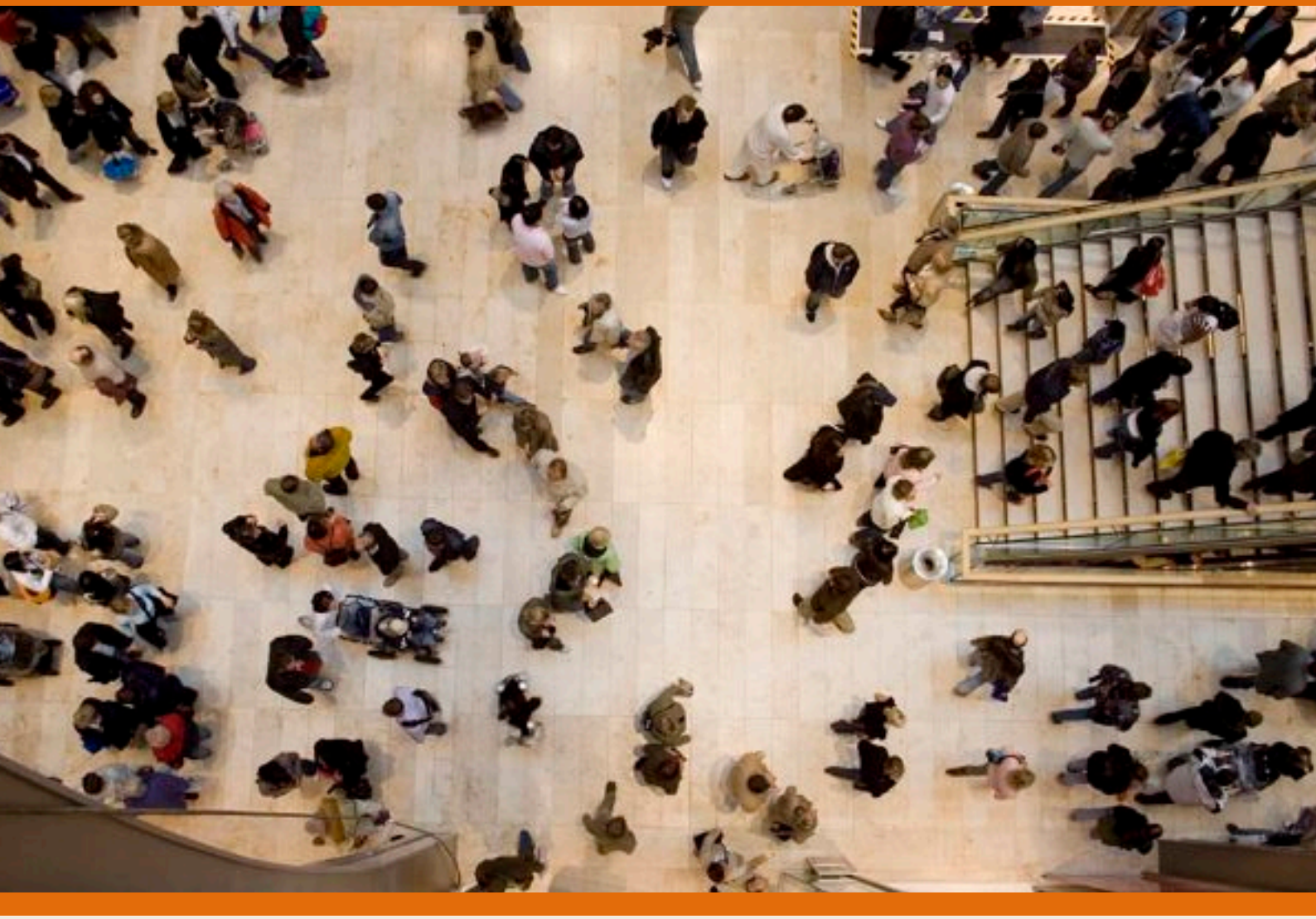
©VisionMobile 2010. Some rights reserved.

July 2010

Making sense of a fragmented world

Mobile Developer Economics 2010 and Beyond

Insights and analysis from the definitive mobile developer survey
Plus benchmarks on the platform development experience



Created by vision mobile

Sponsored by Telefónica DEVELOPER COMMUNITIES

Behind Mobile Developer Economics 2010

Andreas Constantinou, Research Director
Elizabeth Camilleri, Research Partner
Matos Kapetanakis, Marketing Manager

About VisionMobile

VisionMobile is a market analysis and strategy firm delivering market know-how to the mobile industry. We offer research reports, industry maps, training courses and advisory services on under-the-radar market sectors and emerging technologies.

VisionMobile Ltd.
90 Long Acre, Covent Garden,
London WC2E 9RZ
+44 845 003 8742

www.visionmobile.com/blog

Follow us: @visionmobile
Contact us: hello@visionmobile.com

License

Copyright © VisionMobile 2010. Some rights reserved.

Licensed under a Creative Commons Attribution 3.0 License.
Any reuse or remixing of the work should be attributed to the VisionMobile Developer Economics Report sponsored by Telefonica Developer Communities.

Feedback?

For comments, feedback and more information:
www.DeveloperEconomics.com

Disclaimer

VisionMobile believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing on this publication only and the information, including the opinions contained herein, are subject to change without notice.

Use of this publication by any third party for whatever purpose should not and does not absolve such third party from using due diligence in verifying the publication's contents. VisionMobile disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose. VisionMobile, its affiliates and representatives shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

Contents

See also

Mobile Industry Atlas: your competitive landscape of the mobile ecosystem, mapping 1,100+ companies across 69 market sectors. Available in wallchart and PDF format.

www.visionmobile.com/maps

Foreword

Developer Economics 2010 is a global research report delving into all aspects of mobile application development, across 400+ developers segmented into eight major platforms: iOS (iPhone), Android, Symbian, BlackBerry, Java ME, Windows,Phone, Flash/Flash Lite, and mobile web (WAP/XHTML/CSS/Javascript). The report provides an unprecedented range of insights into all the touchpoints of mobile app development, from selecting a platform to pocketing the profits.

Research was conducted between January and June 2010. It was carried out by a team of three researchers, five interviewers, and eight mobile app developers. This major research project represents a first, in terms of the depth and breadth of the research into mobile developer attitudes and expectations. Our hope is that this report will be seminal in forging bridges between the two sides of the ecosystem: the mobile industry decision makers, and the mobile application developers.

Andreas, Eli, Matos & team at VisionMobile
Follow us on twitter: @visionmobile

A word from our sponsor, Telefonica Developer Communities

Welcome to “Developer Economics 2010 and Beyond”.

Telefonica has always sought to base our roadmap on a clear understanding of the wants and needs of developers. We conducted comprehensive developer research in both 2008 and 2009, and we have used the insight gained as the foundation for our current and future developer community plans. Working with VisionMobile to create “Developer Economics 2010 and Beyond” was a natural evolution of this process.

For a long time we have felt the industry has been lacking research that credibly tackled the key issues facing developers. “Developer Economics 2010 and Beyond” was an ambitious project. I am confident that due to the quality of the participants and the breadth of the research, this project has uncovered many of the key issues in application development today.

The participants represent 53 countries and 290 individual companies. More than 40 percent of these respondents have at least five years experience in developing apps, while 78 percent have been developing apps for more than 12 months. Participants include almost 20 Forum Nokia Champions, three Android Developer Challenge Finalists, two Handango Champions, winners of the Vodafone Summer of Widgets Contest, NavTeq Global LBS challenge, the Flash Lite Developer Challenge, and the Blackberry Developer Challenge.

I’m delighted that through Telefonica’s support of this project the results can be freely distributed.

This report breaks new ground, and as such I feel it opens as many questions as it answers. I would encourage you to feedback your opinions on the findings at www.developereconomics.com. If you are a mobile developer and would like to be involved in next year’s report, please contact VisionMobile at hello@visionmobile.com.

James Parton, Head of Developer Marketing, Telefonica
June 2010
Follow me here: @jamesparton
http://www.o2litmus.co.uk/o2blog/

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

About this Research

Developer Economics 2010 provides insights into platform selection, application planning, code development and debugging, as well as support, go-to-market channels, promotion, revenue generation, and hot topics such as the role of open source and network operators.

The objectives behind this research were to analyse the mobile developer experience from two very different angles:

1. Survey the perceptions of mobile developers across the eight major platforms - Android, iOS (iPhone), BlackBerry, Symbian, Windows Phone, Flash/Flash Lite, Java ME and mobile web (WAP/XHTML/CSS/JavaScript) – and through a balanced combination of online surveys and telephone interviews.

2. Benchmark the app development experience across four platforms (iOS/iPhone, Symbian, Android, Java ME) through hands-on development of nine mini applications.

The survey methodology, developer distribution across platforms and regions, as well as the benchmark methodology appears in Appendix 1.

Key messages

- **Commercial Pragmatism.** In the last two years, mobile software and applications have moved from the sphere of cryptic engineering lingo to part of the essential marketing playbook for mobile industry vendors. At the same time, software developers have grown to be much more knowledgeable, pragmatic and savvy about the economic implications of mobile development.

- **Market penetration** is hands down the most important reason for selecting a platform, chosen by over 75 percent of respondents across each and every platform. Developers care more about addressable market and monetisation potential than any single technical aspect of a platform.

- **Platform concurrency.** Most developers work on multiple platforms - on average, 2.8 platforms per developer, based on our sample of 400 respondents. Moreover, among iPhone and Android developers, one in five releases apps in both the Apple App Store and Android Market.

- **Mindshare migration.** In the last two years, a mindshare migration has taken place, with mobile developers moving away from "incumbent" platforms, namely Symbian, Java ME and Windows Phone. The large minority (20-25 percent) of Symbian respondents who sell their apps via iPhone and Android app stores reveals the brain-drain that is taking place towards these newer platforms. The vast majority of Java ME respondents have lost faith in the write-once-run-anywhere vision. Moreover, anecdotal developer testimonials suggest that half of Windows Phone MVP developers (valued for their commitment to the platform) carry an iPhone, and would think twice before re-investing in Windows Phone.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

- **Android as mindshare leader.** Android stands out as the platform most popular with mobile developers. Our survey results suggest nearly 60 percent of all mobile developers recently developed on Android, assuming an equal number of respondents with experience across each of eight major platforms (see research methodology in Appendix 1). iOS (iPhone) is second in terms of developer mindshare, outranking Symbian and Java ME, which were in pole position in 2008.

- **Mindshare vs addressable market disconnect.** Platform characteristics could not be more diverse across installed base and number of apps, revealing a disconnect between developer mindshare and addressable market for each platform. For example, the Symbian operating system is deployed in around 390 million handsets (Q2 2010), and claims over 6,000 apps, while Apple's iPhone has seen 30x more applications while being deployed at just 60 million units over the same period.

- **Developer bias.** Most developers have a head-strong affinity towards the platform(s) they have invested time in, which distorts the perception of platform characteristics; across all eight major mobile platforms we surveyed, respondents felt that the best aspect of their platform was the large market penetration, even if the actual market penetration was relatively small.

Market-related insights

- **Market channels** that were once mainstream, pre-2008, today take only a small chunk of the go-to-market pie for mobile apps. Operator portals and on-device preloading through OEM or operator deals is the primary channel to market for fewer than five percent of mobile developers surveyed. Our findings show that developers resort to either 'native' app stores, or to direct download via their own websites – in addition to the traditional model of bespoke app development.

- **Planning techniques** are ubiquitous for application developers. Over 90 percent of respondents use some form of planning technique, such as beta testing or peer reviewing, for deciding on the target user segment or application features. However, given the hundreds of thousands of mobile apps, we believe that efficient (crowd-sourced) app testing is considerably under-served.

- **App stores** have reduced the average time-to-shelf by two thirds: from 68 days across traditional channels, to 22 days via an app store, according to our research. Moreover, app stores have reduced the average time-to-payment by more than half; from 82 days across traditional channels, to 36 days via an app store. On average, it takes 55 days to get paid via an operator channel, or a whooping 168 days when on-device pre-loading via a handset manufacturer.

- **Short-head app stores.** Despite the hype, there is little use or availability of app stores outside the Apple and Android platforms. Only five percent of Java and just over 10 percent of Windows Phone respondents reported using an app store as a primary distribution channel.

- **Discovery bottleneck.** The key challenge reported by mobile developers is the lack of effective marketing channels to increase application exposure and discovery. Moreover, half of respondents are willing to pay for premium app store placement. Despite their commercial savvy, developers have not taken application marketing into t**heir own hands.**

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

- **Certification.** The most important challenge in app certification is its cost; more than 30 percent of respondents who certify their apps report the high cost of the certification process as the number one challenge. The economics – often 100s of dollars per certification – do not work for low-cost apps, but only for mega-productions.

- **Dubious long tail economics.** App stores are young and surrounded by a hype wave that distorts the reality of average per-capita monetisation. Only five percent of respondents reported very good revenues, above their expectations. Moreover, nearly 60 percent of iPhone respondents had not reached their revenue targets.

- **Popular revenue models.** Ad-funded models are only secondary revenue sources for developers employing app store and portal-based channels. Despite the hype, our research found ad-funded models lagging much behind tried and tested pay-per-download models. Subscription models, meanwhile, mainly apply where the application is distributed via an operator or content aggregator portal; they have made limited inroads into app stores.

- **Role of operators.** Mobile developers view network operators as bit-pipes. Nearly 80 percent of respondents think that the role of network operators should be to deliver data access anywhere/anytime, while only 53 percent considered their role to be delivering voice calls.

- **Operator support.** The majority of developers had not interacted directly with operators, but were very opinionated towards the lack of support. Almost 70 percent of respondents thought there was little or no developer support from network operators. Moreover, industry standards, consortia and joint initiatives (including OMTP and WAC) appear to have captured very little developer mindshare.

Technical-related insights

- **Learning curve and efficiency**. The learning curve varies greatly across mobile platforms. On average, the Symbian platform takes 15 months or more to learn, while for Android the average reported time is less than six months. Moreover, Symbian is much more difficult and time consuming to program than iOS (iPhone), Android or Java ME; our benchmarks show that for developing nine different typical applications, a Symbian developer needs to write almost three times more code than an Android developer, and twice as much code as an iPhone developer.

- **Development environment.** From a technical perspective, top pain points for mobile emulators and debuggers are slow speed and poor target device mirroring. Top pain points for development environments (IDEs) are the absence of an app porting framework, and poor emulator integration.

- **Debugging.** In terms of debugging, our benchmarking shows that Android has the fastest debugging process, compared with iPhone, Symbian and Java ME. Debugging in Symbian takes up more than twice the time it takes on Android.

- **UI tools**. Ability to build compelling UIs is still far from the reach of most mobile developers. Around 50 out of 100 Symbian, BlackBerry and Windows Phone per-platform respondents are annoyed with the difficulty in creating great UIs.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

- **Support.** Our research indicates that the majority of developers (more than 80 percent of respondents) rely on community or unofficial forums for support during software development, while websites are used for support by only 40 percent of respondents.

- **Hidden device APIs.** Access to unpublished or 'hidden' device APIs is a control point for platform vendors, but it is also what developers seem to be willing to pay for – in fact, more so than any other type of technical support. We believe that platform vendors could benefit from tiered SDK programs, where privileged SDKs are available to developers on a subscription plan.

- **Network APIs.** Operator network API programs have so far failed to appeal to developers. Only five percent of respondents thought that the role of network operators should be to expose network APIs. Yet more than half would pay for billing APIs, followed by messaging and location APIs.

- **Open source.** On average, 86 percent of respondents who use open source at work use it within development tools such as Eclipse. Android and iPhone developers are three times more likely to lead open source communities, compared to Symbian, revealing the contrasting pedigree of the developer communities. The single key drawback to open source reported by 60 percent of respondents was the confusion created by open source licenses; we believe education on open source realities can be used as a competitive advantage for developer programs launched by operators and OEMs.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Part 1

The Migration of Developer Mindshare

Part 1. The Migration of Developer Mindshare

Software has played a critical role in transforming the mobile industry since the beginning of the century. Since 2008, mobile software and applications have moved from the sphere of cryptic engineering lingo to part of the essential marketing playbook for mobile industry vendors. Mobile industry players are now vying to win software developer mindshare, in order to add value on top of their devices and networks.

The evolution of mobile software 2000-2015

The mobile industry's relationship with software has evolved through three distinct phases.

The Dark Ages (2000-2004). The first five years witnessed the hype of the mobile Internet and the shift of power – from hardware features to software smarts, and from manufacturer control to network operator control. The incumbent mobile platforms – Symbian, Windows Smartphone, PalmOS, Java ME and BREW – were then seen as the one-way street for manufacturers to 'open up' their phones to the possibilities of the Internet and to modernise their aging legacy platforms. Operating systems were open simply by virtue of published APIs. Mobile development was in the Dark Ages, with information on routes to market and monetisation available only to those working within the inner circles of the mobile industry.

The Renaissance period (2005-2009). The next five years witnessed the hype of the smartphone, the launch of iconic experience products, the app store phenomenon and the displacements caused by open source. New mobile platforms like BlackBerry, Android and iOS opened their doors to developers, while the incumbents transformed (Symbian Foundation) or withered in popularity (PalmOS and Java ME). Mobile widget platforms allowed web applications to be first class citizens. Open source disrupted both the operating system business (killing off UIQ and MOAP, and sidelining Windows Phone) and also the browser business, displacing the top two browser vendors, Teleca and Openwave. This era proved that it's not enough to have open APIs; an open, streamlined developer-to-consumer channel for applications, popularly known as an "app store", is also essential. Mobile development moved to the Renaissance period, in which developers are much more knowledgeable – and pragmatic – about the commercial reality of mobile development.

