Contents

Executive Summary p. 4

Who Developers Are p. 6
Experience, Organizational Affiliation, Working in Teams, Work or Play

What Developers Do p. 13
Development Lifecycle, Applications Developed, Client & Server, Mobile App Development

How Developers Feel p. 18
Challenges, Where Developers Get Help, Why Software Projects Fail, What Excites Developers

What Developers Use p. 22
Tools of the Trade, Programming Languages, Databases and Platforms, Cloud

Conclusion p. 32
Methodology p. 33
Appendix p. 35
Snapshot of Today’s Global Developer

Today’s developers can be portrayed by these core traits:

- **Employed**: 70% Work in enterprises, 45% in orgs >100 employees
- **Experienced**: 68% Have 5 or more years of experience
- **Anxious to stay current**: 57% Challenged to keep up with dev technology
- **Balanced**: 71% Work equally (71% of developers) on business & consumer apps
- **Team player**: 74% Work in teams, 85% in teams <15 devs
- **Energized**: 66% Excited about wearables (44%) and/or robotics (38%)
- **Self-reliant**: 83% Use search engines and independent forums to get help
- **Open source**: 75% Use open source software
- **Polyglot**: 88% Multilingual (programming languages)
Executive Summary

In April 2015, the Alliance commissioned IDC to conduct an extensive global survey of software developers yielding over 850 responses. The survey cut across a broad cross-section of developers engaged in all paramount activities of application development. Some of the key findings are:

- Developers use multiple languages in their programming. 88% have used more than one language and 18% have used more than 5 languages in the past year.

- The causes of software projects failing is a central debate. 48% of developers cited changing or poorly documented project requirements as the reason for failure while 40% cited underfunding or a lack of resources.

- Many more women are joining the developer ranks. While females comprise only 25% of all developers, they are more than 40% of newer developers.

- Mobile software development is now the norm. About 87% of respondents are engaged in mobile development, and 18% cited organizational commitment to a mobile-first approach.
• Java was the most highly cited programming language with 68% of developers possessing moderate or high Java skill levels. This is likely due to its entrenchment in enterprise systems and its use as the “native” language for Android. Javascript, which is built into all browsers, is the second most-cited programming language.

• Wearables, the Internet of Things (IoT) and robotics are the next wave technologies that are most exciting to developers today.

• Developers are anxious about staying current with the latest trends with 57% reporting they are challenged to keep up with new development technology.

• Developers work in teams (74%) and are increasingly working in smaller teams. 60% of developers work in teams of 5 people or less, and 85% in teams of 15 people or less.

• Four in five developers use open source code in software development, and over two in five reported they contribute to open source projects. This is the highest IDC has noted in open source engagement and represents an important milestone for open source.
Who Developers Are

The Alliance Global Developer Insight survey highlights the striking characteristics of these highly skilled, highly sought after individuals.

Developers take their job very seriously, are experienced team players, are passionate about all aspects of software development and are increasingly female.
Experience

The survey shows that on the whole, software developers are very experienced. Developers consider software development to be highly complex, requiring considerable years of training or experience, and thus generally self-identify as an “established” software developer only after acquiring appropriate know-how. More than two thirds (68%) of developers reported as having over 5 years of experience and about an eighth (12%) have 20 or more years of experience. Given that the iPhone SDK shipped in October 2008 and was the genesis of the mobile app economy, we can ascertain that less than one third of all developers started developing during the mobile revolution.

**Years of Development Experience**

- 12% More than 20 years
- 30% More than 5 and less than 10 years
- 26% More than 10 and less than 20 years
- 23% More than 2 and less than 5 years
- 6% More than 1 and less than 2 years
- 1% Less than 1 year

N=855.
Drilling down further with regard to developers’ experience, a clear trend of increased female participation in software development emerges. As would be expected, the most experienced (20+ years) developers are predominantly (86%) male. However, among the newer developers (1-5 years) nearly 30% are female, and 42% of the very newest developers are female.

