Preparing for the Flood: 6 Steps to Build Apps at the Speed of Mobile

appcelerator®
NOW PART OF AXWAY

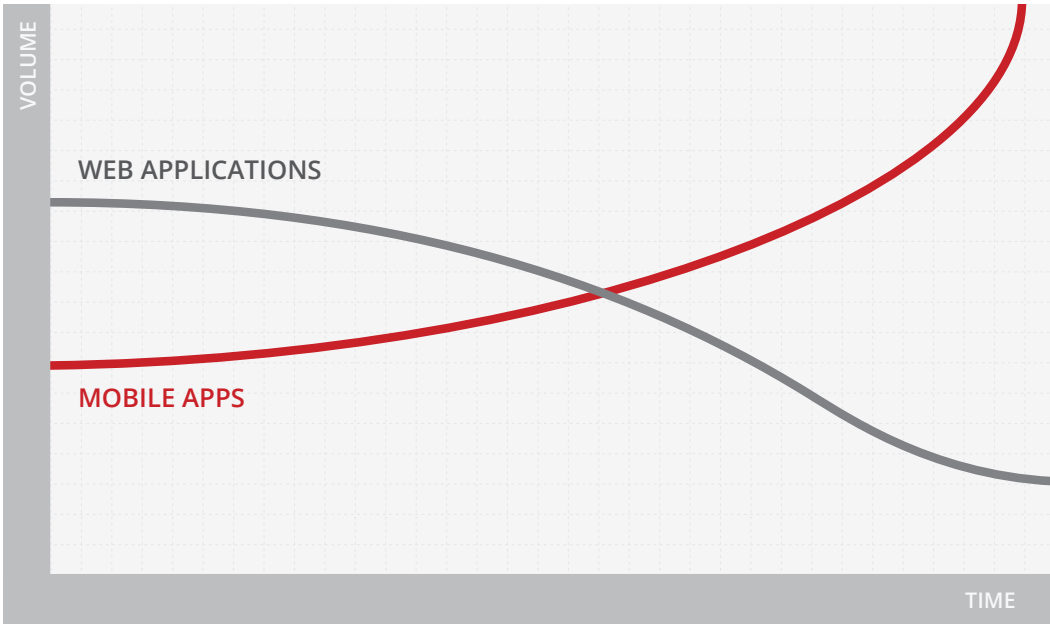
axway
business. in motion.™

appcelerator®
NOW PART OF AXWAY

Preparing for the Flood: 6 Steps to Build Apps at the Speed of Mobile

For most companies, building the first few mobile apps is no sweat. Throw in enough specialized resources, and presto—problem solved.

The real trick is when you realize *all* your software projects will be mobile, that mobile has become the default paradigm, just as the web was before it. Now it's no longer a specialized problem: it's your entire business. And with



© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

that realization comes another—scale-wise, there are dozens, even hundreds of apps on the way. The trickle has become a deluge.

We're fast emerging from the first wave of mobility, where a business might have only a handful of consumer apps. Looming behind this wave is the second, in which businesses count on mobile to transform not only routes to market but their internal processes—and expect development teams to be ready for the challenge.

> "I don't think it's too much hyperbole to say mobile is as big, if not bigger, than the opportunities we had 25 years ago when the first desktop PCs came out. We are in that kind of shift..."
>
> —Jeffrey Hammond, *Forrester Research*

Admitting We Have a Problem

As we noted in *Mastering the Mobile Shift: Three Keys to the New Enterprise*, there's a temptation to see mobile as merely an extension of the web—same basic principles, smaller screen. This is understandable.

From a software development perspective, the web's legacy is baked into our design strategies, the delivery tools we use, how we connect to systems of record, our processes and timelines... basically everything.

But as it turns out, the operational and infrastructural impacts of mobile are big. In fact, the turn from web applications to mobile apps is no less trivial than the last great architectural shift, from client-server to web. Mobile differs from everything that's come before in several important ways:

1. **The range of platforms and devices.** Gone is the Wintel monopoly. No longer are we creating applications to run on a single set of devices of fixed screen sizes. Now the need is for apps that run across a range of devices, a range that seems to expand by the quarter. Our most recent *survey* saw 83.9% of developers reporting their organizations are supporting two or more mobile platforms (iOS, Android, Windows, etc.).

2. **The number and variety of data sources.** While most people are awake to the device explosion, fewer see that an equivalent expansion has occurred on the backend. Good mobile apps are greedy things, hungry for all manner of data from enterprise systems