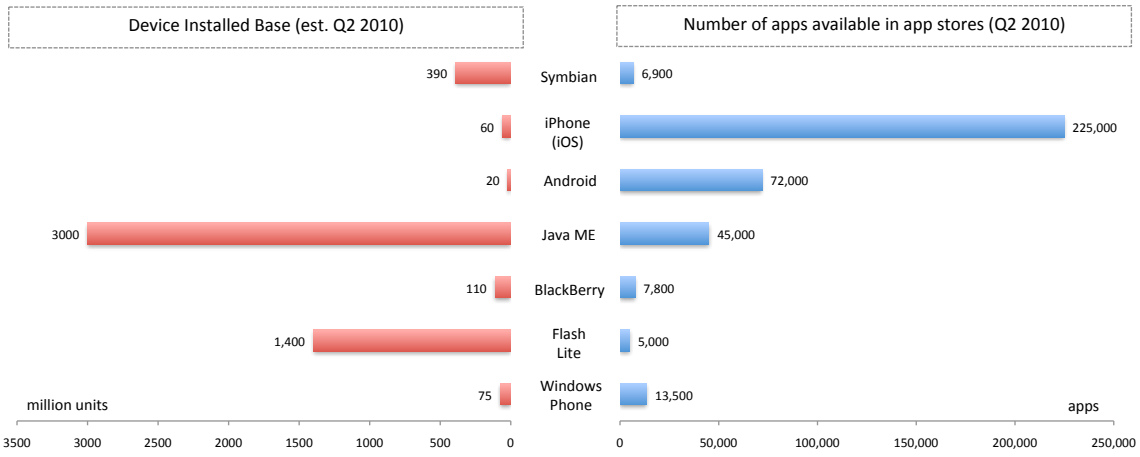
The Industrial Revolution era (2010-2014). The next five years will completely remap the mobile industry landscape. RIM and Apple, two verticalised companies, move into the top five, displacing the incumbents, leaving one Finnish and two Korean companies in pole position. The operating system landscape will consolidate into two tiers; the top-end open to iconic products dominated by Apple and followed by the iPhone clones powered by Android; and the feature-phone market where licensable operating systems (Android and BREW) will finally allow handset OEMs to move away from legacy RTOS platforms. Google's Android will also power a diverse range of new form factors, from picture frames to car dashboards, offering for the first time a simplified platform from which to achieve convergent interconnected services. In this age of Industrial Revolution, mobile developers will be responsible for most of the innovation on mobile devices, and can act independently from the mobile industry powers-that-be – OEMs or network operators – to get their applications to market. In this age, developers have **both power and choice.**

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

The disparity between devices and applications

Mobile developers have a choice today; they can choose a platform among the many available. The key technical and marketing characteristics of these platforms are very diverse across installed base, number of apps, learning curve, development tools and revenue potential.

One of the major disparities is between the device installed base and the number of apps per platform. One would expect that the platforms deployed on the largest number of devices would have the biggest number of applications. This couldn't be further away from the truth. For example, Java ME is available on around three billion handsets, but the platform can boast less than half of the apps available for the much younger Android, shipped in only 20 million devices as of the end of the second half of 2010. Similarly, the Symbian operating system is deployed in around 390 million handsets (end of first half of 2010), and claims over 6,000 apps, while Apple's iOS has achieved 30x more apps over just 60M units.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

These disparities stem from the origins of the two ecosystems; on the left hand side of the chart, the embedded software industry ecosystem has focused on enabling *handset OEMs* to differentiate, while on the right hand side, the Internet/PC ecosystem has focused on enabling *developers* to differentiate. The speed of evolution of these two ecosystems is worlds part; since 2008, the number of applications for the younger iPhone and Android platforms has skyrocketed compared to those for the incumbent Symbian and Java ME platforms.

The migration of developer mindshare

In stock market terms, developer mindshare is one of the hottest "commodities" in the mobile business, one whose "stock price" has ballooned in the last two years. Platform vendors, handset OEMs, network operators, hardware vendors, and infrastructure providers all want to contribute to mobile apps innovation. Yet, there are no "stock-markets" for buying developer mindshare, and very few commercial bridges or matchmakers exist between mobile developers and the mobile industry, or between mobile developers and media brands.

> "The [J2ME] platform has not evolved over the years and it is stagnating."
>
> Java ME Developer,
> India-based software house

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

In general, software developers have a high affinity toward their mobile platform, and high switching barriers exist, due to emotional attachment to their platform and the time they have invested in it.

In the last two years, a mindshare migration has taken place for mobile developers away from the incumbent platforms Symbian, Java ME and Windows Phone, while a substantial number of PC software developers have flocked to iPhone and Android. The large minority (20-25 percent) of Symbian respondents who sell their apps via iPhone and Android app stores reveals the brain-drain that is taking place towards these newer platforms. The vast majority of Java ME respondents have lost faith in the write-once-run-anywhere vision. Moreover, anecdotal developer testimonials suggest that half of Windows Phone MVP developers (valued for their commitment to the platform) carry an iPhone and would think twice before re-investing in Windows Phone.

> "Most Windows Phone developers have started developing for the iPhone – question is whether they will go back to developing for Windows Phone. Half of the Microsoft Windows Phone MVPs have an iPhone."
>
> Anonymous
> Microsoft mobile devices MVP

If we ascribe value according to developer mindshare, or the number of developers who have experimented or worked on any single mobile platform, we find that **Android has the highest valuation.** Our survey findings suggest that Android is the single platform that the most mobile developers have experience with.

> "It feels like we 're back in the dot com era. Everyone wants an iPhone application."
>
> Siddhart Agarwal,
> CEO, Mobicule

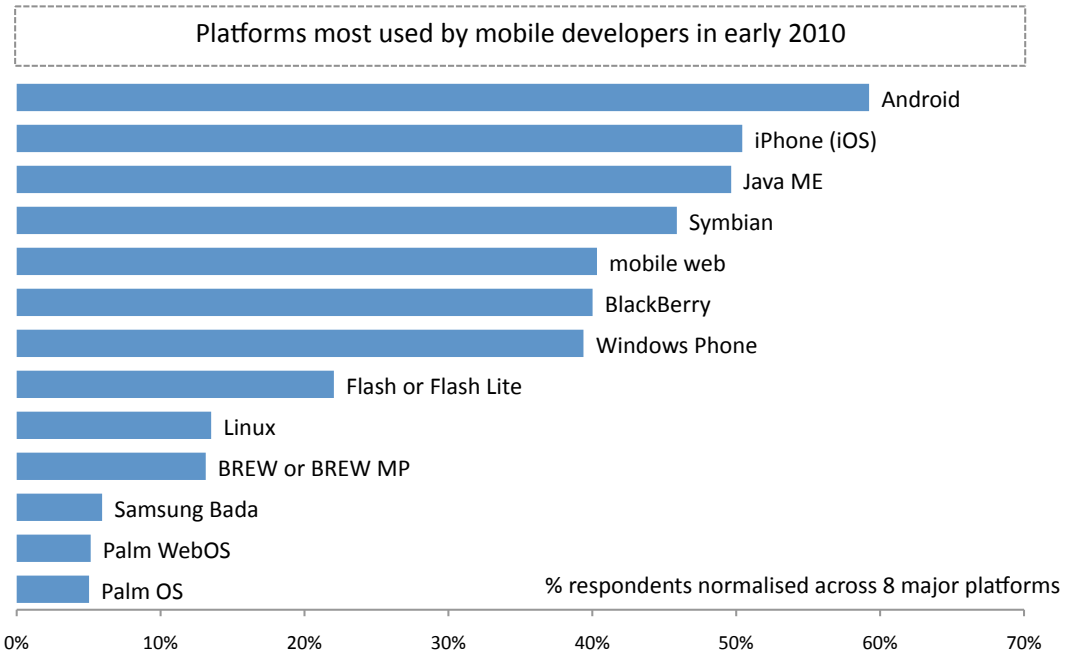
In our study, respondents were asked to base their answers on one of eight mobile platforms (up to two responses were allowed per developer, each on a different platform). Among developers taking the survey as iPhone developers, 56 percent had recently worked on Android as well, while on the contrary only 42 percent of those responding as Android developers had recently worked on iPhone. The next chart ranks each platform according to the level of experience developers had on each platform. By normalising responses to 100 developers across each of the eight major platforms in our survey, it aims to show what the results might have been had we surveyed an equal number of developers on each of the major platforms.

> "Android is better than other platforms in terms of tools, platform features, and it's easier to stand out as a developer."
>
> Aamir Yassen,
> Grand Prize winner of
> Nokia's Calling All Innovators
> Mobile Application Competition 2009

One can easily see that Android stands out as the top platform according to developer experience, with close to 60 percent of developers having recently developed on Android, assuming an equal number of developers with experience on each of eight major platforms. iOS (iPhone) follows closely as the next most popular platform, outranking both Symbian and Java ME, which until 2008 were in pole position.

We believe that Android's lead in developer mindshare ahead of Apple's iOS is down to two factors: first the $99 fee developers have to pay in order to deploy their applications, an entry barrier which reduces the innovation from developing countries. Secondly, the very effective use of open source licensing as a marketing technique to attract developers to Google's Android.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

iPhone/iOS and Android platform adoption have benefited from positive-feedback-loop effects; media and consumer brands are eager to leverage iPhone, iPad or Android applications as a means of acquiring new customers, as part of a multi-channel marketing strategy, or simply to add new revenue streams.

Software development firms have been inundated with requests for iOS and Android apps. “It feels like we ’re back in the dot com era. Everyone wants an iPhone application, while we get very few Symbian requests these days,” recounts Siddhart Agarwal, CEO of Mobicule, a 30-strong mobile software development firm in Mumbai.

The success of Apple and Google in the mobile space is the main contributor to the exodus of innovation from the incumbent platforms.

Our research further indicates that Flash developer mindshare seems to be in decline, despite Flash’s installed handset base of more than 1.3B devices. Adobe’s string of execution failures has meant that the installed base for Flash Lite is extremely fragmented, breaking the write-once-show-anywhere story for media brands who are Adobe’s key customers. At the same time, Flash, the much-touted replacement for Flash Lite, was more than 18 months late, while Flash Lite shipments have stagnated, dropping from 43 percent to 15 percent of handsets sold from 1H09 to 2H09. This leaves Adobe with a rapidly shrinking window of opportunity, primarily on Android handsets, while having been banned from Apple’s growing empire, and slowly seeing the adoption of HTML5, yet another replacement threat for Flash.

Despite the investment Sun and Adobe put into ensuring mass-deployment of Java ME and Flash Lite, they have been marginalized in terms of developer mindshare by Android and iPhone, the new kids on the block. Taking a cue from the lingo of financial investors, past performance in device shipments is no indication of future return, where

> Past performance in device shipments is no indication of future return in developer mindshare.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

developer mindshare is concerned.

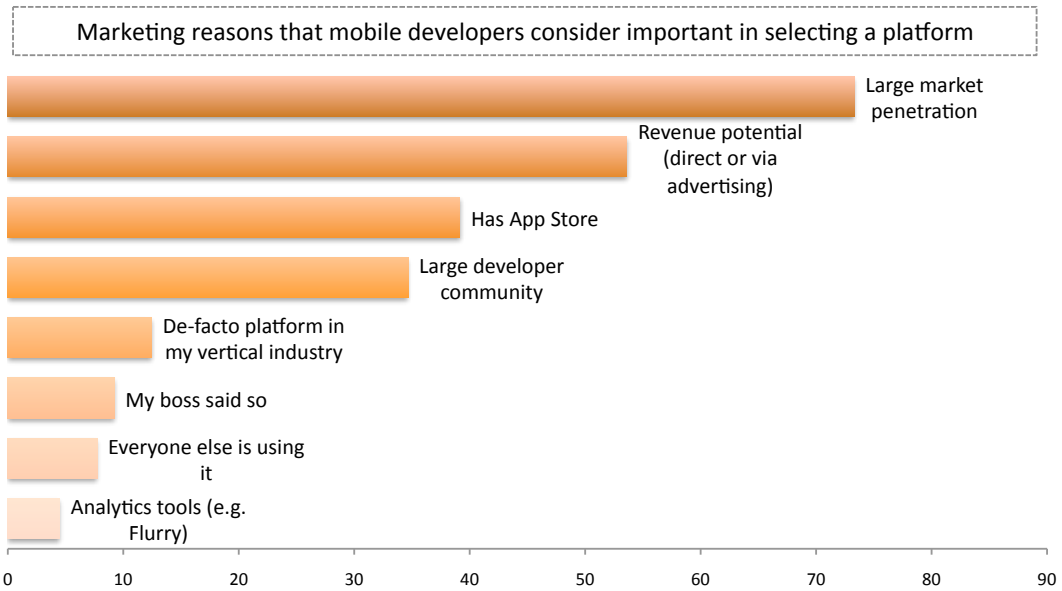
> "Technical considerations are irrelevant. The choice of platform is ALWAYS marketing-driven."
>
> Christophe Lassus,
> Founder & Director
> flirtymob.com

Palm's WebOS is familiar to fewer than five percent of mainstream platform developers, based on our platform-normalised sample of 400+ respondents. But is HP's acquisition of Palm able to increase the relevance of WebOS to application developers? On one hand HP, the number two PC manufacturer in terms of industry profit share according to Deutsche Bank, might inject enough capital to help propel WebOS into an effective competitor to Android. On the other hand, the marriage of HP and Palm misses on key synergies; Web OS won't offer HP clear consumer differentiation, developer mindshare or operator subsidies, as we noted in a recent article. As a result we remain pessimistic on the future of Palm's Web OS.

Commercial savvy and market penetration

Most developers work on multiple platforms, on average 2.8 platforms per developer, based on our sample of 400 respondents. Moreover, one in five iPhone and Android respondents release apps in both the Apple App Store and Android Market.

The question is: in a market crowded with software platforms, how do developers choose between iOS, Android, Symbian, Java ME, BlackBerry, Flash, Windows Phone, mobile web, WebOS or Samsung Bada? For today's mobile developer, market penetration and revenue potential are hands down the two most important reasons for selecting a platform.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities.
June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Large market penetration was chosen by 75 percent of respondents across each of the eight major platforms we surveyed. Revenue potential was the second most important reason, chosen by over half of respondents. In fact, market penetration and revenue potential were more important than any single technical reason for selecting a platform, revealing how mobile developers today are savvy about the economic implications of mobile development.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

The availability of a large developer community ranked fourth among marketing reasons for choosing a platform. It's worth noting that the more experienced a developer, the less important they view the developer community.

We should also note how few developers acknowledge that they choose a platform because 'their boss said so'. Given that freelancers and students comprised only 20 percent of our respondent sample, one might have expected more developers to choose this option.

Emotional bias

Despite the commercial savvy and rationalism, there is a lingering emotional bias towards developers' chosen platform. This is evident from two data points. Across all eight major mobile platforms surveyed, respondents felt that the best aspect of their platform was their large market penetration, even if the actual market penetration did not bear this out. The emotional bias was also evident when we asked what type of mobile apps will prevail in the next two years. Most developers chose the answer that included their platform; most Android, Symbian and iPhone developers selected "native apps" as prevailing, while mobile web developers chose "web apps" and Flash developers mostly chose "cross-platform apps". Note that we consider Android as a 'native' operating system as Android apps cannot be ported on other operating systems.

The only developers who thought that the grass is greener on the other side of the fence were Java ME developers, of whom only a fraction believed in the future of cross platform apps. One could say that the vast majority of Java ME developers have lost faith in the write-once-run-anywhere vision.

One thing is clear; developer needs are extremely varied by platform, and in some cases by region. Anyone wanting to attract and retain developer mindshare needs to build a two-way channel of communication to understand what the 'qualifiers' are versus the 'decision clenchers' for mobile developers.

The vast majority of Java ME developers have lost faith in the write-once-run-anywhere vision

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Part 2

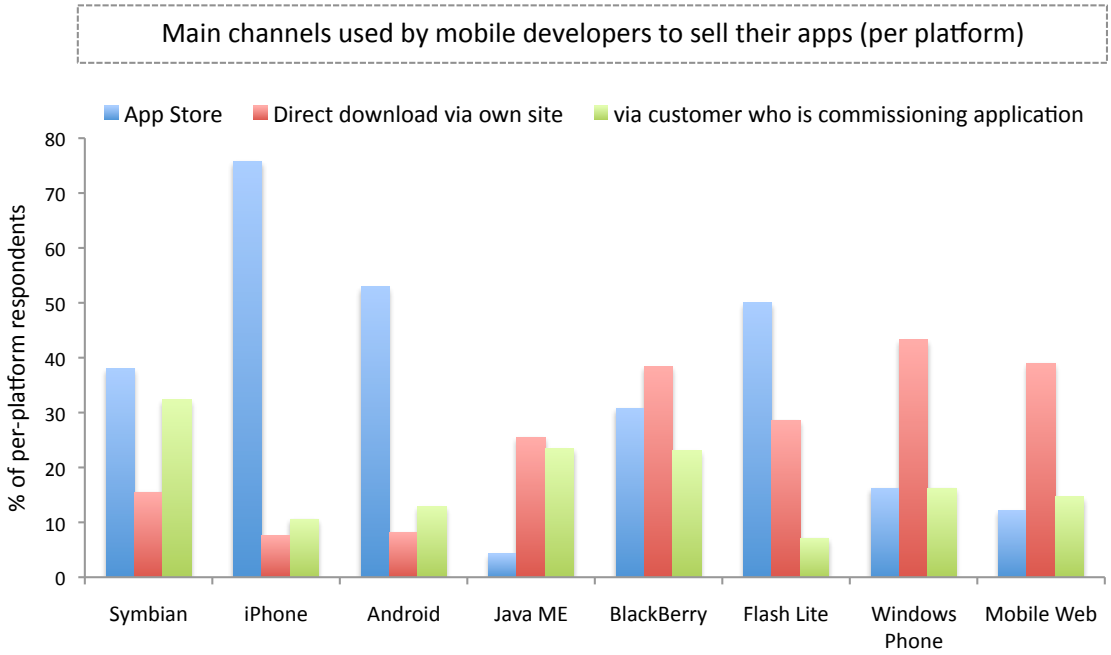
Taking Applications to Market

Part 2. Taking Applications to Market

In this chapter we analyse the developer experience as the application goes to market: planning, testing, certification, submission to market channel (e.g. an app store), shelf placement, promotion, payment and revenue generation.

The decline of traditional go-to-market channels

Developers have a variety of channels through which to distribute their applications. Yet market channels that were once mainstream pre-2008 are now taking only a small chunk of the go-to-market pie for mobile apps. Operator portals and on-device preloading through OEM or operator deals is the primary channel to market for fewer than five percent of mobile developers. Our findings show that developers resort to either 'native' app stores, or to direct download via their own websites, followed in third position by the traditional model of bespoke app development.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

The preferred go-to-market channel for applications varies significantly by platform. The vast majority of iPhone respondents designate an app store as their primary channel for selling apps, followed by around 50 percent of the Android and Flash developers. The most popular channel for Windows Phone developers is direct download via the developer's own web site. That almost half go this route hints at the lack of effective go-to-market channels for Windows Phone apps. This dwarfs the percentages of iPhone and Android respondents who use this channel (10 percent each), since those two platforms have very prominent native app stores.

The app store phenomenon

If there's a single reason for the mass-entrance of developers into the mobile market, it is app stores. We view app stores as direct developer-to-consumer channels, i.e. commercial conduits that streamline the submission, pricing, distribution and retailing of applications to consumers. For a breakdown of key ingredients in the app store recipe, see our Mobile Megatrends 2010 report. App stores have streamlined

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

the route to market for mobile applications, a route that was previously laden with obstacles, such as lack of information, complex submission and certification processes, low revenue shares and regional fragmentation.

Despite the hype, there is sporadic use of app stores outside the Apple and Android platforms. Only four percent of Java respondents used App Stores as their primary channel to market. Windows Phone and mobile web developers find app stores little more relevant, with fewer than 10 percent of such respondents using one as a primary channel for taking applications to market.

