



SELECTIONS FROM

THE DZONE GUIDE TO

MOBILE

DEVELOPMENT

2015 EDITION

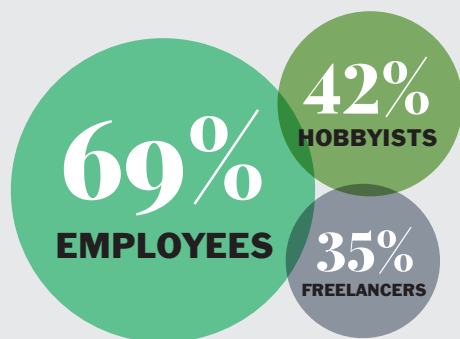
RESEARCH PARTNER SPOTLIGHT



# Key Research Findings

DZone surveyed over 500 IT professionals with some involvement in mobile technology for the 2015 *Guide to Mobile Development*, providing key information on mobile monetization, tool usage, production levels, and platform prioritization. The largest demographic segments are developers (30%) and development team leads (26%). 44% of respondents come from small organizations (under 100 employees) and 56% come from large organizations (100 or more employees). The majority of respondents are headquartered in the US (41%) or Europe (34%).

## 01. TYPES OF MOBILE DEVELOPERS



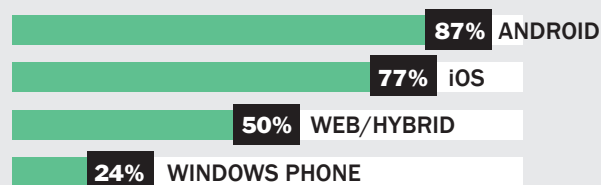
## MORE MOBILE HOBBYISTS AND EMPLOYEES THAN FREELANCERS

An overwhelming majority (70%, up from 51% last year) of those surveyed work in an organization that develops mobile apps, and 69% (up from 48% last year) of all respondents have participated in their organization's mobile development. Freelancing is still not as popular (35% of respondents develop mobile apps on their own, down from 37% last year) as those developing as hobbyists (42%), which have also seen a decrease from last year's results (down from 56%). 19% of respondents don't expect to see a return on investment from their apps, which is down from 30% last year.

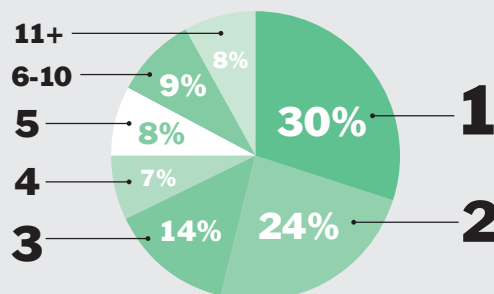
## ANDROID STILL MORE POPULAR THAN IOS, BUT NOT AS MUCH; NATIVE DEVELOPMENT UP OVERALL

Android continues to be the most popular platform for organizations and individual developers, with 87% of respondents saying they are targeting the Android platform. iOS has closed the margin, with 77% of respondents saying they are targeting iOS—the difference last year was 14%. Interestingly, non-native development seems to be decreasing slightly: 50% of respondents say they are developing for web or hybrid (down from 56% last year). Windows Phone remains a distant third, targeted by 24% of respondents.

## 02. WHICH DEVICE PLATFORMS DO YOU OR YOUR ORGANIZATION TARGET FOR MOBILE APPLICATION DEVELOPMENT?



## 03. IN THE PAST YEAR, HOW MANY MOBILE APPLICATIONS HAS YOUR ORGANIZATION RELEASED?



## INDIVIDUAL DEVELOPERS BECOMING MORE EFFICIENT

App development time can vary widely based on a host of factors, but finding out how much time it generally takes organizations and individuals to develop apps is useful for discovering industry opinions about how long an app should take to complete or how large a project should be. The top three answers for organizations were 12 weeks (15%), 8 weeks (12%), and 4 weeks (12%), which is exactly the same as last year. This suggests consistency for app turnaround time within organizations. For individuals working by themselves, timetables gravitate toward 8 weeks (15%), with 4 weeks (13%) and 6 weeks (11%) close behind. These timetables are all smaller than last year's results, which could be explained by a number of factors—more legacy code, more APIs, or more developer knowledge. As for the number of apps completed per year, chart #3 shows statistics for the number of apps organizations are churning out.