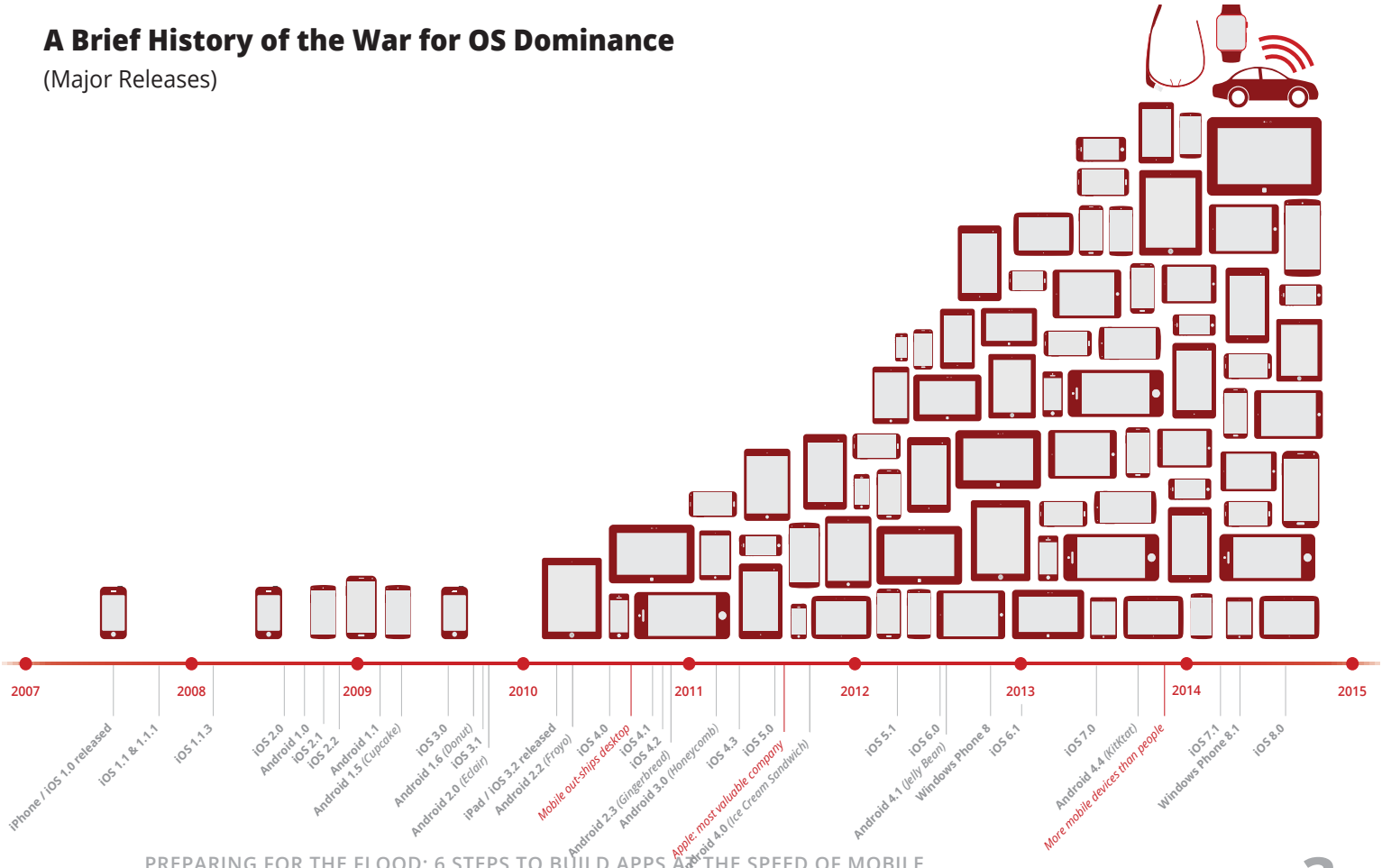
© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

to SaaS repositories, public sources such as social to the looming Internet of Things. And the ways these data are accessed and consumed are quite different.

3. **The rise of the user.** Not long ago, users were second-class citizens. What mattered was the system. Users existed to enter data in precisely the way the system wanted it, and if they got it wrong it was their fault. (Recall the old "user error" jokes.) Not any more. In a mobile world, users expect smart, elegant, context-aware apps that orchestrate the data they need regardless of source, and which run anywhere, anytime on their device of choice. And if your app can't deliver that experience? The user—whether customer, partner or employee—will refuse it.

One upshot of these changes is the way release velocity is moving outside enterprise control. Apple and Google (and soon perhaps, Microsoft) are in an arms race for mobile OS supremacy. This means each vendor pushes multiple upgraded OS versions each year, driving the apps built for that

A Brief History of the War for OS Dominance
(Major Releases)



© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

platform to be upgraded too. This is to say nothing of the expectation of mobile users, who expect a regular stream of improvements and differentiated features to keep their chosen apps relevant.

> "When I talk to a company that asks me about what they need to do in the mobile space from a dev perspective, the first thing I tell them is that you need to build an organization that is capable of building and releasing an app 6–10 times a year."
>
> —Jeffrey Hammond, *Forrester Research*

For anyone building and maintaining more than a couple of mobile apps, this new reality presents a kind of hellish arithmetic. It goes something like this:

```
                          (Total # Apps)
                 (# Mobile OSes Supported)
X                     (Releases Per Year)
-------------------------------------------
# of Deliveries That Had Better Amaze
```

In other words, mobile is driving expectations for superlative app experiences, even as it increases both the number of device+OS combinations we must build for, and the rate of release. This is a dizzying prospect for development shops, many of which were already under strain before mobile appeared on the scene. To solve for this ratcheting complexity, we must radically lower the time it takes to implement and validate new ideas—essentially a reinvention of the application lifecycle for the app, using velocity as the organizing principle.

Velocity as the Organizing Principle

If the goal is to be able to build and release six to ten updates a year for each of your apps, it means examining your team structures and processes and optimizing around velocity. Automate where you haven't already. Identify and eliminate the handoffs that our legacy lifecycles were built around. Here are six such opportunities to strip away latency and miscommunication to create a modern app lifecycle that can move at the speed of mobile.

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

1. Make the app the requirement

A great place to start is the long-standing "bucket brigade" of handoffs from requirements to design to build. For rapid mobile delivery, you must euthanize the requirements elicitation phase, the weighty Product Requirements Document, and the requirements database. These approaches are needlessly time-consuming, and worse, are poor at communicating actual user needs. It's far easier to validate direction by looking at a mock-up or prototype, which is why many teams are adopting UI prototyping as a way of visually expressing requirements.