This contrasts completely with platforms that have 'native' app stores. Over 95 percent of iPhone respondents use the Apple App Store as their primary channel, while the percentage of Android respondents using Android Market is just below 90. Besides the mainstream use of native app stores by Android and iPhone developers, a small outlier of developers use alternative app stores, such as SlideMe for Android, or Cydia Store for iPhone. The findings also reveal a circa 20 percent cross-pollination of platforms, i.e. iPhone developers producing Android applications and vice-versa.

In terms of the incumbent mobile platforms, around 75 percent of Symbian respondents that use app stores, use the Nokia Ovi Store. The significant number (20-25 percent) of Symbian developers who also use iPhone and Android app stores reveals the brain-drain that is taking place towards these newer platforms. This is a particularly critical migration of developer mindshare, considering that the Symbian platform is the hardest to master. Thus, the size of developer investments on Symbian being written off is substantial.

Java ME respondents utilise GetJar most often, followed by Nokia's Ovi Store and the traditional route of operator portals for retailing Java applications.

For BlackBerry and Windows Phone developers that use app stores, native app stores are used by over 65 percent, with outliers using Handango, Handmark and Mobihand to retail their apps. The main cross-platform App Stores, Handango and GetJar, were used by an average of just over 10 percent of respondents across the eight major platforms.

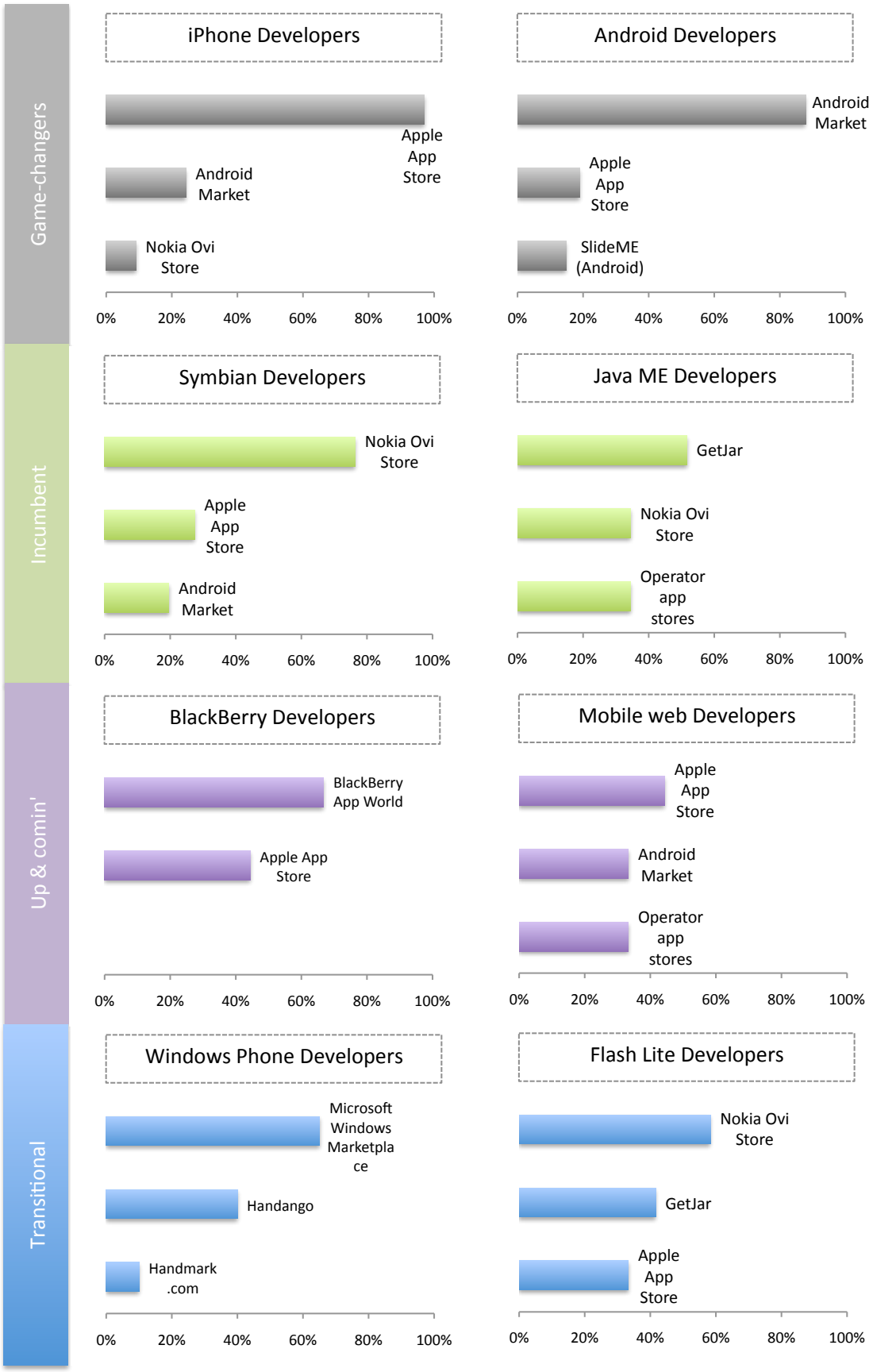
The next set of charts reveal the top three app stores used by developers on each platform – in many cases suggesting that developers are investing in multiple platforms.

Besides the growth of apps, app stores are the cornerstone of another major transformation that has taken place in the mobile industry: the mass-market use of mobile as the next marketing channel beyond the Internet. We would argue that it was app stores that triggered the influx of apps – not the open source nature of Android, or the consumer sex appeal of the iPhone.

> "App stores like Ovi store should be promoted to people in rural areas (India) where there is a large opportunity and they should be trained/informed about downloading apps, usage etc."
>
> Kishore Karanala,
> Senior Software Engineer,
> Teleca India

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Top 3 app stores as voted by mobile developers (per platform)



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

App stores triggered the sheer growth in app numbers and diversity that led to the cliché, "there's an app for that". Another cliché, "the screen is the app," tells the other half of the story. Combined, the app store and touchscreen were the two essential ingredients behind mobile apps as the next mass-market channel beyond the Internet. These two ingredients inspired just about every media and service company to commission companion or revenue-driven apps as extensions to their traditional online channels. In effect, this phenomenon fuelled the app economy, even beyond what app store numbers alone suggest.

Time-to-shelf and time-to-payment

App stores have revolutionised time to market for applications. To research exactly how radically the time to market for applications has changed since the introduction of app stores, we analysed two parameters:

- the time to shelf, i.e. how long it takes from submitting an application to that application being available for purchase
- the time to payment, i.e. the length of time between an application being sold and the proceeds reaching the developer's bank account

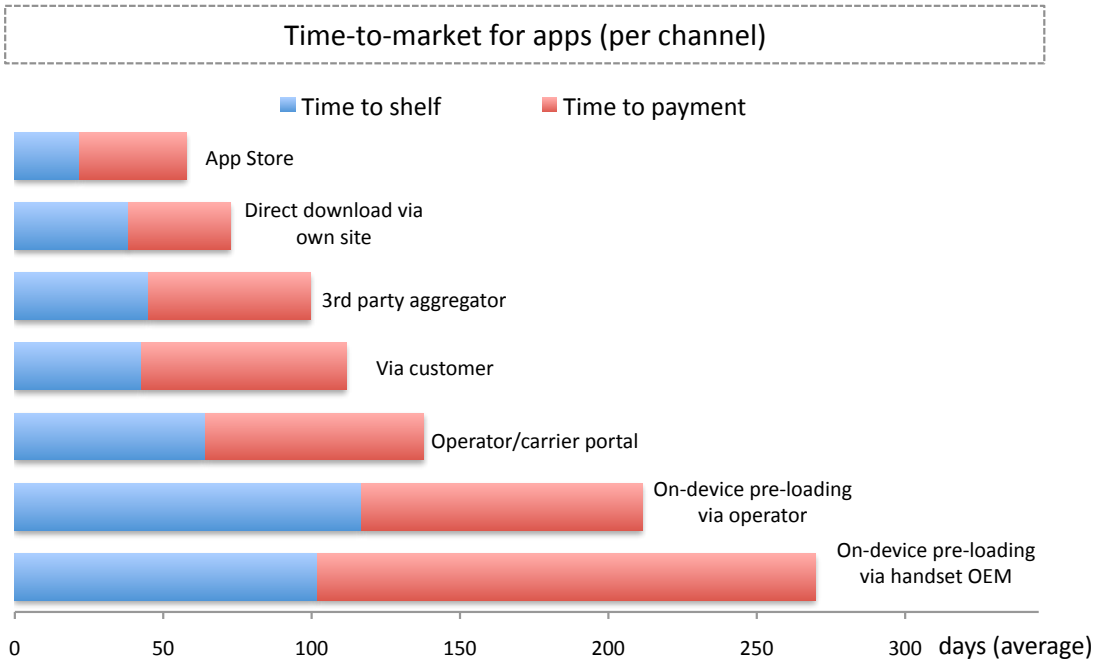
Our findings show that app stores have reduced the average time-to-shelf by two thirds: from 68 days across traditional channels, to 22 days via an app store. These traditional channels have been suffering from long, proprietary and fragmented processes of application certification, approval, targeting and pricing, all of which need to be established via one-to-one commercial agreements.

For example, placing an application on an operator portal takes more than two months, due to the inflexibility of operator processes that are not designed with smaller developers in mind. Moreover, to preload an application on a handset takes more than three months. Meanwhile, three to six months before launch is the typical timeframe for pre-loaded applications that are customised by an operator or third party. As we shall see, it actually takes even longer to get paid for pre-loaded applications, since royalty pay-outs by the operator or OEM are typically delayed by a few more months, which represents the time the handsets spend in the channel.

Time-to-shelf also varies greatly per platform; our findings show Apple's iOS as the fastest platform for taking applications to market, at 24 days time-to-shelf, irrespective of route to market. Windows Phone is almost on par with Android in terms of average time-to-shelf. This may come as a surprise, given that 64 percent of Android developers using an app store report that their apps take less than one week to reach the shelf. However, the majority of Windows Phone developers make apps available via their own websites, or direct to the customer who is commissioning the application, and both of these routes to market are also fairly fast. Symbian applications are by far the hardest to take to market, taking on average over 52 days to reach the shelf from the time of submission – at the opposite end of the spectrum from iPhone applications.

Our research shows how app stores have shortened the time for taking applications to the virtual shelf by two thirds, compared to traditional channels. How about the time it takes for developers to get paid, though? Too often, this is a make-or-break question for mobile developers, as payment delays have a negative impact on cash flow. For small developer shops, poor cash flow can break the bank.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

For developers choosing an app store to retail their apps, almost 60 percent get paid within a month from the sale of the application. In contrast, when using traditional channels, the time-to-payment increases substantially. On average it takes 55 days to get paid via an operator channel, 69 days when preloading an app via an operator and a whooping 168 days (5.5 months) when pre-loading an app via a handset manufacturer.

All in all, app stores reduce the time-to-payment by more than half; from 82 days on average in the case of traditional channels, to 36 days on average with app stores.

The bigger picture that emerges is that the developer's choice of platform impacts the time-to-market for applications, i.e. the length of time from completing an application to getting the first revenues in. The iOS platform is fastest to go to market with, particularly thanks to Apple's streamlined App Store process, while Java ME and Symbian are the slowest,due to the sluggishness of the traditional routes to market used by these developers (in particular via commissioned apps and own-website downloads).

Revenue models and monetisation

Mobile applications have evolved greatly from the days when the first mobile platforms (Symbian, Java ME and BREW) were introduced in 2001-2 – both from a technology as well as a commercial standpoint. Commercially, the routes to market have been radically streamlined, as we discussed earlier, but revenue models have seen only incremental change; pay-per-download (as introduced by Qualcomm's BREW circa 2002) is by far the most popular revenue model used by application developers today, followed by one-off development fees for custom apps.

Our research shows that the choice of revenue model for mobile developers is a question of the channel to market, as shown on the next graphs.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

- App store applications are monetised primarily through pay-per-download models (used by two thirds of respondents) and secondarily through ads or one-off fees for custom applications.

- Apps distributed via web portals (developer, third-party or operator portals) utilise either a pay-per-download revenue model or subscriptions, with ad-funded and one-off development fees being less favoured.

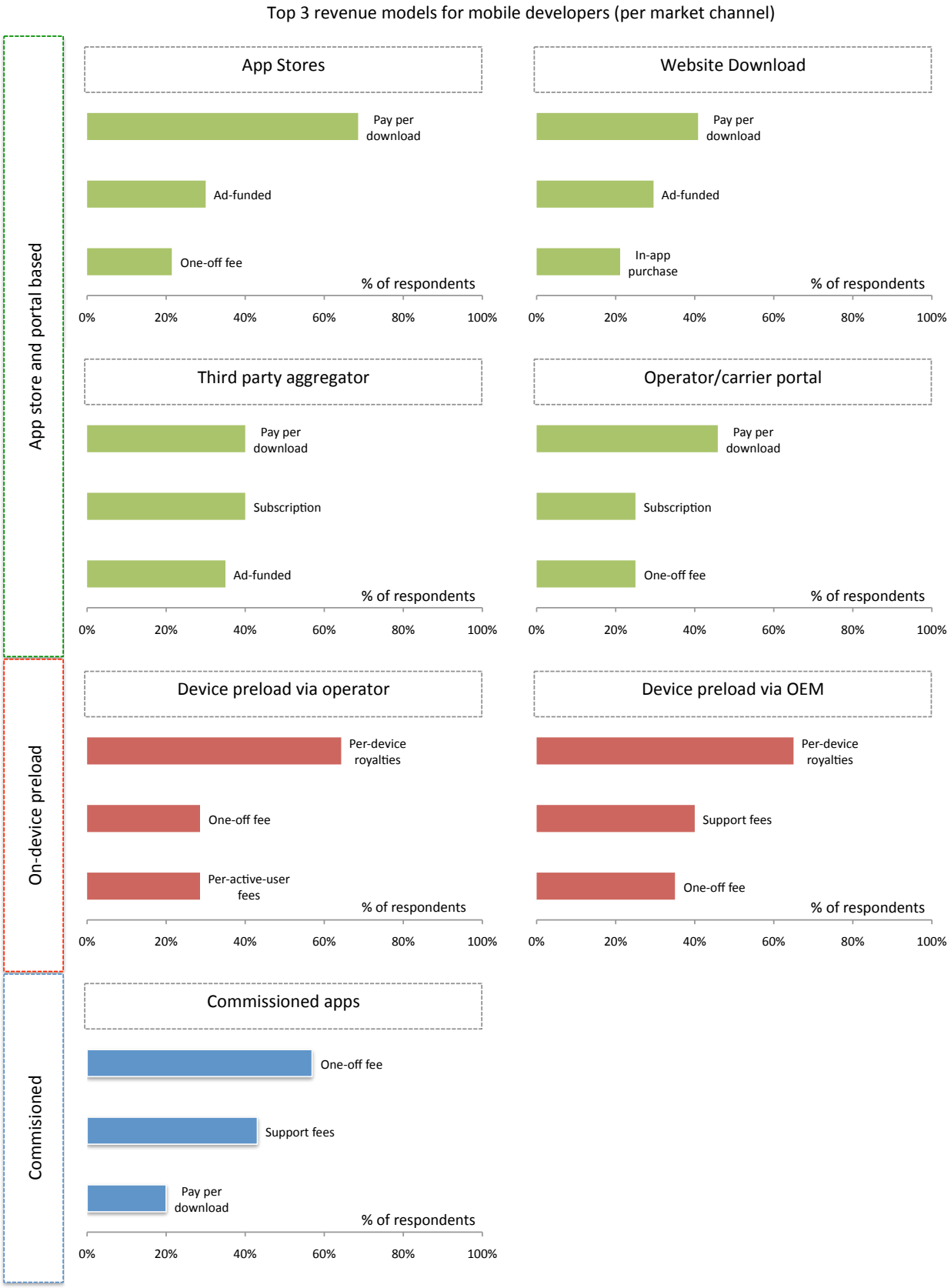
- Apps preloaded on-device by operators or handset OEMs are typically monetised via per-unit device royalties, plus some form of NRE fee (non-recurring engineering or similar one-off fees). For OEM deals, support fees are typically charged. Meanwhile, for operator deals, per-active-user revenue models are employed, in which the application bundling is typically free, but the operator pays a higher fee for each active user of the application.

- Commissioned apps are typically monetised through one-off fees, with support and customisation fees also being common.

Ad-funded models are only secondary sources of revenue employed in app store and portal-based channels. Despite the hype, our research found use of ad-funded models lagging much behind tried and tested pay-per-download models. Subscription models mainly apply when the application is distributed via an operator or content aggregator portal; subscription models have made very few inroads into app stores.

We should also point out that based on our survey findings, the per-active-user revenue model, despite having been much talked about in software circles as the next step in the evolution of software monetisation, is in practice used only in operator device-preload deals. However, we do maintain an optimistic outlook toward the long-term adoption of per-active-user revenue models, as these can be directly passed onto users as subscription fees, and therefore aligned with operator (and service provider) incentives.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

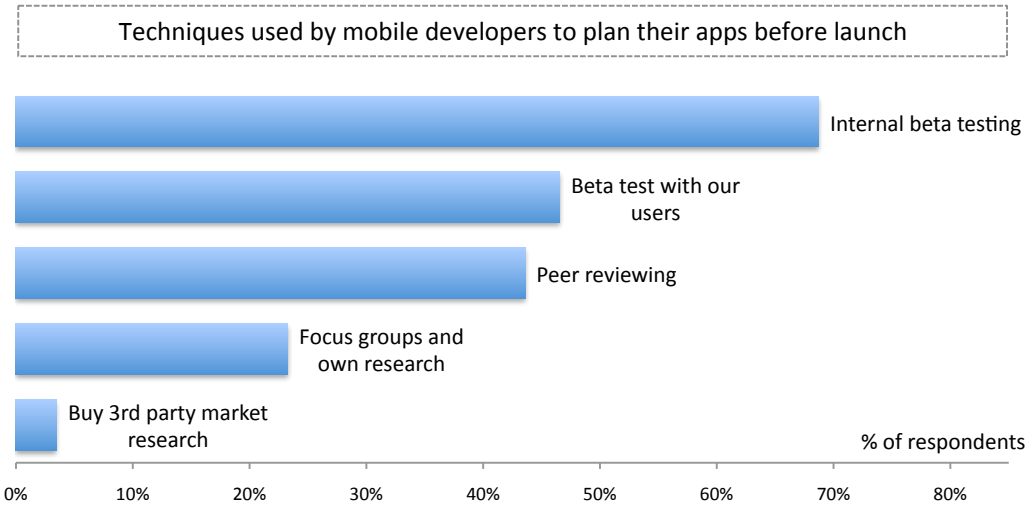


Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Planning an application

Application planning is a core part of taking an application to market. Our research confirms that planning techniques are near-ubiquitous for application developers. Over 90 percent of respondents use some form of technique for deciding on the target user segment or planning application features. The key exception is companies that engage in bespoke app development, and are contracted to execute specific requirements where feature or segment planning is not part of the project.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Internal beta testing is the most popular technique used by the vast majority (nearly 70 percent) of respondents, with beta testing with users and peer reviewing the next most popular techniques. Only 20 percent of respondents use focus groups or research of their own. Overall, North American developers are somewhat more sophisticated in their application planning, with 97 percent using beta testing as a standard part of application development and with broader use of a portfolio of planning techniques as well.

