perfecto mobile

From Traditional Functional Testing to Enabling Continuous Quality in Mobile App Development

Introduction

Today's developers are under constant pressure to launch "killer" apps and release enhancements as quickly as possible – all without sacrificing quality in the process. As a result, agile methodologies featuring Continuous Integration (CI) solutions are rapidly becoming a key element of the modern software development process. When compared to the sequential or waterfall models of the past, the agile CI approach enables organizations to continuously incorporate test feedback throughout the development lifecycle, and incorporating continuous quality (CQ) allows for improved quality while speeding time-to-market.

Continuous Integration is a development practice that requires developers to commit code into a shared repository several times a day. Each commit is then verified by an automated build verification testing. The goal is fast feedback.

Build verification with a limited set of production targets for example leading web browsers (IE, Chrome, Safari and Firefox) lends itself well to this process. However, extending CI practices to mobile app development creates new challenges. Instead of dealing with a small number of browsers that are periodically updated, mobile app dev teams must execute on a far larger set of development targets and keep pace with rapid innovation on both the device and OS level. This means testing apps on a slew of device profiles – the performance of which are all susceptible to variable network conditions.

> **Due to perceived complexities, many organizations are struggling to extend CI to the mobile app development process.**

This whitepaper will serve as a primer for using CI and CQ to enhance mobile app development projects and ensure agility throughout the software development life cycle (SDLC). It details the processes and the types of tools needed to automate best-practice functional, performance and usability testing in an efficient manner.

