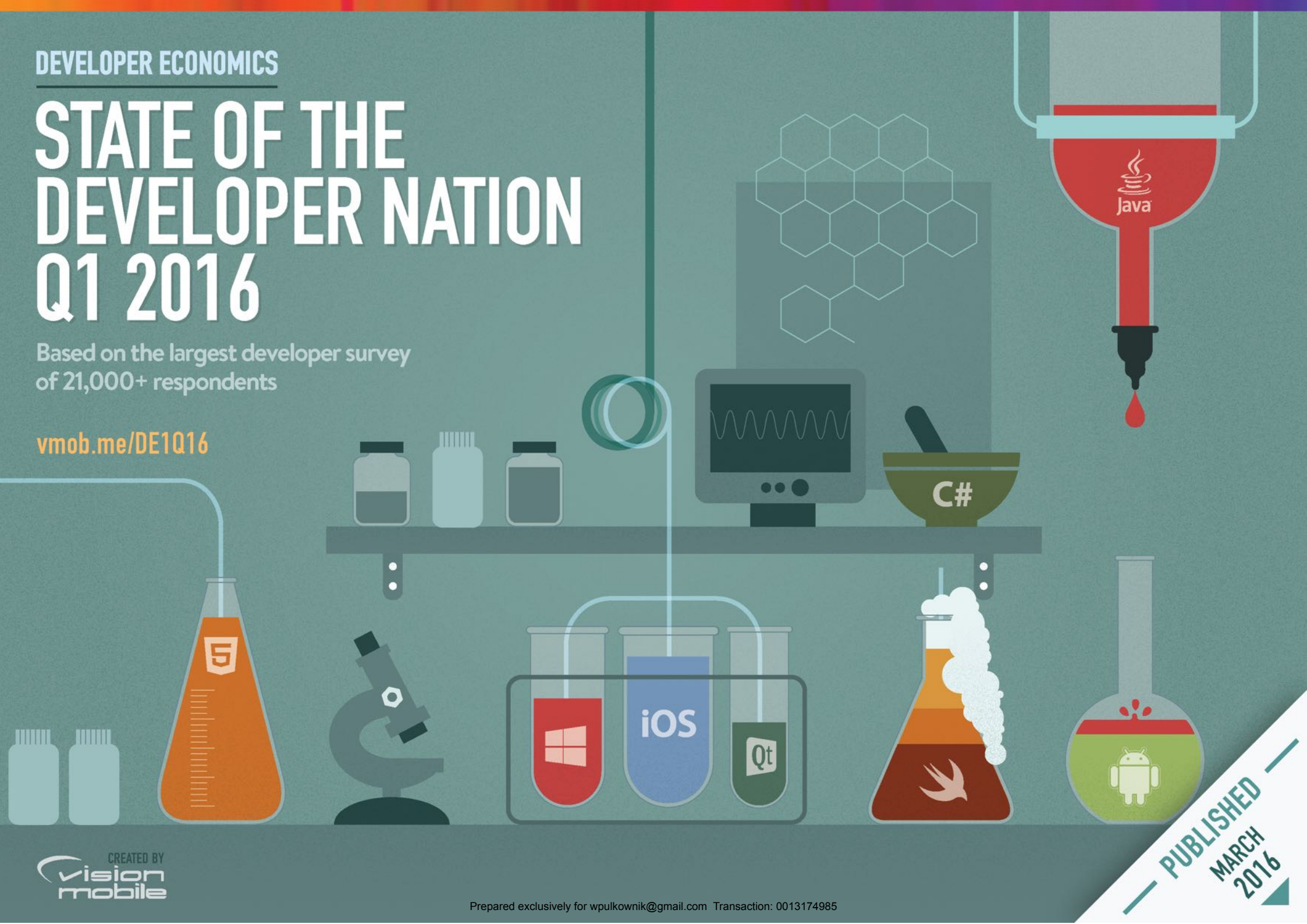


DEVELOPER ECONOMICS

STATE OF THE DEVELOPER NATION Q1 2016

Based on the largest developer survey
of 21,000+ respondents

vmob.me/DE1Q16



CREATED BY
vision
mobile

Prepared exclusively for wpulkownik@gmail.com Transaction: 0013174985

PUBLISHED
MARCH
2016

About VisionMobile™

VisionMobile™ is the leading analyst company in the app economy and the developer ecosystem. Our surveys and app analytics track the changing landscape of mobile, IoT, desktop, and cloud developers.

Developer Economics is the most global app developer research & engagement program reaching just under 22,000 developers in 150 countries. Our research program tracks developer experiences across platforms, revenues, apps, tools, APIs, segments and regions.

Our mantra: distilling market noise into market sense.

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

www.visionmobile.com/blog

Follow us on twitter: [@visionmobile](https://twitter.com/visionmobile)

Terms of re-use

1. License Grant.

Subject to the terms and conditions of this License, VisionMobile™ hereby grants you a worldwide, royalty-free, non-exclusive license to reproduce the Report or to incorporate parts of the Report (so long as this is no more than five pages) into one or more documents or publications.

2. Restrictions.

The license granted above is subject to and limited by the following restrictions. You must not distribute the Report on any website or publicly accessible Internet website (such as Dropbox or Slideshare) and you may distribute the Report only under the terms of this License. You may not sublicense the Report. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Report you distribute. If you incorporate parts of the Report (so long as this is no more than five pages) into an adaptation or collection, you must keep intact all copyright, trademark and confidentiality notices for the Report and provide attribution to VisionMobile™ in all distributions, reproductions, adaptations or incorporations which the Report is used (attribution requirement). You must not modify or alter the Report in any way, including providing translations of the Report.

3. Representations, Warranties and 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.

4. Limitation on Liability.

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.

5. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by you of the terms of this License.

Contents

Key Insights

1. The platform landscape
2. How developers build experience and where it takes them
3. Building bridges across developer communities
4. Language trends over the last 12 months
5. Developer revenue by targeted platform
6. IoT Business models

Methodology

Also by VisionMobile

Find out more visionmobile.com/reports



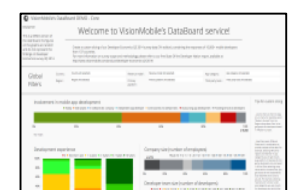
[Cloud and Desktop Developer Landscape](#)



[Best Practices for IoT Developer Programs](#)



[Developer Programs Benchmarking 2015](#)



[Dashboard](#)

Want access to all Premium reports?
[Get an annual subscription!](#)

ABOUT THE AUTHORS

Bill Ray
Senior Analyst



Bill wrote his first mobile app in 1988, and has been failing to make money out of them ever since. He architected set-top boxes at Swisscom and Cable & Wireless, and was Head of Enabling Technology (responsible for on-device software) at UK mobile network O2. He then spent eight years as a journalist at tech publication The Register, before joining VisionMobile as a senior analyst.

You can reach Bill at:
bill@visionmobile.com
@bill4000

Stijn Schuermans
Senior Business Analyst



Stijn is the lead Internet of Things researcher in the VisionMobile team since 2012. He has authored over 20 reports and research notes on mobile and the Internet of Things. He focuses on understanding how technology becomes value-creating innovation, how business models affect market dynamics, and the consequences of this for corporate strategy. Stijn holds an engineering master degree and an MBA. He has over 10 years' experience as an engineer, product manager, strategist and business analyst.

You can reach Stijn at:
stijn@visionmobile.com
@stijnschuermans

George Anadiotis
Senior Research Analyst



George has been active in IT since 1992, having worn many hats and juggled many balls. As a ninja programmer, a lead architect, a team manager, a trusted consultant, an entrepreneur and an analyst, he has provided services to the likes of KLM, SAP and Vodafone, built and managed projects and teams of all sizes and shapes and got involved in award winning research along the way. As a Senior Research Analyst with VisionMobile, George focuses on understanding and analyzing the developer economy in its many facets with the goal of helping developers, vendors and organizations understand its dynamics and navigate the landscape. George has previous analyst experience with Gigaom and his own brand, Linked Data Orchestration.

You can reach George at:
georgea@visionmobile.com
@linked_do

TABLE OF CONTENTS

Key Insights	6	IoT is not mainstream yet - but it is getting there	22
10 Thoughts on the Developer Nation	6	Mobile, Desktop, & Cloud - three platforms bridging into one ..	23
1. The platform landscape	7	4. Language trends over the last 12 months	26
Windows is still the developers' native choice.....	7	Languages for cloud 9, and 10, and 11	28
The race to the pocket is down to two	9	5. Developer revenue by targeted platform	30
Head in the Clouds, but Clouds in the office	10	The big bucks are in browsers, but not the mobile browser.....	30
2. How developers build experience, and where it takes them.....	13	A two-horse race to making money from mobile	33
It takes experience to explore the clouds	15	There's money in them there clouds.....	33
Segmenting the developer community	15	6. IoT Business models	34
3. Building bridges across developer communities.....	18	IoT is tough business.....	34
One foot in the desktop.....	18	The exploration phase	34
Bridge Building	20	The most effective IoT revenue models	36
Mobile, a playground for greenhorns and old hands alike	21	Methodology.....	39
In the future, everyone will be a cloud developer for 15 minutes - or more	21		

About Developer Economics

Welcome to the State of the Developer Nation Q1 2016 report, based on the 10th Developer Economics global survey wave. Produced by VisionMobile, Developer Economics is the leading research program on mobile, desktop, IoT and cloud developers, tracking developer experiences across platforms, revenues, apps, languages, tools, APIs, segments and regions. The 10th Developer Economics global survey wave run in October-November 2015 and reached just under 22,000 developers in over 150 countries.

This research report delves into the key developer trends for 2016, and discusses demographics, platforms, languages, revenues, mobile commerce and more.

The report focuses on six major themes – each with its own visualization, showing how the data lends insight into the developer community.

1. The platform landscape: which platforms are attracting developer attention, and how developers are dividing their time.
2. Who are developers? The 6 segments or types of developers, from Hobbyists to Enterprise IT, and how they have evolved over the last six months.
3. No boundaries: Developers don't stay in boxes, and so mobile, desktop, cloud and IoT developers are often applying their skills across several domains, staying in one box professionally whilst dabbling elsewhere.
4. Language choices and trends: from Java to Swift, how developers' preferred languages have changed over the last 12 months.
5. Developer revenues: Which developers are making the most money, and which business model is paying off best for them.

6. IoT Revenue Models. IoT isn't making money for many developers, but some business models are starting to pay off.

We hope you'll enjoy this report and find the insights useful!

If you have any questions or comments or are looking for additional data, you can get in touch with Matos Kapetanakis, Senior Manager for the Developer Economics Program at matos@visionmobile.com. You can download this free report at www.DeveloperEconomics.com/go

Bill, Mark, Christina, Matos, Alex, Stijn, Andreas, George, Emilia, Dimitris, Vanessa, Sarah, Chris, Michael, Nick, and Dinos at VisionMobile.

Thank you

We'd like to thank everyone who helped us reach 21,000+ respondents for our survey, and create this report:

Our Research Partners – Google, Intel and Microsoft.

Our Leading Community and Media Partners, who are too many to number here – you know who you are!

Also, a special thanks to Yun Joonghyeon, Caven Cade Mitchell, Yonathan Randyanto, Serena Davis, Max Chen, Okeowo Aderemi, Sherwin Binario, Adeponle Segun Patrick, Cindy Simeti, Fabrizio Bernabei, Michael Tedder, Toka Ahmed, Febria Roosita Dwi and C. O. Uwadia for their support.

COMPANIES WHO CONTRIBUTED TO THIS RESEARCH

MARKETING SPONSOR



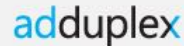
RESEARCH SPONSORS



LEAD COMMUNITY PARTNERS



MEDIA PARTNERS



KEY INSIGHTS

The 10th Developer Economics survey again ups the bar in reaching out to the global developer community. Nearly 22,000 developers, from more than 150 countries, have responded to help build a comprehensive view of the software industry. This year we continued our examination of Cloud and Desktop development, and asked more questions about how developers are approaching the Internet of Things.