Yet, small development firms have limited means today to beta test and peer review their applications with a cross-section of representative users. Given the hundreds of thousands of mobile apps, we believe that efficient (crowd-sourced) testing of apps in a global market of users is considerably under-utilized. . This presents an opportunity for the few solution providers in this segment – Mob4Hire and uTest.com, for example – but also for network operators, who can generate a channel for testing applications with end users, and provide an open feedback support system back to developers.

> "It's like going to a record store with 200,000 CDs. You'll only look at the top-10."
>
> Christopher Kassulke, CEO, HandyGames,

Challenges with taking applications to market

Application distribution may be going through a renaissance period that began in 2008, with the direct-to-consumer model pioneered by Apple's App Store. However, taking applications to market is still plagued with numerous teething problems, as is typical with nascent technology. There are four recurring issues reported by developers: app exposure, app submission (and certification), low revenue share and the challenges with app localisation.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Challenge 1. Application exposure

Our survey found the number one issue for mobile developers to be the lack of effective marketing channels to increase application exposure, discovery and therefore customer acquisition. This was an issue mostly for Flash and iPhone developers, followed by Symbian, Android and Java ME developers. Developers reported persistent challenges with getting traffic, customer visibility or in short "being seen". One developer put it succinctly: "It's like going to a record store with 200,000 CDs. You'll only look at the top-10."

The exposure bottleneck is new in mobile, but an age-old problem in fast moving consumer goods (FMCG). With such large volumes of applications in stock, app stores are taking on the role of huge supermarkets or record stores. As in any FMCG market, app developers have to invest in promoting their products above the noise, because supermarkets won't.

Our research shows that in 2010, developers are relatively unsophisticated in marketing their applications. More than half of developers surveyed use free demos and a variety of social media, i.e. the *'de facto'* techniques for application promotion. Other techniques cited were magazines and influencing analysts or journalists, while promotion through tradeshows was also deemed popular among a fifth of respondents. Less than 30 percent of respondents invest in traditional marketing media such as online advertising or professional PR services.

When asked about what type of marketing support they would be willing to pay for, our survey found half of respondents willing to pay for premium app store placement. This willingness varies greatly by platform, however; developers whose platform features a 'native' app store (iPhone, Android and to a lesser extent Symbian) are almost twice as likely to pay for premium app store placement, compared with developers whose platforms do not (Java and mobile web) as well as Windows Phone. This finding indicates that direct-to-consumer distribution channels are necessarily crowded and therefore developers will be willing to pay a premium to be able to stand out from the crowd – much like how FMCG brands pay for premium shelf space in supermarkets.

> "In past efforts with third party apps Nokia has been much too lenient towards piracy."
>
> Sander Van der Wal,
> Owner, mBrain Software,

Yet with free applications being the norm, developers have to become more creative with promotion and advertising; free applications make up more than half of the Android Market catalogue and 25 percent of the Apple App Store catalogue, according to different reports by Distimo and AndroLib.

There are two types of solutions emerging to cover the market gap of application promotional services. Firstly, there are app discovery and recommendation startups (e.g. Apppopular, Appolicious, Appsfire, Apprupt, Chorus, Mplayit and Yappler), which help users discover applications based on their past preferences or on explicit recommendations from the user's social circle. Secondly, there are white label app store providers like Ericsson that are moving to app mall (shop-in-shop) infrastructure. App malls will allow the creation of 1,000s of application mini-stores, each targeted to niche sub-segments, much like Amazon mini-stores.

However, the gap in application marketing services is widening in 2010 due to the rapid growth in application volume, which is outpacing the appearance of app discovery and recommendation solutions. We believe that application marketing and retailing services remain the biggest opportunity in mobile applications today. The

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

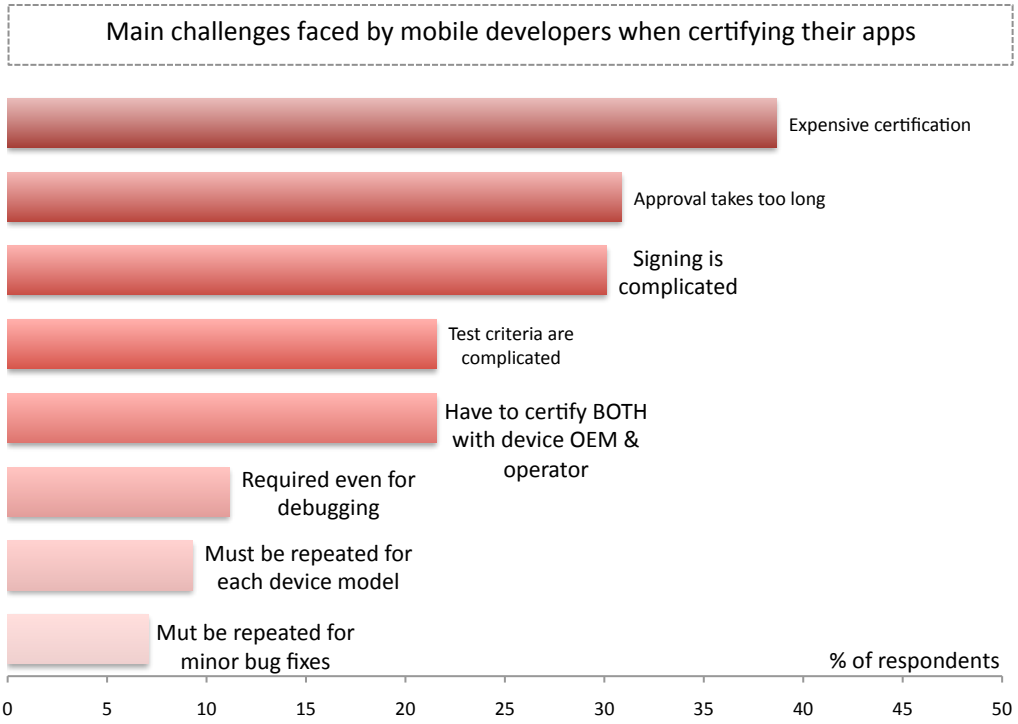
opportunity is particularly fit for network operators, who have a great level of insight into their customer segments (incl. age bracket, spend bracket and roaming profile) and can offer segment targeting for mobile applications as a service to developers.

Challenge 2. Application submission and certification.

Application submission and certification are two of the top five challenges for mobile developers, according to our survey.

Overall, the most important issue related to certification that was raised by nearly 40 percent of respondents is its cost. In some cases, developers report that the certification cost rises to a few hundred dollars per app certification (not per app). Such economics do not work for low-cost apps, but only for mega-application productions. Java developers, for example, report that Java Signed is impractical; developers have to purchase separate certificates based on the certificate authority installed on the handset – and certificates are expensive.

> "The whole signing process and its implications are the biggest threat to J2ME in the future... no one but the signing authorities benefit."
>
> Steven Jankelowitz,
> Java developer



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Moreover, 31 percent of respondents believe that approval of the application takes too long, while 30 percent think that signing applications is complicated.

Signing the application was a challenge for one third of the Symbian and Android respondents that certified their apps. For iPhone respondents this issue was much less important, since only 15 percent of them reported it as such.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

For iPhone developers, the key challenge was not the cost of certification, but the length of time it took for their apps to be approved, as well as the opacity of the approval process, according to anecdotal reports. Others cited the lack of transparency in the acceptance process, which is subject to "Apple's whims," or based on unclear acceptance criteria in the case of Java ME.

> "[Apple's App Store has] unwritten rules for certifying apps"
>
> iPhone developer

Challenge 3. Dubious long-tail economics

The mobile app economy is nothing short of hyped from the successes that have come into the limelight – the $1m per month brought in by the Tap Tap Revenge social app, or the $125K in monthly ad revenues reported by BackFlip Studios on their Paper Toss app. Yet the economics for long-tail developers – i.e. the per-capita profit for the average developer – remain dubious at best.

At least 25 percent of Symbian, Flash, Windows Phone and Java ME respondents reported low revenue share as one of the key go-to-market challenges. Most app stores are still playing catch-up to Apple in terms of the revenue share they are paying out to the developer. As one developer put it, "There has been a bastardisation of the 70/30 rule which has been mis-marketed by app stores; for example with Ovi Store, where operators often get 50 percent of the retail price, so developers gets 70 percent [of the remainder]". Unsurprisingly, the revenue share was not a major challenge for iPhone or BlackBerry respondents.

Moreover, less than 25 percent of respondents stated that revenue potential was one of the best factors of their platform; on average revenue potential ranked last among "best aspects" of each platform, showing how mobile software development is still plagued by poor monetisation in 2010.

> "There has been a bastardisation of the 70/30 rule which has been mis-marketed by App Stores"
>
> Flash Lite developer

The dubious long-tail economics are reinforced by our findings on developer revenue expectations. Only five percent of the respondents reported very good revenues, above their expectations, while 24 percent said their revenues were poor. Note that we didn't poll for absolute revenues, because of the discrepancies across regions, different revenue models and distance of developers from revenue reporting.

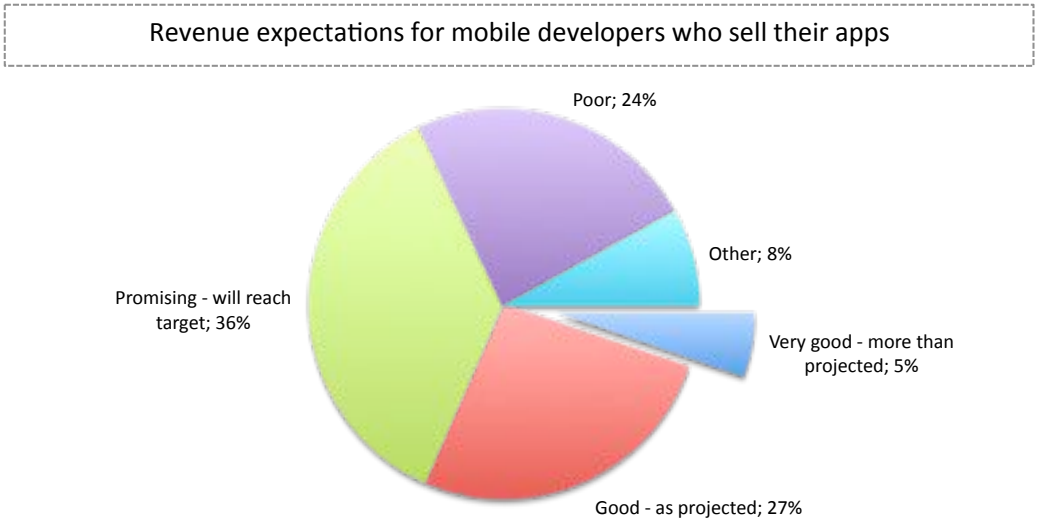
At the same time, there is a general consensus of optimism; 27 percent of respondents said that their revenues were as projected, while another 36 percent said they should be reaching their revenue targets.

There are two effects at play that make for poor long-tail economics. Firstly, the number of 'garage developers' who are creating apps for fun or peer recognition but not money; and secondly, the noise created by the 'app crowd' which prompts developers to drop prices in order to rise to the top of their pack.

There are also platform-specific effects: the unpredictability of revenues, in the case of the Apple's pick-and-choose culture for featured apps; and, the limitations of paid app support for Android, where paid applications are only available to users in 13 countries out of 46 countries where Android Market is available, as of June 2010. Android has also been jokingly called a "download, buy, and return business",

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

referring to how you can get a refund for any paid Android application without stating a reason within 24 hours of purchase – a policy that allows many users to exploit the system. In addition, the applications that are published on Android market are not curated by Google, resulting in 100s of applications that are low quality or are infringing copyright, thereby making it harder for quality, paid apps to make money. Even in economically healthier ecosystems like Apple's App Store, a standalone developer can hope to sell in total an average of 1,000-2,000 copies of an application at an average price of $1.99, which is barely justifying the many man-months of effort that it takes to develop a mobile application by today's standards.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

We maintain that the monetisation potential for the long tail of apps won't be realised until effective policies are put in place to curtail the adoption of free apps – for example by enforcing a minimum $0.01 app price. Psychology experiments have proven time and time again how our perception of value is distorted when the price drops to zero. It is time for app store owners to borrow from cognitive psychology to help boost the long-tail developer economy, rather than compete on number of downloads.

Challenge 4. Localisation.

Another issue highlighted was the lack of localised apps. One developer said characteristically, "There is a big problem for developers in markets with low penetration of English as a second language. Since the platforms are poorly adjusted to localisation, the costs of development grow and thus profitability and attractiveness [drop]. It would be great to see platforms that take action towards easing the challenge of localisation." The lack of localised apps for non-English markets is exacerbated for Android. A search on AndroLib reveals that out of the approximately 60,000 apps on Android Market, there are only about 1,400 apps localised in Spanish and only 1,800 localised in French, as of early June 2010.

The lack of localised apps on Android presents the number one opportunity for alternative app stores like SlideMe, AndAppStore and Mobihand, i.e. to attract communities of regional app developers, or to facilitate localisation of apps to different languages – in other words, to reach where Android Market doesn't reach.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Part 3

The Building Blocks of Mobile Applications

Part 3. The Building Blocks of Mobile Applications

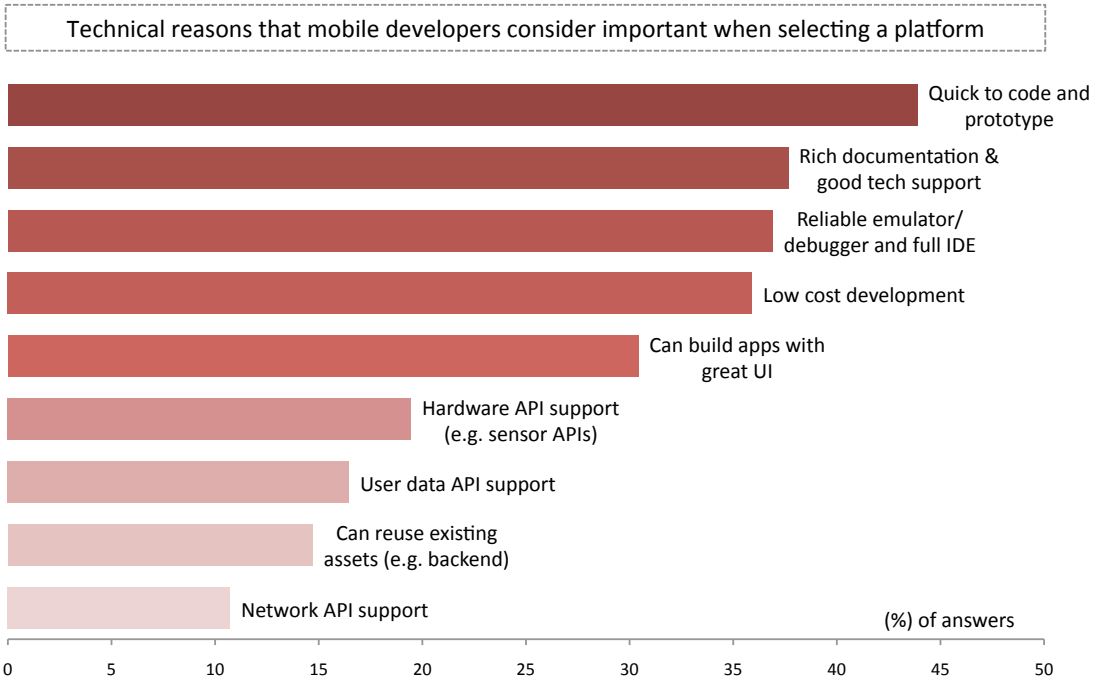
In this chapter we analyse the developer experience during the many touchpoints of application development; learning the platform, coding and debugging the application, building the UI and getting support. We finally look at issues around the adoption of open source.

The fun side of mobile development

Contrary to traditional marketing acumen, developers still care about the 'experience' and 'fun' of developing, as opposed to purely taking an interest in the marketability or revenue potential arising from a platform.

The most important technical reason for selecting a platform was "Quick to code and prototype", selected by almost half of respondents. This finding highlights how the 'fun' aspect of software development (trying things out and getting results quickly) is most important to mobile developers.

We also found that iPhone and Flash developers ranked a platform's ability to build apps with great UIs two times as high, compared to their peers. The inverse is also telling – Symbian and Windows Phone are less appreciative at the importance of UI for mobile apps. Moreover, developers who are new to mobile tend to deem being able to build apps with great UIs to be much more important than do more experienced developers.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

The varying characteristics of the mobile development experience surfaced in our survey when we asked about the best aspects of each platform. Android, Flash and mobile web respondents were impressed by their platform's ability to code and prototype quickly. iPhone respondents were mostly happy with iOS's ability to create a great user experience for their apps. Lastly, Windows Phone respondents were pleased with the platform's emulator and debugger.

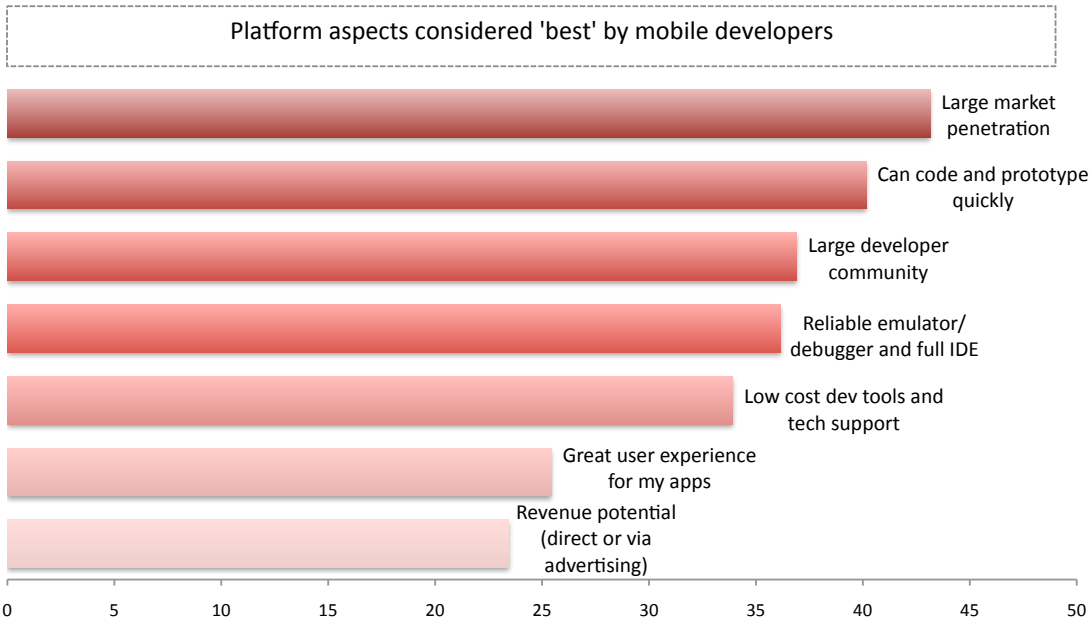
> "Mobile web is the easiest platform to learn, adapt and develop for, esp. for a company that does not have a background in mobile development."
>
> mobile web developer

The pains with mobile development

Bigger discrepancies in the development experience across platforms surfaced when we asked what developers hate most about their platform.