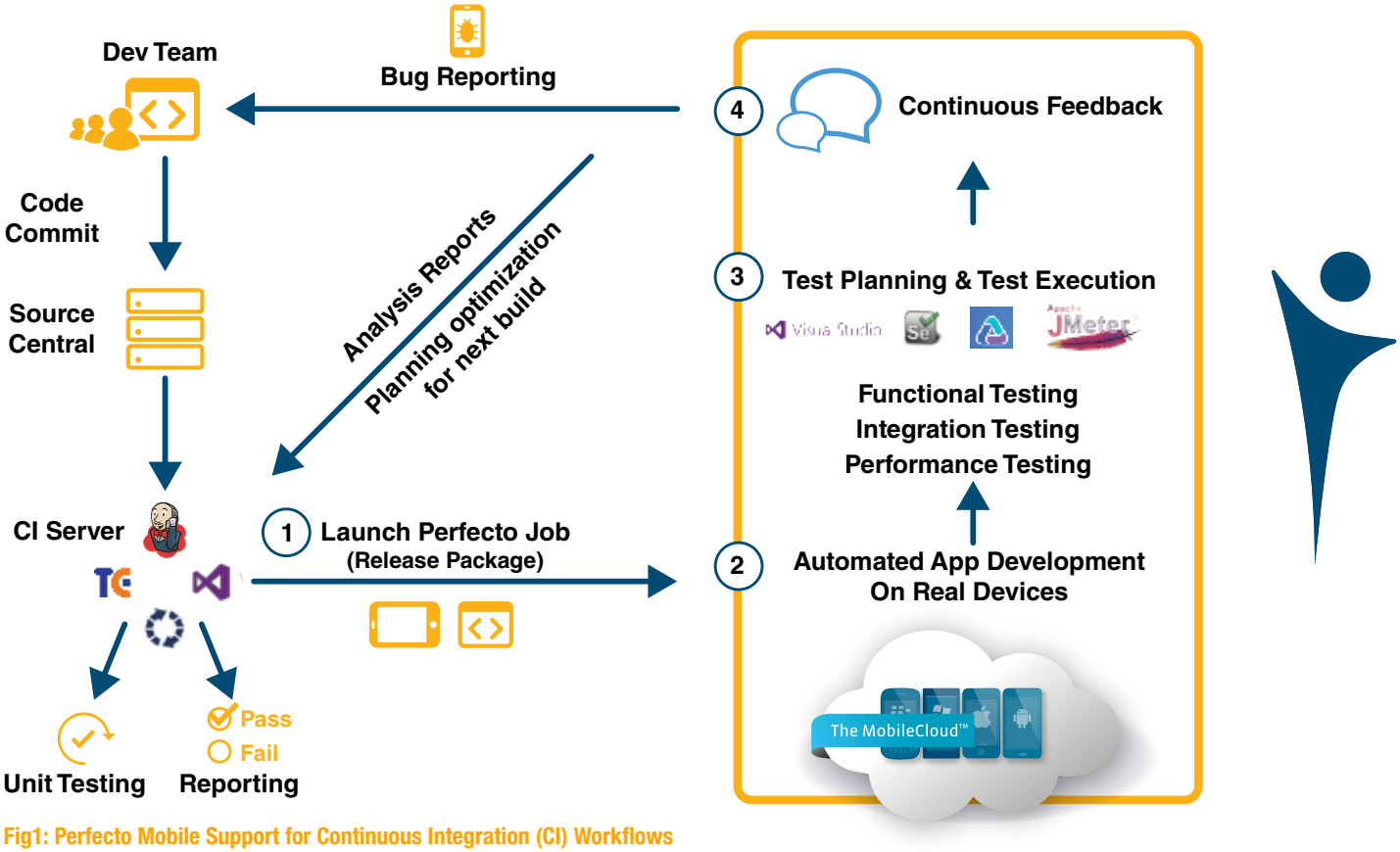


Fig1: Perfecto Mobile Support for Continuous Integration (CI) Workflows

Transitioning From Traditional Waterfall To Agile and CI Age

In classic development models, organizations proceeded from one phase to the next (requirements, design, programming, testing, release, maintenance) and expecting everything would work out in the end. Unfortunately, the waterfall approach proved unable to easily accommodate dynamic market conditions and changing requirements.

Agile methods flip this model sequential approach into a smaller iterative process. Transforming the approach introduces greater flexibility into the development process, speed deployment and increase overall confidence in quality resulting in greater user satisfaction.

To achieve this level of flexibility and allow developers to work on different aspects of a project in parallel, companies needed to be able to manage how code from various sources was validated and aggregated into approved builds. Thus, CI servers were born to automate this process. Solutions such as Jenkins, Team City, Electric Cloud and Visual Studio and others were designed to automatically build and evaluate code in relevant test environments and ensure defects were exposed and corrected before it could be accepted and incorporated in an active build.

Team implementing CI quickly learned that the fundamental success enablers are build and test automation supported by a test ready lab mimicking production. Combined they provide dev teams a scalable process and actionable insight resulting in speed.

Time to Market Requirements

More than others, mobile app developers are constantly driven by both internal and external deadlines. As consumers shift more and more of their actions to mobile devices, programmers are under extreme pressure to continuously innovate and add new features. And, every new phone, tablet or OS version released by Apple, Google, Samsung or others forces a software update. In today's competitive environment, companies cannot afford to miss any target release dates.

> **Without rapid feedback developers are challenged to meet the complexities of requirements evolution, device and OS innovation and most important user expectations.**

Release velocity is the name of the game in mobile app development.

