

For: Application
Development
& Delivery
Professionals

Building High-Performance Mobile Experiences

by Michael Facemire and Jeffrey S. Hammond, July 15, 2014

KEY TAKEAWAYS

Great Mobile Experiences Require Great Performance

Engaging mobile experiences are half user interface, half performance. If one fails, any amount of success in the other is lost.

Performance Must Be A Focus Throughout The IDEA Cycle

Traditional development processes focus on assessing performance at the end, right before delivery. Many mobile teams ship code with “good enough” performance, hoping to improve it in a later version of the app. But mobile consumers often don’t allow developers multiple chances, so a focus on performance throughout the development process is required.

Complexity Hurts Performance

Simplicity of architecture and experience is always a goal in development, but it’s mandatory to deliver great mobile performance. Complexity can creep in in a number of ways, both at the mobile client and back-end services, resulting in key performance failures.

Offline Support Drives Amazing Performance

Engineering client-side caching solutions for offline usage can be incredibly challenging, but the performance benefits are almost always worth it. Keep performance in mind when considering investment into offline scenarios.

Building High-Performance Mobile Experiences

Performance Management: The Mobile Application Development Playbook

by [Michael Facemire](#) and [Jeffrey S. Hammond](#)

with [Christopher Mines](#), [Rowan Curran](#), and Eric Wheeler

WHY READ THIS REPORT

The early days of enterprise mobile app development focused on creating a functional experience and getting it out the door quickly, leaving “quality” as a problem for another day. As organizations mature their development and delivery of mobile applications, traditional software quality fundamentals settle back into the life cycle, with some key changes to testing processes and team structure. App performance is an example of how mobile changes quality assurance. Performance is a key pillar of quality that mobile shops often overlook in their race to market. This report dives into the problem areas that impede mobile performance, and provides guidance on this not-so-black-and-white area of quality.

Table Of Contents

2 **Five-Star Experiences Demand Five-Star Performance**

Three Challenges Of Mobile Performance

4 **Performance Must Be A Focus Throughout The IDEA Cycle**

Identify Performance Hazards In The Mobile Moment

Design For Performance In Likely Contexts

Engineer High-Performance Apps With Radical Simplicity

Analyze The Experience To See Results And Establish Guidelines

7 **Offline Apps Are Great Performing Apps**

RECOMMENDATIONS

8 **Establish A Culture That Demands Performance**

WHAT IT MEANS

9 **Increased Focus On Mobile App Performance Will Spur The Spread of Virtualization Services**

11 **Supplemental Material**

Notes & Resources

Forrester interviewed nine mobile vendor and services companies, including Compuware, EPAM Systems, HP, KPMG Digital Services, Mobiquity, New Relic, Perfecto Mobile, Soasta, and WillowTree Apps.