The ability to build compelling UIs is still far from the reach of most mobile developers. Around 50 out of 100 Symbian, BlackBerry and Windows Phone per-platform respondents are annoyed with the difficulty in creating great UIs.

The other key dislikes cited across platforms were the app porting experience, limited hardware API support and the complexities of code development.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Respondents focusing on the Java platform are the most dissatisfied with their platform, lamenting its limited hardware API support, the challenges in porting apps and the difficulty in creating great UIs. Java developers also seem disillusioned with the promise of cross-platform support; most Java developers thought that the future of app development lies in native, not cross-platform environments.

A CEO at a games development house reported how the Java apps market is only suited for the short-head (as opposed to the long-tail) of developers: "You need to customise [your app] for 1,000+ variants and operator customisations." He continued to explain how the mindshare shift to Android and iPhone are causing

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Java apps to stagnate: “Two years ago, network operators reviewed 20-30 new [Java] titles a week. Now they are down to five titles a week. There is not enough [Java] content [available].”

Symbian developers find it challenging to create great UIs, and are also dissatisfied with the long development times needed to create apps in Symbian. In contrast, they complain the least about both hardware and user data API access.

It comes as no surprise that iPhone respondents have the least number of reasons to be dissatisfied with their platform. The only factor that troubles them to a limited extent (30 percent of respondents) was the need to port their apps. Hardly any iPhone respondents complained about the App Store, the revenue potential or the number of iPhones in the market.

Finally, Android developers are a little concerned about the platform's low device count and, to a lesser extent, tech support & documentation, the challenges of porting apps and the difficulty of creating great UIs.

Learning curve and development experience

The learning curve of mobile platforms varies widely, a consequence of varied design objectives. At the two opposite ends of the spectrum are Symbian, a platform conceived in the mid-90s for embedded devices, and Android, a platform designed in the mid-90s for developing connected applications for mass-market smartphones.

Our research confirms that Symbian is by far the toughest platform to learn, with responses indicating that on average the Symbian platform takes 15 months or more to learn, compared to an average of 7.5 months for other mobile platforms. The slow learning curve on Symbian has a direct impact on the higher resource investments made by mobile software firms to cultivate or hire expensive Symbian talent. At a time when the demand for Android and iPhone custom development is booming, the cost of Symbian investments is getting harder to justify for software firms.

Diametrically opposite to Symbian is Android, which is the easiest platform to learn, taking on average less than six months. Twenty-two percent of Android respondents state that it took them less than a month to learn the intricacies of this platform. Apart from Android, the easiest platforms to master are iPhone, Flash and mobile web.

Between the two extremes, Blackberry development takes on average 10-11 months to learn properly, followed by Windows Phone at nine months, Java ME at 8.5 months and iPhone at just over seven months.

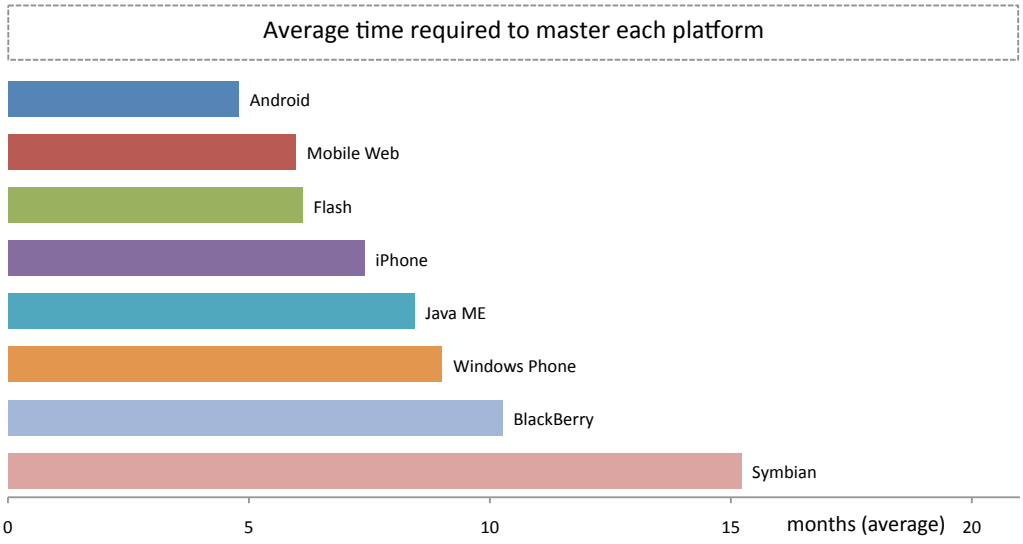
We also analysed the coding experience through hands-on development and debugging of nine reference applications across Symbian, iOS, Android and Java ME.

> “Since I come from a Java ME background, it was initially difficult to understand iOS's syntax and language. But any developer who knows the basics can definitely pick it up.”
>
> iPhone Developer
> Mobicule Technologies, India

Symbian emerged as needing the most tedious development effort to accomplish even simple tasks. For developing nine typical applications, a Symbian developer needs to write almost three times more code than an Android developer. iPhone is based on a complex C-like programming paradigm, but its drag & drop design

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

environment allows for far more effective coding, resulting in half as much code authoring as Symbian. Java ME is lagging slightly behind Android in terms of the overall coding effort.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Our benchmarking analysis further shows that debugging on Android is faster than on any other platform – and two times faster than debugging on Symbian. Second runners to Android are iPhone and Java ME, in terms of debugging effort. For details into our platform benchmarks, see Appendix 2.

Development environments and their challenges

Our research indicated that the most prevalent issue with the development environment (IDE) is the absence of an app porting framework – particularly for Symbian and Android developers. However, it's worth noting that the "fragmentation" problem space is targeted by a number of vendors, including J2ME Polish, Javaground, Mobile Distillery, Mobile Sorcery, OpenPlug, Recursions software and Rhomobile (for a complete list of vendors providing developer tools, see VisionMobile's Industry Atlas on page 2 of this report).

Developers further report a number of issues with the emulator and debugger, both essential parts of the development toolkit. The single most important emulator/debugger pain point, as chosen by more than 40 percent of respondents, was its speed. A slow-start emulator was a very common problem, with a whopping 60 percent of Symbian and BlackBerry respondents reporting this as the main fault with their emulators, a view also shared by more than half of our respondents for Android. On the opposite end of the spectrum, hardly any iPhone or mobile web respondents had this issue.

Another common complaint across all platforms was that the emulator does not accurately mirror the target device, an issue cited by more than 25 percent of respondents across platforms. The performance of device emulation on mobile

> "What I don't like about iOS is that there is a long signing process.. and we need to purchase a 99 dollar IDE in order to develop an application."
>
> iPhone Developer
> India

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

platforms has improved over the years, but is still a moving target, as actual devices will always have differences in implementation, and the latest hardware features that may not have been 'baked' into the emulator binary.

All in all, development tools are a critical component of the development experience, but they leave a lot to be desired. The next table summarises the four key challenges in developer tools across all eight platforms, and the degree to which they were encountered by our survey respondents.

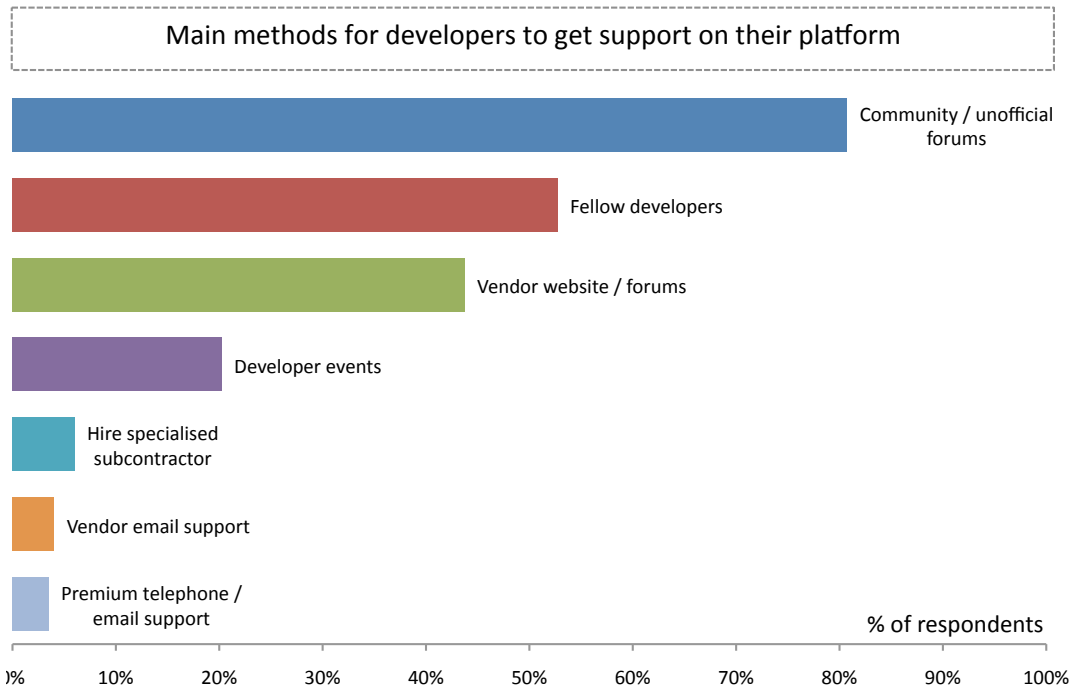
> "There's no standard emulator or debugger for [mobile web]..and error messages are hard to figure out."
>
> mobile web developer

Where do developers reach for technical support?

Developer support programs vary widely across platforms. While some platform vendors have invested heavily in supporting developers through official portals (e.g. Nokia and Qualcomm BREW), other vendors have let the community support itself. For example, Google has set-up forums for Android support, but seldom responds directly to developer queries.

> "Nothing wrong with XCode at all - brilliant IDE that's very fast (esp. compared to Eclipse)."
>
> Alistair Phillips,
> iPhone Developer

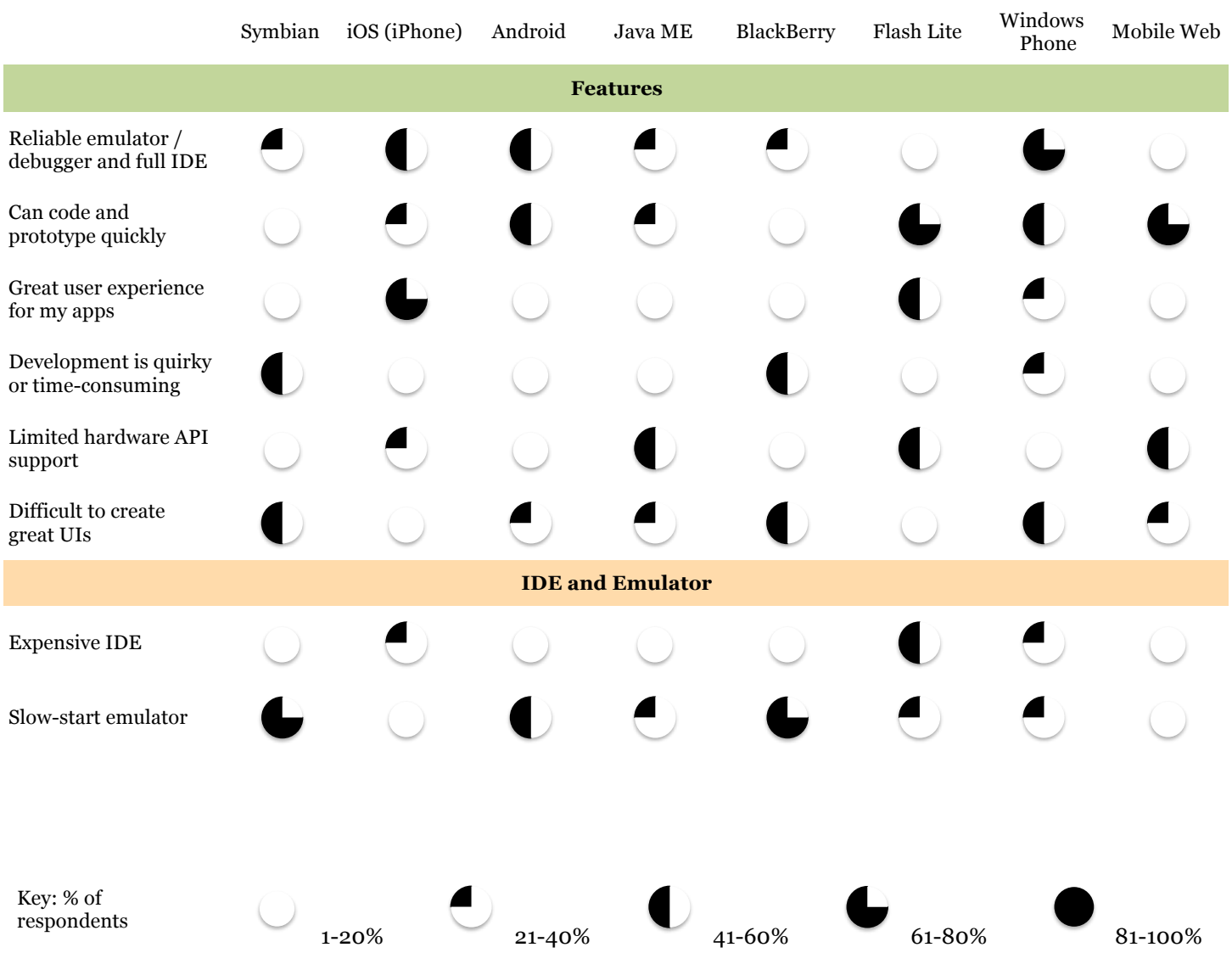
The vast majority – more than 80 percent – of developers rely on community or unofficial forums for support during software development. Android, Java and Symbian developers are the most reliant on community support. Vendor websites are used for support by only 40 percent of respondents.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Platform scorecard: Feature highlights in IDE and emulator/debugger by Platform

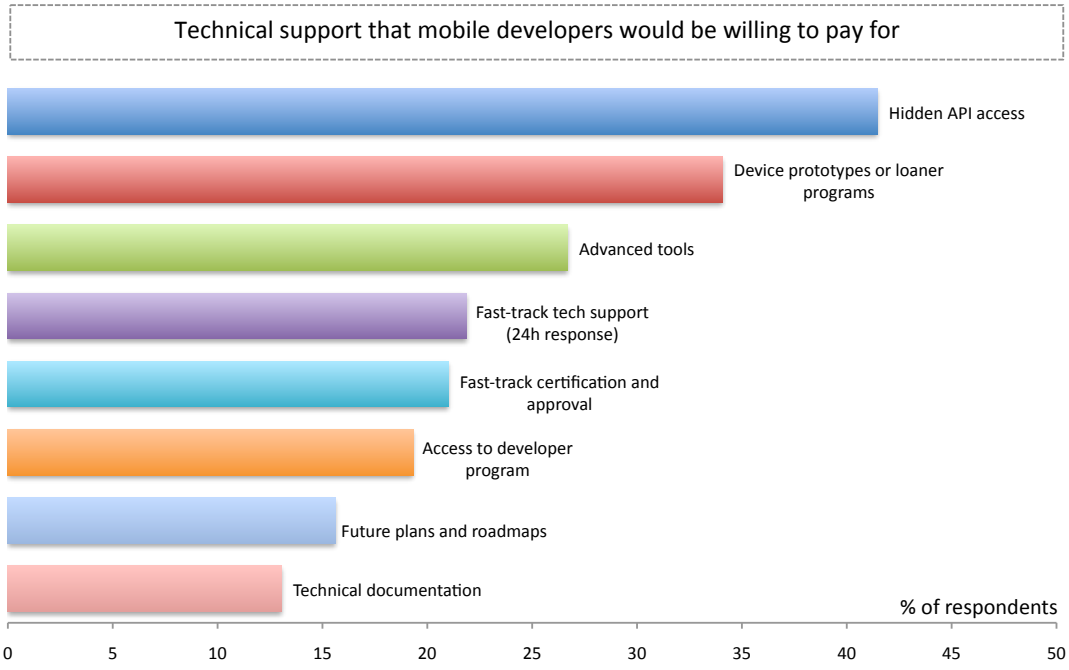


Developer events are a means of getting support used by around 20 percent of respondents, with Windows Phone, mobile web and Flash developers most likely to attend events to get up to speed with their platform. On the contrary the current services offered by platform vendors, such as email or premium telephone support, are not well received by developers, with fewer than 5 percent of respondents using such services.

The findings point out how the most efficient way for platform vendors to support developers is to support the communities first, and let communities support themselves. At the same time this should not be taken to the extreme, as in the case of Google, which seldom responds to direct developer queries on Android official forums.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Our research uncovered further opportunities for platform vendors in paid support programs. Access to unpublished or 'hidden' APIs is a control point for platform vendors, but it is also what developers seem to be willing to pay for – in fact more so than any other type of technical support. Almost 40 percent of respondents are willing to pay for hidden API access, since this would offer them a competitive advantage or allow them to access otherwise unsupported functionality within the device internals. This finding suggests that platform vendors could establish tiered SDK programs, where privileged SDKs are available to developers on a subscription plan. Reaching out to support developers in this manner could create additional revenue streams.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Thirty-four percent of developers surveyed seem willing to pay for device prototypes and loaner programs, a view shared irrespective of the mobile platform developers are targeting. Comparatively few developer programs today offer device loaner and testing facilities, which are simple to set-up, yet monetisable and essential for reducing time-to-market for applications. As such, it's surprising that device loaner programs are not widely established across OEMs and operator developer programs today.

> iOS, Symbian and Windows Phone respondents are the most willing to pay for technical support.

Another interesting insight we came to is that iOS, Symbian and Windows Phone respondents are the most willing to pay for technical support – even if the revenue potential offered across these platforms is very different. Symbian and Windows Phone respondents typically work for large companies (around 40 percent of these respondents work for companies employing more than 100 people) – and therefore would be more prone to getting paid support. In contrast, most iOS respondents are either freelancers or work in small (<10 people)

> "Support is an issue.. takes a bit longer since you have to go through the community."
>
> Jason Delport,
> Paxmodept

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

software shops. This finding highlights the market gap of paid support services for the iPhone and iPad developer ecosystem.

