsunami chips that are of comparable functionality to the Hyperion.

  The production manager promptly sends an email to the sales group, three of whom have major orders planned for delivery over the next two weeks. While driving to Austin, Relationship Manager Jane receives the email using her mobile PIM, and immediately uses her Palm device to check her customers' orders. She notices that two customers are expecting delivery over the next two weeks, and immediately calls them and offers to change their Hyperion order to Tsunami chips. Customer A agrees, and Jane uses her Palm to place the order wirelessly, which immediately goes to Silicon's fulfillment department. Customer B doesn't want Tsunami, but asks if Silicon has any higher end chips that they could integrate instead. Jane looks at customer B's last three orders and notices that they usually purchase three other types of chips. She then searches the current inventory levels for these three products and finds that Silicon has sufficient supply of two of them. Jane secures an offer to split the order between the available high-end chips. She then fills the order on her PDA and sends a confirmation form and thank you notes to her customers for their flexibility.

  After being notified, Silicon's other traveling sales representatives use their RIM devices, phones, and wirelessly connected palmtops to check the supply chain information for four other customers, to ensure that they have adequate levels of Hyperion chips. They then follow up with the appropriate purchasing managers to fulfill orders with the Tsunami chips.

  Silicon's high performance sales team was able to respond immediately to the Hyperion chip set shortage. The sales reps use their mobile applications to maintain their customer relationships and prevent customers from switching suppliers. The resulting customer orders total $4,500,000; they are saved in just under twenty minutes by Silicon's mobile employees.

Mobility applications enable enterprise employees and supply chain partners to respond rapidly and effectively to unforeseen situations, access critical information whenever and wherever they need it, and turn potential problems into opportunities to strengthen customer relationships.

MYTH #2: Mobile Applications = Web Applications

The universal adoption of HTML has driven the proliferation of PC-based Web content. Because of this, many organizations have made a significant investment in Web servers and Web-oriented application architectures. It's no surprise that most would like to believe that the application architectures they have created for PC-based Web applications are sufficient to support the demands of mobile users.

In truth, the issues surrounding mobility are a superset of the issues surrounding Web applications. Effective mobility requires applications that adapt to the mobile user's changing circumstances, including such factors as the type of device he or she is using and his or her physical location. Web applications, on the other hand, are designed to interact with users through a limited range of well-defined browsers running on personal computers. Thus, mobility solutions must build upon the architectures of Web applications in order to deliver a fully optimized user experience.

Mobile Application Architecture

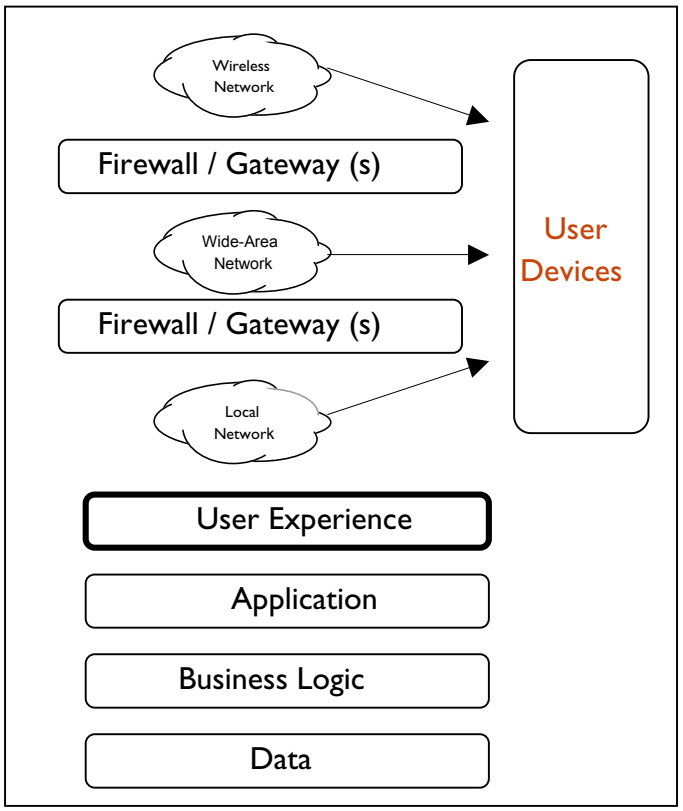
The architecture of a mobile application is open, multi-tiered, and distributed across one or more networks.

Developing and deploying mobile applications involves a number of specific functions that are distributed across one or more interconnected networks.

A generalized mobile application consists of the following components organized into tiers:

- A data warehouse tier where application data is stored
- A business logic tier that enforces business roles on access and manipulation of the data
- An application logic tier that drives the application logic
- A user experience tier that creates a customized user experience for each individual user on a given device
- User devices that present the user experience to the user
- Network, gateways, and firewalls that interconnect the devices and application servers

Figure 1. Multi-Tier Architecture for Mobile Applications



User Experience Development

The user experience layer adapts an application to the immediate needs of a mobile user.