Related Research Documents

[The Engagement Platform's Aggregation Tier](#)
May 27, 2014

[Organize For Mobile Development Success](#)
April 22, 2014

[A Benchmark To Drive Mobile Test Quality](#)
February 20, 2014



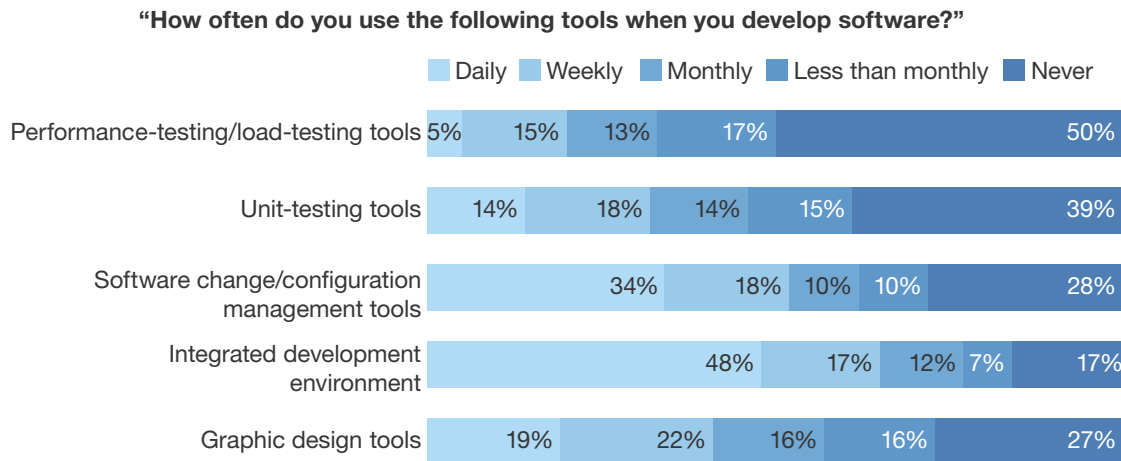
FIVE-STAR EXPERIENCES DEMAND FIVE-STAR PERFORMANCE

Success in mobile app development means creating great customer experiences. The mobile moments an app serves are made up of two parts: first, the user interface, where we see amazing innovations almost daily. The other (often underserved) half is performance. A great user interface will win accolades, but great performance is a baseline customer expectation. The only time performance drives conversation — and drives down app ratings — is when it's poor. Unfortunately this conversation is never a good one.¹ Our research data shows that half of North American and European mobile developers never use performance testing tools, leading us to believe that these unfortunate conversations are happening too often (see Figure 1)!²

And it's not just customer facing apps that need to perform. Employee efficiency and productivity often drive the business cases for internal apps.³ A major worldwide beverage distributor used internal contractors for on-site fountain repair. During the course of each inspection, the field worker would look up the fountain schematic, determine the broken part, and finally look up the part number to order for the repair. The distributor built a mobile app that compressed this process to roughly 11 minutes, but didn't focus on measuring app performance. The distributor's business leaders thought they could do better. By simply improving the performance of the app (using techniques described later in this report), this field service process was further reduced to less than 4 minutes, with the goal of getting down to 2 minutes with an ongoing agile approach to iterative improvement. With thousands of repairs yearly, improving app performance saved millions.

What defines “great” performance in mobile and how do you know if your app is meeting or exceeding these guidelines? The rest of this report will review the techniques and tricks we've uncovered in our conversations with top-notch mobile development teams.

Figure 1 Performance Is Not A Priority For Today's Mobile Development Teams



Base: North American and European mobile and mobile web app developers.

Note: Select responses shown

Source: Forrester's Forrsights Developer Survey, Q1 2013

92121

Source: Forrester Research, Inc.

Three Challenges Of Mobile Performance

In the early stages of the mobile mind shift many development teams look at performance as a challenge to be solved simply by writing quality code. But code quality is only part of the challenge. A high-performing app results when development teams:

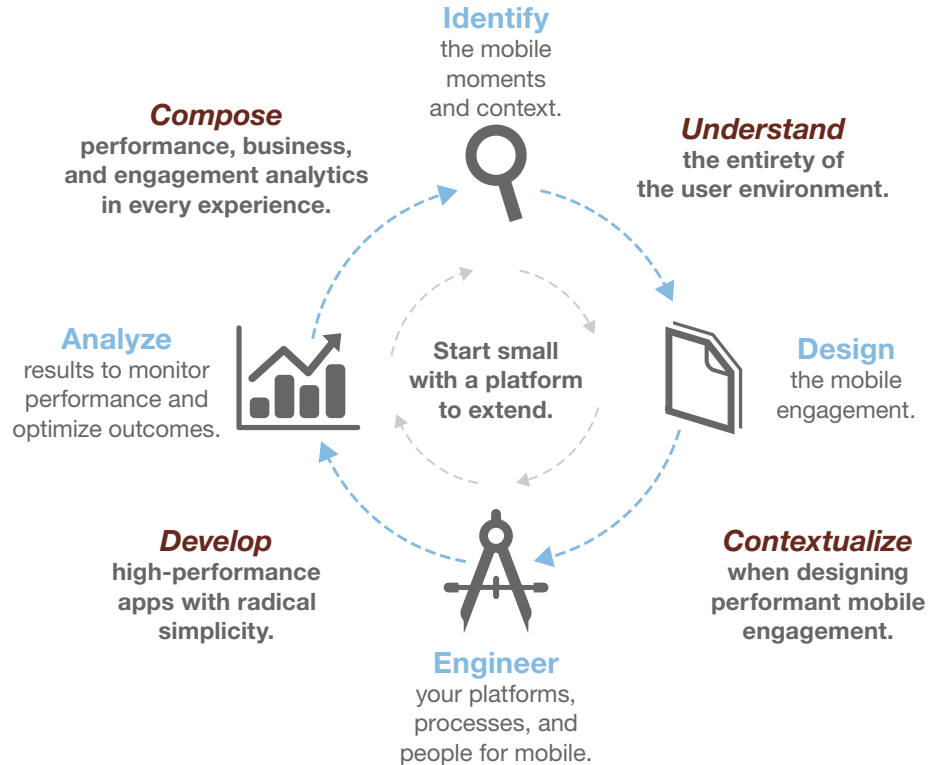
1. **Define performance criteria before cutting code.** Current industry standards say all mobile web pages should respond in less than 1 second regardless of network conditions.⁴ Mobile apps should follow a similar guideline. There are tools that provide competitive benchmarks for apps in a given industry that teams can use to identify average response times and set their best-in-class targets.
2. **Establish metrics to quantify and continuously measure this performance.** Once performance criteria have been established, they need to be quantified and added to the overall quality standards. Too often we hear of developers knowing what the guidelines are, yet something slips through the cracks because these guidelines aren't defined and tested as a part of a continuous testing cycle.
3. **Push the app to exceed these criteria.** Only with defined and quantified performance criteria can developers shift their focus to the code that implements the app. With these first two challenges solved, all development must include performance as a fundamental component of feature delivery.