The substantial increase in female developers in recent years corresponds with female enrollment trends in computer science programs. As a result, we expect the ratio of female developers to continue increasing over time.

**Years of Experience by Gender**

<table>
<thead>
<tr>
<th>Experience</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 20 years</td>
<td>86%</td>
<td>14%</td>
</tr>
<tr>
<td>&gt; 10 and 20 years</td>
<td>80%</td>
<td>20%</td>
</tr>
<tr>
<td>&gt; 5 and &lt; 10 years</td>
<td>73%</td>
<td>27%</td>
</tr>
<tr>
<td>&gt; 2 and &lt; 5 years</td>
<td>70%</td>
<td>30%</td>
</tr>
<tr>
<td>&gt; 1 and &lt; 2 years</td>
<td>71%</td>
<td>29%</td>
</tr>
<tr>
<td>Less than 1 year</td>
<td>58%</td>
<td>42%</td>
</tr>
</tbody>
</table>
Organizational Affiliation

We asked developers where they work and found that some 69% are employees or contractors involved with various types of organizations. The remaining 31% considered themselves independent developers, freelancers or developers working on their own ideas. For this study we labeled these developers as “Unaffiliated.”
Approximately one half of respondents worked for a technology vendor. Only 16% worked in an IT user organization.

Of the 31% unaffiliated developers, 12% were completely independent and working on their own ideas. This is a sizable group which would have been unlikely even 5 years ago. We believe that mobile devices and the app economy are important factors contributing to this growth, though the rise of gaming, and the higher level of freelancing in the economy in general are also contributing factors.

The finding that 69% were affiliated or organizationally employed is consistent with IDC data that estimates that 40% of the world’s 18.5 million developers are hobbyists — those who engage in software development outside of a job or their main job. IDC believes this reflects the increased engagement of business employees in application activities in what is known as the “citizen developer” phenomenon in enterprises.

Improved tools and the increased level of abstraction provided by modern model-driven application development tools are key to this increase. This research is referenced in the Appendix section of this study.
Of the survey respondents who were organizationally employed, 54% were “individual contributor” software developers and 46% had a management role. Two-thirds of organization-connected developers worked in companies greater than 100, which is consistent with census data on the number of employees in firms over 100 employees in size.
Developers work in teams and are increasingly working in relatively small teams. Nearly three quarters (74%) of the survey sample reported they work in teams. 60% of the developers reported that they work in teams of 5 or fewer. Overall, 85% reported that they work in teams of 15 or fewer. IDC believes that this reflects the increased adoption of agile methodologies which benefits from smaller teams.
What Developers Do

While coding, or authoring of software, is the most common activity associated with software development, the discipline spans several different activities that encompass the entire lifecycle of software projects.

Each of the major phases of software development — including requirements gathering, coding, testing, and deployment management — was selected by over half of the respondents. This reveals the broad scope and diversity of many developers’ skills (and of enterprise requirements).
The survey asked about specific activities in software development to understand at a granular level what developers do and how they spend their time. The most popular software development activity is - perhaps not surprisingly - coding, which came in at 71%.

Also not surprising is that functional testing and bug-tracking (which tend to happen after code creation), and requirements analysis and design (which generally happen before), had a roughly equal incidence rate of around 62% and 61%. These responses are generally consistent with prior IDC research.

The developers that selected coding identified that activity as consuming 45% of their time, by far the most time-consuming of all activities practiced.