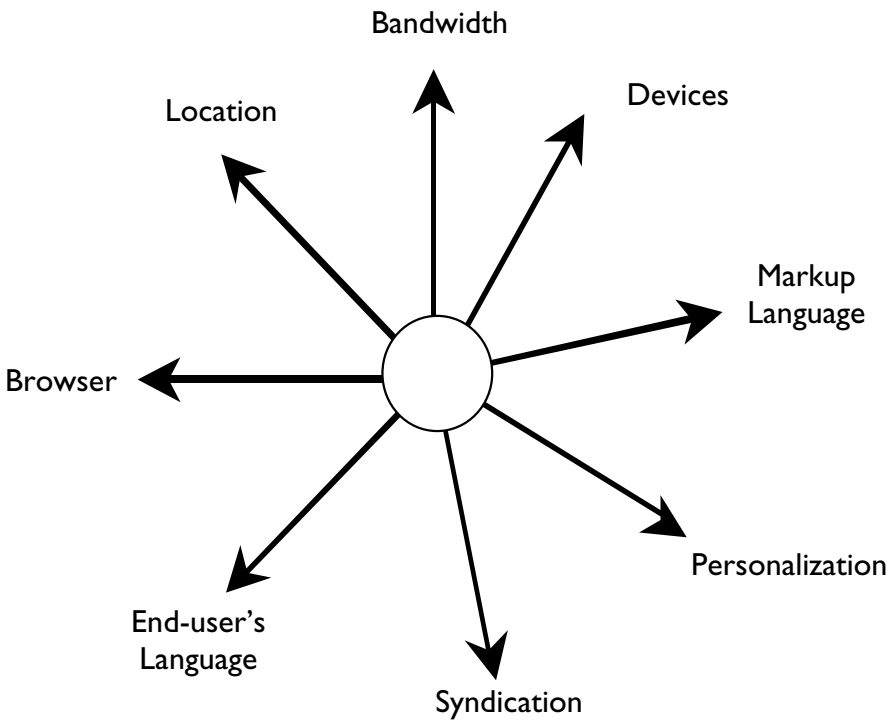
In traditional application development, the destination device was always assumed, and so the user experience component was trivialized as a thin presentation layer.

In contrast, user experience is at the heart of a mobile application. Mobilizing an existing application, or creating a new mobile application, involves separating the elements that determine the application's user experience from the rest of the application components and implementing a comprehensive user experience layer that adapts the application to the specific circumstances of each individual user.

Delivering an optimal user experience for an application involves adapting the experience based on a number of factors that are specific to each and every contact with the application. The role of the user experience layer is to adapt applications to the instantaneous requirements of a given user by taking into consideration information such as:

- The attributes of the device being used
- The client software (browser) used on the device
- The markup languages support on the device
- The bandwidth available to the device at that instant
- The user's preferred language
- The user's personal preferences
- The physical location of the user
- The current date and time

Figure 2. Adapting the User Experience of Mobile Applications



When a user invokes an application, the user experience tier is responsible for identifying these parameters and presenting a user experience optimized for this unique context. For example, devices with jog-dial control or a mouse allow fast navigation through selection lists; devices with just a numeric keypad may be better off with a numbered menu. Low bandwidth networks may make interaction very slow, but sending multiple parts of a form to the user's device all at once alleviates this problem. In each case, the user experience layer adapts the application's behavior and presentation to ensure an optimal user experience.

Mobile applications rely on the user experience layer to optimize user experience based on all of the parameters describing a user's instantaneous context. This is not as simple as merely filtering or transcoding the output of the application layer, as the user experience layer is responsible for actually determining the application's dialog with the user.

The user experience layer creates different user experiences based on location, language, device capabilities, personal preferences, and bandwidth:

Consider a mobile application that presents a corporate telephone directory. The user experience layer could create different user experiences depending upon such factors as location, user's language, device capabilities, personal preferences, and bandwidth:

- A user with a PC on the company's internal LAN would be presented with the first page of the worldwide corporate directory with columns for office location, first name, last name, phone number, and employee picture. Buttons on the page would allow the user to flip forwards and backwards, and sort on any column. Text fields at the top of each column would allow searching and filtering based on full or partial text matches. The columns would initially be sorted based on the user's preferences.
- An English speaking employee with a PDA visiting the company's Munich office and accessing the application using a wireless LAN would see a column of names for the people working in the Munich office with a button to invoke a search of the worldwide directory. Selecting a name would bring up a form with the complete entry for that person, including their picture. Clicking the search button would present a blank form (with English labels) that could be used to bring up a new list of matching item.
- A German-speaking employee at the company's Munich office who is accessing the application using a mobile phone with WAP browser would be presented with a search form (in German) with fields for first name, last name and location. The location would default to "München," and after entering a whole or partial name, the name, phone number, and location of the best five matches would be returned.
- A French-speaking employee using a payphone in Osaka would access the application through a voice interface. The application would begin by asking, in French, if the employee was searching the worldwide directory or just the directory of the Tokyo office. The application would then prompt for the first name and last name and would iterate until the desired directory entry was located.

As these examples illustrate, the user experience layer doesn't simply determine what data is displayed and how it is presented; it also determines how the application interacts with the user.