This State of the Developer Nation report highlights some of the most interesting findings from the survey: the trends in platform and language that are changing the industry, and how developers are meeting those new challenges. In this report we've summarized the key findings, and present some thoughts on how these trends will change the software landscape over the next few years.

10 Thoughts on the Developer Nation

- Desktop development is still the most-popular place to be, attracting 50% of professional developers, and making money for them too.
- The way to the desktop is through the browser, targeted by 44% of desktop professionals.
- Developers are getting less experienced every year. As the population grows so the average experience level is falling.
- Developers aren't sticking to their sectors. Only 34% of professionals are focused on one sector, while the majority are working in at least two, and 7% are professionally involved across all

four - Mobile, Desktop, Cloud, and IoT.

- Objective C has had its day. Six months ago 14% of mobile developers listed Objective C as their primary language, now that number has dropped to 5%. The once-mandated language of iOS developers is under assault from cross-platforms tools, and Apple's new wunderkind, Swift.
- JavaScript and its derivatives continue climbing across sectors. They are now the primary language for 12% of mobile developers, and 13% of those working in the cloud.
- C# is static in mobile, at around 40% mindshare, but Microsoft's language of choice is gaining ground in the Cloud (up 4% in the last six months, reaching 41% mindshare).
- More than half of developers in IoT and Mobile are living below the poverty line, pegged at \$500 a month in revenue.
- IoT developers are broadening their horizons in the search for financial stability. Those who make money are combining revenue streams to make their development pay.
- Most clouds are still kept at home, with 42% of professional cloud developers hosting their own cloud infrastructure.

The State of the Developer Nation is only scratching the surface of the data collected by VisionMobile, but within this report we can see a growing developer community expanding its horizons to encompass multiple computing sectors, and keen to apply the same skills across them all.

1 THE PLATFORM LANDSCAPE

Half of the professional developer community is targeting the desktop, including web-based apps aimed at desktop users, demonstrating that the oldest personal computing paradigm remains the most important.

Other areas, i.e. Mobile, IoT, & Cloud, might get more media attention and newspaper headlines, but the humble desktop is where computer users still spend most of their working lives, so it shouldn't be surprising that so many professional developers are involved in creating desktop applications. The desktop is generally a laptop these days, but even that moniker is becoming misplaced as few laptops are used on laps, and Microsoft has gone a long way in erasing the division between tablet and desktop computing with its Surface range of devices. But desktop computing remains a distinct field, and one with which our developer community is heavily involved.

Those developers aren't necessarily targeting traditional desktop platforms, as 44% of them are creating applications which will run within the web browser, and thus work across all the popular operating systems. Applications running in a browser gain the cross-platform capabilities that web technologies provide, as well as enabling developers to make use of generic skills and high-level scripting languages. Web apps happily combine content using different technologies, can easily be connected to a cloud service, and even provide a foothold in the mobile market as portable browsers

often rival their desktop equivalents in functionality and compatibility.

The price of web development is in performance and access to local resources such as storage or input devices. The web community has gone a long way in addressing both those issues; providing HTML5 Web Storage as a first step towards persistence, WebGL for faster graphics, and WebRTC which provides access to cameras and microphones. With these tools a web developer can create high-performance applications which will work across any desktop platform, and most mobile platforms too.

Windows is still the developers' native choice

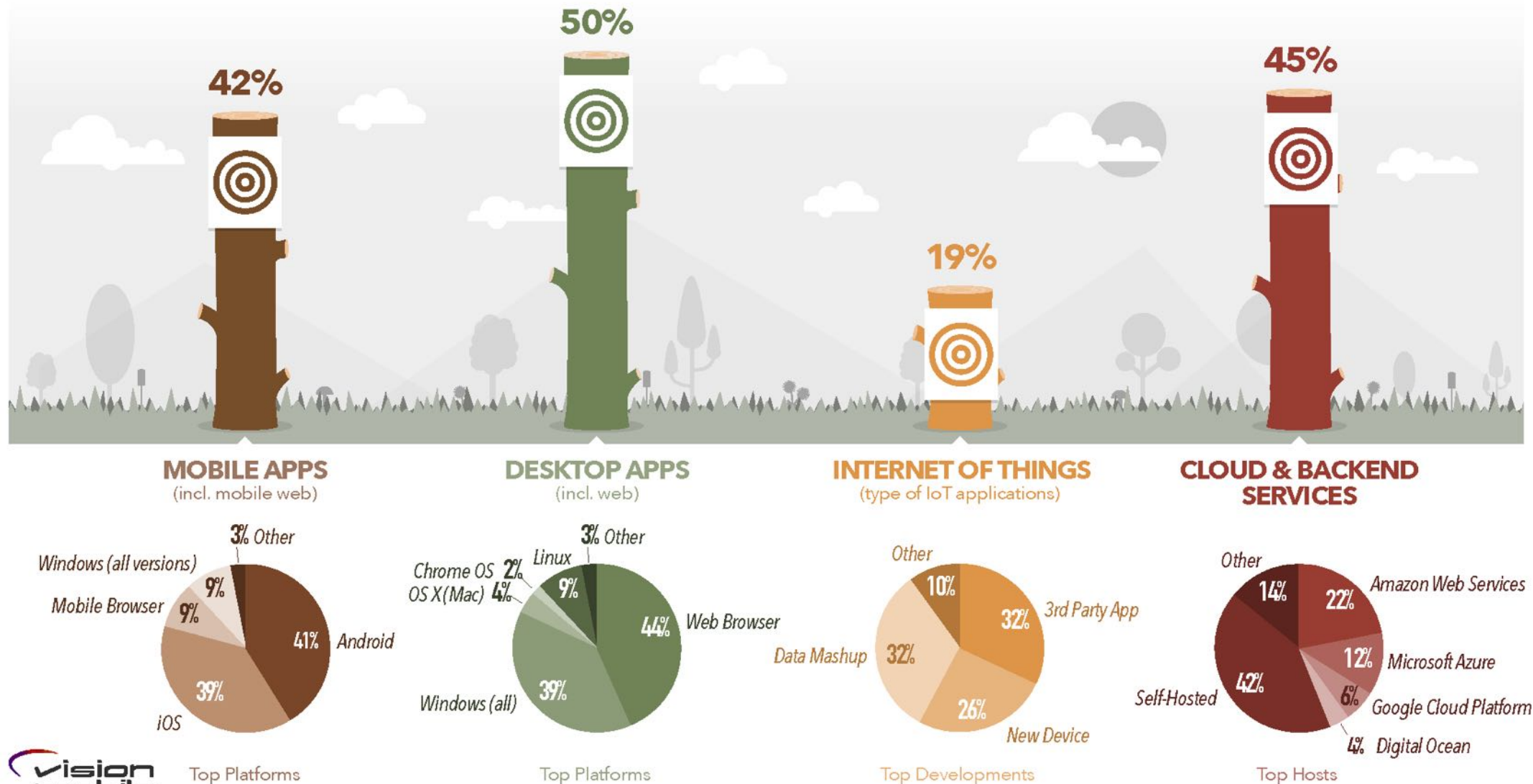
For developers not seduced by web technologies, those who need access to features only available from native apps, or needing the performance that only native applications can deliver, Windows is still the platform of choice. 39% of professionals who target the desktop are writing Windows code to do it, though they are split between traditional and modern versions of the Microsoft platform.

DEVELOPER TARGETS, FROM MOBILE TO CLOUD

Where professional developers are focusing their efforts, by area and platform (n=11,257)

BASED ON
A SURVEY OF
21,000+
DEVELOPERS

Percentage of professional developers targeting each sector



OS X is still very much a minority option, despite the success of Apple's range of laptop computers. Only 4% of professional desktop developers are targeting the OS, though Apple apparently holds 7.68% of the desktop OS market (as of January 2016, according to Net Marketshare). Many developers addressing an audience of OS X users are using web technologies, to ensure compatibility with the dominant player (Windows) while providing support for other minority platforms (Linux et al).

39% of professionals who target the desktop are writing Windows code to do it.

Linux is quite well-represented, being the primary platform for 9% of our professional desktop developers, which is interesting as this group is not made up of voluntary contributors to large projects, but paid developers creating commercial applications. Linux is very popular in servers, but has a negligible market share in desktop computing (1.71% according to Net Marketshare), so the interest from professional developers is harder to understand. There is a significant market in analysis tools, many of which run on Linux, so some of these developers will be involved in creating those tools, but we will need to delve deeper into our data to establish why Linux is so interesting to professional developers.

The other desktop platform with a significant showing is Google's Chrome OS, which is being targeted by 2% of desktop professionals. This number is significant, especially as these developers identified Chrome OS as their "Primary" platform. Google's thin platform has gained some ground in schools, who can utilise old hardware (laptops) by installing a distribution of Chrome OS, so some development will be in that area, but there will also be a good deal of experimentation as developers look to exploit a platform with the weight of Google behind it.

It is also worth noting that the number hasn't increased since our last survey, so there is a core of professional developers targeting Chrome OS, but that core has not grown over the last six months.

The race to the pocket is down to two

Mobile developers are much less interested in targeting the browser, with only 9% naming it as their primary platform. Mobile browsers are very capable these days, but integration with device hardware is more important on mobile devices which haven't settled into a standard form factor in the way that desktop computing has.

Extensions to the web standards allow browser apps access to the microphone, camera, and location hardware such as GPS, but mobile devices have fingerprint readers, electronic compasses, barometers, gyroscopes, and all sorts of specialist hardware enabling the device to respond to a wide variety of environmental conditions. The web standards do not provide access to this hardware, limiting the functionality of browser-based applications.

In addition to the hardware, there is an expectation amongst mobile users that mobile apps will work smoothly together. Mobile applications can trigger each other, and pass information between themselves, which is more complex to arrange with browser-based apps.

Some phone platforms (notably Firefox OS) have sought to extend web-standard development to encompass this functionality, an idea which goes back to Palm's WebOS and beyond (taking in the ill-fated BONTI initiative from the OMTF), but none have managed to gain any significant support, and the market leaders – Google and Apple – have little incentive to support an extended browser feature set.

Apple and Google are clearly the market leaders, both in devices shipped and developer mindshare. 41% of professional mobile

developers are targeting Android, while 39% list iOS as their primary target platform. It is, however, unintuitive to see Android and iOS being so level when the market share in devices is clearly dominated by Android, a reflection of the greater revenue available from the iOS ecosystem.

IDC give Android 83% of the global handsets shipped in the second quarter of 2015, while iOS was only running on 14%. Other market-tracking companies provide similar statistics. It is clear that there are many more devices running Android than iOS, so to understand the issue we have to look at where, and how, mobile developers are realising their revenue.

Apple and Google are clearly the market leaders, both in devices shipped and developer mindshare. 41% of professional mobile developers are targeting Android, while 39% list iOS as their primary target platform.

Much of the growth of Android is in developing markets, and in very low-end handsets. At the time of writing, a functional Android handset can be bought for less than \$50 in the UK (£35), without subsidy or operator lock-in. The TTsims M5 SMART is a perfectly usable smartphone, which demonstrates how the use of commodity hardware, and some compromises on screen size and resolution, can reduce hardware prices. Companies in India are also experimenting with alternative sources of subsidy, such as government grants for distributing educational software, in order to get devices into the hands of the currently-disconnected population.

An Android user who pays £35 for a phone is unlikely to be interested in an app costing £4.99 (the current price of the bestselling game Minecraft). They are also of limited interest to advertisers, who fund most of the Android applications in the Play store: the owner of a £35 phone, perhaps bought with the help of a

subsidy, has little disposable income, so advertising is less valuable.

Apple's iPhone, in contrast, starts at £330 (\$470) for the most-basic model (5s, 16MB, at time of writing). Clearly the owner of such a device won't balk at paying a few dollars for an app, or an in-app purchase, and also has the kind of disposable income that makes them very attractive to advertisers.

An Android user who pays £35 for a phone is unlikely to be interested in an app costing £4.99 (the current price of the bestselling game Minecraft).

This demographic difference goes a long way to explain why the professional mobile developer population is so evenly split between the leading platforms, but it can also tell us how that split is likely to develop over time. The market share owned by Android is going to become increasingly important, as developing economies become more attractive to developers.

