

FOR: Application
Development
& Delivery
Professionals



The Future Of Mobile Application Development

by Jeffrey S. Hammond and Julie A. Ask, January 17, 2013

KEY TAKEAWAYS

Mobile Apps Are A Part Of Larger Modern Applications

A mobile app strategy is not sufficient if you want to be successful in your future mobile development efforts. You need to think of mobile apps as one component of a larger, modern, omnichannel application architecture and structure your development strategy accordingly.

Context Is The Future Of Mobile Development

Mobile context -- the sum total of what your customer has told you and experiences at their moment of mobile engagement -- is the future of mobile development. As development shops increasingly collect context from webs of devices, they'll find opportunities to create superior customer experiences that add value to their companies' bottom lines.

Companies Must Lower Their Cost Of Software Innovation To Survive

As companies target more devices and platforms when building modern applications, client-side development costs will increase. At the same time, they will need to deploy releases faster than ever. The only way to survive this Catch-22 is to lower the cost of testing new ideas and make it quicker and cheaper to separate the good ideas from the bad.

The Future Of Mobile Application Development

Vision: The Mobile App Development Playbook

by [Jeffrey S. Hammond](#) and [Julie A. Ask](#)
with [Phil Murphy](#), [Michael Facemire](#), and Shannon Coyne

WHY READ THIS REPORT

With more than 1 billion smartphones worldwide, and tablets numbering in the hundreds of millions, the scope of the mobile revolution rivals that of the move from monolithic systems to client/server computing in the 1990s. As in that period, we're now seeing massive changes to application architecture as companies modernize the way they build applications. Modern applications require multichannel clients, elastic infrastructure, and elastic licensing; they're composed of multiple service endpoints that are provisioned by developers with direct customer feedback on their convenience. This report, which is the vision report of the mobile app development playbook for application development and delivery (AD&D) professionals, focuses on the changes application developers need to understand if they want to build modern applications that deliver contextual customer experiences. It also looks at how modern applications will evolve over the next few years and what development teams should prepare for now.

Table Of Contents

- 2 **The Mobile App Revolution Transcends Mobile Devices**
- 9 **Technology Innovations Will Accelerate Contextual Computing**
- RECOMMENDATIONS
- 17 **Ready Your Team To Build Modern Applications, Not Mobile Apps**
- WHAT IT MEANS
- 18 **The Biggest Change In Development In A Generation Is Here**

Notes & Resources

Forrester interviewed multiple vendor and user companies over the course of 2012.

Related Research Documents

[Build Five-Star Mobile Apps](#)
November 7, 2012

[Don't Move Your Apps To The Cloud](#)
November 5, 2012

[The Future Of Mobile eBusiness Is Context](#)
May 1, 2012

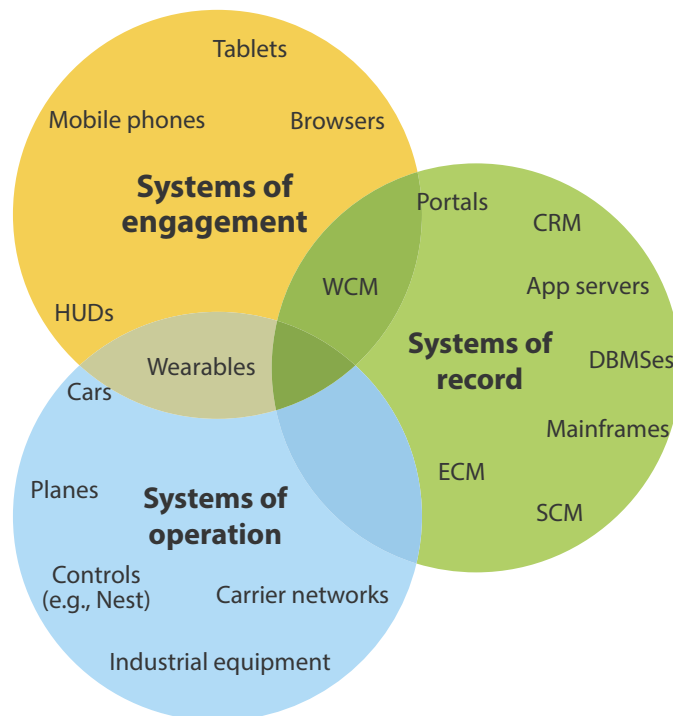
[Embracing The Open Web: Web Technologies You Need To Engage Your Customers, And Much More](#)
January 24, 2012



THE MOBILE APP REVOLUTION TRANSCENDS MOBILE DEVICES

Even the most casual observer would agree that mobile apps are a big part of the future of application development and delivery. With more than 1 billion smartphones worldwide, the mobile revolution rivals the move to client/server computing in the 1990s and the Internet revolution in the previous decade. But the future of mobile app development is more than just adapting to smaller screens, different programming languages, and new operating systems. Mobile app development is part of a larger structural change in the way we build applications. We're entering a new age of application development that creates modern, compelling systems of engagement and links them with systems of record and systems of operation (see Figure 1).

Figure 1 Modern Applications Are Systems Of Systems



89181

Source: Forrester Research, Inc.

Mobile Apps Are The Vanguard Of A New Wave Of Software Innovation

Linus Pauling said it best: “The best way to have a good idea is to have lots of ideas.”¹ But it costs money and time to commercialize an idea, whether it’s a chemical compound, a new business process, or a new software application. In fact, one could argue that the entire discipline of software development is about capturing ideas, codifying them in ones and zeroes, and automating them. If that’s the case, then there should be a correlation between the cost of software development and the cost of business innovation. If the cost of software development declines, this should enable

companies to vet more ideas, which should result in more good ideas that ultimately create value for customers and wealth for investors. In fact, that's what's begun to happen over the past decade — software is figuratively eating the world.²

The Modern Approach Fosters Open, Elastic, Omnichannel Applications

Whether it's Facebook, Google, Netflix, Amazon.com, GitHub, or Instagram, small groups of developers are creating new businesses in record time. If you dig beneath the business models and examine the software architectures that drive these businesses, you can discern a number of common traits that define the modern application infrastructures that these firms deploy to their competitive advantage. Modern application development processes mean that developers introduce changes; they:

- **Go beyond mobile and multichannel to omnichannel.** Simply operating in multiple channels is no longer enough; the software services and applications that these companies create are omnichannel — that is, they work everywhere. Take Netflix; it works on tablets, mobile devices, TVs, laptops, and game consoles. A Facebook post that one person uploaded from an iPhone can be viewed by other users on an Android tablet or in a Windows XP browser. People now expect omnichannel interactions; they're the new normal. This is why you don't need a mobile app development strategy per se — you need an omnichannel client development strategy. Your first step should be to deliver a consistent (although not necessarily identical) brand experience across *all* of the channels your customers use.
- **Venture beyond fixed on-premises hardware to elastic infrastructures.** Before the public cloud, developers had to estimate how much infrastructure their applications would need before deploying their applications. Talk with most experienced developers over beers and they'll confess that application sizing is more art than science, so it's natural to overestimate. They'll tell you that it's better to overprovision hardware than to run out of boxes — especially if the hot new idea they're working on really takes off. It's different with modern applications; the ability to scale up (or down) on demand significantly alters how much time and capital are needed to deploy version 1.0 of a new software service.³ The result: For modern applications that use elastic infrastructure, application sizing is now an iterative activity. Elastic infrastructure is why a modern development shop like Instagram could easily add more than 1 million new customers in only 12 hours when it launched its Android app last year.⁴
- **Combine discrete services into compelling new solutions.** An eternal struggle rages anew inside every development project: Do we build everything the project needs from scratch so that it works exactly how we want it to, or do we reuse or buy whatever components we can to deliver a new service faster? While previous generations of developers tended to fall into the “not invented here” camp, we're seeing a different type of behavior when it comes to modern application architecture. Developers are more comfortable using open source, because they

can fork an open source project and modify it if they need to; and they're more comfortable with third-party web services, because they can isolate dependencies and replace them later if necessary. As a result, modern applications are composites, built from multiple URL endpoints and application frameworks and designed to get to market quickly. And whether it's Nike, Twitter, Spotify, Twilio, Google, or Facebook, there are more compelling services available for developers to remix every day.

- **Use managed APIs everywhere.** While defining APIs is closely related to combining discrete services, we're seeing developers further extend application modularity by using managed APIs not to merely consume services, but also to expose their own services to third-party developers. In the process, many create a "headless" version of their services that others can incorporate into their digital products and software services. To do this, development shops must think deeply about security, authorization, and real-time management of the data that flows into and out of their application through third-party applications. The modern application is much less a front-to-back stovepipe and much more a web of information that flows through multiple layers of RESTful API calls. This thinking leads to corporate service platforms and third-party development ecosystems from companies like MasterCard and CBS Sports that aren't traditionally known as software shops.⁵

Modern Provisioning, Feedback Loops, And Open Source Speed Application Delivery

The changes that mark modern application development are not just structural; we've also seen substantial changes to development processes that complement modern application architecture. These new processes speed up the deployment cycle, reduce the number of organizations involved, and decrease the cost of reuse. Here's how:

- **Modern applications use open source software to grease the licensing skids.** In the old application model, software startups needed to raise millions in venture capital to buy hardware, development tools, and infrastructure software. Projects at established companies faced a similar burden — they had to clear budget committees and build comprehensive, multiyear business cases to justify a significant capital investment. But the proliferation of capable open source development tools and pay-per-use cloud services has dramatically reduced the cost of trying out lots of different ideas. And open source licenses are perfectly compatible with elastic infrastructure; you don't have to check with purchasing to see if you have spare licenses available every time you spin up a new server instance. Open source frameworks are becoming a key component on the client side of modern applications for that same reason: good-quality code with elastic licensing flexibility.
- **Modern developers self-provision applications to avoid I&O roadblocks.** It's no coincidence that many advanced dev-ops practices come from firms that are building modern applications. They must keep up with the rapid release cycles of mobile devices and can't wait for infrastructure and operations (I&O) professionals to make hardware or software available to them. The good news: Developers using elastic open source software licenses and public

cloud hardware aren't held back by I&O delays. Automating application deployments reduces the null release cycle to days, hours, or even minutes.⁶ Take self-provisioning to the extreme and you find yourself working in a continuous deployment environment like Amazon Web Services, where developers deploy new code to production every 11.6 seconds.⁷ On the client side, platform app stores like the Apple App Store, Google Play, and enterprise mobile app management tools complete the developer self-provisioning puzzle.

- **Immediate customer feedback speeds initial value and improvements to customers.** Traditional application development separates developers from their customers with multiple layers of middlemen. Whereas salespeople, product managers, and business analysts spent months or years collecting long lists of future requirements, modern application delivery methods collect requirements via real-time rating systems and receive near-instantaneous public feedback to optimize and speed their development cycles. It's why developers, product managers, and even CEOs from companies like Kayak.com, Brainshark, and NatureShare all watch the ratings and comments in the Apple App Store and Google Play and why knowing how to deploy in-app analytics is now a critical developer skill.⁸

The net result of this change is dramatic — modern applications are complex, multitiered, and omnichannel. They arrive faster, scale up and down as needed, and create value quicker than traditional applications. To achieve long-term success building mobile apps you must embrace modern application development techniques. Ignore these structural and business model changes and you risk creating a new generation of stovepiped mobile apps that are hard to maintain and ill-equipped for the changes that are just over the horizon.

Context Is King When Delivering Modern Mobile Apps

There's a simple formula to evaluate a mobile service: The benefits of the service must outweigh the inhibitors of adoption.⁹ Mobile services must be *immediate*, *simple*, and *contextual*. As your application developers move from early experimentation to developing modern applications, it's important to think carefully about context. Mobile devices enable your applications to tap the customer's current context: Where in the world are they? What else are they doing that an available service can add significant contextual value to? Forrester calls this the customer's *mobile context* and defines it as "the sum total of what your customer has told you and is experiencing at his moment of mobile engagement." A customer's mobile context is:

- **Situational:** The current time, location, altitude, environmental conditions, and travel speeds the customer is experiencing.
- **Preferential:** Historical personal decisions that the customer has shared with the application or with social networks.
- **Attitudinal:** The feelings or emotions implied by the customer's actions and logistics.