![Development Activities and Time Spent](chart)

<table>
<thead>
<tr>
<th>Activity</th>
<th>Incidence Rate</th>
<th>Time Spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>39%</td>
<td>25%</td>
</tr>
<tr>
<td>Design</td>
<td>61%</td>
<td>30%</td>
</tr>
<tr>
<td>Coding</td>
<td>71%</td>
<td>45%</td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td>24%</td>
</tr>
</tbody>
</table>

Development activities, regardless of time spent
Proportion of time spent

N=855.
When asked about the types of apps they have built, 72% of developers selected business applications followed closely by consumer applications at 62% (multiple responses allowed). A smaller but still sizeable 28% selected gaming. An aggregation of the data reveals that 71% of developers are building out consumer and/or gaming apps. The consumer apps revolution continues to build momentum. With this trend, we expect the number of developers engaged in consumer development to move ahead of business development in the near future.
Mobile App Development

The survey found that 87% of organizations where developers worked were involved in mobile development, including 18% of organizations that identify a “mobile first” strategy as their key approach.

Business Approach to Mobile Software Development

- **18%** Committed to mobile-first approach
- **13%** Does not develop mobile apps
- **35%** Has developed some mobile apps for B2C
- **34%** Has developed both B2C and B2E
When examining all respondents who identified as developing mobile client software (i.e., mobile developers), we see preferences between native versus Web platform approaches. Almost 70% of developers are creating native apps while mobile Web and hybrid Web are both about 50%. Third party application platforms such as Xamarin, Kony, Appcelerator, Red Hat Mobile Platform, etc are referenced by 1 in 5 developers. Individual developers still prefer native mobile development.
How Developers Feel

One of the important goals of the survey was to probe developers regarding a variety of important issues, then in future surveys we plan to track their changes and evolution. We included a broad question on the most pressing long-running challenges developers face. The biggest concern by far was the challenge of staying current with the technology of their craft, with a 57% incidence rate.

Most Pressing Developer Challenges

- Staying current with new languages, frameworks and tools: 57%
- Striving to develop high quality, high performance and secure code: 43%
- Overall work/life balance: 39%
- Overly demanding clients that do not appreciate the complexity: 37%
- Finding work that utilizes the skills I have developed: 32%
- Difficulty in communicating with team members and stakeholders: 14%
- Immature tools and platforms that make my work harder: 13%

N=855.
As for where developers go to get technical help, most (63%) said they first utilize a search engine. We have seen a continuing shift in the last decade away from vendor networks (now at 38%) or more traditional sources of information like books and magazines (now at 19%). The second most popular option is independent forums such as StackOverflow. A deeper exploration of the survey data showed that some 83% of developers cited either search engines or independent forums as a primary source, suggesting that developers are a highly independent group and platform vendors must accommodate this trend in their outreach strategies.

<table>
<thead>
<tr>
<th>Where Developers Get Help</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search engines (e.g. Google/Bing)</td>
<td>63%</td>
</tr>
<tr>
<td>Independent dev networks &amp; forums</td>
<td>55%</td>
</tr>
<tr>
<td>Vendor/ product info/ blogs/forums</td>
<td>38%</td>
</tr>
<tr>
<td>Personal networks of friends, etc...</td>
<td>33%</td>
</tr>
<tr>
<td>Formal Training</td>
<td>31%</td>
</tr>
<tr>
<td>Magazines or books</td>
<td>19%</td>
</tr>
<tr>
<td>User group meets and professional societies</td>
<td>11%</td>
</tr>
</tbody>
</table>
Why Software Projects Fail