Within many markets Android will become the default platform, and Apple risks iOS being relegated to the role of fringe player. Apple will play up its role as a premium brand, but challenging the dominance of Android will be hard once it has been established. This is a long-term view, but we expect to see the proportion of developers targeting Android to increase steadily over time.

Head in the Clouds, but Clouds in the office

In commercial cloud computing the dominance of Amazon is impossible to ignore. Recent high-profile endorsements – such as Netflix announcing that the company now has no in-house servers at all – have reinforced the view that Amazon is providing the infrastructure behind the information age, but our research again shows self-hosting to be the most-popular option for those seeking to exploit the advantages of cloud computing.

42% of professional Cloud developers say they are hosting their own cloud service, running servers literally within their own building or renting infrastructure (which may be virtualized as an IaaS offering) as an effective way to provide co-location. These users prefer to manage their own platform, either because they want to ensure compatibility with legacy systems, or because they want to maintain control of the underlying software.

22% of professional cloud developers are using Amazon, almost twice the proportion of their nearest competitor (Google) who vies with Microsoft Azure for second place.

IaaS is the simplest way to take advantage of cloud computing, and the advantages of simply moving an existing installation onto cloud-

provided infrastructure are easily explained and measured. Management can understand the cost saving in co-located hardware, and the disruption to existing systems is kept to a minimum.

The problem is that IaaS doesn't realise all of the advantages that cloud computing can offer, and once a company has invested in creating self-hosted clouds then moving to a commercial platform (PaaS) is a second round of investment which may be harder to justify.

Not that commercial cloud platforms are lacking in customers. 22% of professional cloud developers are using Amazon, almost twice the proportion of their nearest competitor (Google) who vies with Microsoft Azure for second place.

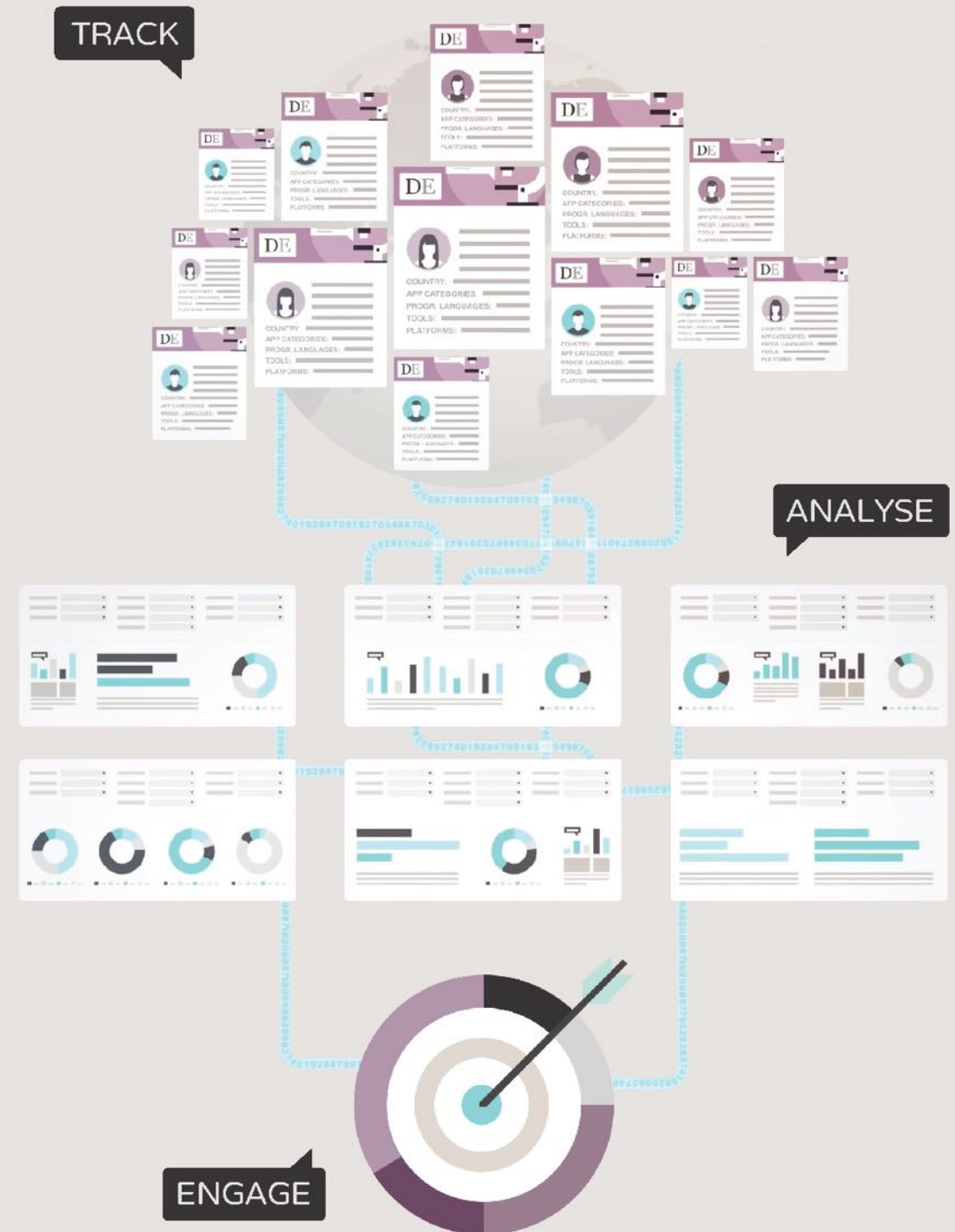
VISIONMOBILE'S DATABOARD

Access thousands of data points

VisionMobile's Databoard is an annual subscription service that allows you to slice and dice our Developer Economics data across 10,000+ developers, via dashboards of interactive charts.

Track developer experiences across platforms, regions, segments and more. Understand how developers make choices and what is the best way to engage them.

FIND OUT MORE
vmob.me/DB



2 HOW DEVELOPERS BUILD EXPERIENCE, AND WHERE IT TAKES THEM

Across all four sectors (IoT, mobile, cloud, desktop), we can see that the proportion of developers with more than six years of experience has fallen significantly, over the six months since our last Developer Economics survey.

This is offset by an increase in developers who've been working in software for less than a year, and a bigger jump in those with 1-2 years of coding behind them. Overall we can see an industry that is growing fast, with new developers entering the business at an accelerating rate, which is pushing the balance in favour of those with less experience.

If we take just Mobile developers, which are a typical example, we can see that the percentage with less than a year's experience has increased from 14 to 17% over the last six months, while the percentage of those who have been creating software for more than six years has dropped from 48 to 40%. Clearly a significant proportion of mobile developers have migrated from less than a year to 1-2 years (the latter rising from 17 to 22%) but even more have joined the industry for the first time, pushing the other numbers down.

This is a good sign for the industry, which needs a constant supply of fresh talent and is still expanding. Companies in just about every industry are realising the importance of software, and are recruiting developers to help them reach more customers, and improve their business processes, resulting in the demographic shift that we see.

This shift will also have an impact on the choice of language and toolsets used by developers, though it will continue a long-evolving

trend towards abstracted logic and visual development. Less experienced developers will likely opt to use high-level scripting languages, such as JavaScript and Python, rather than more-powerful languages such as C++ or even Java. This not only makes development quicker, but also means that cheaper (inexperienced) developers can be used for the duration of the project.

We can see an industry that is growing fast, with new developers entering the business at an accelerating rate, which is pushing the balance in favour of those with less experience.

High-level languages obviously have limitations when it comes to performance, and in many cases have limited utility as developers can only access resources made available through APIs for that language, but as the developer community continues to expand, and demand outstrips supply, the use of such languages will continue to rise.

An increasingly inexperienced developer community will also drive demand for in-job training and the kind of peer-support mechanisms offered by Stack Overflow, Reddit, and their ilk. These resources are already highly valued by developers, and we expect to see their importance increasing, as more developers join the workforce.

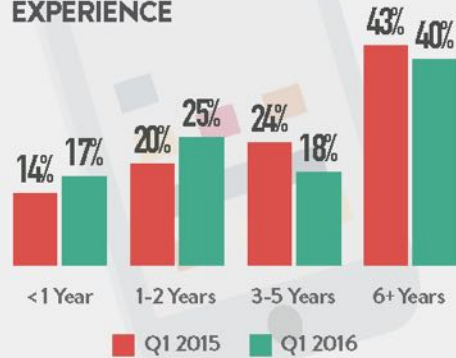
HOW DEVELOPERS BUILD EXPERIENCE, AND WHERE IT TAKES THEM

Experience and segmentation of Desktop, Cloud, Mobile & IoT developers (n=21,690)

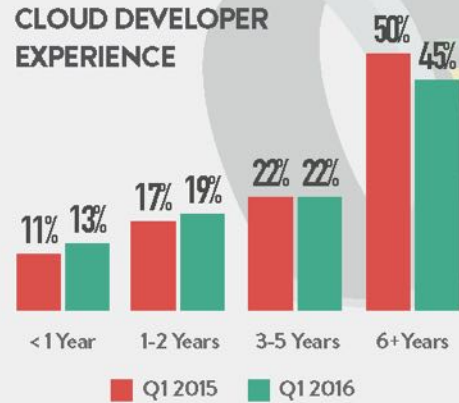
BASED ON
A SURVEY OF
21,000+
DEVELOPERS

The developer community gets less experienced every year

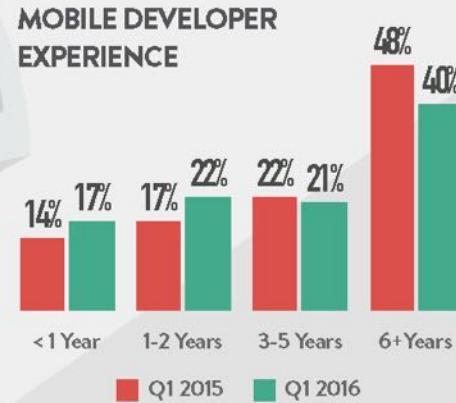
DESKTOP DEVELOPER EXPERIENCE



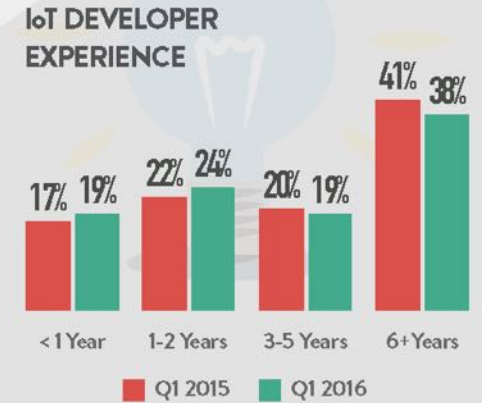
CLOUD DEVELOPER EXPERIENCE



MOBILE DEVELOPER EXPERIENCE



IoT DEVELOPER EXPERIENCE



Developer segments across desktop, cloud, and mobile



HOBBYISTS

Moonlighters building their own apps to learn and have fun



EXPLORERS

Independent developers gaining experience as a side project to seize on future opportunities



HUNTERS

Experienced developers building an app business and focused on the money



GUNS FOR HIRE

Seasoned pros developing apps on commission



PRODUCT EXTENDERS

Companies using apps to promote or extend a non-mobile product or brand



DIGITAL PRODUCT PUBLISHERS

Developer teams using mobile apps to monetise digital content inventory



GOLD SEEKERS

Mobile startups aiming to hit VC gold



ENTERPRISE IT

CIOs and IT managers using apps to increase organisational efficiency and reduce costs



It takes experience to explore the clouds

The most experienced group of developers is that targeting the Cloud. 45% of cloud developers have more than six years of software experience, and while that number has dropped over the last year (from 50%) it is still significantly higher than that of developers working in any other field.

Cloud also has the smallest percentage of developers who've been working with software for less than a year, which would seem to indicate that cloud computing is a step up the career ladder for the aspiring developer, though there are other factors which should be considered before jumping to such a conclusion.