So how can an application developer deliver a contextual experience within a modern multichannel application? Start by recognizing that the customer's context is intimately related to the application state. In order to effectively use all the situational information at your disposal, your developers will need to rethink the way they architect their applications. In fact, in past research, we've noted the emergence of a different system architecture pattern: the web services façade (see Figure 2).¹⁰

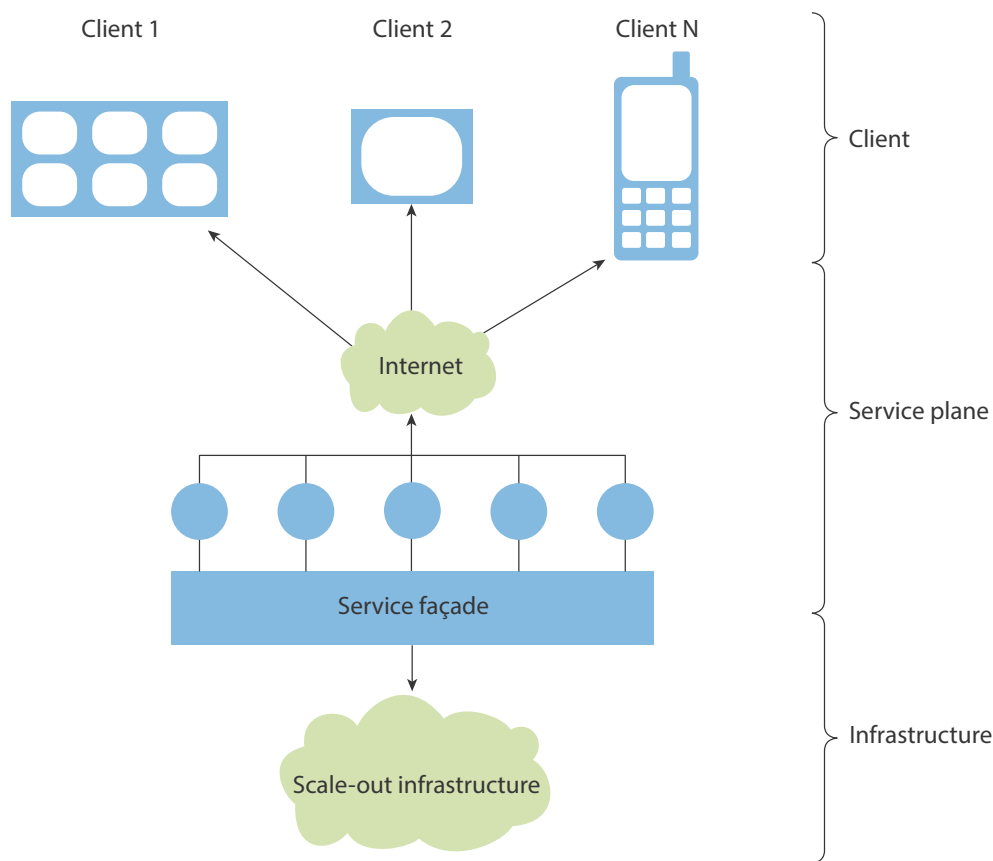
Use of the service façade pattern is increasing because it's well-suited to mobile clients that connect to modern application architectures, for the following reasons:

- **Pushing the in-process state to devices helps establish local context.** It's important for a contextual application to take advantage of all of the sensors and information on the customer's device. It really doesn't make sense to send all this information up to the server tier of a modern application, especially over a network connection with variable latency. It makes better sense to manage as much of a contextual application's state as possible locally, subject to device performance constraints. Think of a request for network access as the new "thinking."¹¹
- **Elastic infrastructure benefits from asynchronous control flows.** High-scaling modern applications built on elastic scale-out architectures tend to employ service endpoints that are both stateless and asynchronous. This allows the applications to use load-balancing techniques to scale automatically by adding additional processing instances in a nearly linear fashion.¹² Efficient use of many different server instances is best achieved when the control layer can minimize the amount of state it needs to track between requests for information or processing resources.
- **Compositional service design allows developers to tune server infrastructure.** One consequence of state management moving to mobile clients is that server-side design can become more atomized — and the hardware and software components of modern application infrastructure are increasingly designed around and optimized for individual services. Evernote is a good example. The service that stores and retrieves customer notes is built on a fault-tolerant server infrastructure where every server instance has a hot backup ready to go in the event a primary instance fails. On the other hand, Evernote's optical character recognition service feeds requests to a pool of servers that perform tasks as capacity allows, in a completely stateless manner.
- **Alternative system patterns replace model-view-controller (MVC).** Over the past decade, web developers have often applied the MVC system pattern to build web applications. But as state migrates to the edge of modern apps, we see a renaissance in application architecture design. This renaissance is driven by the emergence of alternative system architectural patterns like "pipes and filters" and "broker."¹³ Both are compatible with the web service façade, but modern application development shops use pipes and filters to power content-oriented workloads, while broker works well for shops that are building connected and collaborative workloads (see Figure 3).

- **In-memory databases provide immediate access to cached data.** Last-mile connectivity is a significant challenge for modern applications that connect to mobile phones or tablets over 3G/4G networks. Developers can't always depend on speedy connections, even as 4G networks proliferate. In order to maximize app responsiveness, modern application developers stage data as close to the last mile as possible using a combination of in-memory databases, such as Ehcache and Memcached, and lightweight HTTP servers like nginx.

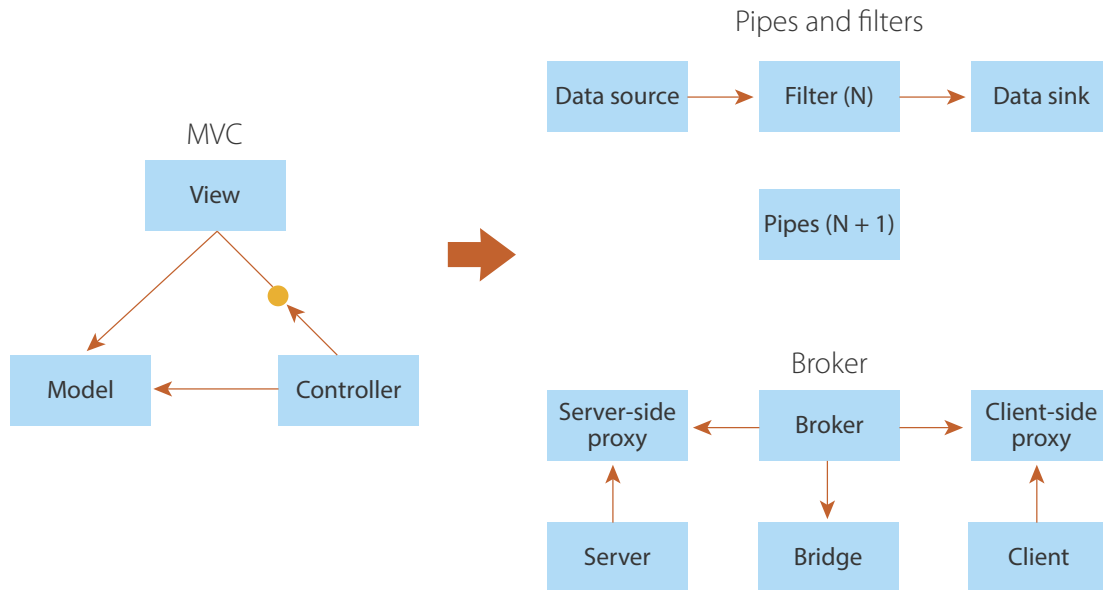
By now, you should realize that the architecture of a modern application looks very different from that of a traditional MVC web application. Most enterprise mobile app developers are only beginning to employ the principles we've articulated above. But analyze highly rated, ubiquitous consumer-facing mobile services in depth and you're more than likely to spot many of these telltale architectural traits (see Figure 4).