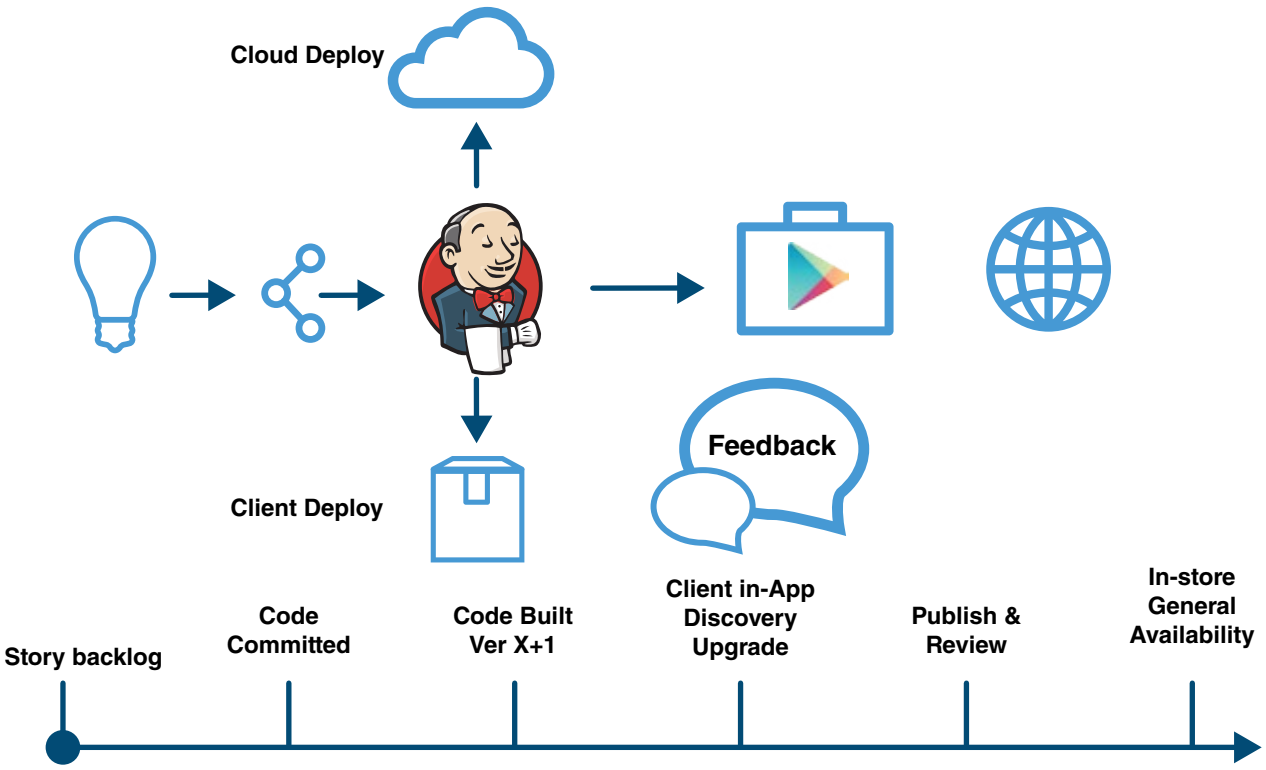


Figure 2: Sample of Mobile CI/CD Cloud based Model (Source)

Extending CI Frameworks For Mobile APP Development

Test Scenarios

A fully designed test plan for mobile apps must include three key scenarios: functionality, performance and usability testing.

- Functionality testing ensures that a wide range of actions, such as login validation, SMS, push notifications, call connections and more operate according to design.
- Performance testing provides insight on bottlenecks as well as network and resource utilization issues.
- Usability testing ensures the look and feel across all user devices. Understanding user experience is vital to the process as it can mean the difference between an adopted mobile app vs. an unused one.