Considering the importance of time to market in such a competitive environment, it is interesting to note that only 20 percent of respondents are willing to pay for fast track certification and support.

> "[I would pay for] any kind of tech support from Google."
>
> Slobodan Ivkovic,
> Lead Mobile Application Developer,
> Esteh Doo

Open source adoption and challenges

In the 2010 shape of developer economics, the notion of open source is closely tied to mobile platforms, with Android, Symbian, MeeGo, Java ME, and WebKit using various forms of open source licensing.

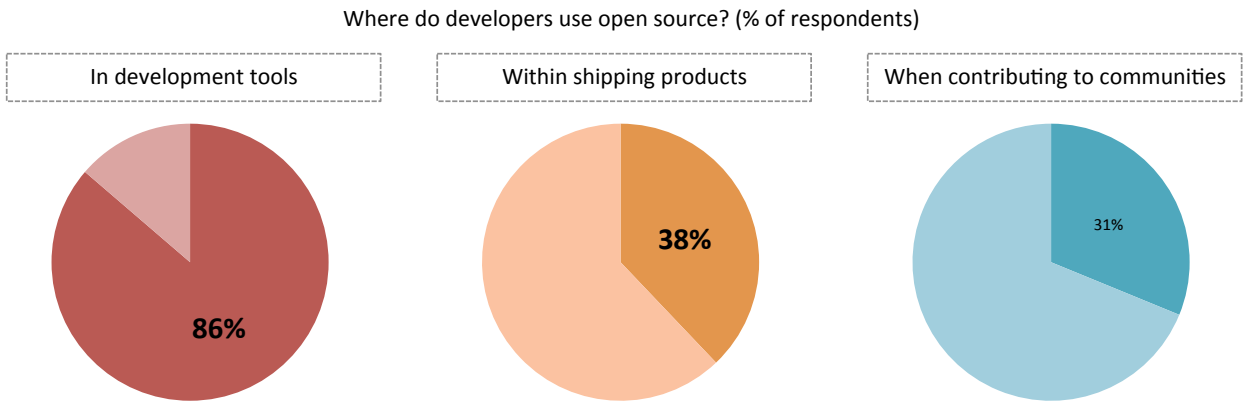
Within the space of just two years, open source has created the biggest disruption the mobile industry has ever seen, second only to the Apple's iconic product series and the app store paradigm. The announcement of Google's Android in late 2007 triggered the sale of Symbian to Nokia and the relicensing of the platform, killed off UIQ and MOAP platforms, marginalised Microsoft's Windows Phone and set new norms for pricing mobile operating system royalties down to zero.

Similarly, the WebKit browser engine, debuted in 2005 by Apple, has resulted in the exit of Teleca's Obigo and the sale of Openwave's browser business in 2007, which were until then the two biggest-selling browsers for mobile handsets. Today WebKit is the *de facto* browser engine for mid/high-end mobile handsets, shipped in more than 250 million handsets as of the end of 2009 (see www.100millionclub.com).

Besides disrupting the operating system and browser business in the mobile industry, open source has acted as a "mindshare magnet" for tens of thousands of developers. Open source has been employed toward this aim sometimes successfully (by Google's Android) and sometimes unsuccessfully (by Sun's Java Phone ME project).

The use of open source among mobile developers

Notwithstanding its mass adoption, open source remains one of the most misunderstood topics in the mobile industry, both in terms of licensing and governance. Our research probed into two aspects of open source: its use and its challenges, both from the mobile developer perspective.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities.
June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

On average, 86 percent of respondents who use open source at work use it within development tools such as Eclipse.The exceptions are iPhone and Windows Phone developers, who are less heavy users of open source development tools. Another popular use of open source is within shipping products (almost 40 percent of respondents). It's worth pointing out that BlackBerry developers are by far the least active users of open source within shipping products, which indicates a commercial skepticism – one RIM will have to overcome as it gradually adopts open source software within its devices starting with WebKit.

Overall, developer involvement in open source correlates highly with background. Android and iPhone developers are three times more likely to lead open source communities compared to Symbian developers. This reveals the contrasting pedigree of the two developer communities; iPhone and Android developers originate from the Internet domain where open source has existed for more than 10 years, while Symbian developers come from the mobile domain, where open source is relatively new.

Challenges and opportunities with open source

We asked mobile developers what were the key drawbacks to open source and the results are convergent. The consensus among developers pointed to open source licensing: 60 percent of respondents thought that the main issue with open source was the confusion created by licenses.

One developer said characteristically: "Corporations are wary of the licensing terms and err on the side of caution – by avoiding open source altogether." More than 10 percent of respondents went as far as to say that open source was a viral threat to software companies. This presents an opportunity for organisations vying for developer mindshare to provide developers with much-needed education on open source licenses, and thereby boost their awareness and exposure among developer communities – especially as open source remains a hot topic in mobile development in 2010 and beyond.

Another reason cited as a drawback to open source by 25 percent of respondents is that one can't make money with open source. One developer said characteristically that "it's hard to convince people to pay for software," while another said, "If your app uses too much open source it's easy to duplicate". Again, these present concerns that will benefit from education on the commercial use of open source and how the use of open source is in fact orthogonal to the business model, as demonstrated by 10s of case studies in the mobile industry.

Another often-cited downside is that open source projects **lack documentation and support,** causing delays to the development and release cycles. Other developers maintained that liabilities arising from the use of open source software can **cause unexpected costs**. Other notable comments were that open source projects lacked **commercial promotion behind them**. Again, these are challenges for which best practices exist already within OEMs and software vendors whose management has defined open source policies and processes.

> "Corporations are wary of the licensing terms and err on the side of caution - by avoiding open source altogether."
>
> Android Developer

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Part 4

The Role of Networks in Taking Apps to Market

Part 4. The role of networks in taking apps to market

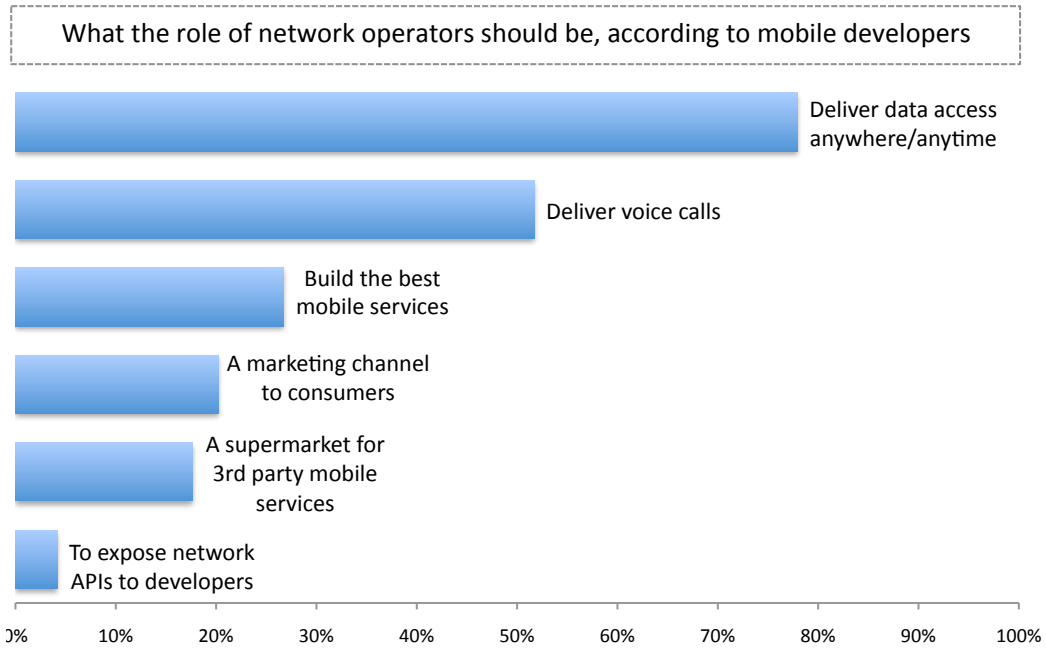
Mobile developers have in recent years become a new target 'supplier' for network operators ("carriers," if you live in North America). Developers are seen as the key to driving innovation within and on top of the network and helping operators profit as distributors of the most popular apps.

Networks have exposed capabilities (e.g. location) for third party services since the beginning of the century, but have only done so for their major (read: multi-million-dollar) content partners.

Since 2004, networks have begun exposing their network capabilities as an offering towards the 100s of mobile content developers (e.g. Orange Partner). Since 2009, networks have opened up enabler APIs to the 1000s of businesses (e.g. Telenor's Mobilt Bedriftsnett) and the hundreds of thousands of mobile application developers – as with O2 Litmus, Vodafone's Betavine, Orange Partner and Telenor Fusion, followed closely behind by North American operators AT&T, Sprint and Verizon.

Hallmarking the importance of the developer to the mobile industry, the headline mobile events of 2010 – Mobile World Congress and CTIA – both featured dedicated developer pavilions that attracted heavy marketing investments from network operators in Europe and North America.

Our research found that while mobile networks are vying for developer mindshare, the opposite is not true; in their majority, developers seem apathetic towards networks.



urce: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities.
ne 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Our analysis shows that a majority of mobile developers view network operators as bit-pipes. Nearly 80 percent of developer respondents think that the role of network operators should be to deliver data access anywhere/anytime, while only 53 percent of respondents considered the role of networks as delivering voice calls. This suggests that developers believe operators have a bigger role to play than just delivering voice calls.

Responses were opinionated and strong, given that the vast majority (95 percent) of respondents voiced their opinion. Moreover, developer sentiment doesn't vary significantly by the level of experience, by platform or by region, so we can assume it is representative of the overall mobile developer community.

It's noteworthy that only 20 percent of respondents see networks as a potential route to market. While network operators are best placed to connect developers to consumers, they are not perceived as a marketing channel to consumers.

What's even more surprising is that only five percent of respondents thought that the role of network operators should be to expose network APIs, hinting at the lack of interest towards network enablers. Some developers were more opinionated: "Operators should get out of the way of developers" was one characteristic comment, which is disheartening for operators spending resources and money in developer programs.

> "What's most important is not understanding demographics, but reaching to these demographics."
>
> Flash Lite Developer

> "[There is a] big gap between intention and outcome. All [operators] talk about supporting developers. But in practice actual support (usable SDKs, decent documentation, support, person to talk to) is lacking."
>
> Malcolm Box, Developer

We believe that the relative developer apathy towards operators is down to three reasons:

A. Lack of operator-driven marketing programs for the long tail of developers, and the slow pace with which network APIs have advanced since the turn of the century. The handful of network API programs available globally are generally in their nascent stages, with many APIs being in beta while the commercially available ones are usually prohibitively priced for small developer shops (Orange Partner and Telenor's CPA programs are good examples).

B. Mismatch between operator ambitions and the market reality. Operators and related initiatives (e.g. JIL) have been keen to develop their own, full-blown app stores – from app ingestion to discovery – when operators haven't yet fixed their key issues, namely 70/30 revenue share when apps are purchased through the operator payment gateway. The golden rule here is that operators can only extract value where they add value – and they have little value to add in app ingestion or application discovery, for example. Moreover, many network API programs focus on enablers like messaging or location, for which better or cheaper alternatives exist. T-Mobile USA's closure of their developer program is perhaps a sign of maturity in this direction.

C. The gap in developers' perception of network value. Much like in the 90s when the first Internet providers had to educate the public on the utility of the Internet, so operators have to convince developers as to the value of the network. As we argue at the end of this section, operators have value to add in two market gaps:

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

micro-billing (at credit-card-like rates) and in helping developers target the right segments through their understanding of consumer behaviour on their networks.

Who should pay for mobile data anywhere/anytime?

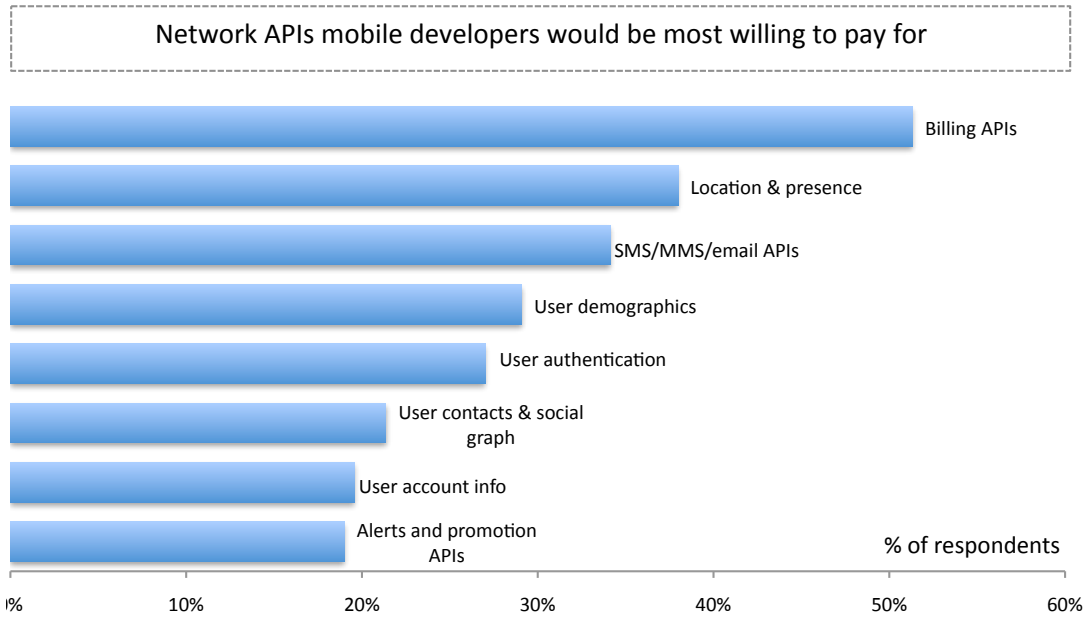
All in all, mobile developers see the role of networks as delivering data access anywhere/anytime. But, given the billion-dollar infrastructure investments operators are making (3G, HSPA, LTE) and the incremental (per-MB) cost of offering data access, the obvious question is: who should pay for mobile data?

Our research found the consensus of developer opinion pointing to a tiered revenue model, where users should pay extra for premium bandwidth. This practice of bandwidth tiering has long been debated within standards groups, but it has been applied only in very limited contexts (e.g., dual flat-rate pricing structures with DoCoMo in Japan). At the same time, pricing plans always need to strike a fine balance between demand and supply; increasing data prices (with tiering or without) might reduce usage and therefore revenues for both operators and app developers.

More novel business models for data access were mostly unpopular in our survey; proposed models included taxing traffic-generating applications, adding a state tax on applications, ad-supported data access, charging app store providers and subsidising bandwidth from the sale of device or application analytics.

What network APIs would developers pay for?

Many tier-1 operators today in Europe and North America are offering network APIs, i.e. a set of programmatic interfaces allowing developers to leverage network capabilities from their applications. Examples are APIs to allow developers to detect device capabilities, location, send SMS or emails and tap into user profiling information.



urce: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities.
ıe 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Our research shows that developer opinion isn't strongly converging to any single API; billing APIs was the only one that attracted interest from more than half of respondents. Also, around one in three developers said they would pay for SMS/MMS/email APIs and location/presence APIs. The only other single API which garnered a positive response from more than 30 percent was the user demographics API, allowing developers to tap into user profiles like age bracket and spend bracket (privacy concerns aside).

> "There needs to be generational shift in management at operators."
>
> Antony Hartley,
> CTO, MoSync AB

Are standards needed for network APIs?

As network API programs are maturing, standards groups are emerging – such as GSMA's One API initiative and the Wholesale Applications Community – to homogenise the technical and commercial access to operator network capabilities.

Our research showed that more than 60 percent of respondents think that unified API specs are needed, followed by around 45 percent of respondents who felt a single entry point for API calls was called for. There was no noteworthy variance for this question across platforms or developer experience levels.

Our findings confirm the current state of the network enablers market. With network API programs still in their infancy, developers are asking for unified APIs and single entry points. As network API programs mature over the next three years, the need for unified pricing and a single commercial framework should dominate discussions within the developer community.

Despite the availability of many network API programs, developers in the post-app store era are all painfully aware of the commercial reality – and the importance of monetisation over and above standards. One developer said characteristically, "Unified 'whatever' is conceptually nice, but as a developer, I really don't care whether I write essentially the same stuff with different API calls twice, as long as I get paid for that."

Are networks supportive towards developers?

We also polled mobile developers for their opinion on the level of support they receive from network operators. Overall, results were quite disappointing for networks. Only 10 percent of respondents thought that operators were adequately supportive towards developers, while almost 70 percent thought that there was little or no developer support from network operators.

The majority of developers had not interacted directly with operators, but were very opinionated towards the level of support they are getting (or not getting) from operators.

Developer comments on this matter were polarised. Anecdotal reports of developers working with operators in UK, Israel or Estonia reflected a positive sentiment. For example, one respondent commented, "Networks like O2 seem to go out of their way with Litmus".

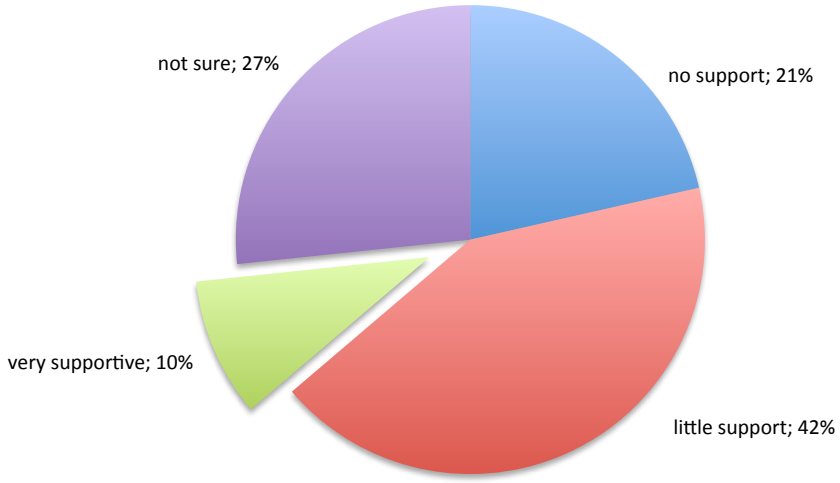
Others were disillusioned with operator support: "Developer support [programs] (e.g. O2 Litmus) do

> "Pay [for network APIs]? Do they want our apps on their network?"
>
> Robert "Ozzie" Osband,
> PHonePHriendly.Com,
> mobile web developer

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

not convert into the creation of commercial opportunities." Or, "In Kenya, Safaricom, the leading mobile operator, has completely refused to grant developers access to the M-Pesa [payment] API." Other responses were equally opinionated: "It's a dialogue [with Vodafone] vs. dictatorship with Orange." or "[Operators] talk a lot, but nothing really happens. This is true for all networks in Spain!"