The goal is to build mobile applications that are comfortable, convenient, and intuitive—the metrics for optimal user experience.

MYTH #3: Markup Languages Define the Application

Markup languages are merely the means for expressing how information is to be presented. Markup languages such HTML, HDML, WML, and VoiceXML simply describe how particular pieces of information are to be handled by a user's device.

Markup languages do not define the scope or nature of a mobile application. Nor do they define an industry or a market category.

Technical Challenges in Building the User Experience Layer

Application developers working on the user experience layer of a mobile application face some significant technical challenges:

- Device, markup language, and client software proliferation
- Network, database, and application integration
- Scalability
- Security
- Push and pull
- Iterative development

Mobile application developers must pay careful attention to all of these issues when opting for a particular development strategy or development tool.

Device, Markup Language, and Client Software Proliferation

Mobile solutions must deliver content and applications that leverage the unique features and capabilities of an ever-increasing number of devices.

While it is important to develop applications in a manner such that supporting new devices incurs no incremental cost, developing in a manner that completely disregards the unique features of particular devices is no solution. Mobility is not just about rendering the same content in the same way to every supported device. Rather, due to varying form factors such as screen size, display, and input mechanisms, successful mobile applications must be able to deliver content and applications that leverage the unique features and capabilities of each device.

- Diverging devices and functionality —There is already a plethora of different devices and different form factors on the market, and the variety and number are only expected to increase over the next few years. Some people estimate that the device lifecycle is currently between nine months and one year. As wireless bandwidth increases, this short lifecycle will only exacerbate the diversity of devices used as vendors add device functionality to leverage these new opportunities.
- Markup language diversity — Varying standards also add to the complexity, including WML, HTML, HDML, XML, and VoiceXML.
- Client software proliferation —While traditional Internet applications had to comply with two predominant browsers, the number of mobile software implementations that must be supported today is manifold and increasing. In addition to determining the markup language that must be used, different microbrowsers also render the same markup language in different ways, and one device may also have multiple browsers. This diversity is driven by the need for device vendors to satisfy the unique needs of different customer segments and to differentiate their products with innovative new capabilities.

Those who want to deploy viable solutions must develop mobile applications that can exploit the unique richness and functionalities built into existing and future devices. They must also integrate them into a single solution that is comfortable and intuitive to use. The variety among target devices can only continue to increase, as vendors are increasingly able to optimize their products for different types of users. Using conventional software platforms (e.g. Web servers) and ensuring multi-device support require the creation of dozens of different applications, which is too time-intensive for most organizations to sustain.

Network, Database, and Application Integration

The user experience layer must enable enterprises to preserve their existing infrastructure and to extend its productivity and functionality.

The user experience layer sits between user devices and backend application components such as data-warehouses, business logic servers, and application servers. Implementation of the user experience components must be independent of, and yet support tight integration with, all of these backend components.

In considering whether to mobilize their applications, enterprises and service providers are particularly concerned about the level of modification required to their existing applications and data sources. Many enterprises are hesitant to introduce mobility if it means abandoning or re-creating their already sizable information technology investments.

The architecture of the user experience layer must support linking to existing and anticipated systems, enabling a system owner to preserve their existing investment, and to extend the productivity and functionality of their existing infrastructure to mobile applications. This future-proofing includes providing hooks for dramatically different data forms and types depending upon the age of the underlying application being mobilized. In addition, different types of enterprises and service providers have adopted different operating system and database platforms, which require unique connectors.

As the functionality of networks and user devices increases, implementations must facilitate the addition, modification, or replacement of applications, networks, and databases. Consequently, as enterprises and service providers adopt new platforms, data sources, applications, and networks, the user experience layer of the mobile applications must accommodate these improvements and changes.

Scalability

The user experience layer must be designed to support growth without creating management headaches.

As utilization of new user devices and networks grows, the load on mobile application implementations will also grow rapidly. This software must be designed and written so that it can be scaled from a small pilot program to the point where every conceivable user is interacting with it perpetually.

The user experience layer must be designed to add users efficiently and effectively and to support significant growth in users without additional management or deployment overhead. The only limit to scalability should be the number of servers in the server farm and the number of processors running each component of the mobile application. Furthermore, the mobile application's architecture should support off-the-shelf hardware or software load-balancing technology in order to simplify integration into existing network operations centers.

Security

The user experience layer must be part on an end-to-end security solution.

With increasing levels of personalization and distributed use over public wired and wireless networks, mobile application implementations must be particularly concerned with securing every aspect of application implementation. As the user experience layer represents the interface between user devices and the mobile application, security considerations play a key role in its design. Mobile applications must not only offer high levels of usability to be effective, but they must also be trusted by users, enterprises, and service providers to access sensitive corporate data or e-commerce services.

The mobile application, through the user interface layer, must secure the data traveling throughout the communication and application process from the user, through the network, all the way to the back-end systems.

Authentication and transport level security are not the only elements of the overall security challenge. Application level security is also necessary to prevent reverse engineering of applications through their URLs and Web forms.