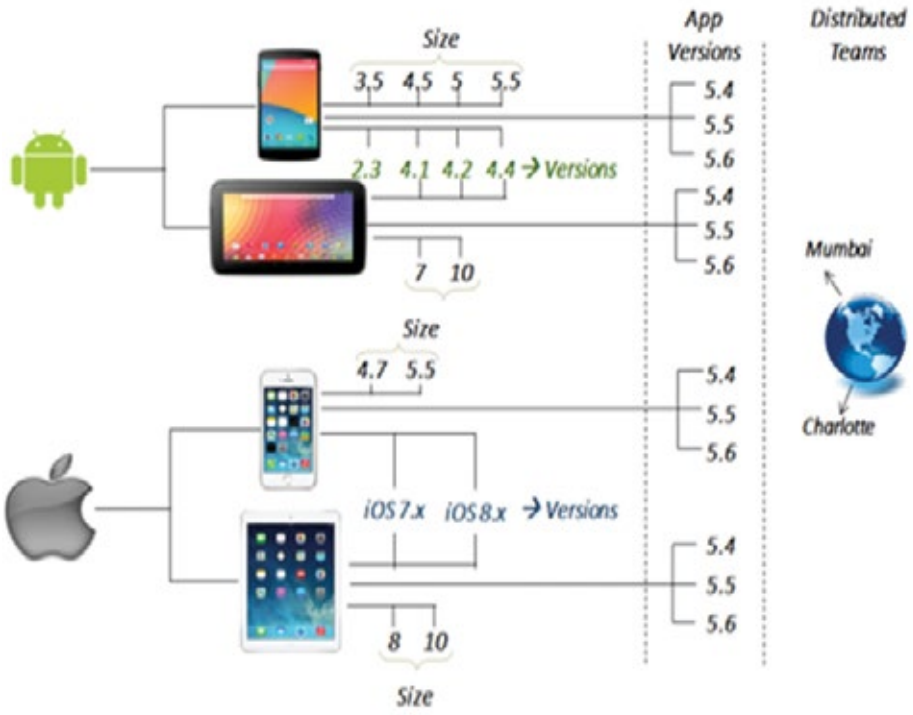


Figure 3: Sample test matrix required to be supported per testing cycle (Source: Perfecto Mobile Customer)

Test Requirements

To support the development process and thoroughly evaluate mobile app code, CI solutions must be able to perform tests on numerous device profiles under various network conditions that change unexpectedly in parallel. Unlike more traditional software development projects, testing mobile apps under real-world conditions can quickly become overwhelming. Consider the following illustration (Figure 3) taken from one of Perfecto Mobile's enterprise customers, showing the complexities inherent in validating each release.

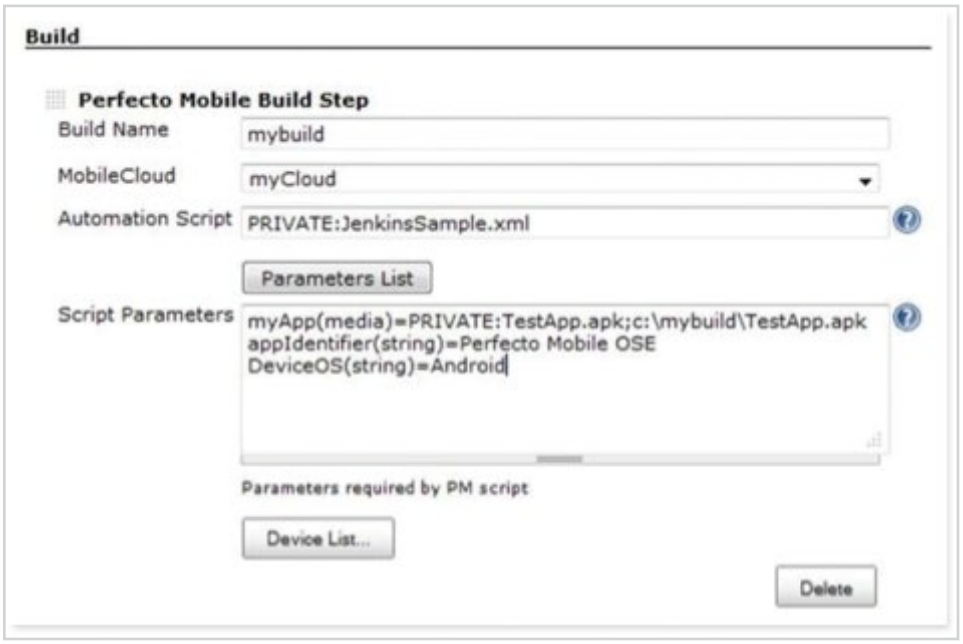
Along with support for multiple device profiles, agile developers need a solution able to effectively evaluate events unique to mobile apps – phone calls, text messages, security pop-ups, loss of network, battery stretch, etc.