The causes of software projects failing is a central debate. The reason most cited for project failure was changing or poorly documented project requirements. Historically, this reason has been relatively small, and the practices around requirements gathering have not been as active a debate. Developers love to code, but also need effective communications systems in place. Potentially tied together are the next reasons for failure around underfunding or under-resourcing (40%) and team management (37%). Tools and platform vendors would be well advised to understand these results to uncover unmet developer needs.
We wanted to gauge what next wave of technologies are exciting for developers. We provided respondents with a list of current and emerging areas in the industry. The lead area of developer excitement is wearables, cited by 44% of the respondents. Robotics was a surprise trend, coming in second. If we combine IoT across both consumer and business, it would come on top at 48%. Another surprise was drones, cited by 19% of the developers. Since drones are essentially a sub-branch of robotics, combining the responses for robotics and drones shows an incidence rate of 46%, second to the combined IoT.
As programming practices have matured, software development tools have proliferated. The areas of mobile, cloud and devOps have enjoyed an explosion of tools and in some cases new categories such as cross-platform mobile advertising tools. While the tool categories listed address different aspects of application development and management, it is interesting to note that some are used more broadly than others.
It is not surprising that the most universal tools are code editors and integrated development environments (IDEs). Historically, IDEs have been the mainstay of professional developers as they provide a central console for managing most aspects of software projects (especially code authoring). It is perhaps this constant integration of added function that have given developers some pause in using IDEs in all situations. Code editors tend to be light-weight, easy to install and start, and focused on the authoring process. In recent years, there has been a flight to lightweight tools and middleware.

The chart reflects the prioritization of tools and middleware for the maturing software industry. For example, certain niche tools like business rules engines have compelling uses in certain situations but — with an incidence rate of 16% — are clearly not universally deployed. Another interesting observation is that relational databases (SQL databases) continue to dominate (50%) compared to recently popular NoSQL databases (15%) which feature different styles of lightweight storage algorithms to support new styles of applications.
Programming Languages

Programming languages are the most technically intensive skills developers must learn. We asked about the number of programming languages developers used in the last year.

Today’s Polyglot Programmer

- 3% More than 10 languages
- 15% Between 5 and 10 languages
- 70% Between 1 and 5 languages
- 12% One language

A Global Survey of Today’s Developers

N=855.
The data shows 88% used more than one programming language, and about 18% used more than five. Historically, software developers build a skill set in a given language and learn it over a number of years. It wasn’t long ago when most developers identified themselves as “a C developer” or a “Java developer.” Today, however, the world is more complex, the tools are more powerful and the learning curves are somewhat faster. There has also been an explosion of new programming languages addressing specific domains of software construction (e.g. Erlang for realtime systems, Node for asynchronous programming, Rust for better multicore programming, etc...) and many have gained respectable levels of traction. To build today’s apps, a development team may need to use multiple languages and some developers are conversant with multiple languages adequately enough to support multi-lingual software projects.

IDC’s model of programming languages categorizes them into five major groups which we refer to as programming language ecosystems. These typically tier along the level of abstraction in the programming mode. We have found historically that developers tend to operate comfortably inside one ecosystem. (See the Appendix for more information about these language groupings and their relative size.)

This survey ranks the most popular programming languages, highlighting two levels of developer skill depth for each language.
The top answer is Java, which is the most popular programming language for enterprise systems and has received a boost in its popularity from its selection as the “native” programming language for the Android platform. Second is JavaScript, which is the standardized language for programming browser front-ends. While JavaScript is often used with other mark-up languages like HTML and CSS, we deliberately focused on JavaScript as it reflects a deeper level of Web development beyond website layout.
SQL (structured query language), which is the lingua franca of relational databases, ranks third—and indeed, a third of all developers have a high level of SQL skill. Other languages that rank highly are C/C++ which is largely used for system level development where performance is critical, and C# which is the de facto Microsoft ecosystem language for desktop and server applications. C# is the language of .NET, so it rises in correlation of the popularity of the Microsoft platform.

To build today’s systems, a development team may have to use multiple languages, and some new developers are conversant with multiple languages adequately enough to support multi-lingual software projects.
The popularity of SQL should come as no surprise given the emergence of relational database systems approximately three decades ago. According to IDC’s Software Tracker database, the RDBMS category is responsible for $30 billion, or a third of all 2014 spending on application development and deployment software licensing and subscriptions. The two largest products in the space today are Oracle Database and Microsoft SQL Server, but both lag behind MySQL.

