

The Mobile Software App Takeover

Jeffrey Voas, IEEE Fellow

J. Bret Michael, Naval Postgraduate School

Michiel van Genuchten, Open Digital Dentistry



MARC ANDREESSEN WROTE a column in *The Wall Street Journal* in September 2011 entitled, “Why Software Is Eating The World” (<http://on.wsj.com/o6yleE>). Part of his reasoning was based on the sheer increase in smartphones: “Over two billion people now use the broadband Internet, up from perhaps 50 million a decade ago In the next 10 years, we expect at least five billion people worldwide to own smart phones, giving every individual with such a phone instant access to the full power of the Internet, every moment of every day.”

In 2011, approximately 850,000 Android devices were activated every day, equating to roughly 310 million that year (www.pcmag.com/

[article2/0,2817,2400782,00.asp](http://engt.co/KiIWNG)). According to IDC, smartphones sales outpaced personal computer sales for the first time in the fourth quarter of 2010, with 100 million smartphones sold versus 92 million personal computers (<http://engt.co/KiIWNG>).

Getting Smarter

However, smartphones aren’t very “smart” without the software apps that give them their usability and versatility. Apps, like all software, need some degree of guidance, regulation, and measurement to ensure a user is receiving proper functionality and quality of service. This problem has existed in software engineering and software development from day 1, and apps are no

different. The void has led to a recent clamor for some way to vet apps and the app stores in which they will reside and be licensed. We contend that this will be a huge problem for the mobile app market as it continues to grow.

Mobile apps have radically changed the landscape for software development and delivery. Today, they’re developed by everyone from teams of engineers whose primary job is software engineering to laypeople from all professions and walks of life who may have no formal education or training in software engineering or computer science. This is a major departure from the past: the mobile software development frameworks and tools now empower the individual to participate in software development.

In addition, “worldwide mobile app store downloads are forecast to reach 17.7 billion downloads in 2011, a 117 percent increase from an estimated 8.2 billion downloads in 2010, according to Gartner, Inc. By the end of 2014, Gartner forecast over 185 billion apps will have been downloaded from mobile app stores, since the launch of the first one in July 2008” (www.gartner.com/it/page.jsp?id=1529214). The bottom line is that the volume of app software being produced is quickly overtaking other types of software.

Volume is key in understanding the current software industry, and understanding the mobile industry is key in understanding how most software is now developed. The unmatched volume of mobile devices will allow the mobile industry to incorporate more and more software functionality into mobile phones. It started with MP3 players and photo cameras and will continue with search, navigation, and payments. Software is eating the world, and mobile software will take over larger parts of the software world as we know it today.

For those who think mobile platforms and mobile applications are trivial, consider that the Android platform contained 12 million LOC in 2010 (<http://source.android.com/source/code-lines.html>). A description of the Real platform indicated that the total platform is up to 20 million LOC and that approximately 100 million Real clients are sold every year.¹

Upcoming Challenges

Both in volume and in size, mobile applications are among the most challenging software systems being built these days. The increasing importance of mobile platforms will have several consequences for the software profession and the software professional:

- Increasing numbers of companies will decide to first develop for the

mobile platform and later port to computers. Adobe popularized this option when its CTO announced “mobile-first”: “Our approach at Adobe is to take a mobile-first view on the new work that we are doing to design for the more constrained environments, then look to ways to enhance that experience for higher performance environments” (<http://blogs.adobe.com/conversations/tag/kevin-lynch>). Educators might decide to take a mobile-first approach as well. Learning software engineering on a limited platform and later applying it to richer platforms could be more effective than the other way around.

- Assuming unlimited CPU and battery life isn’t advisable.² Also, high volumes of smartphones might be in countries where you can’t take infrastructure for granted.
- Flashiness will become as important as functionality and qualities such as reliable operation and trustworthiness. As a lot of the required functionality is now available for mobile phones, more attention is devoted to “funny” and “fancy.” For high-selling apps such as Fruit Ninja or Angry Birds, looks can be more important than mean time between failures and the way in which the requirements were derived.
- Whereas the PC industry over the past 20 years was a one-platform industry, mobile phones will compete in a market with at least two or three platforms that have a volume equaling or exceeding that of the entire PC industry.
- Diversity in technology and vendors will raise the issue of interoperability. Software app developers will have to decide which platforms they want to build for because each one has distinct operating systems, apps stores, security policies, and so on.