An overall security framework based on threat evaluation, risk analysis, policy creation, address authentication, authorization, privacy, integrity and non-repudiation is required of any mobile application to be deployed on a publicly accessible network.

Push and Pull

The traditional Internet paradigm consists of users actively searching for and retrieving information. These actions are greatly facilitated by Web browsers and search engines. On mobile devices, active surfing is very time consuming because of the limitations of networks and device form factors (e.g. constrained displays and input mechanisms). It is difficult both to enter information and to navigate. In short, surfing on mobile devices generally constitutes a poor user experience

One solution has been to send notifications and alerts to end users based upon rules that are either automatic or user-specified. But receiving information only fulfills half of most users' needs; users also need to take action based on the information that is "pushed" to them. With most applications, however, they still need to surf and navigate to take action. What if users could take immediate action based on the alert or notification? The users could receive a notification linked to a form that enabled them to take action instantly. That is the essence of a push and pull application. Mobile application developers need to provide an infrastructure that supports both push and pull of content and data.

Iterative Development

The user experience layer must enable an iterative approach to application development that rapidly incorporates user requirements into successive releases.

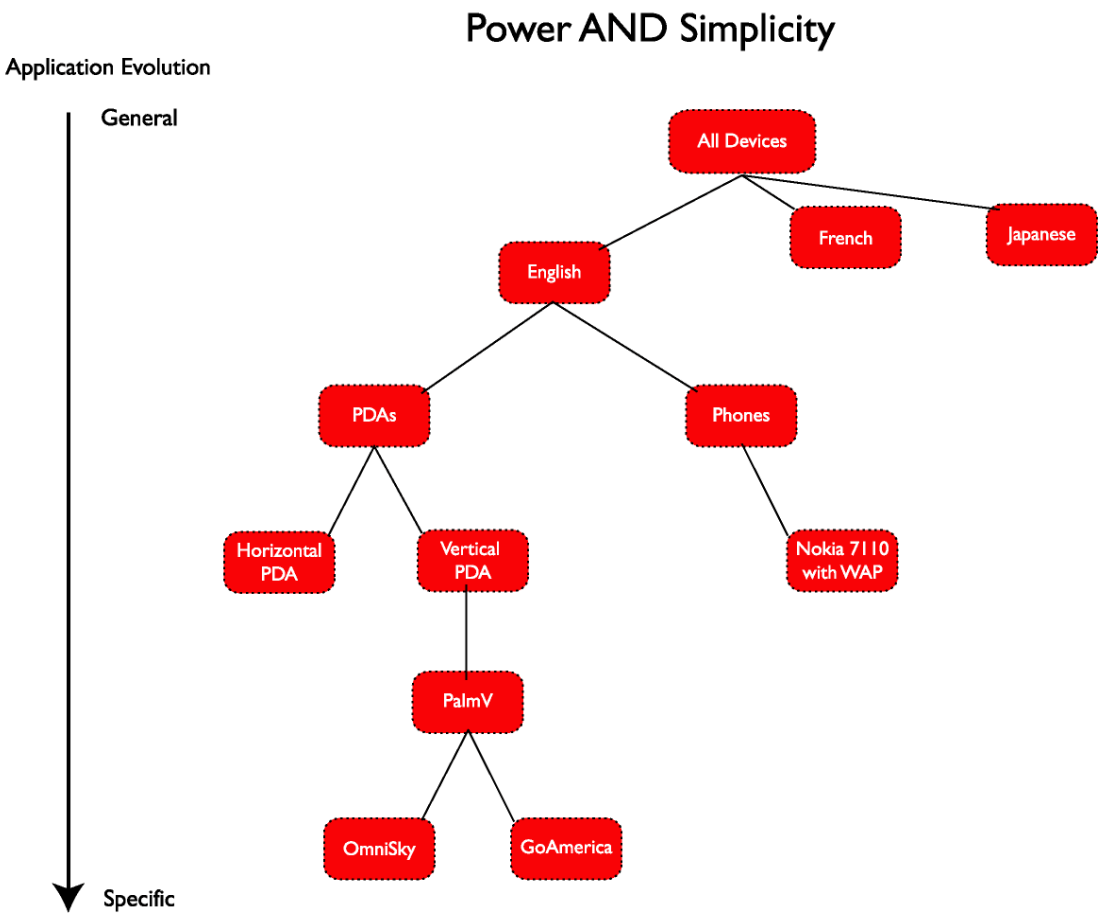
Perhaps the greatest and most important challenge of all is support for a cost-effective, sustainable development framework.

User experience development is not a one-shot process. Instead, it is an iterative process involving a cycle of designing, implementing, and testing. User requirements will continually change as they become accustomed to the application or as new user devices and technologies become available. User devices are moving targets, with new devices being announced on a regular basis and software upgrades and new protocols being deployed on existing devices.

A fundamental tenet of any successful approach to mobile application development is the anticipation of constant updates and improvements. Formalizing this cycle of designing, integrating, managing, and evolving applications allows organizations to adapt and respond to users' needs faster, and to allocate the right business and technical resources in the development cycle. This process also emphasizes rapid development and faster time to market. Even organizations developing mobile applications with short shelf lives stand to benefit from an iterative approach, as this method of development enables them to rapidly and efficiently deliver the small but crucial usability enhancements demanded by their users.