PERFORMANCE MUST BE A FOCUS THROUGHOUT THE IDEA CYCLE

Forrester recommends that app development teams adopt the IDEA cycle to identify, design, engineer, and analyze mobile experiences.⁵ Where does mobile app performance fit into this cycle? At every stage.

Traditional software development processes relegate performance testing to a spot right before delivery, thinking they need to integrate the system before they can accurately measure end-to-end performance. The result was often a web application that loaded more slowly than expected and frenzied work at the end of projects to address systemic problems in the architecture. In the web world, a customer faced with a slow-performing page may simply switch browser tabs while the application loads and eventually switch back to complete their task. Similar mobile context switches are catastrophic (and career limiting for development teams), because users can choose from a plethora of apps (and providers) to complete an undertaking. Additionally, with business stakeholders involved in demos at the end of every sprint, waiting until the last development sprint to focus on performance puts a slow app in front of an important audience many times — and a team on the painful end of a recurring conversation. Move performance focus earlier — not just in development, but also into every phase of the IDEA cycle (see Figure 2).

Figure 2 Focus On Performance During Every Phase Of The IDEA Cycle



Identify Performance Hazards In the Mobile Moment

As teams identify the mobile moments when customers will interact with an app, they must understand the entire customer environment including: potential network availability, timeframe for task completion, and what security criteria are acceptable in this moment. Solutions for these constraints will be defined later in the IDEA cycle; at this point, simply look for the performance hazards that may appear so that they can be included in the app design.

Design For Performance In Likely Contexts

With the potential hazards identified, the design phase of the IDEA cycle should include both user interface and performance design. The latter should include both mobile app and back-end service performance. Using the example hazards identified in the previous phase, dev teams should consider the following during design:

- **If poor network conditions are possible, design cache scenarios.** Nothing enhances the performance of a mobile app more than a proper cache strategy, on both the front- and back-end. This may be as simple as downloading the field service manuals or sales guides the first time the app is used. It may be as complicated as a full store-and-forward offline scenario. Walk through the app use cases with the network conditions in mind; design caches to handle transient network conditions and to optimize around static data and view templates.
- **If an action doesn't require authentication, don't.** Ty Rollin, chief technology officer of Mobiquity sums it up well: "Identification and authentication of users is the No. 1 performance item for a mobile experience." The first interaction a user has with an app is often a login screen, yet some basic use-cases (and some user roles) may not require access to corporate data. Forcing a login for every user and every interaction ensures the initial experience is slow and also requires authenticated service calls that introduce more complexity and more opportunity for performance failure. During design, map users' tasks to back-end data; if three or more of these tasks don't require authenticated access, limit the login requirements to only the requisite flows.
- **If a user's task will be long running, design the front- and back-end to stay active.** With the caches designed above, nothing is worse than using the app, working away from it, and coming back to find out you've been logged out and need to log in again. Even worse, if a large amount of content is pre-cached and your security guidelines specify the removal of on-device data during logout, this becomes an even worse performance nightmare. Design for these scenarios while balancing corporate security and governance policies.⁶