How much support do mobile developers think they are getting from operators?



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

The overall sentiment has been one of developer disillusionment with operators.

Operators are often seen as engaging in one-way communication, providing only requirements but giving little back - in some cases not even offering documentation for their own APIs. "They receive a huge share from the downloads of each product but give close to zero back to the developers!" Other developers see opportunism; "[Operators] are supportive if they can share revenue".

> "If Google became an operator our problems would be solved."
>
> Peter-Paul Koch, www.quirksmode.org

"They often get in the way of developers and stifle innovation in an effort to protect their products and to limit use of their networks to keep costs for new equipment down. They also fragment the device implementations for APIs in an effort to keep developers locked in to support only them, and as a way to differentiate themselves in the market."

Developers also see operators as unreliable. "They seem to change their alignment every other quarter," or, "The marketing initiatives come and go.. not consistent. Have seen this within both big and small developer companies. Example: O2 innovators will be gone in a couple of years... They do not care about developers."

Quite a few developers thought that the level of operator support was unfairly biased towards the major developer houses – indicating how operator developer programs have not adequately evolved to cater to the long tail.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

"The first mobile company to TRULY reach out to web developers will have an edge over the competition, but right now I don't see any candidates. Except for Google, obviously. (And Apple, but they're playing their own game.) If Google became an operator our problems would be solved."

Opportunities for networks

Despite the overall skepticism, we believe network operators have a crucial role to play in the mobile applications market, particularly in three areas:

1. By providing **access, transparency and support**. There are plenty of ways developers have suggested that operators can help through marketing programs (e.g. helping developers map out the demand for new applications), communication (e.g. frank, open feedback), transparency (sharing service roadmaps) or incubator-like facilities. The latter can provide a variety of essential infrastructure to app developers like subsidised market research, handset loaner programs, testing apps within a sandbox network, or free SIM Cards. Our survey also found that more than 40 percent of respondents are willing pay operators for business development. Again, this is an opportunity for operators to connect with developers and help the most promising applications bubble up to the top.

2. By providing **payment gateways** that are cost-effective for application developers, i.e. offer a fair revenue pay-out towards developers, at least matching the standard (70/30) set by app stores. We believe that if networks further extend their revenue share to credit-card-like revenue payouts (95/5), they would see network billing being adopted for entire new market segments – including retail and Internet payments – that have so far presented dormant (but major) opportunities. Operators should also shorten their payment cycles closer to the levels delivered by app stores to help developers manage cash flow.

3. By **helping developers target the right customer segments** for their applications. Network operators can leverage their customer profile goldmine to allow developers to expose their applications to the right consumer segments. For example operators could help developers target 100s of different segments for their applications (e.g. female executives, texting teenagers or travelling salesmen), based on profiling information (age bracket, spend bracket and roaming profile) that are non-personally identifiable.

This is a match made in heaven; networks have detailed usage information (much more so than banks) that could be used to be build up 100s of consumer segments, without exposing sensitive customer information. At the same time, developers see exposure and marketing as the number one challenge for mobile applications today.

Clearly network operators have a long, promising way to go. But at the same time they also have a finite window of opportunity, as Internet players are gradually breaking down the incumbent control points, whether it's spectrum scarcity, network-controlled termination fees or operator-customised devices.

> "If [operators] did really care about app developers, they would go to the same events that app developers go to, rather than trying to do their own events".
>
> Flash Lite Developer

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Appendices

Appendix 1. Research Methodology

Developer Economics 2010 is a global research report delving into all aspects of mobile application development, across 400+ developers segmented into eight major platforms: iPhone (iOS), Android, Symbian, BlackBerry, Java ME, Windows Phone, Flash and mobile web.

The report provides insights into all the touchpoints of mobile app development, from platform selection, application planning, code development and debugging, to support, go-to-market channels, promotion, revenue generation, as well as hot topics such as the role of open source and network operators.

The objectives behind this research were to analyse the mobile developer experience from two very different angles:

1. Survey the perceptions of mobile developers across the eight major platforms: Android, iOS/iPhone, BlackBerry, Symbian, Windows Phone, Flash/Flash Lite, Java ME and mobile web (WAP/XHTML/CSS/Javascript).

2. Benchmark the app development experience across four platforms (iPhone/iOS, Symbian, Android, Java ME) through hands-on development of nine mini applications.

The survey received 401 responses from mobile developers across the globe, across 35 Q&As, and across the entire development lifecycle, making this the biggest mobile developer survey to date.

Out of the 401 respondents, 172 were interviewed by phone and 229 completed the questionnaire online. Respondents were asked to base their answers on one out of the eight mobile platforms noted above. Up to two responses were allowed per developer, each on a different platform.

Participants

The list of respondents mainly comprised developers working for software companies. Around 20 percent of our respondents were either students or freelance developers, while more than 45 percent worked for companies with 10 to 100 employees. The remaining 35 percent of the mobile developers who participated in this research worked for large companies, with over 100 employees.

In total, our respondents included developers from over 300 different companies. The list included developers working in large telecom companies, including software companies Aplix, Opera Software, and Teleca; handset manufacturers Nokia, Samsung, and ZTE; mobile operators Vodafone, AT&T, Deutsche Telekom, and T-Mobile; and, infrastructure vendors Ericsson and Alcatel-Lucent.

The distribution of participants was positively biased towards experienced developers. More than 40 percent of our respondents had at least five years experience in developing mobile apps, while 87 percent had been developing apps for more than a year. Only 13 percent of the sample was comprised of novices at mobile application development, having less than a year's experience in the field.

Moreover, many of the respondents had received one or more developer awards. Almost 20 participants were Nokia Forum Champions, while three were Grand Prize

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

winners of Nokia's Calling all Innovators contest. Our participants also included three Adobe Community Experts, three finalists of the Android Developer Challenge and two Handango Champions, while others had won the Mobile Premier and Navteq Global LBS Awards. Other participants were winners of the Flash Lite Game Contest, the Betavine and Vodafone Summer of Widgets Contest. The list also included a Flash Lite Developer Challenge finalist and the winner of the Indonesia BlackBerry Developer Challenge.

Questionnaire

The questionnaire used in Developer Economics 2010 consisted of five parts, each with a different focus.

- **Background:** Region, years of mobile development experience and range of experience across platforms.
- **Platform selection and features**: Reasons for selecting platform, best and worst aspects of chosen platform.
- **Code development, tools & support**: Learning curve, IDE and debugger/emulator pain points, getting support on the platform and types of technical and marketing support developers are be willing to pay for, and the relevance of standards groups.
- **Taking applications to market**: App certification, use of app stores, main channel for selling apps, channel revenue and challenges, time-to-shelf and time-to-payment, planning, promotion and revenue models for apps.
- **Hot topics: Open Source & Network APIs**: Open source use and drawbacks, role and support of network operators, API requirements and standardization activity.

Data points were captured through mostly multiple-choice questions, with a maximum of three answers per question. Most questions included an open-ended option, allowing us to capture context-specific comments and qualitative aspects of developer sentiment.

All participants were entered into a prize draw for 25 prizes, including handsets, Amazon vouchers and conference passes.

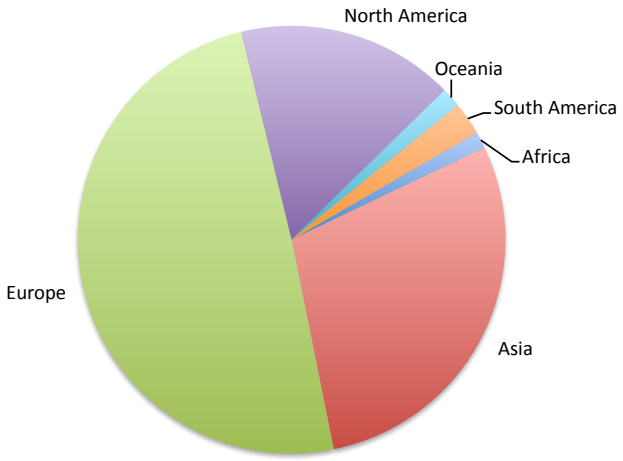
Regional, platform and experience distribution

Our sample of 401 respondents was drawn from a variety of regional and experience backgrounds.

Note that our distribution of respondents is not necessarily representative of the total global distribution of mobile developers across regions or platforms. In interpreting the results of the survey we have often normalised across platforms (i.e. as if we had 100 developers from each platform), as well as checking for response variances based on the level of experience or across regions. The reader may have to apply the insights to their 'reference' developer distribution as applicable in their region or market.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Geographical Distribution



Top-10 countries	
Country	Respondents
India	56
UK	55
USA	43
Germany	29
Canada	21
France	18
Israel	16
Spain	16
China	12
Greece	10

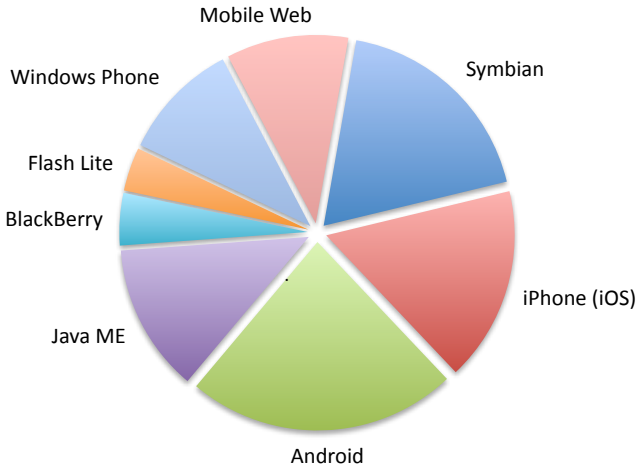
Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Respondents were asked to provide the country they were currently based in, as opposed to their country of origin. Although this research was worldwide, the sample of respondents had a bias towards Europe, as shown in the chart above. Africa, South America and Oceania yielded a small number of respondents.

Platform distribution

Developers were asked to base their answers on the platform that they spent most of their time on. The choice was between the eight major moble platforms, as noted earlier. The platform chosen by the largest number of respondents was Android, followed by Symbian and iOS (iPhone). Regrettably, BlackBerry and Flash Lite were both underrepresented, having been chosen by a fraction of our respondents. The next graph shows the distribution of platforms selected by our 400+ respondents.

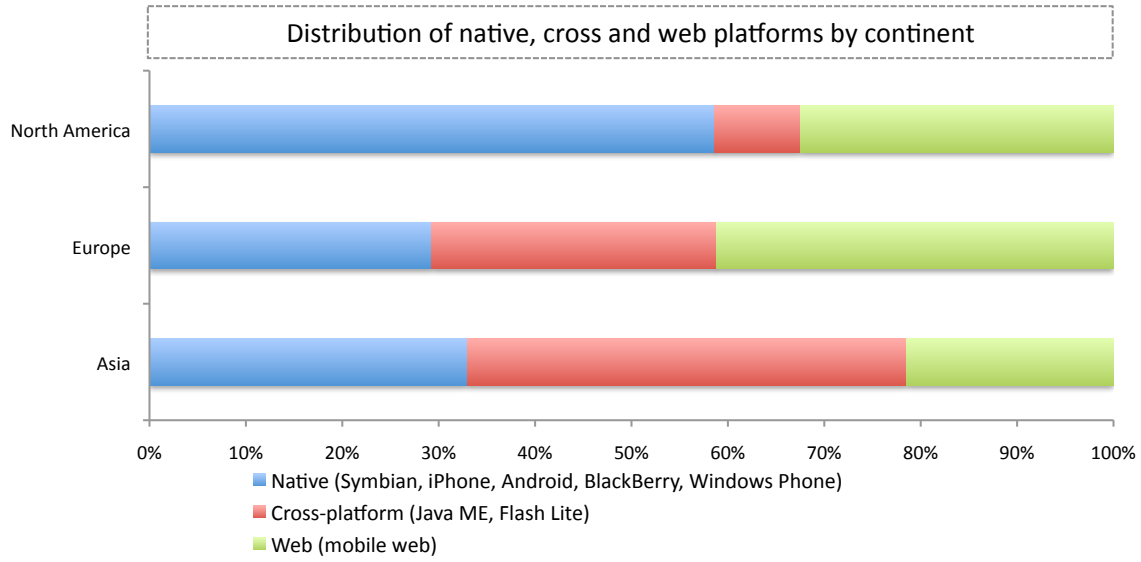
Platform Distribution



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

In terms of geographical distribution, native platform developers (Symbian, iOS, Android, BlackBerry and Windows Phone) who participated in our survey mostly originated from North America, while cross-platform developers (Java ME and Flash/ Flash Lite) made up nearly half of the Asia-based respondents. Developers based in Europe had a more balanced distribution between native, web and cross-platform developers.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Appendix 2. Comparative platform benchmarks

Today's mobile platforms are extremely diverse in their characteristics in a host of different ways, all of which impact the developer experience. Take for example the diversity of coding experience across development languages (C++, Objective-C or Java), environments and tool-chains (xCode, Eclipse, Carbide), emulator performance, on-target debugging performance, programming APIs and idioms (iOS frameworks, Symbian macros, Android Intents or Java ME profiles), available class libraries, supporting SDKs, information available in forums vs communities, and many more.

To research the finer aspects of the mobile development experience, we benchmarked the four key mobile platforms – iOS (iPhone), Android, Symbian and Java ME – against the most common developer tasks: coding, emulator debugging, device debugging, and support resources (SDK, official and community forums).

We structured our research by asking eight developers to prototype nine mobile applications – from 'hello world' to networking, multimedia, sensor and addressbook applications. We then asked each developer to keep track of the time taken in key tasks, including coding, debugging and reading SDKs, as well as key output metrics like lines of code written. To balance out the level of development experience, our panel for each platform consisted of one novice developer (who had not programmed on that platform before) and one expert developer (who had at least one year experience on the platform).

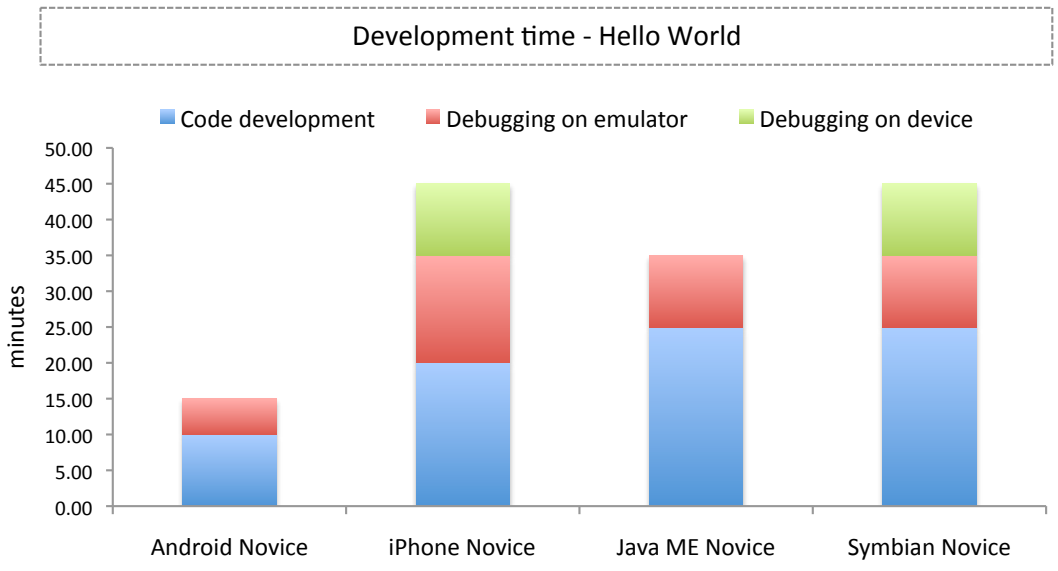
The developers were given application specs in the form of a screen mockup and a description of the application controls and behaviour, as shown in the next example.

We broke down the research results into five key metrics that provide generalisable insights into the key platforms. We next walk through each metric and the insights generated.

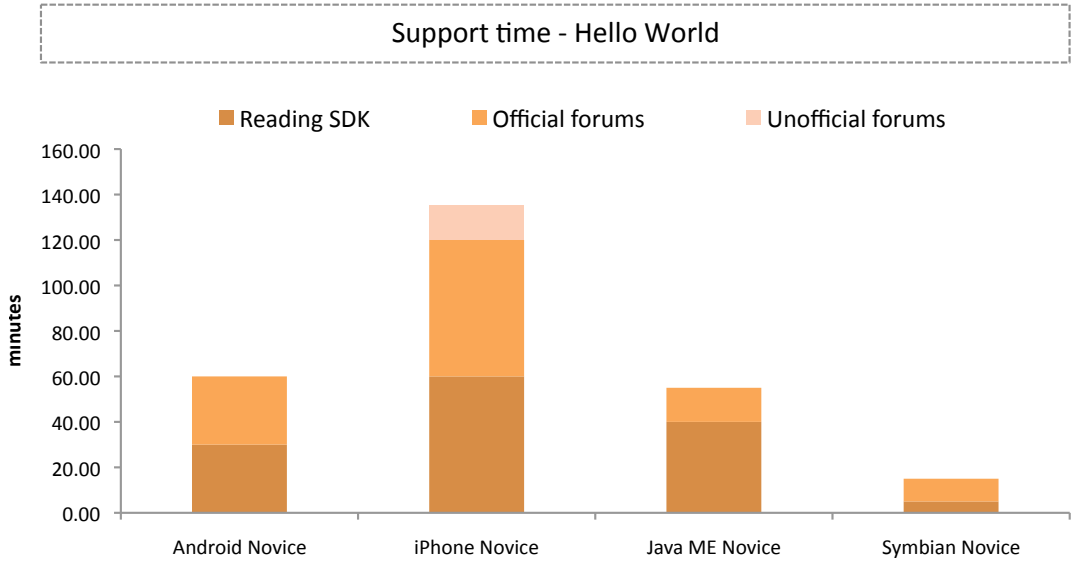
Which platform is the quickest and slowest to get started on?