45% of cloud developers have more than six years of software experience, significantly higher than developers working in any other field.

Last month VisionMobile published a study looking at developers targeting Cloud & Desktop Platforms¹, which found a similar disparity but was able to drill down more into the education and experience of Cloud developers. That report concluded that Cloud developers were comprised of two demographics: software developers with considerable experience who have moved into cloud computing in search of new challenges, and recent graduates who studied cloud computing at university and are keen to apply their skills.

The latter group is reflected in these new figures, which see 13% of cloud developers having less than a year's experience – a modest increase from six months ago – but notably the proportion of cloud developers with 3-5 years of experience has remained static: despite dropping in all other development areas.

¹ <http://www.visionmobile.com/product/cloud-and-desktop-developer-landscape/>

Considering the drop in the proportion of developers with more than six years' software experience it is clear that some developers with a few years' experience are moving into cloud computing, but the majority of developers moving into the cloud come from the two groups at either end of the experience spectrum.

Segmenting the developer community

In desktop development we can see that the percentage of developers experimenting with different platforms has dropped considerably over the last six months. The declining excitement over Windows 10 has contributed to this, as the platform becomes better known, and more-widely adopted, so those exploring as a hobby become professional programmers within the field. We can see that reflected in the rise of Enterprise IT, which almost matches the decline in hobbyist desktop developers.

Microsoft's decision to aggressively push Windows 10 onto consumers and enterprises has created a demand for professional development to ensure a smooth migration. Windows 10 is, largely, backwards compatible, but individual applications need to be tested and optimised, and in many cases can be modified to take advantage of new features unavailable in previous versions of Windows.

The expanded use of Windows 10 means that Explorers (independent developers gaining experience) are still attracted to desktop development, hoping to discover the next must-have utility or application, and we see that effect manifest when we look at the figures for Gold Seekers (startups aiming for VC gold) interested in desktop development. Gold Seekers are still a minority on every platform, though they have always been most prevalent on mobile, where the promise of app-store riches and a handful of high-profile successes fed their dreams. Six months ago that trend was in full evidence, with 5% of mobile developers falling into the Gold Seeker category, compared only 1% of those targeting the desktop. Now we

see that trend reversed, as 3% of desktop developers can be identified as Gold Seekers, compared with only 2% of the mobile community.

These are still small numbers – Gold Seekers make up a small part of the community – but the growth in those targeting the desktop is significant. Windows 10 is being seen by many as an opportunity to sell new services, while the Microsoft Store brings the mobile distribution paradigm to desktop computing. An integrated billing mechanism, in-application billing, and a measure of copy protection, offer developers a much-improved distribution experience. These have been available since the ill-fated Windows RT, and were part of Windows 8, but the wide scale adoption of Windows 10 has brought them to the attention of many more developers.

Despite continuing headlines about high-profile successes, these are not driving the Gold Seekers into mobile, perhaps because the market is becoming increasingly competitive and some cynicism has crept into the Gold Seeker community, and the VC community which funds it.

It is interesting to note that the proportion of Explorers in mobile has risen, showing that there is a community exploring how the latest generation of mobile devices and services can be harnessed to make money.

Mobile is the only area where we see a drop in Enterprise IT, and it is only by a single percentage point, so it should better be considered a stable community. This lack of growth probably reflects the consolidation of platforms and better support from platform vendors. Those vendors; Google, Apple and, to a lesser extent Microsoft, provide a good deal of enterprise functionality on their respective platforms, and few companies are creating internal applications for Blackberry or the other alternative platforms, resulting in a community which is static in size.

In cloud we see a significant rise in those working in Enterprise IT, which reflects the wider adoption of cloud computing across enterprises. We can also see a similar level of growth in the number



of cloud developers working as Guns for Hire, contracting their skills out to companies who don't have (or don't want to have) the in-house experience in cloud development. While some companies do want to avoid building an in-house team who might be redundant at

the end of a project, this growth in Guns for Hire more likely reflects a general shortage of cloud programmers with the experience companies need.

3 BUILDING BRIDGES ACROSS DEVELOPER COMMUNITIES

When we look at the developer communities around each sector (mobile, IoT, cloud, desktop), our survey confirms the day-to-day reality for most: the divisions between sectors are increasingly fading, as more and more developers become active across more than one.

That's not to say that each sector does not possess its own defining characteristics, but bridges are used between them to make life easier for the increasing population of developers working across sectors. So let us see how developers are involved in each sector, and the bridges that help them transcend the divisions.

One foot in the desktop

Desktop is the most mature of development areas. It's been more than 30 years since Microsoft Windows introduced the modern desktop metaphors to the general public and desktop computing went mainstream. The technology continues to evolve, but desktop remains the area that is the most common and well understood for developers. An impressive 84% of professional developers globally are involved in desktop development, and desktop is the development area with the highest proportion of professional involvement. 65% of professional developers are, professionally, involved in the desktop.

But are these developers an endangered species? Since 2014 mobile computing has exceeded desktop in both sales and usage. There are regional variations, but the trend is impossible to ignore. Even

though desktop sales and use remain substantial, most industry observers would say that we are squarely in the era of mobile, if not beyond. If developers are aware of this trend, which they most certainly are, then why are they sticking to desktop development?

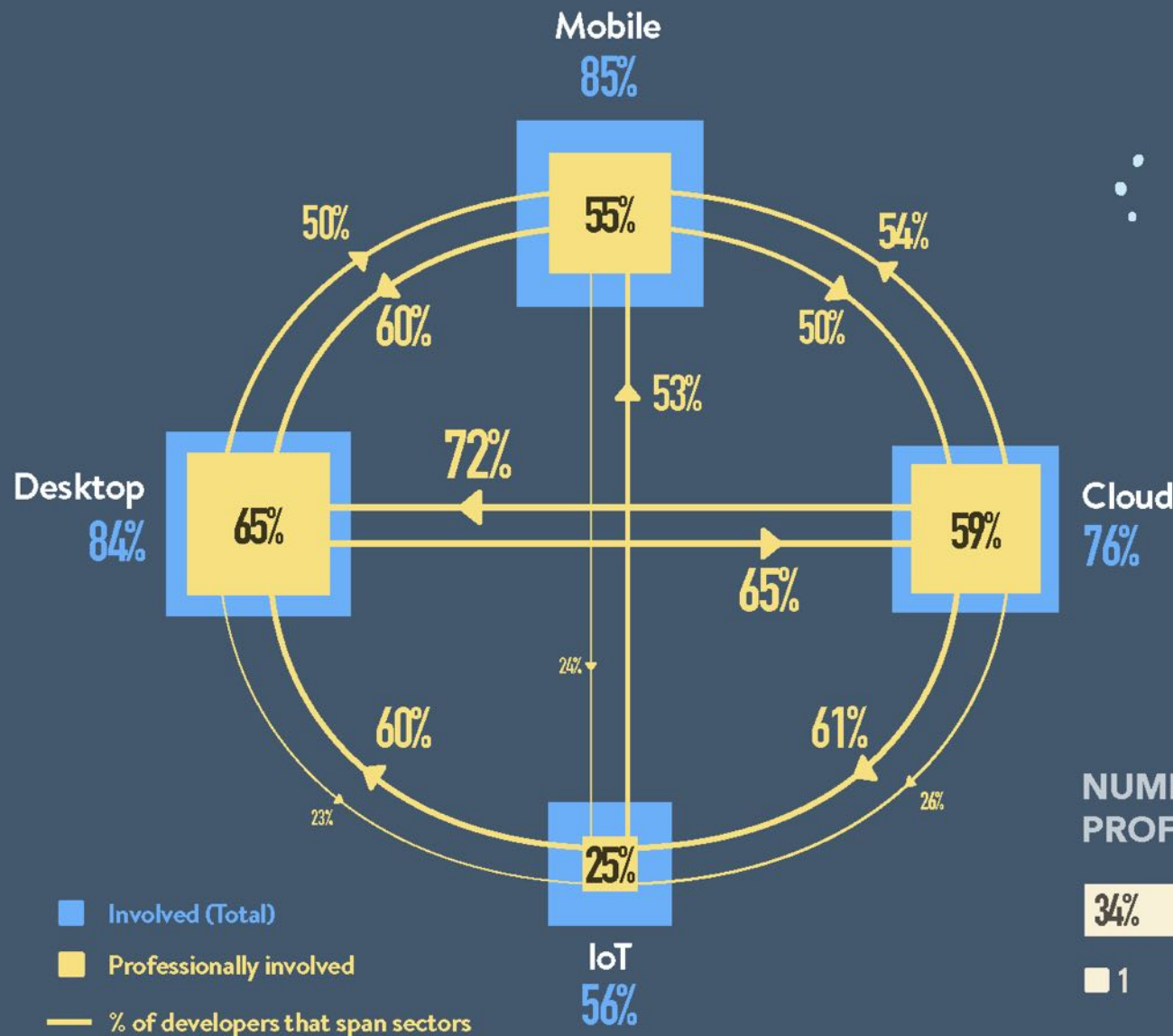
To begin with, there is the legacy factor. Old habits die hard, and the desktop development crowd is an experienced one, second only to cloud developers in this respect. The desktop is ripe with tools, technologies, frameworks, and languages to choose from. Competitive evolution has driven development, from "traditional" technologies such as C/C++, to latter-day ones, such as Java and Maven, and their modern-day incarnations such as server-side JavaScript and Gradle. The desktop-development workforce has invested heavily in learning and refining these technologies, and is understandably reluctant to move away.

That in itself would not amount to much, if there was no money to be made in desktop. That is not the case however: desktop developers are more likely to be above the poverty line (earning more than \$500/month) than those working in mobile or IoT, and a quarter of them are in the top-earner category (earning more than \$25,000/month).

GENERALIST DEVELOPERS ARE CASTING A WIDE NET

Developer involvement per development area(n=21,690)

BASED ON
A SURVEY OF
21,000+
DEVELOPERS



NUMBER OF DEVELOPMENT AREAS PROFESSIONALS ARE INVOLVED IN



vision
mobile

Still, professional desktop developers are not putting all their eggs in one basket: 65% are also professionally involved in cloud development, 50% are also professional mobile developers, and 23% are professionally involved in IoT. If we factor in the additional amateur involvement in those areas then we arrive at the conclusion that desktop developers have only one foot in the desktop, as over 80% are also involved in mobile and cloud, and 51% are involved in IoT.

An impressive 84% of professional developers globally are involved in desktop development, and desktop is the development area with the highest proportion of professional involvement.

Drilling into professional desktop developers moonlighting in other development areas, we see that 34% of them are involved as amateurs in mobile, 33% in IoT, and 15% in cloud.

In the case of mobile this can be explained by the development bridges that exist between the two areas, making the transition an increasingly smooth one. In the case of cloud, we can attribute the figure to the remaining minority of professional desktop developers who are not yet proficient with cloud computing, and are exploring in order to gain exposure.

What is impressive, however, is the high percentage of desktop professionals who are experimenting with IoT development. This crowd of IoT enthusiasts may very well be toying with IoT for their own enjoyment; as the IoT gadgetry seems to have a certain playful appeal to big boys and girls alike, but at the same time they are acquiring skills that may come in handy, professionally, at a later date.

Conversely, amateur desktop developers are spending their working (professional) life mainly in mobile (59%), and to a lesser extent in cloud (40%), and IoT (32%). While the percentage of desktop

amateurs professionally involved in mobile reflects the overall figure for mobile amongst professionals (55%), cloud is under-represented as an employment area (59% overall), and IoT is over-represented (25% overall).

So cloud professionals do not seem to find the desktop particularly attractive as an experimentation area. This may be understandable if we consider the fact, as pointed out in the recent VisionMobile Cloud & Desktop Developer Landscape 2015 report², that a significant proportion of the cloud community is made up of experienced developers who have moved into cloud computing as the field has developed over the last ten years. Those developers are very likely to have a background in desktop, as desktop is the more mature development area, rendering experimentation with the desktop of little interest.

IoT professionals, on the other hand, seem willing to try their hand in the desktop. As we have seen in section one, IoT is the sector in which developers have the least experience in programming overall. Therefore, IoT developers are less likely to have experience in the desktop as well, and more likely to experiment in order to make up for what they lack in desktop exposure.