Figure 2 Multichannel Design And The Service Façade Pattern



Source: August 2009 "Enterprise Web 2.0 Patterns" report, prepared exclusively for Forrester Leadership Boards. For more information about Forrester's executive programs, see: <http://www.forrester.com/LeadershipBoards/>

Figure 3 Alternative System Patterns Power Modern Mobile Applications



Source: November 5, 2012, "Don't Move Your Apps To The Cloud" Forrester report

89181

Source: Forrester Research, Inc.

Figure 4 Traits Of A Modern Application

Architectural trait	Rationale
RESTful APIs (usually employing lightweight payload formats like JSON or XML)	APIs must be asynchronous and consumable across multiple platforms (many of which lack the resources to consume heavier protocols like SOAP).
Use of in-memory databases	Reduces latency at the intersection of mobile clients and infrastructure.
Pervasive use of open source software	<ul style="list-style-type: none"> • Makes it easier and cheaper to scale up or down on demand. • Simplifies licensing.
Services deployed on elastic infrastructure	<ul style="list-style-type: none"> • Makes it easier and cheaper to scale up or down on demand. • Relieves the pressure on the inexact process of estimating infrastructure needs before deployment.
Sharded SQL DBMSes or NoSQL DBMSes	Makes it possible to support millions of customers with commodity hardware using a scale-out approach.
Composed of independent service endpoints	<ul style="list-style-type: none"> • Individual services can change independently. • Applications can continue to function if an individual or primary service fails.
Uses asynchronous communication	<ul style="list-style-type: none"> • Reduces complexity. • Improves performance by eliminating blocking at the infrastructure layer.
Uses dynamic languages in concert with static languages like Java and .NET	<ul style="list-style-type: none"> • Simplifies programming constructs. • Allows applications to evolve without recompiling services.
Uses lightweight process communication frameworks like node.js and nginx	<ul style="list-style-type: none"> • Reduces resource consumption. • Effectively uses smaller processing instances and smaller thread pools.

89181

Source: Forrester Research, Inc.

TECHNOLOGY INNOVATIONS WILL ACCELERATE CONTEXTUAL COMPUTING

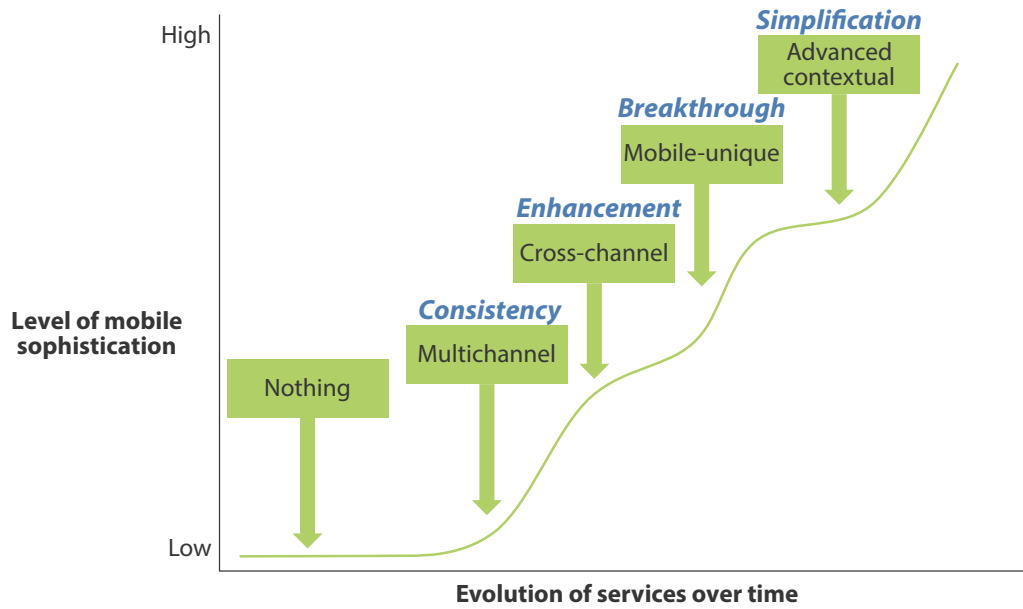
So what does the immediate future hold for modern application developers? A lot — there’s a wave of innovation coming that will increase the pressure to think beyond the current art of the possible. And while a lot on these changes will happen on client devices, modern application developers also need to keep tabs on evolving infrastructure capabilities.

Moore’s Law And New Sensors Will Increase The Power Of Client-Side State

New processing capabilities and sensors are changing how users control their devices and display content — quickly changing the parameters of what is possible in the mobile context (see Figure 5). Our interviews with technology leaders at more than a dozen device manufacturers reveal that the following technology trends will push the evolution of context and convenience (see Figure 6):