### Most Popular SQL Databases

<table>
<thead>
<tr>
<th>Database</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>64%</td>
</tr>
<tr>
<td>Microsoft SQL server</td>
<td>49%</td>
</tr>
<tr>
<td>SQLite</td>
<td>39%</td>
</tr>
<tr>
<td>Oracle DB</td>
<td>37%</td>
</tr>
<tr>
<td>ProgressSQL</td>
<td>23%</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>17%</td>
</tr>
<tr>
<td>SAP/Sybase adaptive server</td>
<td>8%</td>
</tr>
<tr>
<td>SAP Hana</td>
<td>6%</td>
</tr>
</tbody>
</table>

N=427.
MySQL’s success is rooted in its availability as open source starting in the early days of the Web. Today MySQL has become the database most widely used by developers.

Open source availability also plays a major role in the adoption of the relatively new group of non-relational database products, collectively known as NoSQL databases. These data stores, which cater to simpler data structures often accessed directly from inside application code, come in various types. We asked developers which groups they use the most. Of those developers that use NoSQL databases, three quarters (75%) use document-oriented databases (e.g. MongoDB, Couchbase, etc.).
The survey results highlight the importance of open source for today’s developers. Indeed, open source has gone mainstream even in relative conservative sectors of enterprise IT. Approximately 77% of developers said they use open source, and 43% were involved in contributing to open source. Furthermore, while Linux desktops account for less than 1% of all desktops, they are used by 12% of software developers in our survey.
Cloud

The rise of cloud computing is evident in our survey. We asked developers which types of back-end systems and styles of cloud computing they use. 50% are using some form of Infrastructure-as-a-Service (IaaS) clouds. The growth of Platform-as-a-Service (PaaS) adoption — now 29% — is accelerating, which is significant because PaaS can be considered more productive for developers but is less mature.
Conclusion

The Alliance Global Developer Insights survey shows an increasingly diverse developer universe that is dealing with the increasingly stressful and complex demands of modern business. The survey provides a broad view of the nature of modern application development, highlighting the increased focus on front-end development, the rise of the consumer app developer and continued adoption of agile methodologies as organizations focus on smaller software development teams. This is encouraging, as agile drives towards faster development and it is generally accepted that smaller teams work faster. The survey also shows that mobile application development is becoming the norm and that developers are excited about IoT, wearables and programmable robots and drones.

The Alliance Global Developer Insights survey shows an increasingly diverse developer universe that is dealing with the increasingly stressful and complex demands of modern business.
Methodology

The Alliance Global Developer Insights survey is focused on the individuals who are actively involved with software development, by performing any of the various roles across the software development life cycle:

• Determining software user requirements and systems analysis
• Designing and architecting software (including UX design and prototyping)
• Coding and maintaining software
• Testing and bug-fixing software
• Deploying and maintaining released software

Survey respondents were recruited by both the Alliance and IDC to complete the survey online in a three week period from late March to mid April.
By design, the survey was intended to be broad and cover all developers across all platforms, including desktop, Web, mobile, cloud and server back-ends. There was no specific focus on any programming language, or developer ecosystem. Developers from the consumer space and the enterprise space were allowed to take the survey without restriction.

A total of 855 developers responded to the survey, with 78% reached in the USA and the remaining 22% from 50 other countries. While IDC and the Alliance opened this survey to all geographies, the respondent base was more highly concentrated in the USA. According to IDC estimates, USA normally accounts for only 19% of world-wide developers. The male-female split in the survey is representative of the developer community found in the United States.

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world’s leading technology media, research, and events company.
Appendix

IDC’s long-running tracking of worldwide developer populations pins the total number of developers at 18.5 million. This number is based on a bottom-up methodology of estimating developers according to labor and education statistics available for a number of countries.