Bridge Building

The most obvious example of bridges between the desktop and other development areas are programming languages. Sectors would, once, have had their own programming languages. Even Java, the poster-child for write-once-run-anywhere, came in variants for each sector (Enterprise, Standard, Micro, and JavaCard), but languages have become increasingly flexible in their application.

These days the accumulated experience of desktop developers, in (say) Java or .NET, can enable a smooth transition into other sectors. Java and .NET both offer a wide array of scenarios where they can be

² <http://www.visionmobile.com/product/cloud-and-desktop-developer-landscape/>

used for mobile, cloud, and IoT development. Scripting languages have also become multifunction, with JavaScript applications being compiled down for mobile phones, embedded (IoT) devices such as the Intel Galileo, and cloud hosting platforms, as well as running on traditional desktops.

Another prime example of a development bridge is Android for the desktop. Android applications have already taken their first steps into the desktop world, with the BlueStacks Android emulator (for Windows and Mac) claiming more than 100 million downloads, while desktop-oriented implementations of Android are emerging from China in the form of RemixOS and Phoenix OS, intending to create a fully-featured desktop experience on top of the open-source Android core. This renewed interest in Android for the desktop can help mobile developers get a foothold there too.

For the most part, the desktop is an area in which the technology is more or less stable, so the fact that experimentation with it is limited (for example, only 21% of professional mobile developers are involved in desktop development as amateurs) is no surprise.

Mobile, a playground for greenhorns and old hands alike

The growth of mobile as a platform for applications and content has been exponential, driven by largely Apple and Google's entry into the market in 2007. As mobile development has become more fashionable, and delivered iconic success stories, it has become the entry point for a large part of the less-experienced developer workforce.

As we've seen in section one, the flow of new software developers into mobile development has resulted in a significant drop in the average experience level. So mobile development has become, to some extent at least, a playground in which inexperienced developers can quickly put together applications.

Hobbyists and explorers still form a large constituency (43% in total) in mobile development. That could explain why mobile attracts the bulk of developers, but only a relatively small proportion of them are professionally involved: 85% of all professional developers are involved in the mobile sector, but only 55% of them are professionally involved in mobile.

Apparently the mobile success stories are not only attracting newbies, but more seasoned developers as well. 33 - 34% of developers professionally involved in other sectors are, at the same time, mobile amateurs.

85% of professional developers are involved in the mobile sector, but only 55% of them are professionally involved.

The fact that mobile developers come from different backgrounds is also reflected in the languages they use (see section four). Some prefer scripting languages and frameworks such as JavaScript, others go with Java or .NET languages including C#, or with iOS-specific ones such as Objective C.

So the mobile development field seems to be attracting all sorts of developers, with varying degrees of success: more than half of mobile developers are below the poverty line, and only 18% are in the top-earners (see section five).

In the future, everyone will be a cloud developer for 15 minutes - or more

The cloud is different from mobile and desktop in the sense that it is not a distribution channel for applications but rather constitutes the infrastructure supporting the deployment of services such as storage and user management for applications. Therefore, as the use of cloud

infrastructure is becoming the norm, overlap with other development areas is substantial.

This overlap is particularly pronounced between Desktop and Cloud, with 65% of professional desktop developers professionally involved in cloud development and another 19% being involved in an amateur capacity. As highlighted in our recent VisionMobile Cloud & Desktop Developer Landscape 2015 report³, developers are increasingly using cloud services to build their Desktop applications, as well as adding cloud components to them: based on data from the previous VisionMobile survey, 49% of cloud and desktop professionals are involved in both sectors, and 16% of desktop professionals were cloud amateurs. So we see this trend not only continuing, but in fact growing.

As the use of cloud infrastructure is becoming the norm, overlap with other development areas is substantial.

Of course, having a designed-for-desktop website with a backend hosted on-premise, or on rented physical servers, has been the dominant development paradigm for more than a decade. However, it is the shift from this model to the use of IaaS/PaaS, with the benefits that it entails (elasticity and on-demand utilization), which has signalled the rise of the cloud era.

The trends seen in the Mobile sector appear amongst Cloud developers, as the cloud gains momentum amongst mobile developments. 54% of professional mobile developers are professionally involved in cloud development, with an additional 19% being involved in an amateur capacity. Many mobile applications today rely on backend services in the cloud, either custom ones or leveraging MBaaS platforms such as Google Firebase or Appcelerator Arrow.

³ <http://www.visionmobile.com/product/cloud-and-desktop-developer-landscape/>

IoT developers are also extensively involved in cloud development, whether they are building custom solutions or utilizing cloud IoT platforms offered by the likes of Oracle, Google, and Salesforce. 65% of professional IoT developers are professionally involved in cloud development too, with an additional 20% playing with the cloud as amateurs.

Good backend developers need to have a solid background that goes beyond development fundamentals to include advanced topics such as load balancing, distributed programming, and performance optimisation. As such, they have traditionally been well paid and recruited from the elite, and the cloud has not changed that. Not only are cloud developers the ones with the most experience, but they are the best paid too: only 37% of them fall below the poverty line and 32% are top-earners.

IoT is not mainstream yet - but it is getting there

Of all development areas, IoT is the newest one and has yet to attract the quantity of developers that other development areas do, both professionally and otherwise. Although more than half (56%) of professional developers are dabbling with IoT, only 25% of them are involved in a professional capacity.

The commercialisation of applications for IoT is far from clear, in contrast to mobile, and the IoT revenue models for developers are still hit and miss. More than half of IoT developers are below the poverty line (making less than US\$ 500/month), and only 12% are in the top earners (making more than US\$ 25K/month).

There is evidence that as the industry matures it is already changing: 23 to 26% of developers professionally involved in other sectors are also professionally involved in IoT, and 32 to 33% of them are involved in IoT as amateurs. But what are these IoT amateurs building, and how are they building it?

While tools for IoT development do exist, the majority of them are stand-alone rather than constituting an integrated framework. At the moment, most mainstream IDEs only offer basic support for IoT. There are few examples of end-to-end IoT development environments, such as Electric Imp or Codebender, which couple into cloud platforms to make for a more-compelling development experience.

Although more than half (56%) of professional developers are dabbling with IoT, only 25% of them are involved in a professional capacity.

Mobile, Desktop, & Cloud - three platforms bridging into one

The overlap in professional development involvement across Mobile, Desktop, and Cloud is comprehensive. A significant number of Professionals working in each area are also involved in the other two.

The tendency towards convergence also shows in the number of platforms with which developers are involved: 34% of all developers are professionally involved in 2 development areas, and an additional 23% are involved in 3 development areas, making for a total of 57% of all professional developers working across sectors.

To address the rising complexity of having to deal with code that executes in different environments, that typically needs to be in separate projects, and that is often written in different languages, a number of tools and approaches have emerged to bridge the gaps and make life easier for developers.

IDEs such as Visual Studio and Eclipse are increasingly able to support development across mobile, desktop, and cloud. The idea of IDE as a platform, once championed by Eclipse with its plugin

architecture that enabled it to host many different 3rd-party tools, seems to have caught on. Visual Studio, for example, has made a pivot from an IDE geared towards a specific platform to a multi-purpose tool that offers services for sharing code, tracking work, and shipping software across languages, platforms, and even IDEs.

While there has been a large fraction of developers who shun IDEs in favour of simpler text editors with syntax highlighting, there are limits to this approach that become more pronounced as the complexity increases. Therefore the importance of having adequate IDE support cannot be underestimated.

JavaScript frameworks such as React, Angular, and Ember, are trying to unify development efforts across development areas - another take at the "learn once, write anywhere" approach. Since JavaScript has moved to the backend with Node.js, and conversely web clients have become more powerful and pushed more functionality to the frontend, a new approach has emerged: Isomorphic applications. Isomorphic JavaScript applications can run both client-side and server-side, with the backend and frontend sharing the same code and utilizing the same rendering framework.

But what is perhaps even more important than any tool and technology is the philosophy of and attitude towards convergence, which has been shaped to a great extent by what is termed DevOps. DevOps was coined to describe the convergence between software development and operations, extending continuous development as prescribed by the Agile movement to continuous integration and release.

In order to accommodate continuous releases, DevOps encourages automation of the build, integration, change, configuration, and release processes, as well as developer knowledge of and involvement in these processes. DevOps relies on a wide array of tools such as Docker, Ansible, Jenkins, and Nagios, to support these processes.

The rise of the DevOps movement has been pioneered by cloud shops, as much of what used to be considered infrastructure is now

part of the code. The cloud has been instrumental in bringing the separate development sectors together, and with 63% of developers implementing DevOps practices, DevOps has trained developers to work across boundaries.

The evolution of technological bridges, and the shift in attitude towards the software development lifecycle and methodology, have given birth to a new developer culture. In this culture, developers are not expected to remain confined in their narrow field of expertise, but rather to be knowledgeable across a wide array of technologies and application fields. Developers participate in activities throughout the development lifecycle and across development sectors.

As the three most-popular development sectors are now overlapping, they can be exploited to build and deploy applications that target them all at the same time, at least to some extent. A common backend in the cloud and unified front-ends for mobile and desktop, is the promise of the brave new world of development.

As the three most-popular development sectors are now overlapping, they can be exploited to build and deploy applications that target them all at the same time.

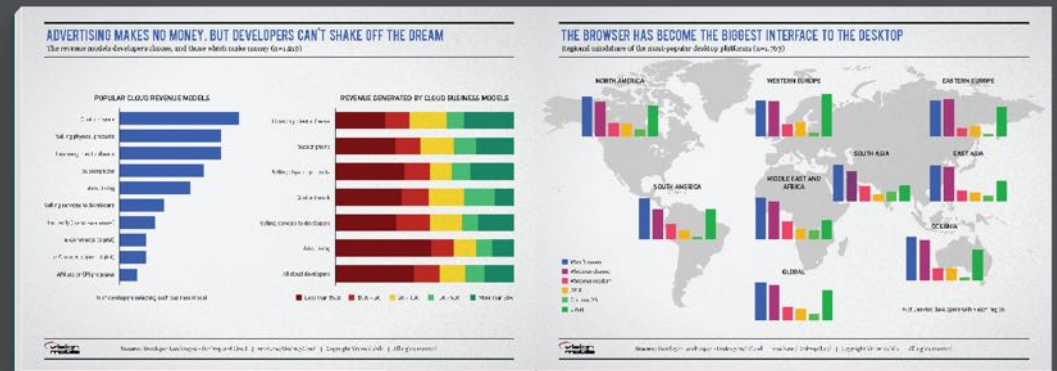
CLOUD AND DESKTOP DEVELOPER LANDSCAPE

The practice and profile of the desktop and cloud developer communities

The overlap between cloud and desktop computing isn't just in the tools and facilities, but also in the audience using them. Cloud developers are creating desktop applications, and vice versa, as the skills are increasingly transferable.

In this report we examine the cloud and desktop developer communities in detail, providing an insight into the applications being built, the tools used to build them, and the how developers are making money from them.

FIND OUT MORE
vmob.me/DesktopCloudDev



4 LANGUAGE TRENDS OVER THE LAST 12 MONTHS

Over the last year the use of scripting languages, JavaScript, CoffeeScript, & TypeScript, has grown significantly as development companies address the cost of coding by making greater use of faster-to-learn languages.

In mobile we can see the use of Java has dropped considerably since the start of 2015. Java is now the primary language for 22% of mobile developers, which is down from 29% a year ago. Java is the preferred language for Android development, promoted by Google who provides APIs to support all the latest hardware and integrates the language into the Android Developer Studio. Despite that it seems interest in Java is slowly declining amongst mobile developers.

That drop is probably attributable to the rise of cross-platform development, which often makes use of scripting languages to create multiple distributions of executable code suitable for different mobile platforms. A developer using React Native, for example, creates applications using JavaScript (with some custom extensions allowing access to mobile-specific sensors and suchlike), these applications are then run through a compiler which creates (native) executable applications for different platforms including iOS and Android.