- **Today's coolest and most high-end features will become commonplace.** Technologies found in high-end smartphones will migrate into lower-end phones as the cost of components comes down through scale and consumer demand. Mobile phones are already packed with phenomenal technology that firms underuse, including GPS, accelerometers, gyroscopes, magnetometers, Near Field Communications (NFC), and high-resolution displays. With the use of networks and other tools, the amount of contextual data developers can collect and use will continue to skyrocket.
- **New sensors will reveal more about the user's environment.** Barometers, microbolometers, and chemical sensors will find their way into phones as use cases and business cases evolve. The phones will also act as de facto routers, relaying or interpreting information from other machines or from attachments with sensors. They will offer new information ranging from a consumer's altitude, speed, temperature, and degree of light or darkness to orientation relative to the North Pole or center of the earth. Dual cameras have already appeared in phones, enabling depth perception and the creation of 3D content.
- **Motion, voice, and touch will redefine the user interface.** A combination of improved display technologies and simplified development tools for motion sensors will shift the control of mobile phones from touch to motion and voice. In the near term, the responsiveness of touchscreens will improve. Going forward, however, phones will be controlled more and more by voice, gestures, speed, pressure, or the presence or absence of light.¹⁴ As a result, developers will need to think about supporting ever more sophisticated access mechanisms. That headless API layer you've no doubt already built will really come in handy when you need to start integrating with Siri or through the next generation of a Kinect controller, right?
- **Blistering multicore processors will enable responsiveness.** Ever mightier microprocessors, faster graphics accelerators, and increased memory will continue to improve the performance of mobile devices. This will allow for increasingly advanced computational functions, and more sophisticated processing of audio and video data. Mobile phones are already beginning to interpret conversational voice and directed commands — tasks that previous generations of mobile services offloaded to servers due to the computational complexity involved.¹⁵ Embedding complex voice and motion control along with HD media will fundamentally change what is possible with a mobile device. This raises the importance of modularized application services, so that shifting the processing of individual application components doesn't require a complete infrastructure rewrite.

Figure 5 Mobile Services Are Maturing



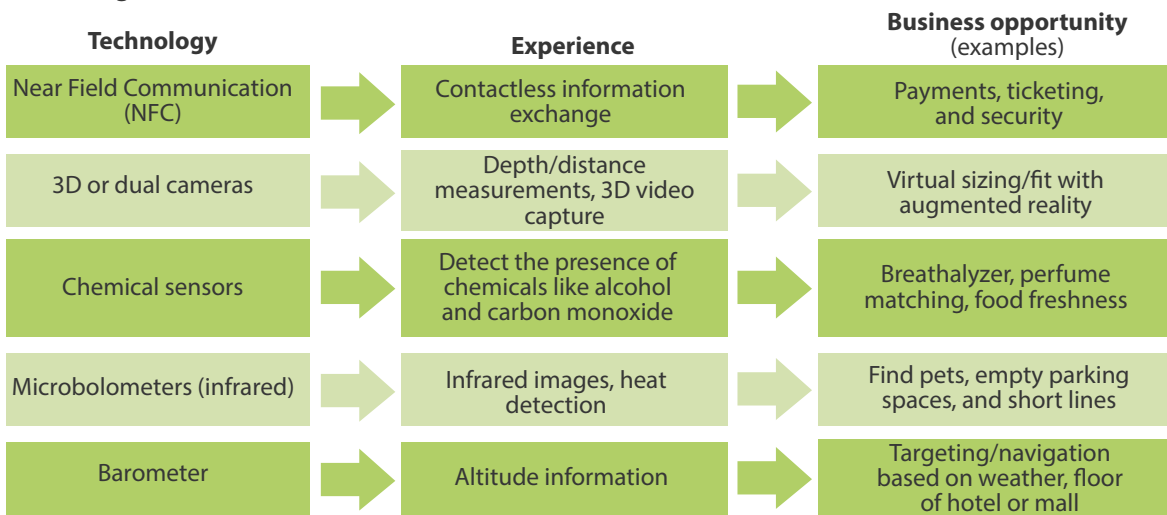
Source: May 1, 2012, "The Future Of Mobile eBusiness Is Context" Forrester report

89181

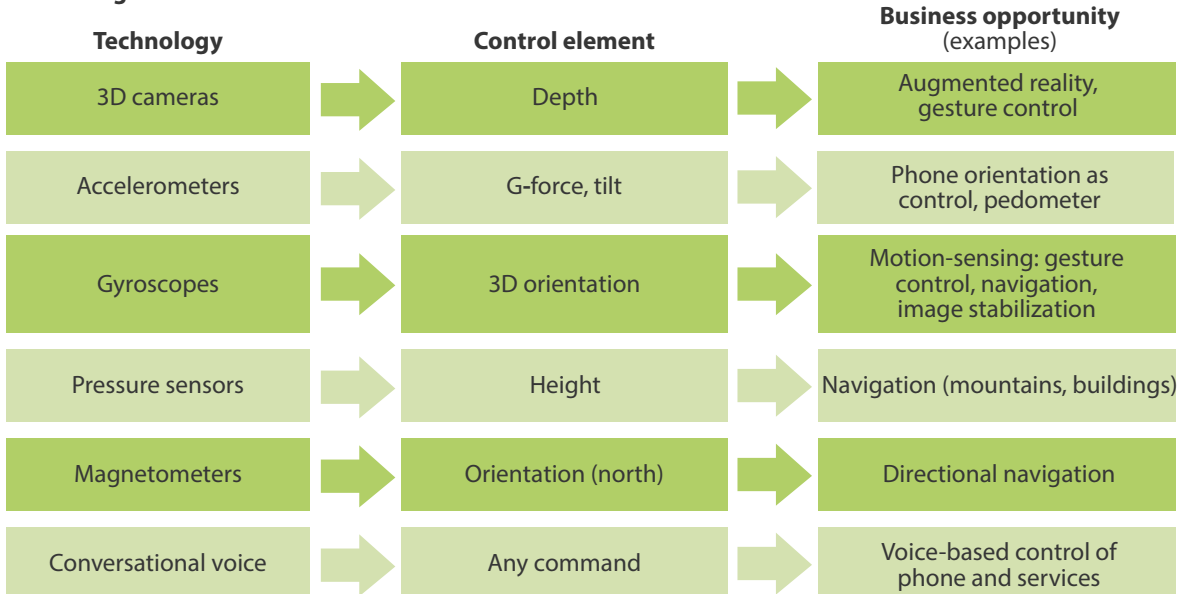
Source: Forrester Research, Inc.

Figure 6 New Technologies Enable New Opportunities

Technologies that drive contextual information



Technologies that enable motion control and voice control



Source: Interviews with A.M. Fitzgerald & Associates and Yole Développement
 Source: May 1, 2012, "The Future Of Mobile eBusiness Is Context" Forrester report

89181

Source: Forrester Research, Inc.