When it comes to expanding CI to mobile, stand up a mobile-ready test environment able to satisfy best-practice recommendations:

- Scalable test automation – Build verification testing that delivers parallel execution over many device/OS combinations able support the full range of mobile user scenarios including native, web and hybrid applications, voice, SMS VOIP and streaming services as well as advanced functions such as speech recognition, location-based services and more.
- Production replica – Emulating conditions is insufficient; to truly evaluate functionality, performance and usability tests must be conducted on a full range of real devices connected to active networks worldwide
- Integration with existing development and test workflows – Don't change, but rather extend your existing source control, development environments, frameworks and CI tooling.
- Actionable feedback – To speed bug fixes and issue resolution, look for solutions that provide screenshots and videos that clearly demonstrate what went wrong and where
- Scalability – Test environments must meet numerous scaling dimensions – devices, test scope (functional and non-functional), network conditions to streamline the release process.
- Always On, Always Ready – Constant test readiness is required to meet the hourly, nightly, weekly build frequency of distributed teams frequently. SLA-back cloud-based labs are often the only viable answer.

Seamless Integration

As an example, developers should be able to use the Jenkins Build Job function to execute pre-defined tests on an active device pool and automate the code validation/check-in process.



The only way to automate code validation from within the CI environment is through seamless integrations that enable developers to access test capabilities from within.

Figure 4: MobileCloud Jenkins Plugin – Build Configuration Screen

Build the Right Process

For mobile dev teams, architecting CI practices requires balancing the right verification scope over the common sequence of hourly, nightly and weekly CI builds.

	Hourly	Nightly	Weekly
Duration	15 - 60 mins	4-6 hours	24-48 hours
Text Scope	Unit, integrations and partial smoke	Add partial regression, lightweight network virtualization and load	Complete regression, extend performance
Device Coverage	±6	±12	±20

Figure 5: Architect CI practices by balancing frequency, duration, test scope and device coverage

For mobile, fast feedback that delivers on its promise must shift performance and other non-functional testing left. Poor performance is the second most common app review complaint by users. Varying network conditions is uniquely mobile aspect impacting user experience. Strong insights are accessible early in the development process to begin understanding how performance is impacted when network connectivity is assumed to be WiFi, 4G or 1G.

Furthermore, actual traffic analysis should largely dictate the range of devices exercised as part of the CI process. In addition, teams must constantly scan the market for expected new device and operating systems. Adequate device coverage is a critical component of acquiring the fast feedback teams require to meet demanding market conditions.

Bottom Line

It is clear that the testing mobile app functionality, performance and usability is complex relative to other projects. However, this does not mean that CI practices cannot be extended mobile. Instead, it requires a mobile-ready automation and an always on, always ready lab providing access to real, globally distributed and carrier-connected devices.

Continuous Integration is becoming the cornerstone of building better apps faster resulting in the customer engagement teams are striving to achieve. Continuous Quality is driving the difference between user engagement and abandonment.

To read more on the Perfecto Mobile CI solution, please refer to the MobileCloud Jenkins Plugin page, download the plugin, or contact us at sales@perfectomobile.com.

About Perfecto Mobile

Perfecto Mobile is transforming the way enterprise organizations go mobile, enabling them to develop, test, deploy and monitor their mobile applications and services and go-to-market with confidence. Perfecto Mobile's cloud-based MobileCloud™ Platform and end-to-end mobile quality product suite enables users to remotely access a large selection of real mobile devices connected to local cellular networks around the world and leverage them throughout the mobile application delivery lifecycle – from development, functional and performance testing to monitoring and support. The MobileCloud™ is available either as an enterprise private cloud or a sharable public cloud.

perfecto mobile
perfectomobile.com
781.280.4111