The impact of such tools has been even greater on Objective C, which was for a long time the prescribed language for iOS development. In the beginning of 2015, 14% of mobile developers said that Objective C was their primary language, but 12 months later that proportion has dropped to 5%.

Objective C has also been under assault from Swift; Apple's new language of choice. Swift was launched in June 2014 and quickly gained a dedicated following. By the end of that year (when our Q1 2015 survey was carried out) 2% of mobile developers said that Swift was their primary language, while another 21% were using it. A year later we can see that Swift has gained a significant following, with 5% now claiming it as their primary language, but the proportion "also using" Swift has declined to 17%, as developers decide to embrace the language wholeheartedly, or not at all.

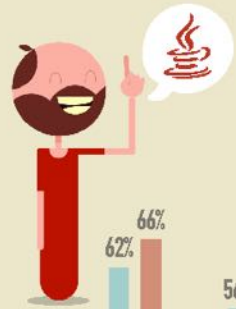
The increasing use of cross-platform tools has also impacted HTML5, which has gained functionality as mobile browsers got better support for some of the latest features of the standard. 12% of mobile developers now list HTML5 as their primary development language, up from 10% a year ago, but more interesting is the 54% who now list HTML5 as a language they "also use". This makes HTML5 the most widely used language in mobile, even if the majority of those using it are relying on another language for the bulk of their coding.

HTML is, of course, a layout language, designed for positioning text and graphics in an interactive arrangement. Developing in HTML is

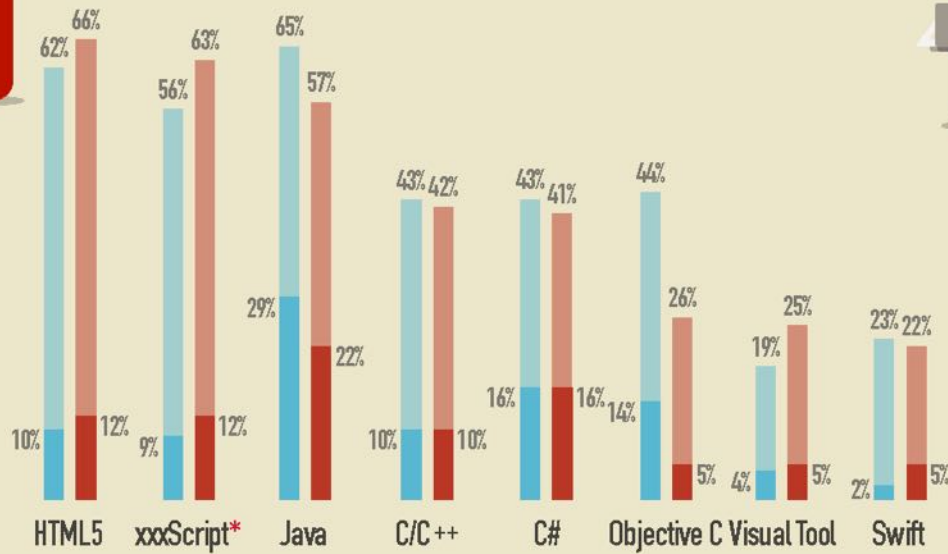
THE CONTINUING RISE OF HIGH-LEVEL LANGUAGES

% of developers using each language, of which as a primary language (mobile n=18,112, cloud n=14,382)

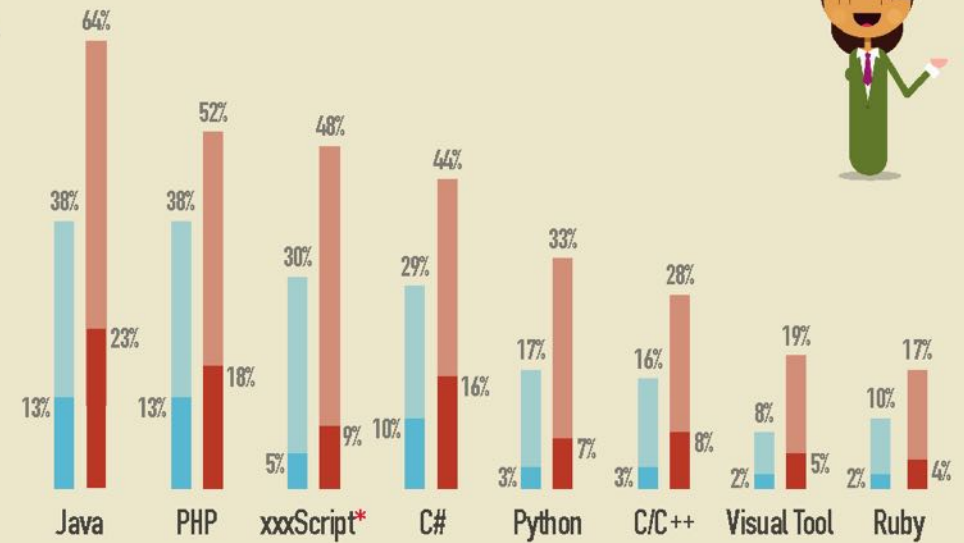
BASED ON
A SURVEY OF
21,000+
DEVELOPERS



Languages Used by Mobile Developers



Languages Used for Cloud Development



*JavaScript, TypeScript and CoffeeScript

very fast and intuitive, and HTML content can be rendered on just about any platform, making it ideal for creating user interfaces which will be used across multiple platforms.

Some cross-platform tools use HTML5 layouts to create native applications, but an increasing number of mobile applications are being run from within the device browser (9% of mobile professionals are targeting the browser as their preferred platform), and these factors combine to make HTML5 such a popular language with mobile developers.

Interest in Java is slowly declining amongst mobile developers. That drop is probably attributable to the rise in cross-platform development, which often makes use of scripting languages.

Microsoft's C#, on the other hand, remains static with 16% of mobile developers selecting it as their primary language. C# is mostly used to develop applications for Microsoft platforms, or cross-platforms apps through Xamarin (which is now part of Microsoft). As Windows has failed to gain any significant market share in mobile, the use of C# has not increased. We might expect to see C# losing support amongst mobile developers, reflecting the lack of market penetration, and that is what we see amongst developers "also using" C# where the proportion has dropped to 25% from last year's 27%.

Clearly C# is hanging on thanks to the power of Xamarin and those developers who still need to target phone handsets using Windows. The latter is clearly a declining minority, so the former is obviously an important group to the future of the language. C# has proved a popular language, but has historically been tied to the performance of Microsoft platforms. On Desktop and Cloud we can see it gaining ground, but amongst mobile developers it remains static.

Also interesting is the rise in the use of Visual Development Tools. This is not huge, only a 1% rise in primary use, and 5% in "also using", but it is steady. Such tools have long been ideal for simple,

but functional, mobile applications such as those used in data collection and other vertical markets, though they have evolved to deliver more functionality over time. Modern incarnations include B4X and RAD Studio, which claim to be suitable for a wide range of mobile applications. 20% of mobile developers now say they use a Visual Tool, alongside a traditional programming language, and we expect to see that proportion increase slowly as the tools evolve to meet the requirement of a broader developer community

Languages for cloud 9, and 10, and 11

In cloud programming what we see is a consolidation in language choices, with all the popular languages gaining mindshare at the cost of the minority options. Languages such as F# and Lua have all declined from the already-niche position they had a year ago.

Java maintains the largest mindshare for those developing server-side code, as a primary choice and part of their programming portfolio. 23% of cloud developers now claim Java as their primary language, up from 13% a year ago, and the trend is even stronger in the "also using" list where Java has jumped from 25% to 41%.

From our previous studies (notably Cloud and Desktop Developer Landscape 2016) we know that the choice of language is heavily influenced by the choice of cloud provider, with some providers pushing developers in a specific (linguistic) direction with supporting APIs and toolsets. We can see this with Microsoft Azure, which boasts the highest proportion of C# programmers, and Heroku, which attracts Ruby developers.

In that context Java is interesting, as it is used quite consistently across all the popular cloud providers (only Google boasts above-average use of Java in its clouds). Cloud developers using Java therefore have great flexibility in selecting a cloud host, and the increase in the use of Java may reflect a desire to maintain that flexibility in the face of a changing market.

We can see the effect of cloud hosts in the rise of C#, which (as a primary language) has jumped from 10 to 16% over the last year. C# came out of Microsoft in 2000, and is still managed (and owned) by the company, so it isn't surprising to see the fortunes of the language closely following the fortunes of the various Microsoft platforms. Azure, Microsoft's cloud service, has been performing well over the last year, and that shows in the increased use of C# amongst cloud developers who are obviously more interested in fast development than language portability.

Cloud developers using Java therefore have great flexibility in selecting a cloud host, and the increase in the use of Java may reflect a desire to maintain that flexibility in the face of a changing market.

We also see a decent increase in those “also using” C# to create server-side code, which is important as it represents a population who might be pulled into using the language more often, and thus become more likely to endorse Microsoft platforms in general.

Another area we see growing fast is scripting languages such as JavaScript, and Python, which see significant jumps in both primary and “also using” use. These languages not only permit fast development, but can also reduce development costs, as they are more-widely known so programmers are less expensive (though high-quality JavaScript developers will still command high wages).

In common with all high-level programming techniques, the cost of this improved productivity is performance and utility. Performance has, traditionally, been the most-significant barrier to the wider use of

scripting languages, though browser innovation has done a lot to address that in recent years.

Cloud computing does lend itself to such situations, as the performance can generally be improved through the purchase of more processing power. Additional resources can be brought online as they are needed, so an application which is underperforming can quickly be brought up to speed without having to engage more developers or re-architect the solution. This only works to a point, as large projects created in scripting languages can quickly become unwieldy, but the scripting languages are addressing many of these issues and there is plenty more room for growth which we expect to see over the next few years.

Contradicting that trend we also see growth in C/C++ as a primary cloud language. These low-level languages have always had a role in cloud computing, used by a minority of developers across most of the popular cloud providers, but when the industry push is towards greater speed of development and embracing high-level scripting languages it is a surprise to see C/C++ gaining ground.

C and C++ do provide the best performance, enabling fast analysis of big data, and that accounts for some of the growth seen. The rest of the growth can be traced to general growth in cloud computing, and the fact that 42% of cloud developers are hosting their own cloud service (see section one). Many of these self-hosted cloud servers are using Linux, in its various forms, and making use of open-source software developed for that open-source platform. Developers seeking to extend the functionality of those applications, or wanting to learn more about how they operate, are required to use C/C++ and therefore are keen to apply those skills more-broadly to their cloud development.

5 DEVELOPER REVENUE BY TARGETED PLATFORM

The majority of developers in most sectors are making less than \$500 a month in revenue, an unsustainable amount we consider below the application poverty line.

For many developers that isn't a problem, as they are Hobbyists or Explorers, more interested in learning about the technology than generating revenue. However, those interested in making real money are finding that some platforms pay distinctly better than others.

IoT developers are most at risk of falling below the \$500-a-month mark. 65% of them aren't making a sustainable income, and more than half (53%) are making less than \$100 a month. IoT is an area dominated by Hobbyists and Explorers; developers who are building skills in working with the technology, either professionally or as a side project, and this demographic reflects what is happening across the IoT industry. Few will now deny that IoT is going to have an enormous impact on computing, and society as a whole, but exactly where the money will be made is still very open to question so it isn't surprising to see so much experimentation going on.

There is a significant minority who are making good money, more than \$25,000 a month in revenue. These developers are likely working in vertical industries where IoT is more mature, and interoperability is less important, so deployments have moved ahead of generic applications and networks. When we look at the most-popular verticals, within that 12% who are earning more than \$25K, we see the value is in adding functionality to an existing product. Developers creating 3rd Party Applications make up more than half the group

(52%), while new devices and data mashups share the rest with business models impossible to describe in a few words. In fact, winning revenue models for IoT are so interesting, that we will be returning to these later in this report (see section six).

The big bucks are in browsers, but not the mobile browser

Desktop developers are more likely to be making money, above the poverty line, than those working in Mobile or IoT, but we see that more than 45% are still making less than \$500 a month. Despite that, we do have a healthy population of desktop developers who are earning a sustainable revenue, and a quarter of them are in our top-earner category pulling in more than \$25,000 a month.