By making the user experience layer independent of the other back end layers, organizations can leverage and build from the same software base to accommodate changes rather than having to rewrite the underlying application. This flexibility allows enterprises or service providers to gain the widest availability and coverage of device support, while maintaining full functionality of the application or service.

Figure 3: Application Evolution through Iteration



MYTH #4: One User Platform Will Prevail

The diversity of user devices and communications standards is exploding with no hope for convergence. User devices now comprise PDAs, mobile phones, Auto PCs, eBooks, kiosks, IVR systems, specialized dispatching devices, and Web-based TVs; moreover, the list is growing at an exponential rate with no hope for slowing down. The diversity is driven simultaneously by consumer demand for devices that optimize their unique preferences and manufacturers' abilities to provide more device types, form factors, and capabilities.

Decisions about which devices to support cannot logically be decoupled from the users themselves. Organizations may choose to support applications on a specific device, but this strategy will not support the devices that are being used by business partners, suppliers, and customers. Furthermore, mergers and acquisitions bring new user communities, necessitating an inevitable reimplementation of device-specific mobile applications. Thus, limiting mobile application access to a single device or a select list of devices is rarely a feasible strategy; at best, it is a temporary solution that will cost an organization much more in the long term.

Three Generations of Mobile Application Development

Approaches to implementing mobile applications and the critical user experience layer have actually evolved through three distinct generations:

- Device-centric
- Application-centric
- User-centric

While forward-thinking software solutions providers are already developing third-generation, user-centric mobility applications, first- and second-generation solutions that fail to address the crucial issues of user experience are still being brought to market. Enterprises that are planning for a mobile deployment must recognize that software based on the development paradigms of the earlier generations will likely be caught in a technological dead end without the ability to evolve or extend into the user-centric paradigm. Moreover, since second generation technologies almost always create unusable applications, even a short-term solution may be a complete failure from the end-users' perspective.

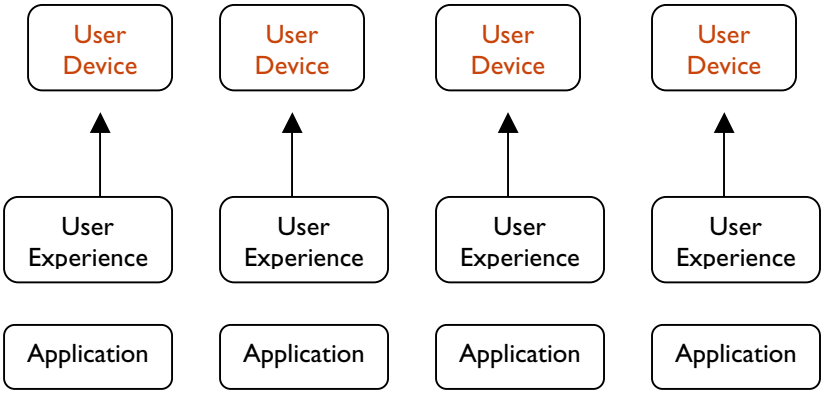
Device-centric Approach

The device-centric approach was slow, expensive, and difficult to modify and upgrade.

The device-centric approach involved treating each individual user device as the focal point for a distinct version of a given application. In this approach, separate applications were developed for distinct devices. Developers would build separate applications for traditional personal computers, specific models of PDAs, and specific mobile telephone products. This development approach, and the associated platforms, were doomed because (i) they did

not allow for rapid application deployment, (ii) they made application upgrades very expensive and difficult, (iii) they limited the ability of application developers to respond quickly to new user requirements, and (iv) they were very costly. The fact that each version of an application represented a different device forced developers to choose between adding features to all versions or writing new versions to support new devices.

Figure 4. Device-centric Approach



MYTH #5: One Size Fits All

When it comes to user experience, "one size fits all" is a doomed strategy for mobile applications.

Offering a user experience that reflects the lowest common denominator of device capabilities is likely to please no one. For example, if applications were written to assume that every device had a four-line monospaced display, users of PDAs, laptops, and PCs would find it tedious to use. Likewise, an application written to assume a PC-sized display would be unusable on a small screen.

Application usability controls the number of users and a marginal change in usability doesn't result in a marginal change in the number of users. If the user experience isn't good enough for a given user on a given device, they simply won't use it. In many cases an 80% solution is appropriate for 0% of users.