## CROSS-PLATFORM TOOLS USED BY NEARLY HALF OF MOBILE DEVS

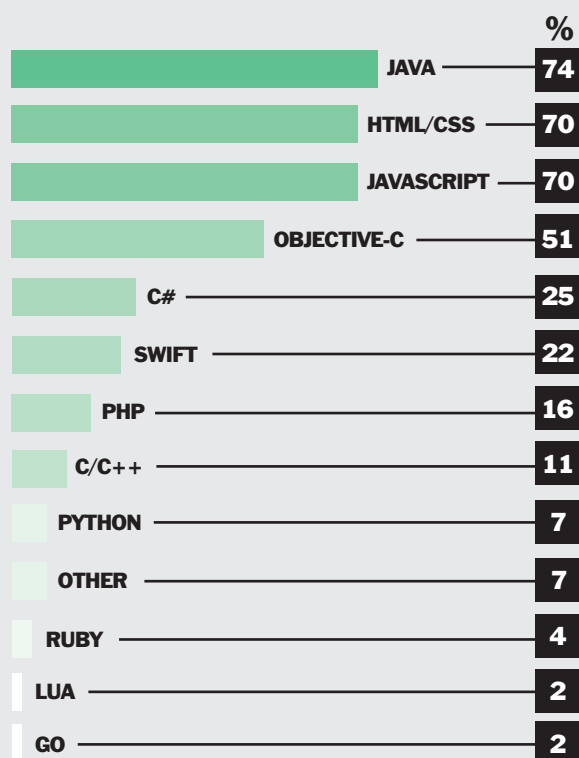
51% of respondents say they or their organizations are using cross-platform tools such as Apache Cordova/PhoneGap, up from 41% last year. This is the most

popular type of mobile tool, with IaaS/PaaS as the second most crucial development utility at 29% usage (up from 20% last year). For programming languages, Java is very popular in this group of enterprise developers (74%), more popular, in fact than HTML/CSS (70%) or JavaScript (70%). Objective-C for iOS is fourth with 51%, while Swift is being adopted quickly (already at 22%). C#, the language of Windows Phone, is fifth with 25%. The mobile usage percentages for the rest of the programming languages can be seen in the chart below.

## DEVELOPERS STILL RELY ON COMMISSIONED APPS FOR REVENUE

Commissioned apps (41%) continue to be the number one way respondents expect to make money as a mobile developer. The actual purchase of the app (26%) is the second most popular way developers expect to make money from their app. Marketing/brand awareness of another service (23%) and monetizing online content (20%) are other strong options. 19% of respondents said in-app purchases are the way they expect to see a return on investment. Unfortunately, 27% of respondents said they do not expect to see a return on their investment.

### 04. WHICH PROGRAMMING LANGUAGES DO YOU OR YOUR ORGANIZATION USE FOR DEVELOPING MOBILE APPS?



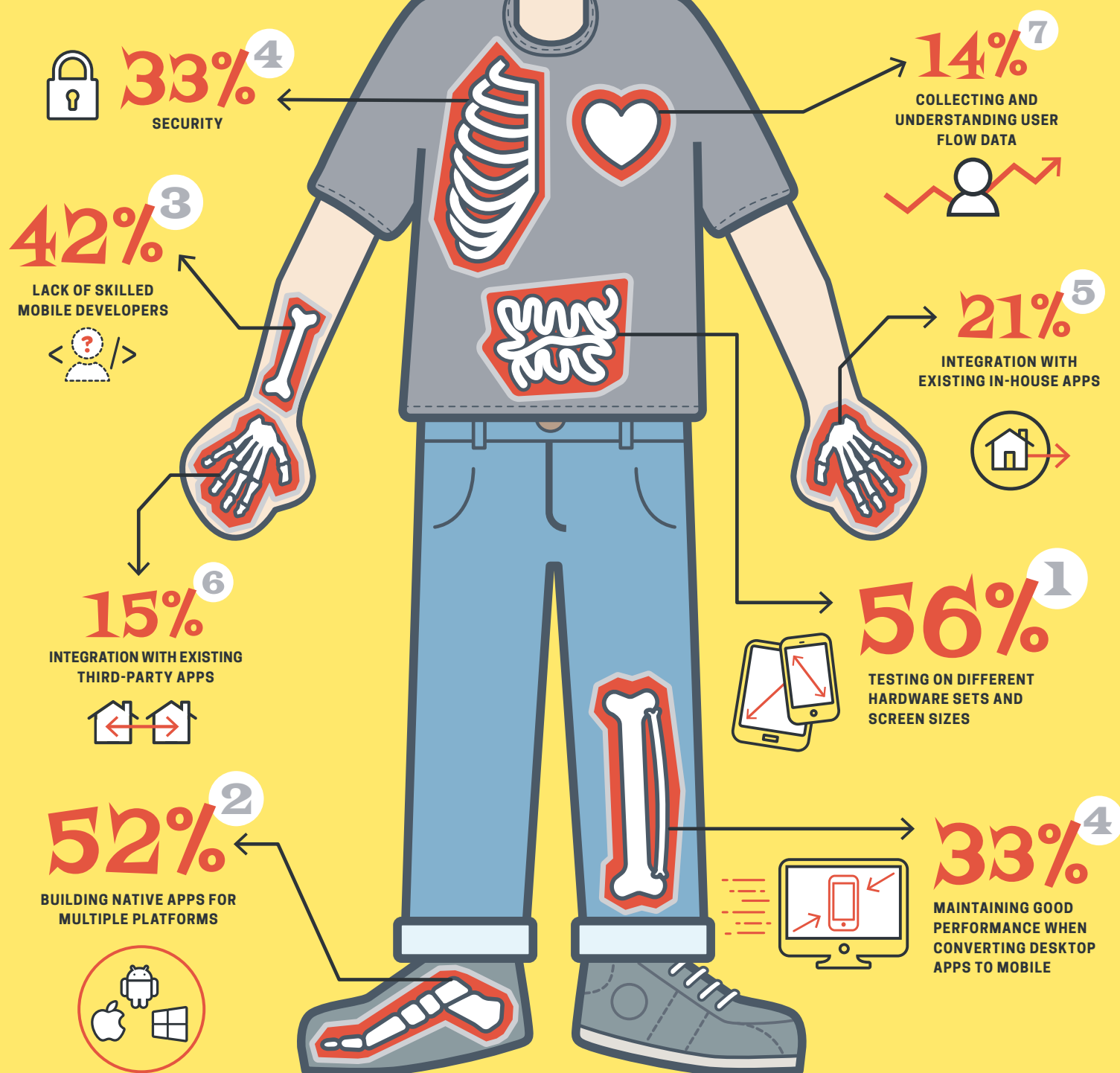
### 05. WHERE DO YOU OR YOUR ORGANIZATION EXPECT TO GAIN MONETARY VALUE FROM YOUR APPS?