Another phenomenon taking shape

today is the clamor for “mobility.” Everyone wants to have access to all information at all times and in all places, which mobility and wireless offer. However, mobility requires assets such as access to clouds or other forms of storage and retrieval. It also has disadvantages such as localization when someone wishes to remain stealth.

Furthermore, discussions about bring your own device (BYOD) make trust, security, and privacy concerns even greater for mobile devices of any flavor and variety. For example, the difficulty in building a smartphone with a fixed set of requirements pales in comparison to what it would mean if every employee in an organization could select his or her own mobile device, hook it to the organization’s enterprise, and then let the enterprise bear responsibility for anything (malicious or non-malicious) on each device. It’s difficult enough to reengineer a commercial off-the-shelf mobile device for very specific policies and requirements—in BYOD, it simply isn’t feasible. However, there is a clamor for such to occur for its alleged cost benefit without much consideration for the likely long-term security and privacy pitfalls.

The implications of BYOD have been brought into the limelight through the large amount of press coverage of the phone hacking scandal involving *The Guardian* (<http://nyti.ms/IpeWi3>). This, combined with widespread fears of industrial and state-level cyberespionage, along with attacks on all forms of computing devices, has raised the public’s awareness of trust issues. Although we recognize that there is no silver bullet to addressing issues of trust, there are trusted computing mechanisms that are becoming mainstream in the mobile device arena, particularly at the hardware- and firmware-level, such as the Trusted Platform Module and self-encrypting storage devices. In addition, more than lip service is being paid to cloud security these days, which is a

good sign, given the symbiotic relationship between “the cloud” and mobile devices and their apps.

In This Issue

To delve deeper into these topics, we selected three articles that cover the benefits and challenges of the mobile software market in different ways. In the first article, “Mobile Content as a Service: A Blueprint for Vendor-Neutral Cloud of Mobile Devices,” the authors discuss an alternative idea to the current thinking of storing data content in clouds and then having different devices access that content from different locations. Their idea is to maintain data content in the device where it was first created, leading to a mobile device cloud in which devices, together with the content and resources they host, are treated as “first-class cloud citizens.” The authors’ proof-of-concept implementation, presented in the article, is based on standard Web protocols, and the underlying design is configured for various contexts such as individuals who have several mobile devices.

In the second article, “Developing Software for a Mobile Service in India,” the author explains how mobile phones have penetrated the so-called BRIC (Brazil, Russia, India, and China) countries over the past decade. Specifically, he discusses why creating software for BRIC countries offers a new set of software development and delivery challenges. The author argues that it can’t be assumed that there exists an infrastructure in which everybody has PC access to the Internet to upload or download applications or other data content. The specific software challenges this creates and the ways in which these challenges have been overcome is discussed via a real-world example from a Bangalore company, OnMobile, which provides mobile software products and services to 92 mobile operators and media companies in 52 countries.

ABOUT THE AUTHORS



JEFFREY VOAS is an IEEE Fellow, associate editor in chief of *IT Professional* magazine, and IEEE Division VI Director for 2012. Contact him at jeffrey.m.voas@gmail.com.



J. BRET MICHAEL is a professor in the computer science and electrical and computer engineering departments at the Naval Postgraduate School. Contact him at bmichael@nps.edu.



MICHEL VAN GENUCHTEN is vice president at Open Digital Dentistry in Switzerland. Contact him at genuchten@ieee.org.

In the last article, “Cloudberry: HTML5 Cloud Phone Platform for Mobile Devices,” the authors discuss the Cloudberry project at Nokia Research Center, which has built an HTML5-based cloud phone software platform for mobile devices. A *cloud phone* is a mobile device in which all the device’s user functionality is downloaded and cached dynamically from the Web, including all the applications and even the entire top-level user interface for the device. The authors claim that a central benefit of cloud phones is that nearly any customer-facing feature of such a device can be changed on millions of devices anywhere and almost instantly. From the user’s perspective, the authors argue, the key benefit is multiple device ownership—the ability to access the same applications and data effortlessly from different devices without tedious

installation, data copying, or synchronization hassles. Their article provides an overview of the Cloudberry system, focusing on its overall vision, usage, technical architecture, and evaluation.

We hope you enjoy this issue as much as we enjoyed putting it together. Feel free to contact any of us with your feedback on this lively topic. 

References

1. L. Bouchard, “Multimedia Software for Mobile Phones,” *IEEE Software*, May/June 2010, pp. 8–10
2. R. Murmura et al., “Mobile Application and Device Power Usage Measurements,” to be published in *Proc. 6th Int’l Conf. Security and Reliability Eng.*, IEEE, 2012.