> "A prototype is worth a thousand meetings."
>
> — Mike Davidson, *VP of Design,* Twitter

This needs to be taken a step further though. Today, the speed with which you can iterate on an app when using technology like the Appcelerator Platform means there is no longer the need to manage separate and distinct artifacts for requirements, design, prototype, and the app itself—each with its own lifecycle and ceremonies. For instance, an early version of the actual app UI can serve as a visual prototype that can be put in the hands of stakeholders and quickly refined.

By treating it as a single evolving artifact, all energy and feedback is directed into the actual product instead of intervening documents. Not only are waste and delays striped from the process, it eliminates the "telephone game" of misinterpretations and resulting rework that have long plagued software development.

2. Embrace cross-platform development

Of course, the first and most visible challenge enterprises hit when pursuing mobile is coping with the multitude of devices and operating systems. In certain instances, for certain kinds of apps, using HTML5—i.e., for web or hybrid apps—can relieve some of these challenges. But unfortunately it's not a cure-all *(see sidebar)*. Where higher levels of user experience, performance, and security are desired, native apps still reign supreme. Cross-platform native development—the ability to build fully native apps for every mobile OS from a single code base—dramatically reduces the effort for building great apps without compromising on the user experience.

(If this sounds too good to be true, read on. The Appcelerator Platform provides exactly this native, cross-platform development. Our recent customer ROI analysis has shown 60-90% code reuse, depending on the scenario, and a 40% time-to-market improvement with 52%+ lower development costs[1]—a huge gain for this tactic alone.)

WHAT ABOUT HMTL5?

Doesn't HTML5 solve mobility's cross-platform headaches using common web skills? Unfortunately, HTML5 is really an umbrella term for a basket of web standards that require formal submission, review, approval, ratification, etc. This means it advances at its own pace, rarely in coordination with the various platforms and browsers on which it must run. Today, the differential in HTML5 feature support across browsers is upward of 30%.[2] Furthermore, as the release of iOS 7 showed, conforming to a separate body of general standards isn't always a top priority for the platform providers, who understandably want to differentiate the capabilities of their own operating systems rather than write to a generalized mean.[3] Finally, HTML5 web apps (and their hybrid cousins) are cut off from all but a fraction of the APIs available to native apps, leaving a much smaller palette of capabilities for developers to work with. The result of all these limitations is often a significantly compromised user experience. This is not to say HTML5 can't be effective for certain kinds of apps. It's particularly suited to content-driven apps such as news, where information is continually updating and the intermediary of an app store only creates a bottleneck. But the above limitations mean HTML5 looks bound to remain one more tool in the toolkit, not a silver bullet.

[1] Forrester Research Inc. "Total Economic Impact of The Appcelerator Platform" July 2014

[2] "BII REPORT: Why Facebook Defriended HTML5-For Now." *Business Insider*. N.p., 24 Oct. 2012. Web. 27 Sept. 2013.

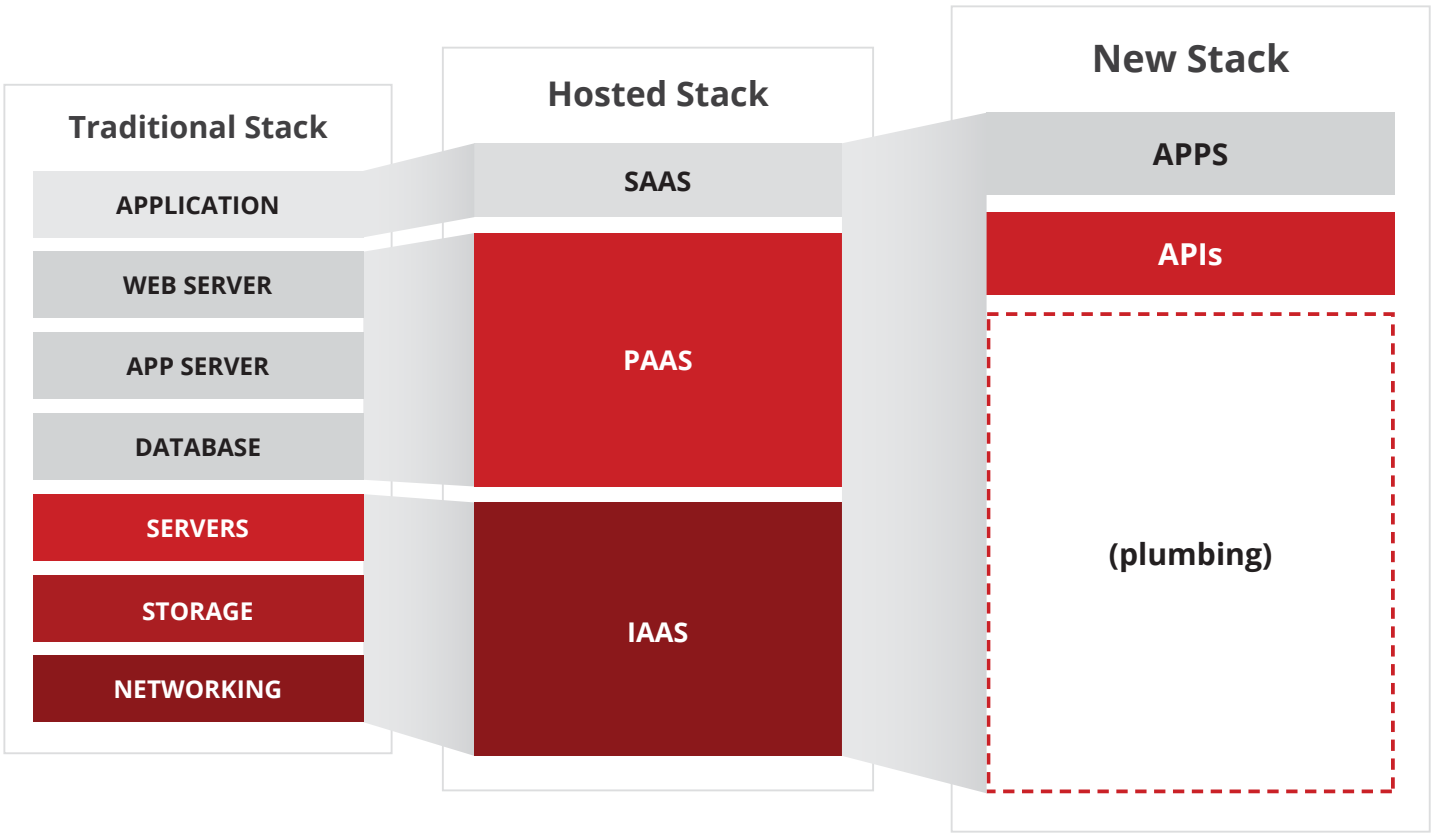
[3] Krill, Paul, and Galen Gruman. "Bad News: IOS 7's HTML5 Is Full of Bugs." *InfoWorld*. InfoWorld, 27 Sept. 2013. Web. 30 Sept. 2013.