Engineer High-Performance Apps With Radical Simplicity

Simple apps are high-performance apps. Software developers covet elaborate solutions to complex problems. Simplifying the problem (and thus the result) will establish a higher likelihood of app success. With that in mind, keep an eye on the following during the engineering phase of the IDEA cycle:

- **Divert a user's attention during long load times.** Users do not want to wait to interact with your app — distract them if necessary, but be transparent. Platform games were the forerunners in this tactic, they used cut scenes to distract players while large binaries were loaded into memory. Mobile apps can use a similar model, but transparency is critical. If the network is congested or a third-party component is causing the delay, consider alerting the user and they can take appropriate action such as switching to a different Wi-Fi hotspot.⁷
- **Rethink security for mobile.** The average mobile app session lasts roughly 72 seconds.⁸ With that in mind, teams cannot waste precious seconds with a forced login. Innovative mobile security paradigms are starting to appear that can help; for instance, some use biometrics (e.g., Apple Touch ID), physical geolocation, or not requiring logins on trusted networks. Back-end session timeouts also need to be considered. When policies force caches to be cleared on logouts and repopulated on logins, short timeouts that were acceptable on the web must be lengthened for long-running mobile tasks.
- **Watch for back-end performance problems.** When an app runs slowly, everyone is quick to blame the client-side mobile developer. Yet often the answer lies deep within the technology stack supplying data to the app. When creating new (or modifying existing) services for an app, start with a stubbed-out version that responds immediately with fixed, non-computed data. Set a performance guideline based on this use case, and monitor transaction times during the course of development. Alternatively, some teams we interviewed proactively injected long delays in back-end APIs during test runs to ensure their front-end could handle those delays appropriately. Tools are available to monitor these “full-stack” traces as well, such as CompuWare dynaTrace and Nastel jKool.
- **Don't turn the app into a mobile client enterprise service bus.** Modern development teams are building apps against a composition of many back-end APIs. This is good trend, but can be taken too far; don't compose 100 APIs on the client to show a single value in a view. Instead, create an API on the back-end to do this composition. It's possible that this composed data doesn't often change, allowing for caching and further performance improvement.
- **Don't make serial API requests.** If data from one API is needed to request the next, make a single request from the client with the context required to compose the full set of requests on the back-end. Avoid this solution as well if possible. Asynchronous back-end computing models, which are limited in their usefulness in these serial request-response cycles, are becoming popular.
- **Scrutinize third-party code for relevance.** It is never too soon to start removing complexity from an app, excluding common-but-complex code. Be careful when including libraries in mobile apps; ensure your use case justifies it. Often these packages are doing much more than needed and therefore dragging down performance. Code inspection tools and performance profilers are essential in finding these offenders.

Analyze The Experience To See Results And Establish Guidelines

The analyze phase of the IDEA cycle is the one that most development teams associate with performance, yet too often it's the first time they analyze performance. Our guidelines recommend a change in that behavior, yet post-production performance monitoring will still be critical.⁹ Application performance monitoring is just the first step; if performance is critical to your organization do the following:

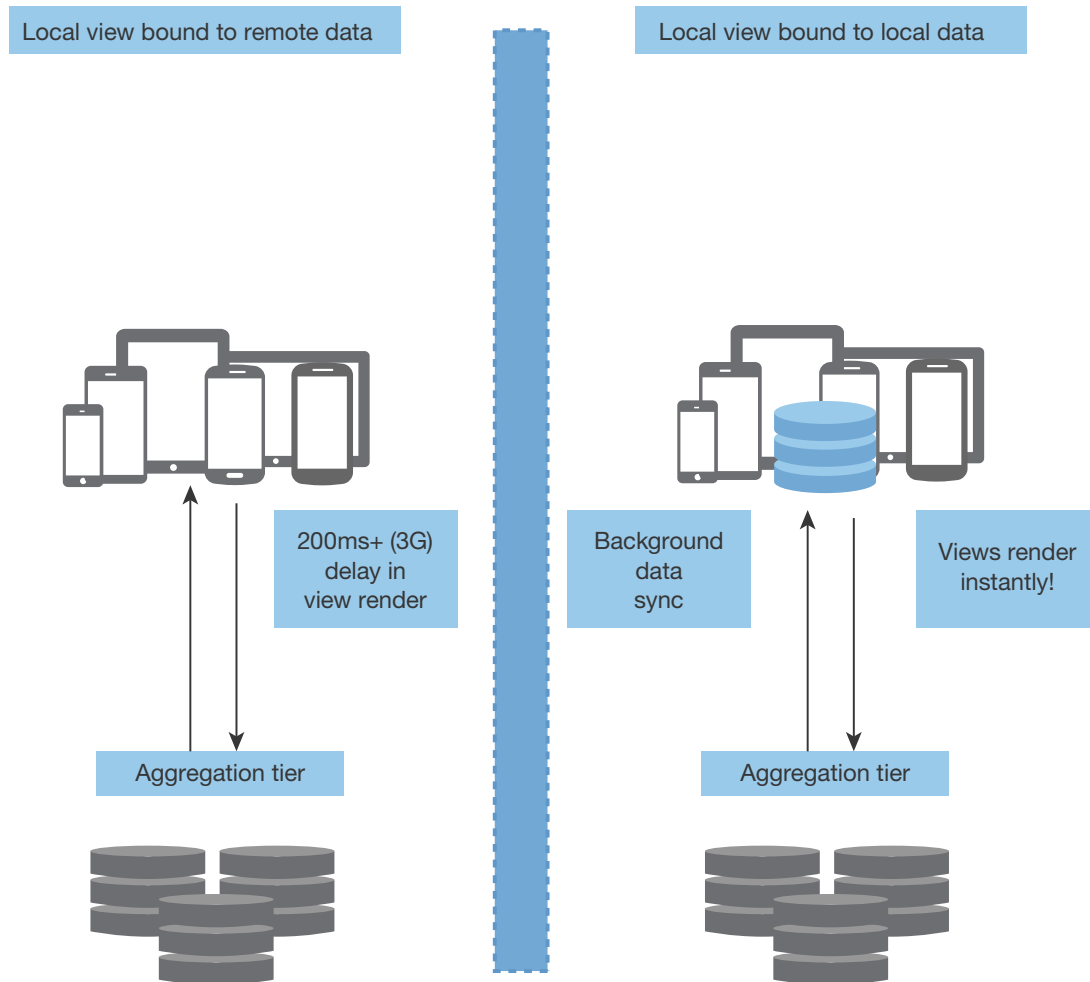
- **Compose performance, business, and engagement analytics for every experience.** Poor performance results are often relegated to the bottom of the development backlog. Instead, create dashboards or use analytics packages that can report of the performance status of business and engagement analytics for mobile experience. Look for correlations between app performance and gains or losses in business/engagement metrics that make fixing performance issues problems much easier to justify to business sponsors.¹⁰
- **Use an analytics platform that allows *multiple* people to monitor performance.** More than one person in the organization should be concerned with performance, yet mobile development platforms often only allow one user to see these technical statistics at a time. Nearly all vendor analytics packages allow for multiple users — do not overlook this investment.
- **Pay attention to the outliers; these will drive future app direction.** One team we interviewed noticed that their app saw the most traffic during the day, but showed poor performance at night, which seemed counter-intuitive. Due to their inclusion of robust analytics in the app, it was determined that server-side compute power was being throttled at night due to a service-level agreement that was put in place before mobile was ever a consideration. A change in this agreement was quickly made, but would have never been found if it weren't for active analysis.

OFFLINE APPS ARE GREAT PERFORMING APPS

Many organizations specify “offline access” as a required feature of their app, but quickly remove that feature when the development teams responds with the estimate of what it will take to architect and build offline support. There's no doubt that offline data synchronization is one of the most difficult challenges developers face when building multi-tier systems (and some would contend it's never been completely solved). If your development team is considering it, know that it will greatly increase the perceived performance of your app.