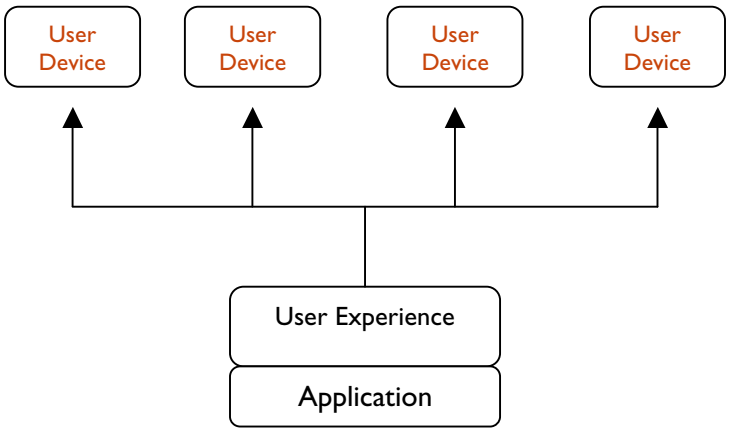
Application-centric Approach

The application-centric approach delivered lowest-common-denominator solutions on all devices.

The application-centric approach involved attempts to build "once size fits all" mobile applications for multiple devices. These applications typically attempted to extend Web content to wireless devices. Unfortunately, changes to the source Web pages very often caused the mobile applications to break down. The applications were based on technologies such as transcoding and Webscraping, and they were very often custom solutions or involved proprietary programming languages. The application-centric approach produced mobiles applications that were inherently unstable and poorly integrated into the enterprise system architecture. From the user experience perspective, a single interface was

defined that was intended to satisfy the majority of users' needs for a given context. This "lowest common denominator" development approach to user experience was doomed because it simply didn't deliver a satisfactory experience on any device.

Figure 5. Application-centric Approach



MYTH #6: Traditional Application Delivered Over a Wireless Network Provides an Acceptable Mobile User Experience

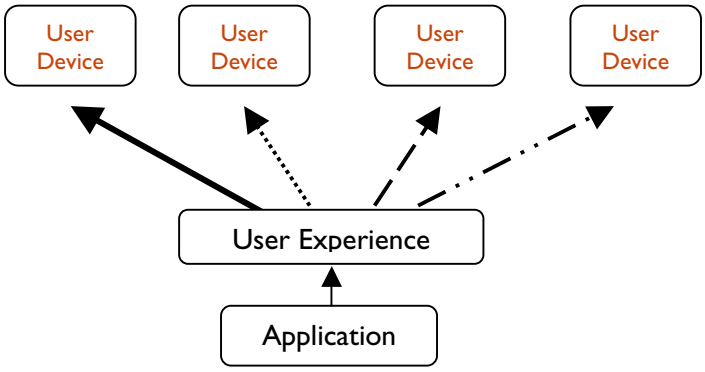
In order to create an optimized user experience, it isn't enough simply to ensure that that an application functions on all the devices and platforms for which it is intended. User experience is intrinsically tied to the context in which the application is being used. If the experience isn't customized and personalized for the user's needs at a particular place and point in time, then the development approach was not sufficiently user-centric.

User-centric Approach

The user-centric approach allows developers to optimize user experience for all devices and all user contexts without rewriting the application for each device.

The user-centric approach, the current, mature approach to mobile application development, makes user experience the focus of development rather than the device or the application. This approach is centered on platforms that allow mobile application developers to refine the user experience for different contexts without having to write a separate application for every device.