3. Abstract the plumbing

Great apps need access to great data to be useful. But the word "integration" makes both developers and managers shudder. Getting mobile apps to talk to systems optimized around web protocols (or older) is a time-consuming and painful process. And frankly, it's boring! The reason we got into mobile was to create compelling experiences for our users, not spend our days knitting together backend data sources.

A mobile API strategy is the answer. This is more than API management, which addresses the management and cataloging of APIs. True speed comes with having a simple way to create mobile-optimized APIs for easy data access. A solid API strategy lets teams spend less time worrying about the minutia of each layer of the traditional stack—the plumbing—and focus instead on creating great user experiences.

This isn't nearly as difficult as it might sound—not with the right solution. For more, see: *APIs & MBaaS: How to Extend Your Enterprise Architecture for a Mobile World.*



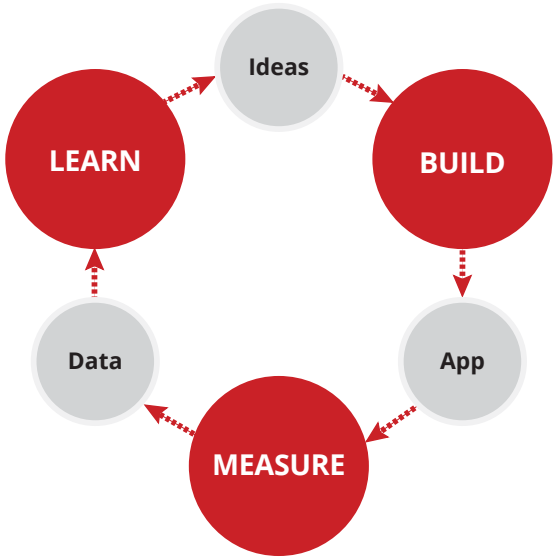
Wanted: the means to build great experiences across any device, quickly.

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

4. Adopt an MVP mindset

Agile taught us that short, frequent release cycles are better than long, artifact-heavy ones. With the rise of mobile, organizations have to go a step further, focusing on minimum viable product (MVP) releases.

The essential idea of MVP is to identify the smallest increment of valuable functionality and get it into the hands of your users, quickly, and then—just as quickly—improve it based on its actual use. As Eric Reis, author of *The Lean Startup* puts it, MVP allows teams to gather maximum amount of validated learning about customers with the least effort.[4]



By using feedback from actual users to guide decisions, you're no longer managing by educated guess or the assumptions of a few "subject matter experts." You can be more confident that what you are building is the right stuff—and when it isn't, you can adapt and change course quickly in the very next release.

5. Tighten the feedback loop with analytics

The operative MVP phrase is "validated learning." It is this feedback—fast and continuous—that fuels the mobile app lifecycle. Some teams constantly scan app store reviews for feedback, but relying on this alone has drawbacks:

- only a small subset of users are represented
- only extreme opinions are captured and the information is often incomplete ("I love it!" or "I hate it!" often without an understanding of why)
- user reviews are ultimately a lagging measure of success

To wring additional speed from the app lifecycle, analytics should be used to shorten the feedback loop and let you find any problems before your users do. Real-time data can show every dimension of app adoption and usage,

[4] Reis, Eric, "Minimum Viable Product: A Guide" *Startup Lessons Learned*, 3 August 2009

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

including a trove of new information never before available—things like user location, device type, app version, OS version and device orientation, just to name a few. This lets you pinpoint the exact user type and scenario encountering a problem. Analytics can even identify the culprit file and line of code causing the issue so it can be quickly fixed and redeployed.

Moreover, mobile analytics also arms you with the hard "validated" data on exactly how you should improve your apps for better business results. We outline the five key metrics you should be measuring in *Stop Guessing, Start Measuring: 5 Mobile Metrics for Great Apps*.