Apps that make network calls to populate data within views will lag when the network lags because the views are bound to network connections. Apps that support offline use cases will bind the views to a local data source (either the SQLite database or the local file system) and use asynchronous background processes to update these local data stores (see Figure 3). Doing this ensures that all views *immediately* render with the data that's stored locally. Supporting robust transactional offline use cases can be incredibly challenging, but there are commercially available solutions starting to appear.¹¹ The performance benefits of this offline support may very well justify the investment.

Figure 3 Offline Use Cases Drive High Mobile Performance



92121

Source: Forrester Research, Inc.

RECOMMENDATIONS

ESTABLISH A CULTURE THAT DEMANDS PERFORMANCE

Development teams typically want to focus on mobile app performance, but all too often business sponsors prioritize new features over sustained performance engineering. Our research found that organizations that cultivate a culture that demands performance don't fall prey to these prioritization challenges. To create this culture, focus on the following:

- **Performance teams are most effective when they include developers.** All developers know where the challenging parts of their code are, and great developers thrive on optimizing away their past mistakes. Inject developers into the performance team to refactor problems and to act as a knowledgeable sounding board for other development teams, either on a full-time or rotational basis. Additionally, when placing people on this team, find problem solvers that can think outside of the box. This is more critical than years of finding the same performance issues in common source code.
- **Start with some quick wins.** A culture of performance can only grow with nurturing from the business and technology management sides of the house. Investigate high-usage, known revenue-driving or cost-saving apps; show a performance win here that's easy to quantify to start the positive momentum.
- **Allow performance successes to go viral.** First, share successes with the business as stated above, and then have these leaders share their excitement with the organization through internal social channels or via your mobile community of practice. These wins, along with continual support from internal evangelists, will keep the mantra of high-performance apps top of mind for everyone in the organization.
- **Don't make the performance guru the speaker for the performance organization.** It's not uncommon that the individual who's best at finding performance flaws lacks tact and subtlety in communicating them. The result is that the message of poor performance is delivered with a whip as opposed to a helping hand. Competition runs strong in development teams; telling one that their baby is ugly (in terms of performance) will lead them to ignore the bearer of this bad news in the future. No developer wants to write poorly performing code, so it's usually a question of not enough time or not enough skill. Both issues can usually be addressed in a constructive manner, using a mentorship model.

WHAT IT MEANS

INCREASED FOCUS ON MOBILE APP PERFORMANCE WILL SPUR THE SPREAD OF VIRTUALIZATION SERVICES

The increase in velocity and the proliferation of platforms will change how we test and monitor apps in two ways.

First, mobile apps will continue to be tested while in production, and this feedback needs to align with feedback given during development. Whenever possible, therefore, the monitoring tools used in development should be the same ones used in production. Otherwise you'll run into impedance mismatches and too much manual interaction will be required to marry the two data sets.

Second, the tools used to ensure mobile app quality will themselves need to react to real-world conditions that are unique to the users of your app(s). Tools are emerging now that will coalesce to form a complete mobile quality platform, built on five pillars:

- **Device virtualization brings the set of global devices into your lab.** Testing against a global portfolio of devices (especially Android) is cost prohibitive, if physically possible at all. By providing virtual access to real devices in real environments, this becomes closer to a reality. The next revision of this technology will allow us to create virtual environments on the device, e.g., loading it up with 100 apps, limiting the memory, throttling CPU, and more. How will we know what environments to create? The user base will tell us through analytics!
 - **Network virtualization ensures the app performs in all network conditions.** The performance of an app that relies on network connections to populate views is highly variable based on current network conditions. These need to be tested against all foreseeable network conditions during development, leading to the need for network virtualization tools like HP Shunra.
 - **Service virtualization keeps back-end and front-end development schedules separate.** Front-end development teams need to deliver on a separate cadence than back-end service teams. Simple service virtualization tools will be used to enable mobile clients to build against services that may not be developed yet, or against services that charge per request and therefore should be virtualized during development.
 - **Data virtualization is required anywhere real-world data is governed.** Service virtualization is one half of the data access story; but even a virtualized service has to provide data. That data is either private or prohibitively expensive to use in many industries. This ushers in the need for data virtualization, implemented as modern test data management tools. These will fit hand in glove with modern service virtualization tools.
 - **User virtualization changes for mobile.** Your mobile app users are unique to your competition, and often unique across the suite of apps you build. To optimize the user experience for a uniquely personalized experience, tools will be built to virtualize a user profile, or segment. This will include the business processes they run, parts of the app they use, times of day they use it, data sets in use, etc. Virtualizing this and making it available in both pre- and post-production can optimize real-world user habits in the cross-channel brand experience.
-