Figure 6. Application-centric Approach



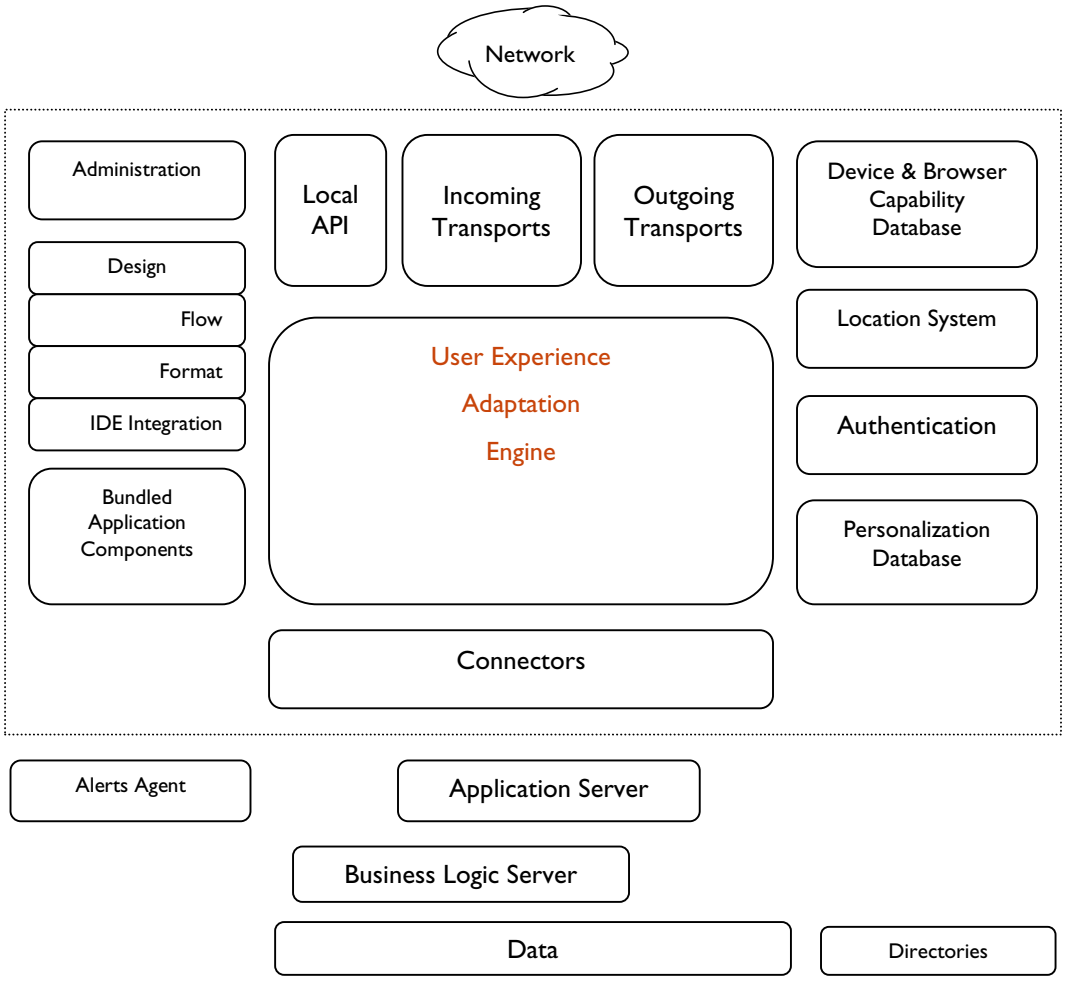
User Experience Platforms

Mature user experience platforms provide a complete set of tools for designing, integrating, managing, and evolving mobile applications.

Some organizations have started to develop and deploy their own mobile applications. Most of these companies are learning firsthand about the challenges and complexities of going mobile. There are few true standards; new devices are proliferating at an accelerating rate; users must be supported across different devices using different networks; applications must be tested over all technology permutations; users want a concise, effective experience, which suits their needs.

Fortunately there is an alternative to building the mobile applications, and the user interface layer in particular, from scratch: user experience platforms for mobile applications with tools for designing, integrating, managing, and evolving the user experience and runtime software for deploying, securing, and administering one or more layers of a complete mobile application.

Figure 7. User Experience Platform Components



A generalized user experience platform consists of the following components:

- Connectors—to integrate with other servers and application agents.
- Incoming and Outgoing Transports—to communicate with user devices.
- Device and Browser Capability Database—to identify the device, client software, and markup language used by a particular user and to retrieve all of the relevant properties in order to optimize the user experience.
- Location System—to determine the user's physical location in order to adapt the user experience based on location.
- Authentication—to determine the identity of the user.
- Personalization Database—to track user preferences in order to adapt the user experience to the needs of an individual.
- Design Tools—to develop the user experience for a specific mobile application.
- Bundled Application Components—pre-built mobile application components and user experience programming to speed development.
- User Experience Adaptation Engine—runtime software for operating the user experience layer.
- Administration Tools—to setup and manage the User Experience Adaptation Engine and all of the other runtime components.

Given the significant technical challenges associated with building a user experience layer and deploying a mobile application, most organizations would be wise to invest in an appropriate user experience platform. In particular, a mature application platform based on the user-centric development paradigm will be needed to quickly and easily deploy mobile applications that will meet the needs of users as needs and devices evolve.

Selecting a User Experience Platform

User experience platform vendors must make developing this part of the mobile application as easy and cost-effective as possible without constraining the ability to provide completely optimized solutions.

Central to the design of an effective user experience platform is a user-centric development paradigm. The following seven key criteria should be reviewed when evaluating user experience platform products:

- Good network citizen
- Comprehensive capabilities database
- Open development framework and tools integration
- Rapid Application Development (RAD) tools
- Reusable application components
- Testing and runtime logging tools
- Support for global deployment

A user experience platform that excels in all these areas will allow an organization to quickly and effectively develop and deploy highly usable mobile applications.

Good Network Citizen

The best platforms are designed to work optimally in any network context, whether hosted or internal to the enterprise.

Support for enterprise, service provider, and network operator environments is an essential feature. While some platforms are designed exclusively to be hosted by a service provider, and others are designed specifically for use by an enterprise customer on their own premises, the best products are those designed to work in any network context. The more broadly applicable a given platform is, the more attractive it is as a development platform and the easier it is to hire developers that are proficient in using it.

Many enterprises have developed proprietary applications, utilizing a combination of CRM, ERP, EIP, and EAI vendors. As a result, many enterprises are hesitant to outsource their internal applications and to relinquish control to wireless ASPs. Furthermore, outsourcing applications to a WASP would result in the continued internal management of the organization's wireline requirements while depending upon the WASP to handle anything wireless.

Good platforms allow deployment both on enterprise networks and in hosted contexts such as with ASPs. Solutions only designed for use by ASPs are extremely limiting.

Comprehensive Capabilities Database

User experience platforms must simplify the task of optimizing the user experience for new devices as they are introduced

Platforms should include comprehensive capabilities databases that track the capabilities, properties, and anomalies associated with the world of user devices, client software, and markup languages.