6. Validate with velocity

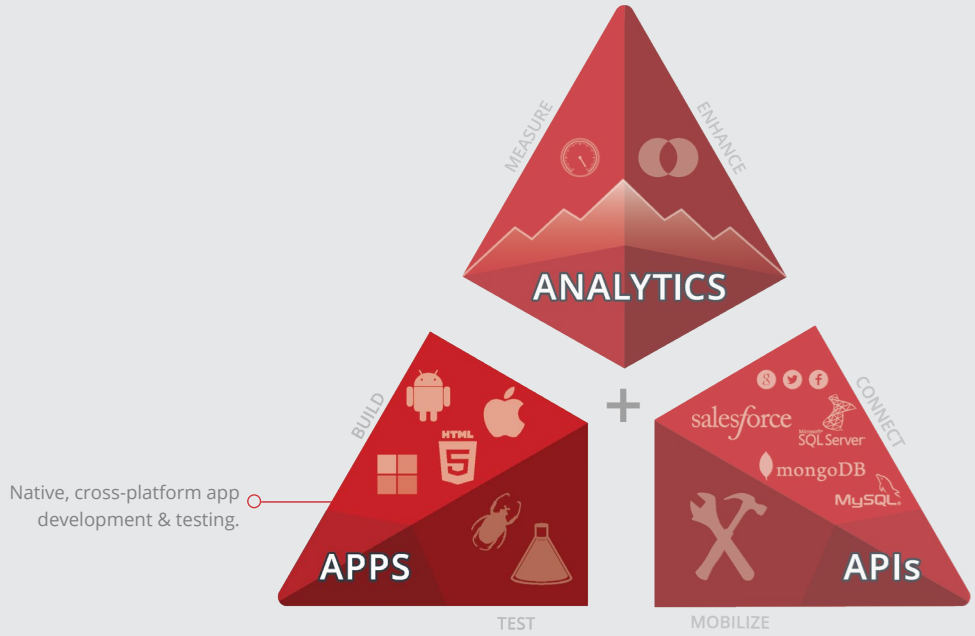
Faster delivery is of little value if quality is low. But overwhelmed by the urgency to release, mobile teams can lose sight of core tenants of good development such as ensuring quality and performance. Overlooking or omitting these things for the sake of speed actually slows you down longer-term in the form of buggy apps and an unwieldy portfolio. The dilemma is how to incorporate these fundamentals and still deliver at the wildly faster pace of mobile.

Conventional wisdom suggests you must trade quality for greater speed. This no longer holds true. Our MVP approach means we are delivering smaller increments of functionality, so it is far easier for testing activities to remain in lockstep with development. The rise of continuous delivery techniques—which emphasize automation and constant verification (rather than treating testing as one big push at the end)—has shown that achieving speed *and* quality isn't a zero-sum game. Continuous testing becomes even more compelling given mobile's expanded array of device and OS combinations and the ability to run the same automated script across all these variations.

Test automation should be thought of as a first line of defense (coupled with a solid analytics strategy). The goal isn't to have an exhaustive catalog of every conceivable test case, but rather coverage of your essential flows. For each code change, your continuous integration process should trigger automated scripts that progressively verify your app at the various levels—e.g. unit, build, regression, performance & security, and acceptance. If a problem is found at any step, it's fixed immediately and reenters the continuous test flow. This enables rapid feedback, for faster, higher-quality releases without getting bogged down in endless cycles of manual verification.

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

THE APPCELERATOR PLATFORM



Have Your Cake and Eat It Too

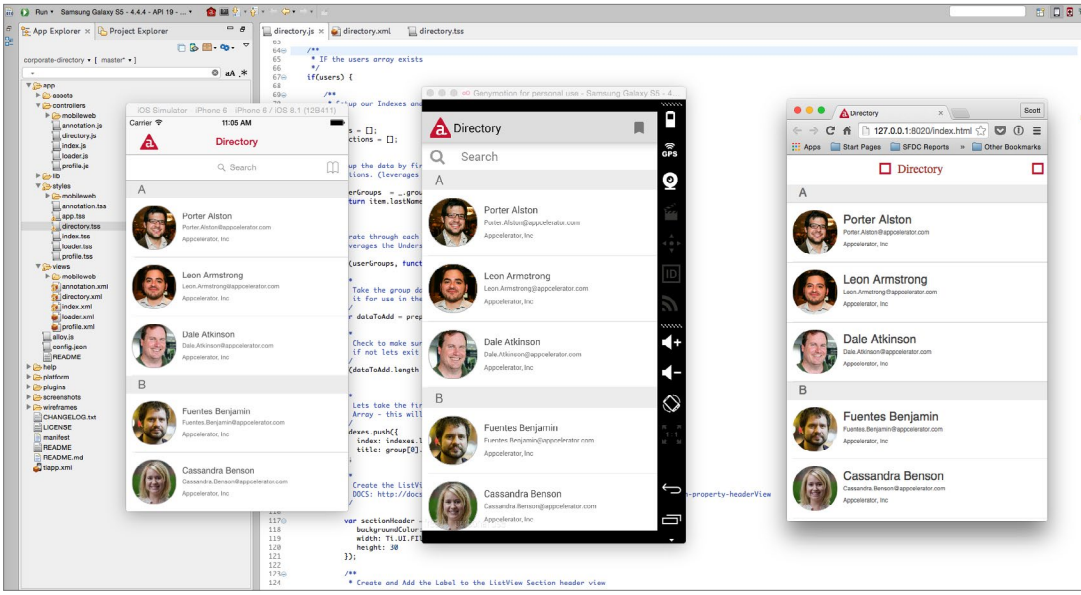
One of the most persistent mobile myths is that while native apps deliver better performance, security and usability, they are also more costly and time-consuming to build. The Appcelerator Platform is engineered to kill this myth for good. With the Platform, you can build great, fully native mobile apps for every mobile operating system, **all from a single JavaScript code base**.

APPCELERATOR PLATFORM : APPS :

IDE and Titanium® SDK

The Eclipse-based IDE and Titanium SDK provide an open, extensible development environment for creating cross-platform native apps. It delivers fully native experiences (as well as hybrid and HTML5) on all major mobile operating systems from a single JavaScript codebase. Developers

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.