We measured the time it takes for a novice developer to install the SDK and develop a simple Hello World application.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

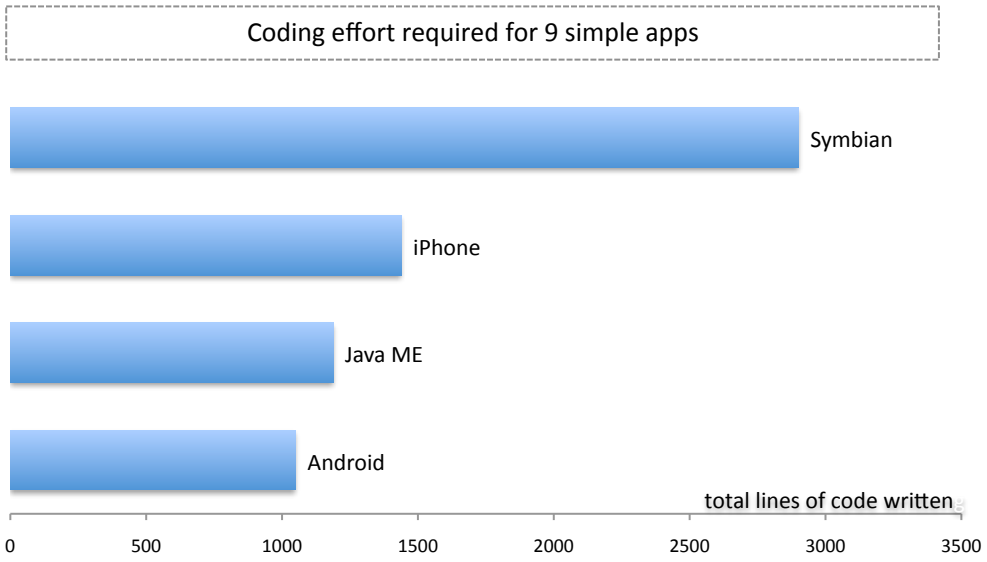


Surprisingly, iPhone is the hardest platform to get started on, taking two to three times longer to set up for a novice developer than other platforms. The iPhone developer spent three hours in total to set up the environment and develop the Hello World app, almost half of which was spent in looking up documentation.

Which platform takes the least and most coding effort for common applications?

Symbian is well known for its quirky development idioms and the tedious C++ development effort needed to accomplish even simple tasks. This was confirmed quantitatively by our research; for developing nine typical applications, a Symbian developer needs to write almost three times more code than an Android developer. iPhone is also based on a C-like programming paradigm, but its drag & drop design environment allows for far more effective coding, resulting in half as much code writing, compared to Symbian. Java ME is lagging slightly behind Android in terms of the overall coding effort.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

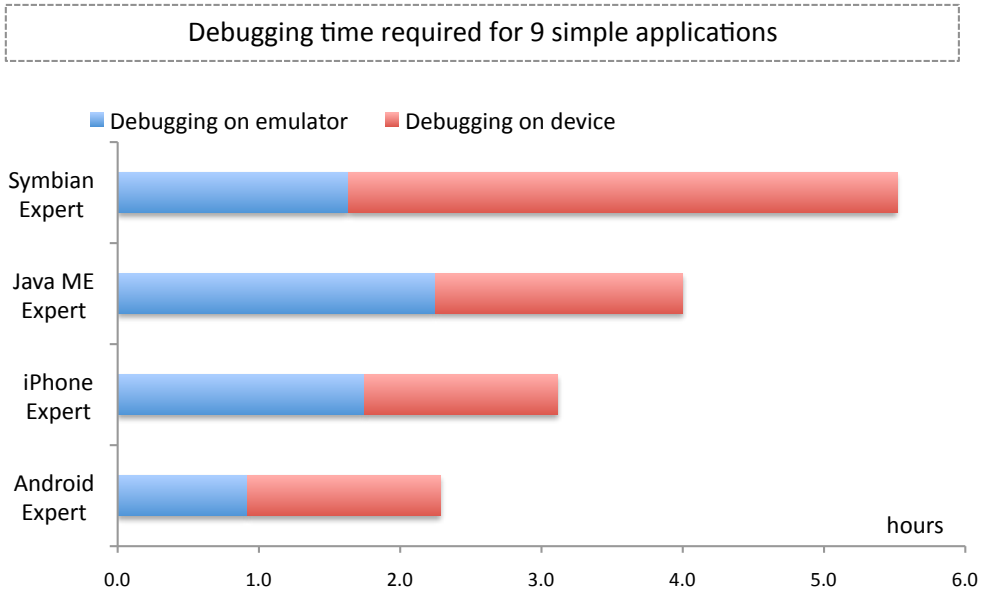


Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Which platform is the fastest and slowest for debugging?

To understand how the debugging effort varies across platforms, we compared the time spent in debugging by all expert developers who were already experienced with the platform. We also compared time taken for on-emulator vs. on-device debugging.

Results show that debugging on Android is faster than on any other platform – and in fact twice as fast as debugging on Symbian. Besides showing its age as a decade-old platform, Symbian presents many challenges with on-target debugging, since the emulator behaviour often differs compared to when the application is tested on the actual device.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Second runners to Android are iPhone and Java ME in terms of debugging effort. Despite Java's simple language, structure and garbage collection, on-target behaviour

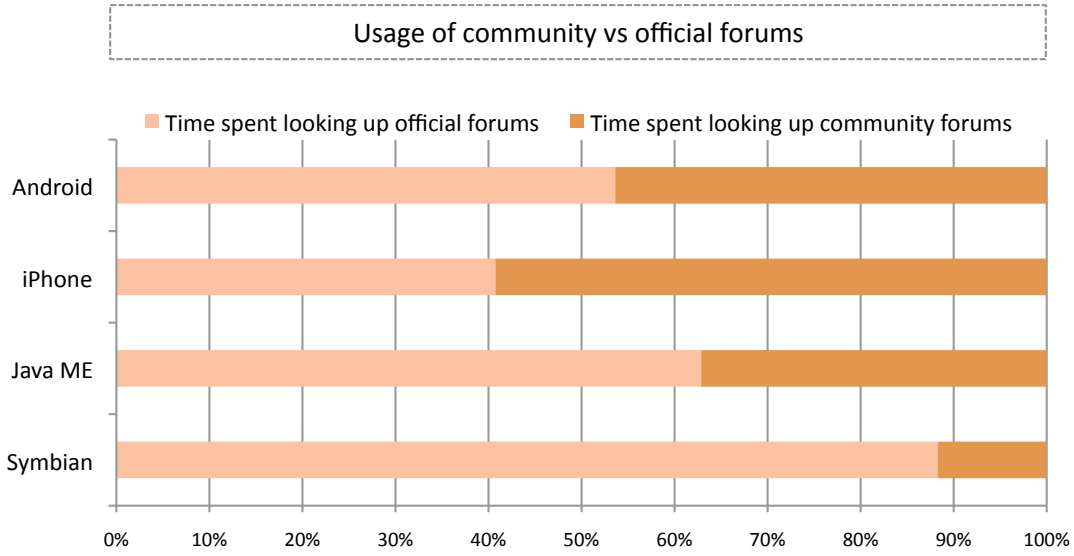
© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

is again dependent on the device and JVM vendor, which increases the application debugging time.

4. How do platforms differ in terms of the documentation & support?

Each platform vendor has a different strategy for in-sourcing documentation as opposed to letting the community support the platform. These variations show up in our research as we looked at how much time developers spent in official forums and community forums.

The level of developer enthusiasm and the huge community that has formed around the iPhone has made community forums the main source for developer support. Our benchmarks show that iPhone offers the strongest community-driven support, followed by Android. On the opposite end of the spectrum is Symbian, where Nokia has done an exemplary job of supporting developers, including a Forum Nokia Wiki and a best-in-class devices database containing hardware specs and platform details.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

How are platforms suited to developing common applications?

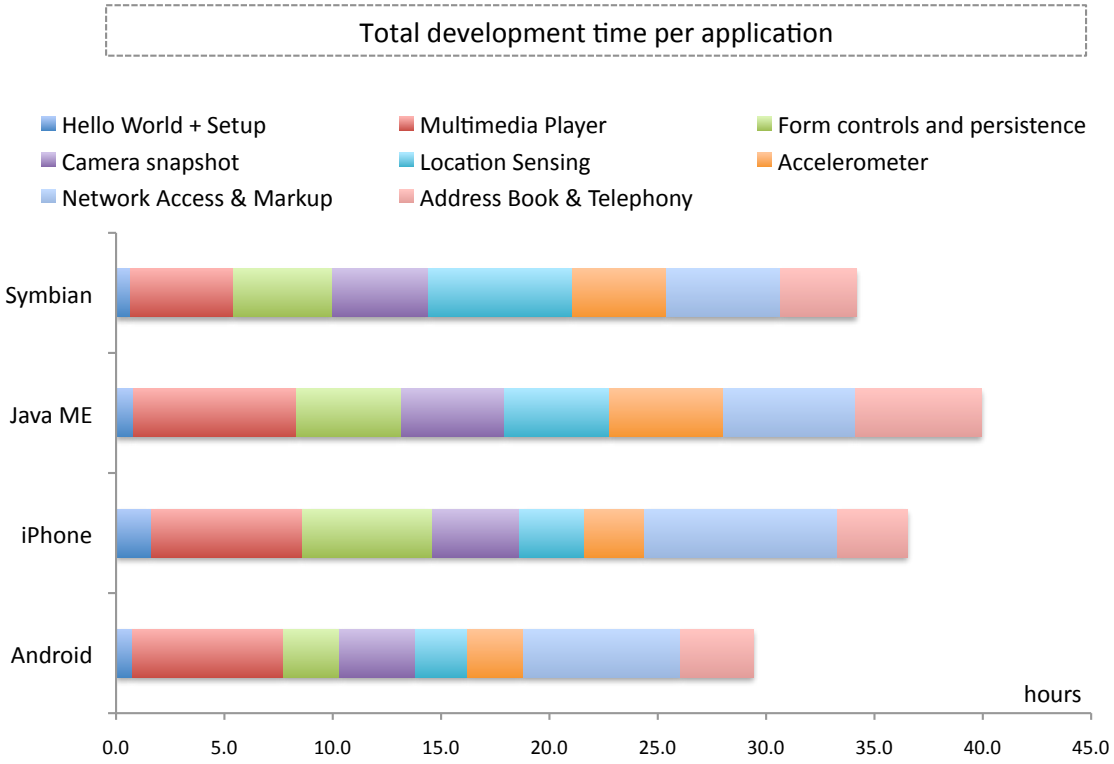
To compare and contrast platforms across a wide range of application use cases, we asked our eight developers (one novice and one expert per platform) to complete a series of self-contained applications: hello world, multimedia player, form controls and persistence, graphics & background task, camera snapshot, location sensing, accelerometer, network access and markup and an address book application. The results show large variance across the applications.

The iPhone expert was the fastest in completing the accelerometer and the address book application. The Android expert completed the location sensing and todo list (form controls and persistence) applications ahead of his peers, while the Symbian expert completed the multimedia player task faster than the others. Finally, the Java expert was faster than his peers in a number of tasks, being the first to complete the hello world and setup task, the graphics & background task, the camera snapshot and the network access & markup tasks.

The Java novice did not complete any of the tasks ahead of his peers, while the iPhone novice completed just one task, namely address book & telephony ahead of

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

the other novices. The Symbian novice took less time than the other novices in completing hello world and setup, the multimedia player and the network access and markup tasks. Finally, the Android novice completed the largest number of tasks faster than his peers, being the first to develop the form controls and persistance, the graphics and background, the camera snapshot, the location sensing and the accelerometer tasks.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

We conclude that for developing typical use cases, Android is consistently the fastest platform to develop on, with few exceptions (the multimedia player and network access use cases) across both novice and expert developers. Surprisingly, iPhone overall takes most time for application developers, even more so than Symbian. This is probably due to our selection of application tasks; we tested for common use cases and not complex tasks where code size increases with complexity, and debugging becomes a significant percentage of the development cycle (contrary to the average 10 percent of the development cycle in our tests).

It's also worth pointing out that Java ME had the most pronounced differences between the novice and the expert, with the novice taking three times longer, or 43 more hours, to complete the whole set. The average difference between novice and expert on the other three platforms was 16 hours.

Using the above data, we can say that when developing common applications, each hour of work for a given Android developer, irrespective of level of experience, equals 1 hour and 10 minutes for a Symbian developer, 1 hour and 20 minutes for a Java ME developer and approximately 1 hour and 30 minutes for an iPhone developer.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Appendix 3. Developer contests and standards groups

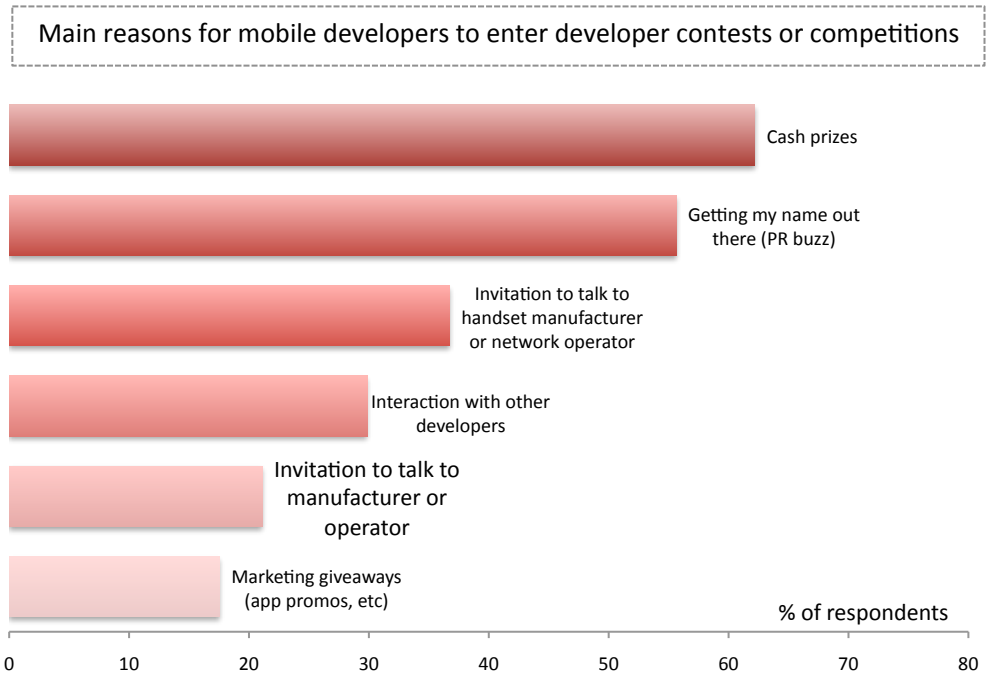
Developer competitions and contests

2010 has been the year of the developer, witnessing a mass introduction of developer events, contests and competitions – not only from platform vendors and handset manufacturers, but also hardware manufacturers (e.g., Qualcomm), network equipment providers (e.g., Alcatel Lucent, Ericsson) and network operators (e.g., Telefonica, Vodafone, Verizon).

So what does it take to get developers to participate? Why do developers attend competitions and contests and what makes them successful?

Our survey found that cash prizes and getting their name out there are the main motivation for developers' participation in competitions and contests – much in line with Maslow's theory on the hierarchy of needs.

Yet views differ significantly by platform. Symbian, Android and mobile web developers take the high road and prefer fame over money. Windows Phone, Flash, BlackBerry, iPhone and especially Java developers prefer a nice cash prize.



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Potential for business development is the next biggest motivator, with more than 35 percent of developers very keen to leverage events to build stronger relationships with handset manufacturers and network operators. There is an opportunity here for mobile operators to not only create competitions to generate innovative services for their customer base, but also attract the attention of some of the best developers in the market, for a potentially small investment.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

Standards groups: are they important to developers?

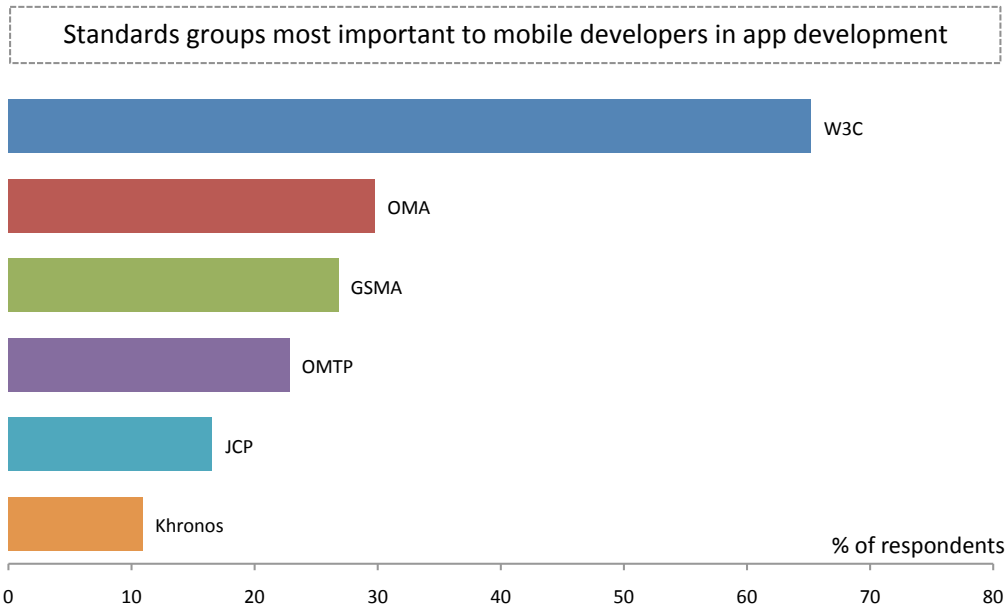
Despite their multitude, mobile industry standards, consortia and joint initiatives seem to have captured very little developer mindshare.

To some mobile industry insiders, this will come as no surprise; standards bodies like the OMA and GSMA have not invested in developer outreach until recently. Furthermore, their recommendations affect network technology, which is far removed from the mobile developer experience. Surprisingly though, OMTP (with their BONDI initiative) has generated significant attention within the mobile industry, a sentiment which seems to not be shared by the average mobile developer working outside the hype-circle of the industry.

Overall, the only standards consortium that is of significance to mobile developers is the W3C, which is seen as important for mobile development by 60 percent of all respondents (and over 90 percent of mobile web developers). Ironically, W3C maintains the only non-mobile-focused group of standards applicable to the mobile industry. This finding points to how standardisation efforts within the mobile industry have traditionally sidelined the role of third party developers, and are now waking up to this reality of developer indifference.

> "Most [standards groups] do not do anything other than get in the way of progress."
>
> Shawn Fitzgerald,
> Java ME Developer

Second runners were the OMA, OMTP and GSMA which are seen as important to only 20-30 percent of developers. Java developers were the only group of respondents who favoured a different standards group, namely the Java Community Process (JCP).



Source: Mobile Developer Economics 2010 and Beyond. Produced by VisionMobile. Sponsored by Telefonica Developer Communities. June 2010. Licensed under Creative Commons Attribution 3.0 License. Any use or remix of this work must retain this notice.

Some developers are even disillusioned with standards groups; one said, "Most [standards groups] do not do anything other than get in the way of progress," while another stated, "[Standards] are coffee+cookie parties". Clearly standards initiatives in the mobile industry have a long way to go before convincing developers of their value.

© VisionMobile 2010. Some rights reserved. Sponsored by Telefonica Developer Communities

knowledge. passion. innovation.