Using their built-in capabilities database, platform products are able to automatically determine the appropriate presentation of information for the device and browser in use and to configure and transmit the data, information, and/or content appropriately. The built-in device knowledge provides the basis for an excellent first iteration of a mobile application that can then be improved upon by the developer.

Platform vendors that provide their customers with updates to the capabilities database help their customers avoid the cost of rebuilding or creating new applications for new user devices. The costs avoided are not only monetary but, more importantly, include concessions in functionality and user-interface.

MYTH #7: XML = Mobile Application Solution

Many software products sold for mobile application development are based on XML. So many, in fact, that for some XML has become synonymous with mobile applications. The popularity of XML reflects the fact that it is an essential standard in the development of open, scalable, and adaptable development platforms for mobile applications.

However, developers must be wary when evaluating their options for tools because mere support for XML is not sufficient in determining the efficacy of a particular product. Not all XML-based products are open, scalable, or adaptable. Some platforms force developers to learn and use proprietary markup languages defined using XML.

While it's unlikely that any product not based on XML will be successful in the long term, use of XML is no guarantee of success.

Open Development Framework

Open standards ensure that applications are extensible, optimizable for all devices, and easy to support and maintain.

An open development framework—one based on open standards such as XML, XSL, Unicode, WML, WAP and HTML—ensures that developers are not forced to adopt proprietary standards or protocols. It also allows developers to continue to use their preferred tools rather than be forced to adopt a proprietary development language or tools that are specific to a particular platform.

Use of an open architecture with support for industry standards makes the platform extensible, allowing for the addition of new devices, data/content, and services. Even more importantly, an open framework does not hide any code or characteristic that could prevent a developer from optimizing an application for a particular device.

The other benefits of standards-based platforms relate to the corresponding support infrastructure. It is easier to get training on these tools, locate professional services companies to employ the tools, hire employees that work with them, and take advantage of shared work with a community of other developers. A platform that allows easy integration with off-the-shelf tools developers are already familiar with maximizes this benefit.

MYTH #8: Time-to-Market or Quality: Pick 1

There is an assumption that developers must choose between platforms that provide comprehensive solutions and platforms that provide fast turn-around using proprietary languages or templating and transcoding of existing Web applications. However this belief doesn't take into account the capabilities offered by mature user experience platforms and doesn't reflect the fact that development of mobile applications is not a one-shot process.

User experience development is an iterative process of ongoing refinement and improvement. Using an appropriate platform, developers can deploy a mobile application quickly and then refine it over time.

Rapid Application Development (RAD) Tools

Future-proofed RAD tools allow developers to compress time to market while taking advantage of open standards and iterative development practices.

Time to market has become a primary metric for enterprise success and must be supported by rapid mobile application development. The appropriate platform will enable developers to compress their time to market by focusing on the user experience; it will also alleviate the need for enterprises to deal with lengthy testing cycles and compatibility matrices.

Platforms offering graphics design tools and integration with popular IDEs and development languages provide the most productive development environment. The key to selecting the right platform is ensuring that its rapid application development (RAD) tools serve to integrate and complement the use of standards-based mark-up languages and popular off-the-shelf development tools rather than to replace them. Future-proofed RAD tools must allow developers to code in the open standard languages for different devices so they retain full control while benefiting from the platform's ability to support iterative development and specialization of code for different user contexts.

Reusable Application Components

Platforms that include or have available application modules that can be integrated in to new applications also provide a big boost in time to market.

Testing and Runtime Logging Tools

The explosion of devices, mark-up languages, client software, and other parameters around which the user experience is customized represents a vast test matrix that cannot be managed through manual testing alone. As such, comprehensive built-in support for automated testing is essential.

The runtime portion of the platform should allow administrators to start and stop individual servers, add or remove resources without disrupting service, monitor performance, carry out automated tests, log activity, and perform system back-up procedures.

Support for Global Deployment

Platforms should be available for worldwide application deployment. Comprehensive support for worldwide alphabets and languages is important. Developers should be able to roll out a mobile application that adapts itself to the language of any user who accesses it. For example, developers using convention application development environments typically write distinct versions of applications for handling foreign character sets. Instead, the platform should make it easy for developers to handle multiple languages with a single application.

Conclusion

Users want mobile applications that:

- Support existing and future devices
- Optimize the experience for each individual device and user
- Extend mobile application in a secure, scalable fashion

The world of mobile application development is surrounded in misinformation and misunderstandings. At the center of any mobile application is a user experience layer that can adapt the application to the individual needs of a particular user with a particular device in a particular location.

Mobile application development has evolved from being device-centric or application-centric, before maturing to a user-centric paradigm.

Choosing the right user experience platform is essential. Given the long-term, iterative nature of mobile application development choosing the wrong platform means having to replace it at the cost of further delay, disruption, and reimplementation. Choosing a platform that relies on an obsolete development paradigm or a proprietary language means eventually hitting a technological dead-end.

User Experience Platforms for Mobile Applications simplify the process of developing user-centric mobile applications by encapsulating the technology for back-end integration, network access, and capabilities databases while providing development tools that permit rapid deployment and ongoing tuning of the user experience.