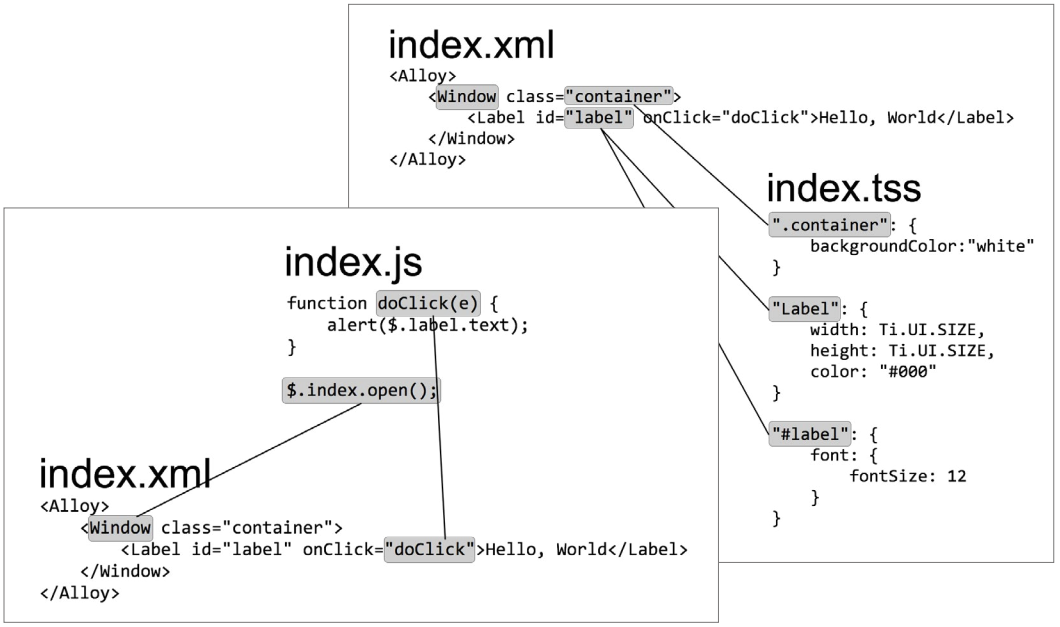
Eclipse-based IDE and mobile device simulators and emulators.

can rapidly build, test, package and publish mobile apps to both public and enterprise app stores. With live prototyping, an on-device debugger, code analyzer, performance profiler and integrations to common source code and build systems, developers have a single place to build great apps.

BENEFITS

- 60-90% re-use of code across device platforms
- Delivers fully native apps, bringing rich user experience, best in class performance and enterprise-grade security
- Immediate support for each new OS release
- Seamless integration to existing continuous delivery systems (e.g. SCM, build, and MDM solutions)

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.



MVC Framework allows you to separate the UI definition from styling and logic

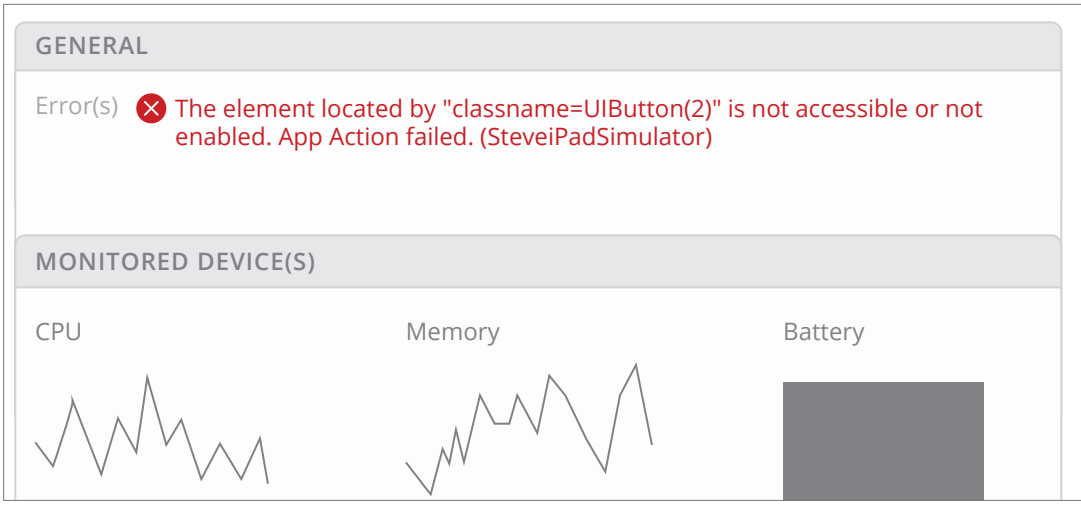
Integrated MVC Framework

The Platform's integrated MVC Framework drives rapid development of high caliber mobile apps by facilitating a model-view-controller architecture. Using XML and CSS syntax, it provides a simple model for separating the app user interface, business logic and data models. The MVC Framework significantly decreases the amount of time and code required to build an app, making the code easier to read, manage and re-use. It is built on Node.js with Backbone.js and Underscore.js support.

BENEFITS

- Designers can prototype in days, with full reuse in the production app
- Developers can build fully reusable components in JavaScript
- Easier to read and maintain code

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.



Live testing and error reporting all within the Appcelerator Platform.