SUPPLEMENTAL MATERIAL

Methodology

Forrester's Forrsights Developer Survey, Q1 2013, was fielded to 2,038 software developers located in Canada, China, France, Germany, India, the UK, and the US from companies of all sizes, as well as students and freelancers. This survey is part of Forrester's Forrsights for Business Technology and was fielded during February 2013 and March 2013. ResearchNow fielded this survey online on behalf of Forrester. Survey respondent incentives include points redeemable for gift certificates. We have provided exact sample sizes in this report on a question-by-question basis.

Each calendar year, Forrester's Forrsights for Business Technology fields business-to-business technology studies in more than 17 countries spanning North America, Latin America, Europe, and developed and emerging Asia. For quality control, we carefully screen respondents according to job title and function. Forrester's Forrsights for Business Technology ensures that the final survey population contains only those with significant involvement in the planning, funding, and purchasing of IT products and services. Additionally, we set quotas for company size (number of employees) and developer type as a means of controlling the data distribution. Forrsights uses only superior data sources and advanced data-cleaning techniques to ensure the highest data quality.

Companies Interviewed For This Report

CompuWare	New Relic
EPAM Systems	Perfecto Mobile
HP	Soasta
KPMG Digital Services	WillowTree Apps
Mobiquity	

ENDNOTES

- ¹ Facebook is a great example of a company that experienced the growing pains of mobile performance. As detailed in their blog post, the drive towards speed in the mobile app led them through three major app revisions on iOS alone. Source: Jonathan Dann, "Under The Hood: Rebuilding Facebook For iOS," Facebook Engineering, August 24, 2012 (<https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920>).
- ² For information on mobile web performance considerations, see the May 20, 2014, "[Is Your Responsive Site Performing?](#)" report.
- ³ For information on mobile strategy see the January 22, 2013, "[Putting A Price On Your Mobile Strategy](#)" report.

- ⁴ Source: “Google Developer PageSpeed Insights” (<https://developers.google.com/speed/docs/insights/mobile>).
- ⁵ For a more in-depth description of the Forrester Research IDEA Cycle, see the January 24, 2014, “[Re-Engineer Your Business For Mobile Moments](#)” report.
- ⁶ On device encryption coupled with “sensible” data fading policies (2-7 days) are tactics that can extend the life of data on device, for example.
- ⁷ Verizon famously pointed out when the underlying ISP rate-limited their data during the net neutrality wars. This may be an extreme example of being transparent! Source: Jacob Kastrenakes, “Netflix Tests Warnings That Blame Internet Providers For Poor Streaming Quality,” The Verge, June 4, 2014 (<http://www.theverge.com/2014/6/4/5778770/netflix-warning-members-when-service-providers-cause-bad-speeds>).
- ⁸ Source: Matthias Bohmer, Brent Hecht, Johannes Schoning, Antonio Kruger, Gernot Bauer, “Falling asleep with Angry Birds, Facebook and Kindle — A Large Scale Study on Mobile Application Usage,” University of New Mexico Department of Computer Science and Engineering (http://www-users.cs.umn.edu/~bhecht/publications/bhecht_mobilehci2011_sleepbirds.pdf).
- ⁹ For more information on monitoring mobile apps, see the June 28, 2012, “[Build Great Mobile Apps That Drive Engagement](#)” report.
- ¹⁰ For more information on engagement and business analytics in the mobile space, see the November 18, 2013, “[Measuring Mobile Apps](#)” report.
- ¹¹ These challenges will be addressed in a future document by the author, and will include analysis of emerging solutions offered by CouchDB, Zumbero and Alpha Software.

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 the 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 Application Development & Delivery Professionals