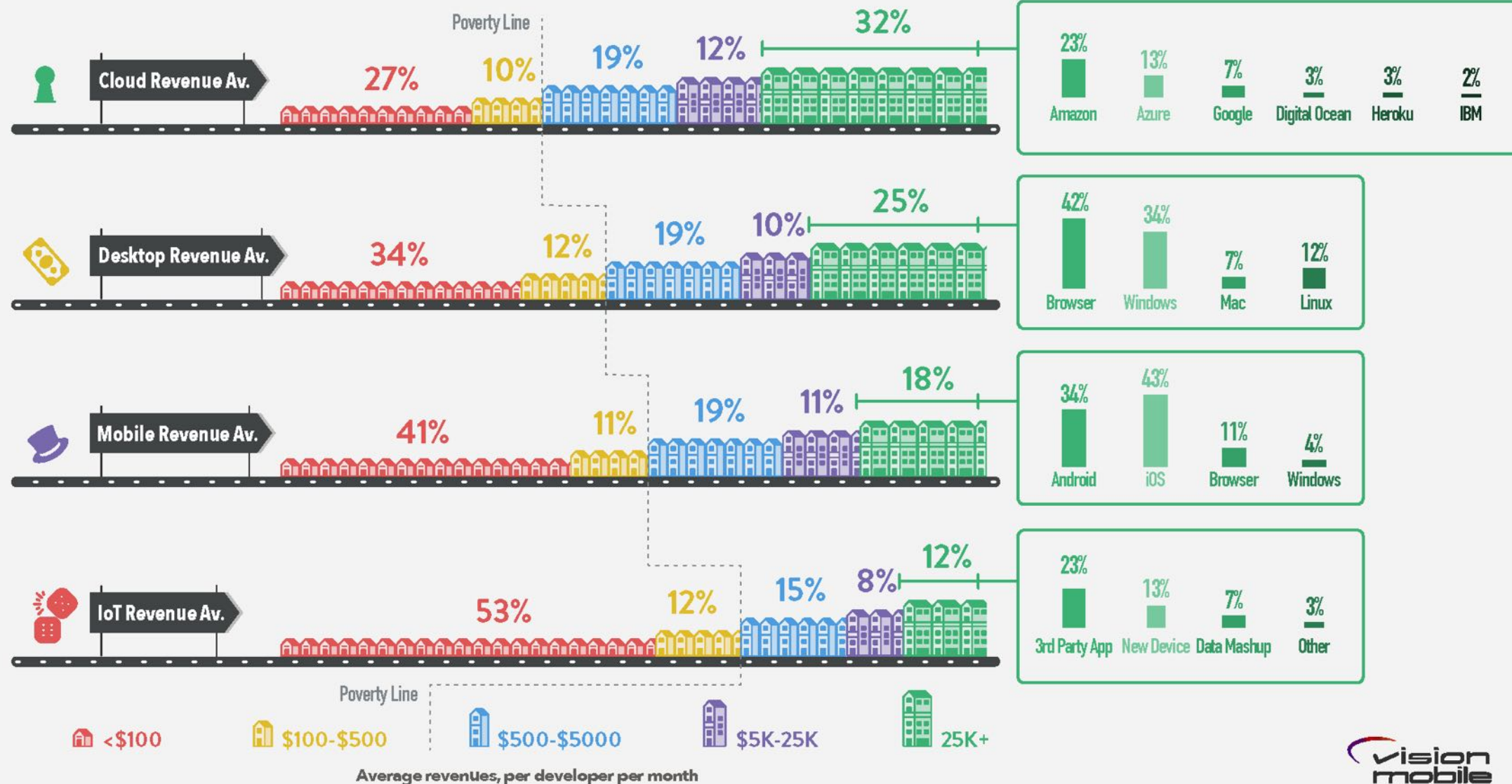
When we look closer at those top earners we can see that 42% of them are targeting the browser as their platform of choice, rather than creating applications that run on a specific OS. The advantages of browser-based applications are manifold: integral cross-platform capability, fast WYSIWYG development, and sandboxing to prevent catastrophic failure. These features combine to make the browser the preferred interface for 44% of professional desktop developers, but web development has also become more important as web sites have evolved.

DEVELOPER REVENUE BY TARGETED PLATFORM

% of developers in each revenue bracket, on average (n=1,951)

BASED ON
A SURVEY OF
21,000+
DEVELOPERS

The biggest money comes from the clouds



Most subtle are the applications asynchronously loading web content, and managing the content rendering based on company preferences and customer profiling. Historically this kind of functionality would have been delivered by server-side code creating a page which was simply rendered by the client-side browser, but with the development of browser technologies there is a demand for rich browser-based applications, which is being met by desktop developers.

When we look closer at those top earners we can see that 42% of them are targeting the browser as their platform of choice, rather than creating applications that run on a specific OS.

Modern web programming involves a great deal of development which adds no obvious feature or utility but instead improves the user experience at a low level. A better experience means selling more products, or content, even if the user isn't obviously aware of the complex development which powered the process.

A good example of this is the menus used by the Amazon retail store. At a glance these are simply cascading menus with the sub-menu popping up when a mouse loiters over a top-level item, but in fact a good deal of complexity is used to ensure the menu pops up on demand, and doesn't disappear as the mouse is moved towards it (passing over other menu items). JavaScript is used to track the movement of the mouse, and predict where it is heading, to establish what the user is trying to do. Amazon has clearly put an enormous amount of work into tweaking the user experience, all of which involves a good deal of development, executed in the web browser.

Even more extreme is the custom-shoe design offered by Converse, which provides a 3D render of shoes based on customer selection of threads and fabrics, turning a retail web site into an interactive experience selling a premium product.

Given the popularity and financial possibilities, of browser development amongst desktop developers, it is perhaps surprising to

see that trend absent in the mobile developer community. Only 11% of mobile developers making more than \$25,000 a month are targeting the browser, making it a minority option compared to the dominant platforms of Android and iOS. Clearly there is a significant difference in how users approach mobile browsing, and thus how developers perceive it - mobile browsers certainly don't match the performance of their desktop equivalents. But in terms of features they are a close match.

Mobile commerce is certainly important, though retailers often report that while users browse for products in their phones and tablets the final purchase is, more often than not, completed on a desktop computer, and this may reflect how users perceive the mobile browsing experience. Mobile browsers have, historically, been limited in their functionality, and while they have now largely caught up with their desktop equivalents they need to surpass that level to compare with the functionality provided to native applications. Integrated payment systems, for example, are provided by all the popular mobile platforms, but are not available from within the mobile browser, immediately limiting the functionality, and the revenue possibilities.

Only 11% of mobile developers making more than \$25,000 a month are targeting the browser, making a minority option compared to the dominant platforms of Android and iOS.

Developers still seem to be struggling with how to add value to the mobile browsing experience, but as they learn more, and mobile browsers become more integrated into the mobile experience, it seems likely that the advantages of browser-based applications will outweigh their limitations and we expect to see the proportion of mobile developers making a high income from browser-based applications to increase over time.

A two-horse race to making money from mobile

With mobile browsers being pushed into the minority, and Windows 10 failing to make a significant showing, the high-earning field is dominated by Android and iOS with 34 and 42% respectively. iOS developers are doing best, showing that targeting those who buy Apple products can be a highly effective strategy. Android has a much larger market share, in raw terms of handsets shipped, but the demographic that buys Apple products has (almost by definition) greater disposable income and can therefore be effectively targeted. This demographic is also more interesting to advertisers, leading to greater revenue for ad-supported apps.

While we expect to see the mobile browser attracting more attention from high-earning mobile developers, it seems unlikely that the balance between Android and iOS will change significantly in the foreseeable future. As Android handsets swamp the low-end market we will see a modest increase in Android developers making a lot of money, purely by having a bigger market to address, but in developed markets Apple will continue to be more profitable as long as Apple devices command the premium customer base.

There's money in them there clouds

The most-profitable development is clearly in the cloud, 32% are in our top-earner category pulling in more than \$25,000 a month. We have already established that cloud developers are the most-experienced community and in other studies (notably Cloud and Desktop Developer Landscape 2016⁴) we have noted that cloud developers have a higher level of professional qualification, so it shouldn't be entirely surprising that they are making more money for their companies and themselves.

The most-profitable development is clearly in the cloud, 32% are in our top-earner category pulling in more than \$25,000 a month.

When we look at where those top-earners host their clouds we see a distribution which quite-closely matches the market share owned by each provider. We haven't included self-hosted, as we wanted to compare commercial hosts, but if we had then self-hosted would account for 40% of those earning more than \$25K a month, accurately reflecting the 42% of cloud developers who (primarily) host their own clouds.

From this we can deduce that the choice of cloud host isn't a significant factor in achieving a sustainable financial return, any host will do.

⁴ <http://www.visionmobile.com/product/cloud-and-desktop-developer-landscape/>

6 IOT BUSINESS MODELS

While the Internet of Things is hailed as the next big thing, the revenue opportunity for IoT developers unfortunately doesn't look much better than that of mobile developers. It's tough to make a decent living in the Internet of Things.

Loyal readers of our State of Nation reports will remember that a majority of mobile developers operates under the app poverty line of \$500 per app per month - an amount that's unsustainable - and that situation has not changed though there is hope on the horizon.

IoT is tough business

Given that many IoT solutions involve device sales, with higher revenue potential but also a higher cost structure than apps, we could put our cut-off point at \$5,000 per month from IoT solutions. Only 1 in 5 IoT developers (20%) who are interested in making money, and who have reported their revenue in our survey, earn more than \$5,000 per month from their IoT activities. If we consider only those developers who have actual existing revenues - as zero-earners might intend to make money but not have serious efforts in place yet - we're still left with only 1 in 3 developers (34%) earning more than our minimum.

And there's more bad news. The hurdle to start earning revenue is increasing. Our data shows a 13-percentage point shift in the last 6 months from low earners (between \$1 and \$5,000 per month) to zero-earners (\$0 per month, even though they're interested in making money). Making any money at all is becoming more difficult.

Luckily, the number of IoT developers earning over \$5K per month has remained stable.

While the Internet of Things is hailed as the next big thing, the revenue opportunity for IoT developers unfortunately doesn't look much better than that of mobile developers.

The exploration phase

Of course, Internet of Things developers are not silently accepting this situation. Our data shows that they are actively experimenting to improve their situation. It's still early, but in these experiments we may find some hints on how IoT developers can make progress - both individually and as an ecosystem.

For one, successful IoT developers started using more revenue models in parallel in the second half of 2015. The overall average shifted slightly from 1.64 to 1.71 revenue models per developer, but this is a result skewed by zero-earners, who are both more numerous and use fewer models, not more, compared to the previous survey.