# MOBILE DEVELOPMENT PAIN POINTS

Let's face it — mobile development is tough. Developers encounter several problems, often unforeseen, throughout the development process. Facing these problems can set back development by days, weeks, or even months. Not anticipating these issues can cause a lot of frustration and can lose you money.

We asked more than 500 software developers and architects to tell us what causes them pain when they develop mobile applications. Multiple answer selections were allowed. The results are summarized below. The number next to each icon/label indicates the percent of respondents who indicated that they encounter this pain point during mobile development.



# Mobile App Testing Checklist

## Device Checks

- ☐ Can the app be downloaded?
- ☐ Can the app be uninstalled?
- ☐ Can the app be updated?
- ☐ Can the app be updated when multiple updates exist?
- ☐ Can the app be downloaded on multiple operating systems?
- ☐ Can the app be redownloaded?
- ☐ Does the app behave as intended when the device is connected to power?
- ☐ Does the app behave as intended when the device is not connected to power?

## UI / UX

- ☐ Is the user experience as intended?
- ☐ Is the user interface appropriate for every form factor?
- ☐ Is the user able to navigate between various pages within the app?
- ☐ Do all buttons conform to standards (e.g. Share button shares, Trash Can deletes, etc.)?
- ☐ Is the UI consistent with its Desktop equivalent (if applicable)?
- ☐ Does the app allow for touchscreen gestures (e.g. zoom, pinch, swipe, etc.)?

## Integrations

- ☐ Does the app access the device's camera?
- ☐ Does the app access the device's sensors (e.g. accelerometer, gyroscope, etc.)?
- ☐ Does the app access any peripheral devices (e.g. Bluetooth headphones, keyboard, etc.)?
- ☐ Does the app integrate with other apps (e.g. Twitter, Facebook, etc.)?

## Storage

- ☐ Does the app write to a memory card?
- ☐ Does the app read from a memory card?
- ☐ Does the app write to the cloud?
- ☐ Does the app read from the cloud?
- ☐ Does the app function if storage is full?
- ☐ Is the correct data being stored?

## Network

- ☐ Does the app function on Wi-Fi?
- ☐ Does the app function on Cellular data (including 3G, 4G, LTE, EDGE, etc.)?
- ☐ Does the app function on Bluetooth?
- ☐ Does the app function when transitioning between networks?
- ☐ Does the app function in Airplane mode?
- ☐ Does the app function when not connected to any network?

## Security

- ☐ Is any personal data stored on the device?
- ☐ Is all app data removed if the app is uninstalled?
- ☐ Can app data be accessed outside of the app?
- ☐ Does the app authenticate?
- ☐ Can the user be locked out of their account (if applicable)?
- ☐ Can the user reset their password (if applicable)?
- ☐ Does the app require any permissions?

## General App Functionality

- ☐ Does the app perform the functions it is designed to perform?
- ☐ Does the app perform functions it is not designed to perform?
- ☐ Is the user prompted to turn on services (e.g. location services, Wi-Fi, etc.) if necessary?
- ☐ Does the app run on multiple operating systems?
- ☐ Does the app run on multiple devices?
- ☐ Is navigation clear and intuitive?