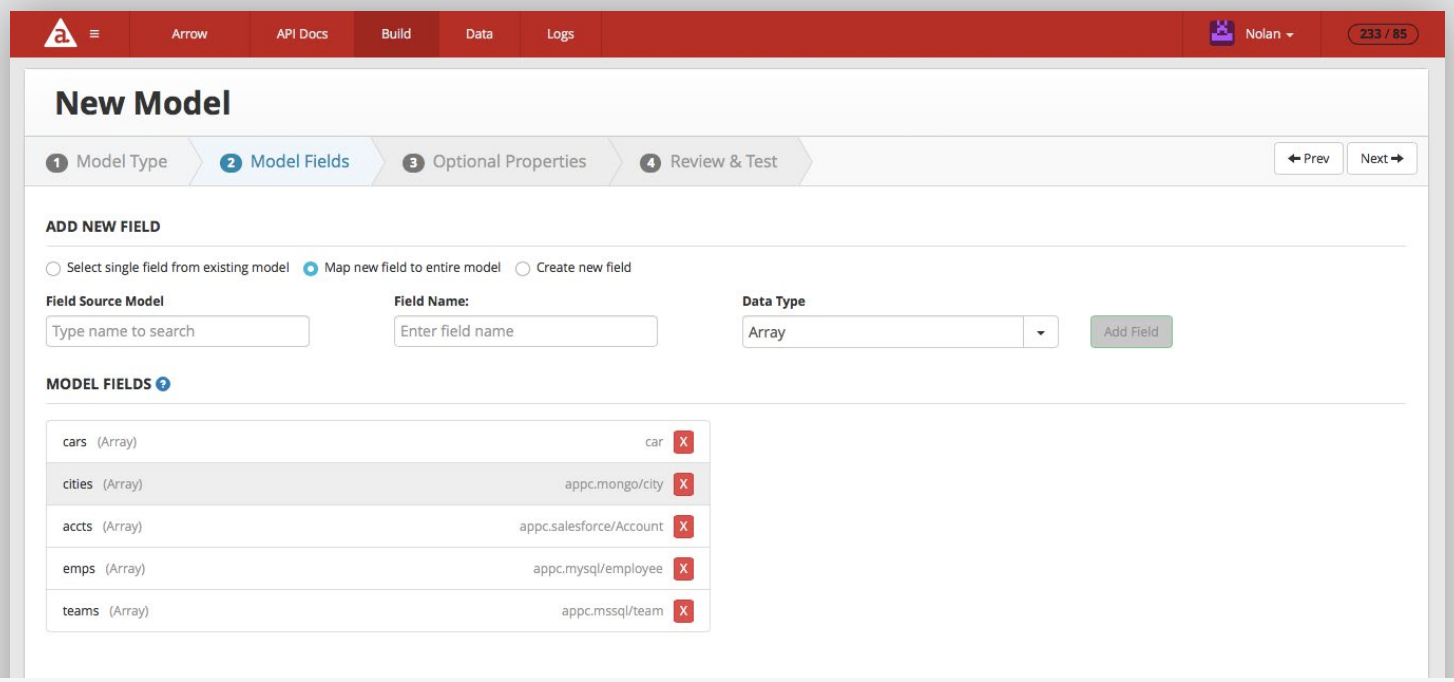
Mobile Test Automation

Coupled with the development environment, the Platform provides complete mobile test automation for fast, through assessments of quality and app behavior across platforms. It includes a test management environment for creating, managing and maintaining automated tests as well as the ability to execute tests on real devices in your lab or on devices in the cloud. It features automatic context sensitive capture and reply (including support for continuous touch gestures such as pan, pinch, zoom, and scroll on iPhones, iPads, iPods and Android devices), unattended replay of regression test suites, and integration with build automation tools for continuous testing.

BENEFITS

- Eliminate labor-intensive and error-prone manual testing
- Increase test coverage on all devices and OS's by 10x
- Decrease test time by 90% and app project costs by 40%
- Create durable test assets that don't rely on character recognition and don't require jailbreaking

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.