New Devices Will Change How Developers Build Modern Application Clients

It seems like today's mobile app development is pretty well defined: Build your app; make sure it looks good on a 4" smartphone screen and a 10" tablet; and submit it to an app store. It's not going to be quite that easy in the near future; a wide range of new client form factors and changes to the nature of the "app" itself will increase the need for flexibility — especially on the client. Collectively, these new devices will significantly expand the potential for collecting contextual data about your customers (see Figure 7). Here are some ideas of what changes you'll face:

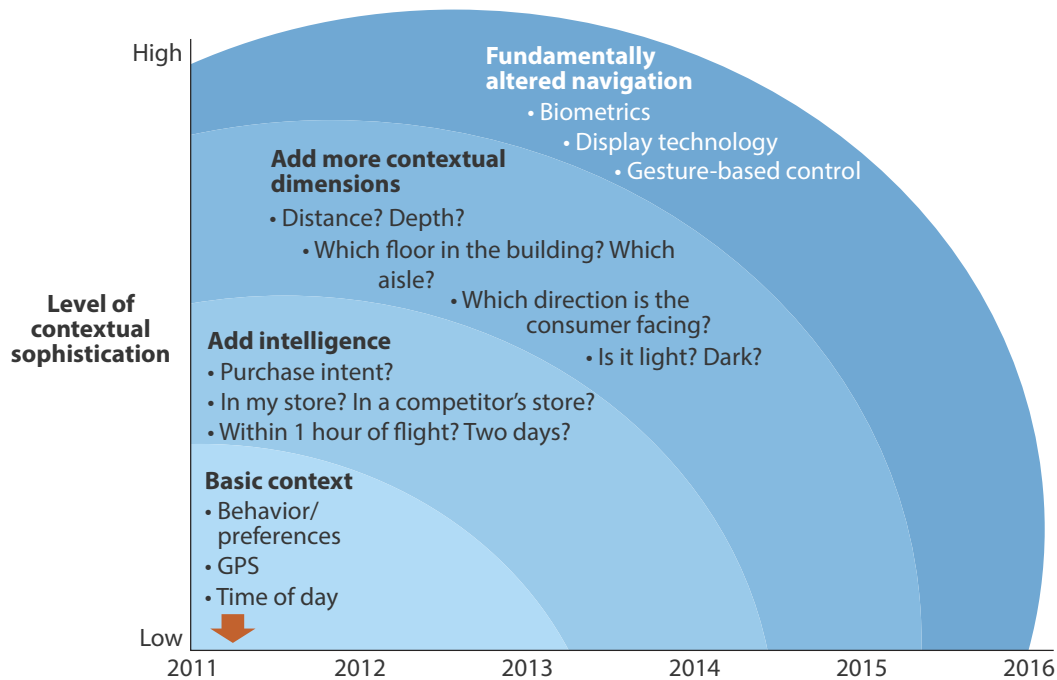
- **Apps will augment voice input — and then prioritize it over touch.** We've already seen mobile developers clamor for API access to Apple's Siri and Google Now. Third-party alternative SDKs are available that let you build voice input into your app today. But why would you want to? Well, consider the times when direct touch interaction isn't convenient. For a running or fitness app, a phone is likely to be strapped to a customer's sweaty arm, and looking down at your phone screen while running — or operating a vehicle — can be dangerous. Modern applications will respond by allowing users to interact with their device and get useful work done while keeping their eyes and hands off of it.
- **User interfaces (UIs) will need to adapt to heads-up interfaces.** The standout event at Google I/O 2012 was a demonstration of Project Glass — specifically, the delivery of a package to the roof of Moscone West from a blimp by skydivers and BMX pros, with a real-time view beamed into the conference keynote. Hype aside, expect to see heads-up displays (HUDs) go mainstream in the next five years as Moore's law pushes processors to the point where the form factor can be made powerful, lightweight, and perhaps even stylish. Augmented-reality apps that don't work when delivered through a phone or tablet will be transformative when ported to a device like Google Glasses — but that will only create more challenges for developers. We're used to monopolizing the user's attention with opaque screens and attention-grabbing controls. With HUDs, we'll have to adapt to peripheral cues and the addition of tactile and aural feedback, and make sure that we don't impair a customer's perception of real-time space.
- **Bigger (and smaller) touch input devices will require adaptive UIs.** Today, most app developers prioritize a few popular devices, like the iPhone, the Samsung Galaxy S III, and the iPad. Cherry-picking devices will become more of a challenge as device form factors and platforms proliferate. Android tablets are already gaining significant worldwide market share, and the 7" tablet is now well-established alongside its larger 10" cousin. And with Windows 8 now shipping, developers can expect to find a whole range of larger touch-sensitive devices, like HP's ENVY series. But device surfaces will grow even larger and move beyond specialized devices as the cost of multitouch monitors falls — to the point where touchwall computing becomes broadly available.¹⁶ Developers will need to scale their user interfaces — a 4" experience will be very different from an 84" experience. Principles that we currently see in responsive web design will pave the way for responsive client design across omnichannel and multidevice form factors.

- **Standalone mobile apps will evolve into pluggable mobile services.** The days of the standalone app, with its own icon on the screen and self-contained sets of services, may soon be over. We're seeing a trend where platform vendors are offering more and more platform-specific services that developers leverage. Whether it's Apple Passbook, Google Wallet, or some other offering, expect mobile apps to become ever more deeply integrated into devices. Another example that points toward this trend is Microsoft's Windows Phone 8 hubs, where developers tap into other common services around people or productivity. RIM goes even further in its upcoming BlackBerry 10 operating system, where the hub, peek, and flow UI metaphors further erode the walls between apps and services. Regardless of whether these specific platform innovations are successful, expect a relentless push toward integrated client-side services instead of standalone mobile apps.

- **Wearables and connectables will herald a fast-changing local network of customer context.** Devices worn on or near the body that sense and relay information will be embedded in clothing, accessories, and even the body itself.¹⁷ First-generation wearables like the Nike+ FuelBand and Fitbit will give way to internal biomedical instruments, shoes that generate power to recharge devices, and golf clubs that provide swing telemetry. Connectable home alarm systems, automobiles, and scales will function in a similar way. Developers will tap into these new information sources, with a phone or tablet acting as a local combination of router and remote-control device. Client-side developers will release updates faster than ever to dynamically add support for new devices in the extended local network.

- **Better web support will advance the economics of the hybrid application model.** With each release, popular mobile operating systems get better at supporting HTML5 and its attendant APIs. We think that this reality, plus the proliferation of connected devices and form factors, will continue to put pressure on application development teams to keep labor costs in check. Teams can respond in three ways: Reduce the number of devices and form factors they support; use a cross-platform tool that generates native clients; and use a mix of native platform-specific code and WebView/UIWebView controls to build hybrid apps that blend the capabilities of native platforms with the cross-platform portability of the Web.¹⁸ We think that the second and third responses are inevitable as platform support demands increase, and that hybrid apps will get more and more sophisticated and capable as a result.