Developer Source Populations

- **18.5 million** Total Software Developers (orange circle)
- **7.5 million** Hobbyists & Citizen Software Developers (oval outside of circle)
- **11 million** Professional Software Developers (circle)
- **18 million** ICT Operations & Management Skilled Workers (oval outside of circle)
- **~1.3 billion** World’s Students (Secondary & Tertiary)
- **~3.3 billion** World’s Labor Force
- **29 million** Total ICT-Skilled Workers (blue oval)
There are 11 million professional developers in the world and 7.5 million hobbyists or informal developers (students, hackers, game and lone mobile devs, early stage startup devs, etc.). The top countries in software developers are the US, China and India which account for almost 40% of all software developers in the world.

There are many ways to segment developers, but one important way is by the types of languages they are skilled in. The programming language and model landscape can be classified on a continuum of abstraction where the lowest are close to the machine and require a lot of detailed skill and complexity to program and the highest operate at the business or domain level and require less technical skill but strong business knowledge and facility with general logic and data relationships. The programming languages and models tend to group and cluster around certain ecosystems which are highlighted in the rectangles of following figure.
Developer Skills Ecosystems

In this figure, the x-axis describes the suitability of the various language groups for front-end or back-end application development. The y-axis represents increased abstraction or distance from a hardware machine-model of programming. These ecosystems of skills are important because developers tend to identify with them and are more likely to learn languages within the ecosystem than outside of it. Most developers tend to have skills in multiple languages and ecosystems however, but generally have primary affinity to a certain level of abstraction.

Source: IDC, 2015
Examples of the programming languages that dwell in the various ecosystem are detailed in following figure.

## Language Groups in Detail

<table>
<thead>
<tr>
<th>Ecosystem</th>
<th>Compiled-to-Machine Languages</th>
<th>Web Platform Languages</th>
<th>Business &amp; Data Languages</th>
<th>Java Platform Languages</th>
<th>Microsoft Platform Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>• Full ahead-of-time compilation process</td>
<td>• Runs in Web browsers or web servers</td>
<td>• High abstraction</td>
<td>• Java and clones</td>
<td>• Microsoft languages</td>
</tr>
<tr>
<td></td>
<td>• Generates machine code (not bytecode)</td>
<td>• General scripting &amp; interpreted languages</td>
<td>• Manipulates data objects</td>
<td>• Generates Java or Java bytecode</td>
<td>• Support Windows Platform</td>
</tr>
<tr>
<td><strong>Usage Scenario</strong></td>
<td>• Commercial software, Embedded applications Mobile apps Scientific programming Legacy work</td>
<td>• Web &amp; mobile applications Server-side scripting Systems management</td>
<td>• Business &amp; process apps and BPM tools Model-driven app platforms Mobile apps Citizen developer opportunistic &amp; RAD</td>
<td>• Industrial-strength back-ends Enterprise apps Commercial/ISV apps Embedded &amp; mobile</td>
<td>• Client-side Window platform apps Business apps Commercial/ISV apps Embedded &amp; mobile</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>• C/C++ Objective C/Apple Cocoa COBOL Fortran Pascal/Modula Delphi Ada PL/1</td>
<td>• HTML XML JavaScript Flash/ActionScript CFML/CFCscript PHP, Ruby, Python, Perl, LUA, Lisp</td>
<td>• SQLPL-SQL, T-SQL PowerBuilder Uniface, Unify Progress 4GL Cache/MUMPS Pick Basic SQR, RPGS, R, SAS Macro</td>
<td>• Java/Android dalvik</td>
<td>• C# Visual Basic VB.NET VBA/Access Visual Foxpro F# Windows PowerShell</td>
</tr>
</tbody>
</table>
About The Alliance
The Application Developers Alliance is a non-profit global membership organization that supports developers as creators, innovators, and entrepreneurs. In partnership with its more than 200 corporate members, the Alliance promotes the continued growth of the industry and advocate on behalf of developers on public policy and industry issues. The Apps Alliance serves a growing network of over 60,000 developers globally. Corporate members include: mobile app publishers, platforms, wireless carriers, hardware manufacturers, ad networks, enterprise tools and service providers.