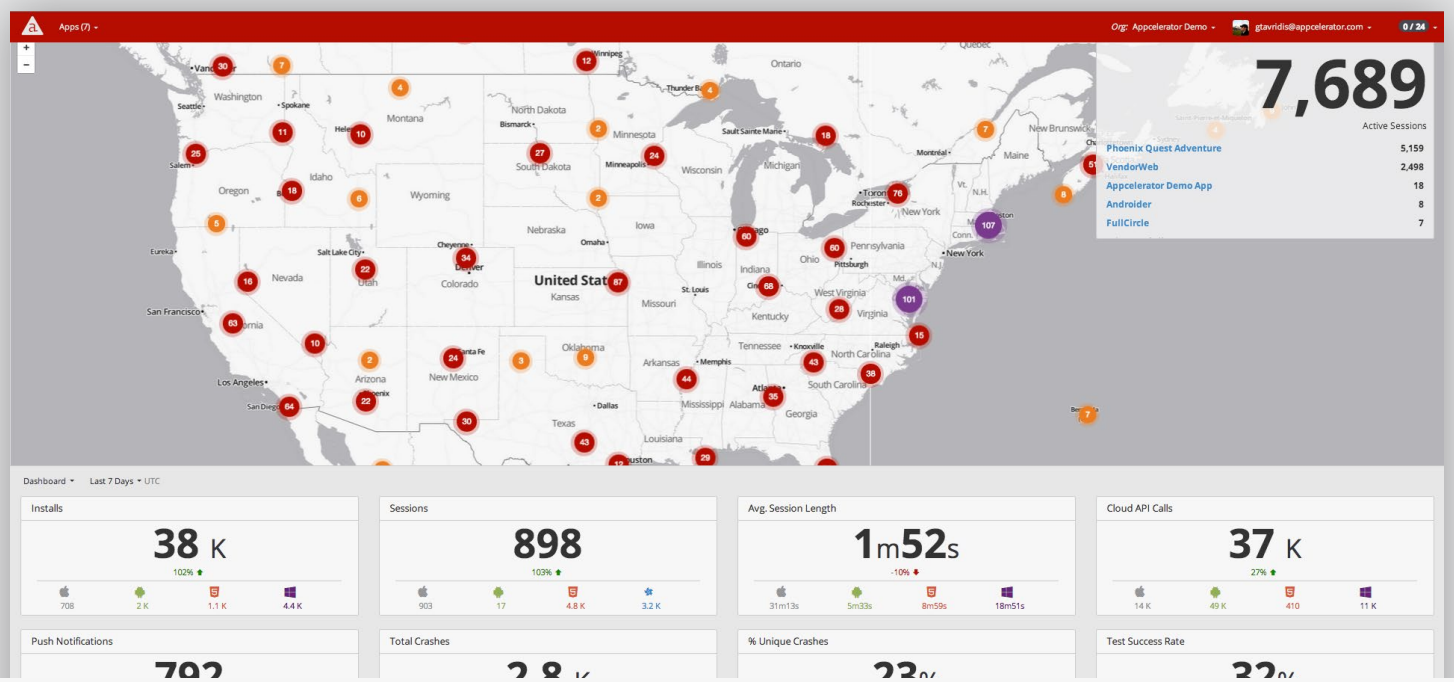
APPCELERATOR PLATFORM : APIs

Mobilize any data source in minutes

The Appcelerator Platform includes Arrow, a next-generation MBaaS for accessing data sources with unprecedented ease. Arrow provides a powerful new framework for building APIs with an elastically scalable cloud service for running them. With Arrow, you can quickly connect, model, transform and optimize data for any client, whether native or web.

- Create custom, reusable connectors to any data source. Optimize payload size and data format for mobile
- A scalable cloud object store to instantly deploy data models with zero setup effort and deliver data to any client
- Prebuilt connectors for Salesforce, MS Azure, MS SQL, MySQL, Mongo DB, Box, Swagger and others
- 20+ turnkey mobile services including, push notification, geo-location, photo storage, authentication and key value pair store
- Flexible deployment options: public, virtual private or private

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.



APPCELERATOR PLATFORM : ANALYTICS

Improve apps continuously with real-time analytics

The Appcelerator Platform delivers a new breed of analytics, helping to ensure that users of every stripe have a smart, responsive, and memorable app experience. This includes both a mobile lifecycle dashboard and a dashboard for executive insights for complete transparency of activities across the entire mobile app portfolio.

- Single 'pane of glass' into all lifecycle activities (test, APIs, app usage, performance, etc.)
- Instantly see user acquisition, engagement, conversion, and retention
- Proactively find an fix problems
- Enhance the user experience based on real insights

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

From Feature Bloat to Purpose-Built

As Gartner has astutely noted, "apps" and "applications" are fundamentally different things.[5] Applications are prized for their long pedigree of features, while apps are valued for their purposefulness—doing just a few things well. Apps can evolve faster and deliver value quicker than applications. Expectations for elegant experiences and constant improvement of apps far exceed those of traditional applications where simply functioning correctly was a high standard. Consider the implications of these differences:

- **Fewer features means faster cycles**—the bite-size nature of apps naturally lends itself to iterating and improving much faster than we ever could with monolithic applications. It also means smaller teams with fewer handoffs, less overhead, and better collaboration.
- **Context heightens engagement**—the 24/7 attachment to our mobile devices, their awareness of factors like location, and new capabilities such as push notification allow us to rethink business processes and provide meaningful interactions in a way the blind, desk-bound PC application never could.
- **Fast, modern architectures**—freed from the constraints of legacy web architectures, mobile encourages (requires?) companies to establish an API strategy for fast, simple access to data so developers can focus on innovation and great user experiences.
- **Data-driven decisions**—the rise of analytics capabilities that can be auto-deployed into our apps means we no longer have to make decisions based on guesswork and gut-feel. We now have the real-time information on app behavior, user sentiment, and business goals. This upends the traditional notion of "requirements gathering" and tells us precisely what should go into the next release.

The net result is profound. More than "just another channel," mobile changes the way we design, build, distribute, and optimize software. With the right steps (and right solutions), your organization can master the modern, mobile lifecycle—and your users will thank you.

© 2016 APPCELERATOR, INC. ALL RIGHTS RESERVED.

Why wait? Start free.

Get full trial access to the Appcelerator Platform. Everything you need to build, test, connect and measure great mobile apps.

Sign up for a free account

SHARE THIS WHITEPAPER:

RECOMMENDED READING:

Mastering The Mobile Shift: Three Keys To The New Enterprise

APIs & MBaaS: How To Extend Your Architecture For A Mobile World

Stop Guessing, Start Measuring: 5 Mobile Metrics for Great Apps

About appcelerator® NOW PART OF AXWAY

Mobile apps are fast replacing web applications as the way we buy, share, search, learn and collaborate. Appcelerator® helps companies of all sizes solve for this new reality with a mobile engagement platform to drive great mobile app experiences, quickly, cost-effectively and at scale. The Appcelerator Platform speeds time to market with native cross-platform app development & testing, simplified access to data via mobile-optimized APIs, and comprehensive real-time analytics to power user engagement and measure success. With 100,000+ mobile apps deployed on 300+ million devices, Appcelerator's solutions are backed by one of the world's largest mobile ecosystems, including better than 700,000 mobile developers and hundreds of ISVs and strategic partners, among them Accenture, Cognizant and CSC.

For more information, visit **www.appcelerator.com**.