Figure 7 The Potential Of Contextual Data In Modern Applications Will Evolve Over Time



89181

Source: Forrester Research, Inc.

Advanced Demands For Contextual Support Will Drive Further Infrastructure Evolution

Even though smartphones and tablets will continue to become more powerful, there are still situations where it will make sense to include server-side processing to aid device context. The entire customer context will be made richer by adding historical data and the raw processing power of thousands of server nodes. Here are some of the ways that we think that modern application infrastructure will evolve to support context:

- **Omnichannel experiences will include more cross-channel and unique capabilities.** Many firms find it hard enough to create consistent omnichannel experiences. But as the use of context grows, leading development shops will go beyond a goal where consistency rules. In some cases, cross-channel experiences will use multiple devices to provide a superior experience — for example, your bank could send a confirmation to your mobile device whenever you use your debit or credit card, or your automobile could send real-time diagnostic information to your tablet to offer an early warning of failing parts. Other development shops will further tailor experiences for individual channels, such as adding an augmented reality interface for a HUD device or adding mapping services and navigation for any device with an embedded GPS. The infrastructure behind these clients will need to adapt to the context-gathering capabilities of the device that a customer is using and shift contextual data gathering as that customer's access methods change.

- **Big Mother in your pocket: Predictive analytics will help services learn about customers.**

If you've used the latest version of Google Android, you're already familiar with Google Now. Google Now attempts to get you just the right information at just the right time, whether it's this morning's weather, traffic patterns to guide your commute, or the score of your favorite football team's last match. It's like having your mother reminding you to wear your galoshes on a rainy day. What's going on under the covers of this service is both powerful and a bit scary. It's making guesses about a customer's current context and refining future guesses based on whether previous ones successfully drove a user interaction. To the extent that customers are creatures of habit, over time the service should get better at predicting the next action it should take.

Mobile apps integrated with predictive analytics are a double-edged sword. Done correctly, these predictive applications will undoubtedly add convenience to an experience — like a “Big Mother”; Done poorly, there's the potential for shenanigans: A predictive application could be a vehicle for the next generation of subliminal advertising or as a corporate Big Brother watching what customers do. After all, it “guessed” that you really wanted sushi tonight at the bar three blocks down the street — right?

- **Complex event processing will manage and enrich the intersection of context.** While customer-centric context on a single device is useful, that usefulness grows when it intersects other contexts. It starts with the cross-channel context of the customer's own devices, but it's bigger than that. Imagine if the inbound airplane for your next flight could signal that it needs maintenance, and that context could be shared with all of the premier-status flyers on that flight, allowing them to take preemptive action. Imagine if your hotel chain could use a geofence to detect when you're five minutes from arriving in the lobby and automatically start processing your hotel check-in.¹⁹ The intersection and the union of contexts will make modern applications ever more convenient. No individual mobile device will be able to handle the processing of events created by complex intersections of context. However, the infrastructure they link to will serve as connective tissue — processing complex events and relaying the notifications directly to the customer. That's if developers can efficiently integrate complex event systems with machine-to-machine communications, local wireless networks, and carrier infrastructure.

- **Real-time collaboration services will improve connected app capabilities.** We've seen great multichannel apps emerge over the past five years — but most of them are single-user experiences that synchronize data. Expect to see in-place document sharing, collaborative editing, communication, real-time location, co-browsing, real-time communications, and low-latency messaging services make it possible for developers to easily construct immersive collaboration applications where multiple customers collectively interact with each other. There are obvious applications in games, office productivity, and trading systems, and we'll also see traditional enterprise applications reimaged as real-time expert systems designed to speed business processes while identifying and removing the barriers that prevent business flow.

RECOMMENDATIONS

READY YOUR TEAM TO BUILD MODERN APPLICATIONS, NOT MOBILE APPS

If your process of readying for your mobile future only involves retooling your development shop to add some Objective-C or JavaScript developers, you've missed the real watershed that mobile devices and tablets represent. We're moving to a world of multiple endpoints, hybrid-use personal and corporate devices, and lightning-fast client device turnover. Success will create immediate demand for processing and infrastructure, but the reference templates of success are not yet clear. Nor can we see how long those templates will endure once we discover them. You have to be fast and you have to be flexible, but above all you need the capacity to experiment and weed out the bad ideas from the good as quickly as possible. You need to:

- **Think like a software development company.** For years, the best software development companies have treated their custom development efforts like a research portfolio with a pipeline process. As a consequence, they tend to organize their development shops differently and use different (more agile) processes. If you're going to be competing with modern software companies, you might as well organize your efforts to be as efficient as the best of them.
 - **Act like a startup.** It's important to understand how companies like Instagram, and OMGPOP could build massive valuations with small development teams in a matter of months. Don't pay attention to the amount of these valuations or whether they have a basis in fact. Rather, pay attention to how these companies went from having a good idea to supporting millions of customers in a matter of months. It takes a combination of architecture, elastic infrastructure, and lean development processes to do it right. It also takes a corporate culture that empowers development teams — which is one of the hardest parts for enterprises to accept.
 - **Use the modern architecture resources at your disposal.** Your company's use of the public, elastic cloud infrastructure is inevitable — there's no reason it can't be as secure and fault-tolerant as your current application infrastructure. There's also no compelling reason to resist the other modern strategies, like self-provisioning for development and testing and elastic licensing with open source software. Each of these strategic components has proven its worth in mission-critical systems; why not bring them together as key components of your next-generation architecture plan?
-

WHAT IT MEANS

THE BIGGEST CHANGE IN DEVELOPMENT IN A GENERATION IS HERE

If you hadn't already guessed it, the move to modern application architecture is big. Development shops have gotten somewhat complacent producing MVC-based apps that were developed on big middleware servers and tested on just a few browsers running on Microsoft Windows clients. The easy days for software development shops are coming to an end:

- **Client-side development costs are going up — way up.** Expect to support at least three client platforms and multiple form factors over the next five years. Prepare to budget 20% to 30% more for front-end UI development tasks in your multichannel projects — and even more than that if you're building native clients.
- **Expect massive changes to enterprise architectures.** Most existing systems of record aren't built with asynchronous cores and don't support a scale-out, elastic architecture. It would take years to retool these current systems into modern applications, so in many cases it will be better to wall them off with the web façade pattern while building a new cross-channel message- and content-passing control tier on top of them. Concentrate any changes to systems of record around areas where you need to improve the latency and currency of data, and then rearchitect the systems behind the web façade as time and budget allows.
- **The wall between dev and ops will crumble.** As the construction of modern applications moves out onto the public cloud and public devices, the traditional organizational model that separates development from operations will break down. Developer self-provisioning will rebalance the relationship in favor of developers, but it will carry with it greater responsibility for security and performance. Expect more developers to be on call for application support in the new model. Expect operations personnel to become integrated into development teams and to start their work at the inception of an idea.