## Interruptions

- ☐ Does the app function when the user receives a phone call?
- ☐ Does the app function when the user receives a text message?
- ☐ Does the app function when the user receives a push notification?
- ☐ Does the app function when the user receives a battery notification?
- ☐ Does the app function when the user receives a voicemail notification?
- ☐ Does the app function when the device is locked?
- ☐ Does the app function when the device is unlocked?

## Accessibility

- ☐ Does the app perform the functions it is designed to perform?
- ☐ Does the app perform functions it is not designed to perform?
- ☐ Is the user prompted to turn on services (e.g. location services, Wi-Fi, etc.) if necessary?
- ☐ Does the app run on multiple operating systems?
- ☐ Does the app run on multiple devices?
- ☐ Is navigation clear and intuitive?

# Mobile Testing Automation for a CI World

As the mobile economy grows, so does the pressure to provide excellent products. Mobile users demand high quality apps, ruthlessly abandoning or deleting those with bugs. The highly competitive market also demands that mobile teams release apps faster and more often. Add to that the challenge of supporting different device types across a fragmented market, and mobile teams are left with a daunting amount of work to ensure quality, consistency, and velocity.

Automated mobile testing done in a CI workflow can help development teams identify errors quickly and

early on in the development cycle to ensure their app works as expected on the devices and platforms their customers are using. It also reduces the need for time-consuming manual testing by traditional QA teams, leaving time for QA professionals to run more valuable exploratory testing programs. Taking advantage of the ability to run automated tests in parallel is key to reducing build time and shortening dev cycles.

I am a big fan of open source, so the mobile testing platform I recommend for running functional tests is Appium, the world's leading cross-platform mobile automated testing platform (supported by Sauce Labs and a thriving community of open source developers). Appium, which could be described as 'Selenium for mobile apps,' is designed to support testing of native, hybrid, and mobile web apps across many mobile device and platform combinations, and comes with many other benefits; for example, it's easy to integrate into CI and allows teams to use any programming language or test framework.



**WRITTEN BY JONATHAN LIPPS**

DIRECTOR OF ENGINEERING, ECOSYSTEMS, SAUCE LABS

## Mobile Testing Platform by Sauce Labs



*Sauce Labs provides a cloud-based testing platform for native, hybrid and mobile web apps. Users run Selenium, Appium, and JS unit tests written in any language on over 500 browser/OS combinations, 80 mobile emulators, and hundreds of the newest, most popular devices with no wait time.*

### CASE STUDY

Bleacher Report, a leading sports news site, has a monthly audience of more than 16 million unique users with more than 50% of site traffic originating from mobile devices. Previously the QA team tested their native applications manually, but without an automated testing process, their mobile apps had bugs after deployment. Unsatisfied with the process, they searched for a better solution. Recently, they started using Appium with Sauce because they could use it with their existing framework and didn't have to learn a new language or set of commands. They now regularly run 12 native application tests and 7 mobile web tests, each checking specific end points in various scenarios. Because they're run in parallel, the test suite only takes approximately 2 minutes. The mobile web tests are run at 4 different times with different parameters to cover iPhone, iPad, and in-app experiences for both.

### CI TOOL SUPPORT

- Jenkins
- Travis CI
- CircleCI
- Bamboo
- TeamCity

### PRICING

By number of VMs and test run minutes

### CUSTOMERS

- Yahoo!
- Capital One
- Twitter
- Travelocity

### LANGUAGE SUPPORT

**JAVA** **JAVASCRIPT**  
**NODE.JS** **PERL**  
**PHP** **C#** **RUBY**  
**PYTHON**

### APP CATEGORY

- Web
- Native
- Hybrid

### TESTING SUPPORT

- Cross-browser testing
- Real mobile devices
- Mobile emulators
- Mobile simulators
- Appium
- Selenium

### OPEN SOURCE

Yes

**BLOG** [sauce.io](http://sauce.io)

**TWITTER** [@saucelabs](https://twitter.com/saucelabs)

**WEBSITE** [saucelabs.com](http://saucelabs.com)

A brief history of web and mobile app testing.



#### BEFORE SAUCE LABS

Devices. Delays. Despair.

#### AFTER SAUCE LABS

Automated. Accelerated. Awesome.

Find out how Sauce Labs  
can accelerate your testing  
to the speed of awesome.

For a demo, please visit [saucelabs.com/demo](https://saucelabs.com/demo)  
Email [sales@saucelabs.com](mailto:sales@saucelabs.com) or call (855) 677-0011 to learn more.



*Testing at the speed of awesome.*