IOT BUSINESS MODELS

Where are IoT developers planning to make money (n=3,386)

BASED ON
A SURVEY OF
21,000+
DEVELOPERS

Making money in IoT is tough, but developers are finding the way

Average monthly revenue generated by IoT developers
(excluding those not interested in revenue)



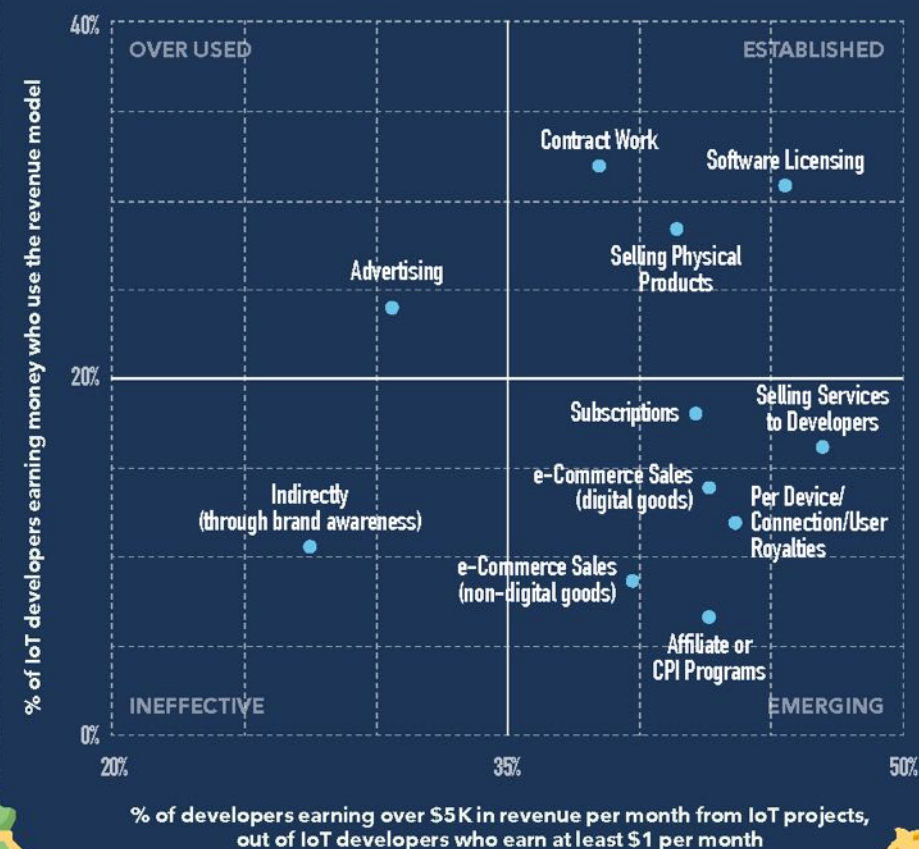
Number of revenue models used, by revenue generated



Difference in popularity, Q2-Q4 2015



Revenue model Effectiveness



For those who do earn money, the use of multiple revenue models has increased more strongly. In addition, more successful developers now use more revenue models than less successful ones, whereas before, developers across all revenue tiers used about the same amount of revenue models. The most successful developers use close to 2.5 revenue models on average.

For those who do earn money, the use of multiple revenue models has increased more strongly. In addition, more successful developers now use more revenue models than less successful ones.

Successful developers seek new, additional revenue streams on top of the ones they've got. Maybe they are productizing a consulting-type business. Maybe they can monetize a piece of technology by opening it up to other developers. These developers can light the path for others in showing which models work, and how to build a sophisticated, effective IoT business.

Developers are not just using more revenue models, they're also adjusting which models they use. Most of the widely used revenue models have dropped in popularity, presumably due to the large group of zero-earners using fewer models. e-Commerce in IoT is on the rise, as we've highlighted in our Commerce of Things report⁵; doubling in popularity from 3% to 6%.

Selling services to developers has also doubled in popularity, from 5% to 11% of IoT developers now using it. This might be the secondary revenue model of choice for developers who already have a successful product and want to monetize it further. This trend also shows that the ecosystem is building up and maturing, with more developer tools entering the market.

⁵ <http://www.visionmobile.com/product/commerce-of-things-2015/>

The most effective IoT revenue models

To gauge which revenue models hold most promise for IoT developers, we're introducing the IoT Revenue Model Effectiveness Quadrant. On the vertical axis: popularity, i.e. the percentage of IoT developers using each revenue model. On the horizontal axis: potential, measured as the portion of IoT developers using each model who earn over \$5,000 per month from IoT projects. For both metrics, we have excluded developers who are interested in making money, but earn exactly \$0 per month. Collecting any revenue at all is a big hurdle on the journey towards profitability and business success; excluding zero-earners ensures that we focus on the people who are truly serious about their business.

Together with selling to other developers, subscriptions, e-commerce sales (including affiliate programs) and royalties are the biggest opportunities for developers who want to climb the ladder.

Note that the average percentage of developers earning over \$5K (34%) is the middle of the X-axis on the scatter plot. Effective revenue models are considered to be those with higher probability of earning over \$5K. This includes most of the revenue models, due to the propensity of 'richer' developers to use more revenue models.

Software licensing, contract work and selling physical products (e.g. IoT devices) remain the top revenue models in IoT, in terms of popularity and solid earning potential. This is the bread and butter of many IoT developers. Contract work has the lowest barrier to entry, as it requires no more than your skills and wit to get started, but it is also the least scalable of the three. In other words: it can give you a solid income relatively easily but it is less likely to make you rich. Selling physical devices is the default mode for device makers, but it doesn't provide desirable recurring revenues.

Together with selling to other developers, subscriptions, e-commerce sales (including affiliate programs), and royalties are the biggest opportunities for developers who want to climb the ladder: highly effective, but not yet popular. In many cases, these models will generate recurring revenues, a key ingredient of stable, sustainable businesses. We've discussed e-commerce opportunities extensively in our Commerce of Things⁶ and IoT Megatrends 2016⁷ reports, for those who would like to dig deeper into this subject.

Indirect monetization (e.g. for product extenders) is not correlated with high revenues from IoT projects. It's unclear whether this

simply doesn't work, or whether it's difficult to allocate gains from the IoT part of the product or line-up from other activities that might have boosted company-wide revenues. Advertising, as it is in mobile, is the mermaid of revenue models. It pulls developers in with a siren song of media headlines, and it's easy to start using, but, barring exceptional examples, it is almost impossible to build a stable IoT business based on advertising revenue. In some cases, it might make for a worthwhile secondary revenue source, but we wouldn't bet the house on it.

⁶ <http://www.visionmobile.com/product/commerce-of-things-2015/>

⁷ <http://www.visionmobile.com/product/iot-megatrends-2016/>

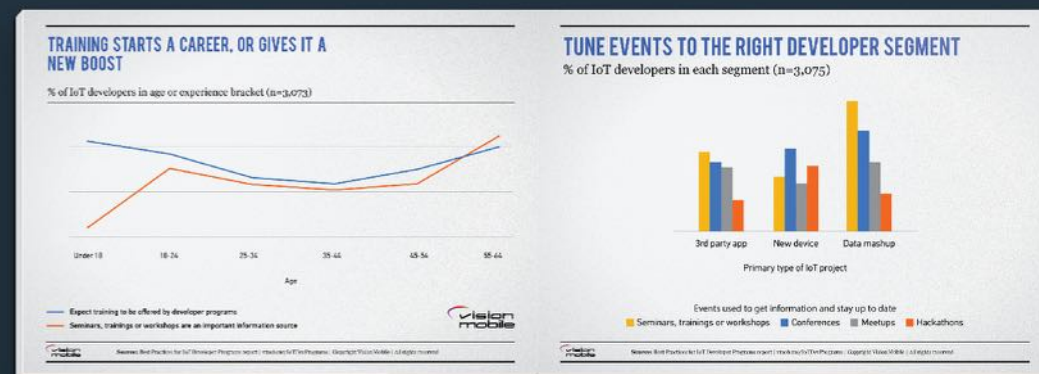
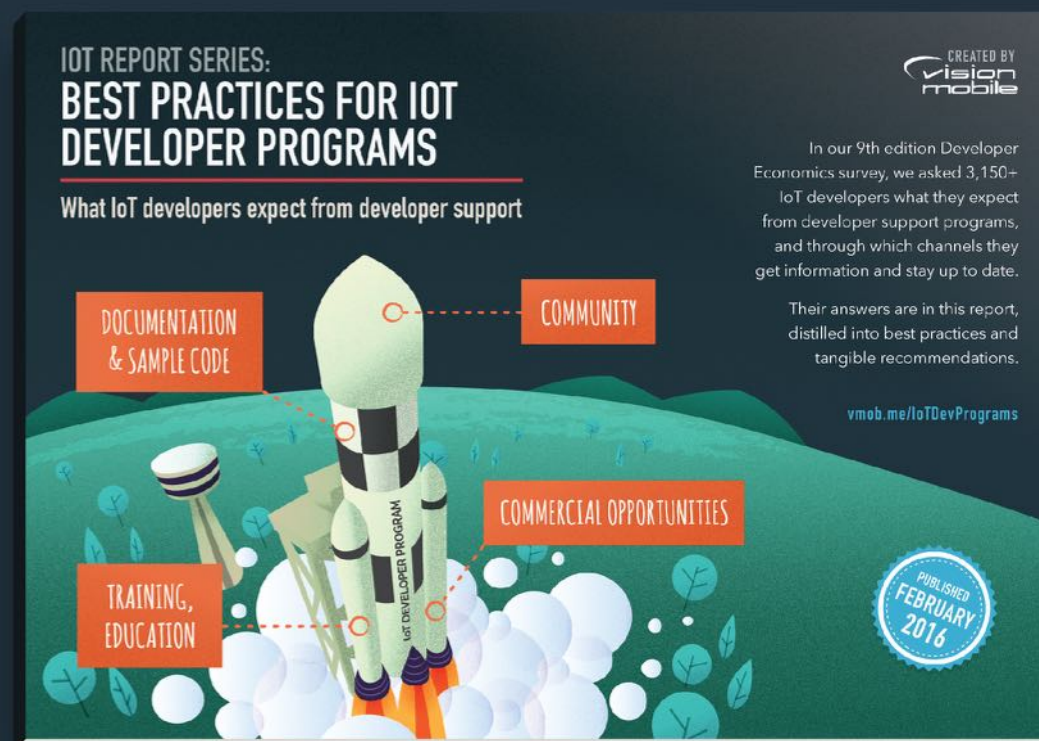
IOT REPORT SERIES: BEST PRACTICES FOR IOT DEVELOPER PROGRAMS

What IoT developers expect from developer support

Find out what IoT developers value most in a developer support program, based on a survey of 3,150+ IoT developers.

Internet of Things developers can be an important ingredient in your business model and this report will help you understand what IoT developers expect from your program.

FIND OUT MORE
vmob.me/IoTDevPrograms



METHODOLOGY

Developer Economics 10th edition reached an impressive 21,941 respondents from 150+ countries around the world. As such, it is the most global research on mobile, desktop, IoT and cloud developers combined ever conducted. This report is based on a large-scale online developer survey designed, produced and carried out by VisionMobile over a period of seven weeks between October and November 2015.

Respondents to the online survey came from over 150 countries, including major app and IoT development hotspots such as the US, China, India, Israel, UK and Russia and stretching all the way to Kenya, Brazil and Jordan. The geographic reach of this survey is truly reflective of the global scale of the developer economy. The online survey was translated into 11 languages (Russian, Vietnamese, Spanish, Chinese (simplified and traditional), Korean, Portuguese, Japanese, French, Indonesian, Italian) and promoted by leading community and media partners within the app development industry.

To eliminate the effect of regional sampling biases, we weighted the regional distribution across 8 regions by a factor that was determined by the regional distribution and growth trends identified in our App Economy research. Each of the separate branches: mobile, desktop, IoT, and cloud, were weighted independently and then combined.

The survey gathered responses from developers across mobile, desktop, cloud, and IoT platforms, using everything from vi-written C to visual development tools to create native, and cross-platform, applications aimed at a wide variety of platforms.

To minimise the sampling bias for platform distribution across our outreach channels, we weighted the responses to derive a

representative platform distribution. We compared the distribution across a number of different developer outreach channels and identified statistically significant channels that exhibited the lowest variability from the platform medians across our whole sample base. From these channels we excluded the channels of our research partners to eliminate sampling bias due to respondents recruited via these channels. We derived a representative platform distribution based on independent, statistically significant channels to derive a weighted platform distribution. Again, this was performed separately for each of mobile, IoT, desktop and cloud, using targeted vertical markets rather than platforms for IoT and cloud hosting providers for cloud.

As we have shown in our Developer Segmentation report, there is no average developer: Our outcome-based segmentation model of eight developer segments, shows that the choices and views of developers may vary wildly according to their desired outcomes from their development activity. Hobbyists, who just want to have fun, and Explorers, who are learning and testing the market, think very differently as compared to professional developers such as Hunters, who are after direct app revenues, and Product Extenders, who are using apps and digital services to promote their other products. We have therefore also weighted our results to minimize sampling bias for segment distribution across our outreach channels to derive a representative developer segment distribution. By combining the regional, platform and developer segment weighting we were able to minimise sampling biases due to these factors. All results in the report are weighted by main platform (or market), region and developer segment.

distilling market noise into market sense