ENDNOTES

- ¹ Linus Pauling was a 20th-century American chemist who is the only person ever awarded two unshared Nobel Prizes. During his influential career in chemistry and biochemistry, he made hundreds of discoveries across numerous fields.
- ² Source: Marc Andreessen, "Why Software Is Eating The World," *The Wall Street Journal*, August 20, 2011 (<http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html>).
- ³ For a deeper treatment of how the economics of elastic infrastructure works, see the May 22, 2012, "[Drive Savings And Profits With Cloud Economics](#)" report.

- ⁴ Source: Anthony Wing Kosner, “Instagram’s Exploding, Adds a Million Android Users in 12 Hours and Raising \$50 Million,” *Forbes*, April 7, 2012 (<http://www.forbes.com/sites/anthonykosner/2012/04/07/instagrams-exploding-adds-a-million-android-users-in-12-hours-and-raising-50-million/>).
- ⁵ Source: MasterCard Developer Zone (<https://developer.mastercard.com/portal/display/api/API>) and CBSSports.com Development Center (<http://developer.cbssports.com/>).
- ⁶ The null release cycle is the time it takes to build, test, and deploy a new version of your application if you were to only change one line of code. For a description of this and other release management concepts, see the February 7, 2011, “[Five Ways To Streamline Release Management](#)” report.
- ⁷ Amazon has achieved an 11.6-second null release window. Source: “Velocity 2011: Jon Jenkins, ‘Velocity Culture,’” YouTube, June 20, 2011 (<http://www.youtube.com/watch?v=dxk8b9rSKOo>).
- ⁸ For more information about the importance of direct and real-time customer feedback, see the November 7, 2012, “[Build Five-Star Mobile Apps](#)” report.
- ⁹ For more on app convenience and the mechanics of mobile context, see the May 1, 2012, “[The Future Of Mobile eBusiness Is Context](#)” report.
- ¹⁰ For full details on the web service façade, see the January 24, 2012, “[Embracing The Open Web: Web Technologies You Need To Engage Your Customers, And Much More](#)” report.
- ¹¹ In the previous generation of Microsoft COM-centric distributed architecture, applications were designed in a way that made it more expensive to call out of an application’s “in-process” stack, so developers tried to avoid using dynamic libraries in performance sensitive code that would require them to “think” out of their current apartment model. Developers in this era quickly learned not to think if you could avoid it.
- ¹² A common example we’ve observed is where a modern application deployed on Amazon Web Services used Amazon’s CloudWatch service to watch a number of Amazon EC2 instances that all take load from an Amazon elastic load balancer. Developers set CloudWatch to fire up a new server if the demand/response time rises above a defined threshold or spin servers down when demand drops.
- ¹³ Source: Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*, Wiley, 1996; Enterprise Integration Patterns (<http://www.eaipatterns.com/PipesAndFilters.html>); and OpenLoop Computing (http://www.openloop.com/softwareEngineering/patterns/architecturePattern/arch_Broker.htm).
- ¹⁴ Accelerometers, gyroscopes, and magnetometers are already present on individual chips in some phones. They will migrate to a single chip. With the addition of a barometer, a single chip will have 10 degrees of freedom. The significance of using a single chip lies in the speed of calculation and ease of development. Device manufacturers will be the first to leverage these technologies. Source: Yole Développement (<http://www.yole.fr/>).

¹⁵ Apple's Siri voice assistant encodes voice commands and sends them to an off-device cloud service unless the command can be handled locally. Source: Andrew Nusca, "Say command: How speech recognition will change the world," *Smart Takes*, November 2, 2011 (<http://www.smartplanet.com/blog/smart-takes/say-command-how-speech-recognition-will-change-the-world/19895>).

Google has announced that Google Voice Search is now processing language queries on board the device.

¹⁶ Touchwalls exists in many forms today, including the 82" multitouch display manufactured by Perceptive Pixel. Source: Perceptive Pixel by Microsoft (<http://www.perceptivepixel.com/>).

Alternative technologies like picoprojectors are set to take the technology even further by making it possible to project an interactive surface onto a bare wall. Source: "SixthSense," Pranav Mistry (<http://www.pranavmistry.com/projects/sixthsense/>).

¹⁷ For more information on how the market for wearables is evolving, see the April 17, 2012, "[Wearable Computing](#)" report.

¹⁸ All major mobile platforms offer a headless browser control that allows developers to embed HTML and JavaScript capability into a native application. In iOS, this control is called the UIWebView; in Android and WinRT, it's called a WebView.

¹⁹ A geofence is a virtual fence that detects when a device with location-based support enters or exits the area it defines. Applications use this data to start processes, allow (or shut off) network access or instrument usage, and alert users to important information. Geofences commonly use GPS or A-GPS technology.

About Forrester

A global research and advisory firm, Forrester inspires leaders, informs better decisions, and helps the world's top companies turn the complexity of change into business advantage. Our research-based insight and objective advice enable IT professionals to lead more successfully within IT and extend their impact beyond the traditional IT organization. Tailored to your individual role, our resources allow you to focus on important business issues — margin, speed, growth — first, technology second.

FOR MORE INFORMATION

To find out how Forrester Research can help you be successful every day, please contact the office nearest you, or visit us at www.forrester.com. For a complete list of worldwide locations, visit www.forrester.com/about.

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Focuses On Application Development & Delivery Professionals

Responsible for leading development and delivery of applications that support your company's business strategies, you also choose technology and architecture while managing people, skills, practices, and organization to maximize value. Forrester's subject-matter expertise and deep understanding of your role will help you create forward-thinking strategies; weigh opportunity against risk; justify decisions; and optimize your individual, team, and corporate performance.

« **ANDREA DAVIES**, client persona representing Market Insights Professionals

