Has Everything Been Invented? On Software Development and the Future of Apps

Alessio Malizia, Universidad Carlos III de Madrid, Spain
Kai A. Olsen, University of Bergen and Molde University College, Norway

Because we’re continually offered an abundance of new apps, we may fall into the trap of thinking that everything in software has been invented.

“Everything that can be invented has been invented.” Although this infamous 1899 quote has been attributed to Charles H. Duell, then director of the US Patent Office, he actually never said such a silly thing. What Duell said to the US Congress was something quite different, that America’s future success depends on invention (http://tinyurl.com/3pjas6u). This clearly remains the case today, not only for America but for most countries. Invention and innovation continue to prosper, not least in the software industry.

The past decade has given us new tools and new ways of disseminating applications. As a consequence, we may fall into the trap of thinking that everything in software has been invented.

SOFTWARE INNOVATION
Some time ago, we had a creative idea for a new app, a location-dependent “reminder.”

The idea was to attach a message to a location, instead of to the date and time. With this app, the next time you go near a hardware store, your smartphone would beep and tell you to get a new hammer, a message you may have entered some time ago. Or, when you read about a famous restaurant in Madrid, you store its homepage URL in the reminder. A year later, when you exit a train in the center of Madrid, the information about the restaurant will pop up on the phone.

To develop this app, we searched the Web for relevant information on APIs and so on, and started programming. Then we found that a similar app was already available for Google’s Android operating system. It just hadn’t been out when we first got the idea for our app.

Well, you can’t beat them all, so we initiated work on a new idea, another breakthrough app for smartphones. We developed an initial design, but then we found an iPhone app that did something similar. Very annoying!

A contributor to DanTech, a blog on tablet and smartphone innovations, relates a similar experience. “I have independently come up with several ideas last year, and all of them are either patented by large PC companies like Apple and Microsoft, or similar ideas have already appeared in new startups ...” (http://tinyurl.com/3jdbu4).

What has happened? Has everything that can be invented been invented?

INNOVATION MODELS
Traditionally, large companies have used a centralized approach to maintain control over innovation. Bell Labs is a good example. In “Bell Labs and Centralized Innovation,” Tim Wu describes how Bell Labs was a “scientific Valhalla” for researchers and engineers working there (Comm. ACM, May, 2011, pp. 31-33). They were free to pursue their own areas.
of interest and got the resources they needed to do so.

In companies such as AT&T, innovation was delegated to an internal research center, often the pride of the company. They were willing to invest heavily in the hope that at least some of the projects would be a success. But centralized innovation has some drawbacks, especially in getting the inventions to the marketplace. A good illustration is Xerox’s failure to exploit the GUI research at Xerox PARC.

The open source software movement is an interesting alternative. While the big companies adopt a monolithic approach, open source

**The sandbox environment offers tools, APIs, and other resources that simplify development.**

is based on a decentralized model. In this model, entrepreneurs with good ideas can set up a development project, then they invite the rest of the world to participate.

The motivation is to develop something in common, something innovative that is more flexible and better than what the private companies offer, and also something that’s free for anybody to use. Amazingly, many excellent products have come out of this movement, including OSs (Linux), Web servers (Apache), programming languages (PHP), and browsers (Firefox). Even the basic parts of the Internet can be attributed to open source development.

While the open source movement has demonstrated its efficiency in large systems and software modules, it might not be as effective for developing end user apps. The problem could lie in marketing—how to get apps to the customer.

In “Entrepreneurial Innovation at Google” (Computer, Apr. 2011, pp. 56-61), Alberto Savoia and Patrick Copeland describe another model for innovation. Google invests in an internal entrepreneurial model that encourages employees to innovate. The company, with its flat organizational structure, supports innovative employees by offering services, data resources, and tools. It offers free time for entrepreneurship and provides additional resources for the most interesting projects. Most importantly, through Google Labs and similar facilities, it offers a venue where new products can be tried in the marketplace.

**VIRTUAL GARAGES**

HP’s birth took place in a garage in 1959. Many other electronics, computer, and software companies had similar humble beginnings in which the entrepreneurs themselves put in the time and money required to get a company off the ground.

Today, we’ve moved from the physical to the virtual garage. A virtual garage is provided with extensive toolkits in the form of OSs, development packages, and open source code. In this way, device manufacturers have opened their systems up for third-party developers, giving them access to all the underlying functionality of the PC, tablet, or smartphone.

Thus, implementing our proposed reminder app is no big deal. An entrepreneur with a Java development tool has access to everything that’s needed to develop the app, including access to display, files, and location coordinates. In principle, there’s nothing new here. Apple invited third-party developers to contribute to the Macintosh nearly 30 years ago. Other manufacturers have also opened their systems for independent developers. What’s new is the support that’s offered for getting new apps to the end user.

**THE SANDBOX**

Traditionally, in engineering terms, a sandbox environment consists of a controlled set of resources for trying a new app without the risk of damaging critical parts of the system.

As an innovation model, we can take this further, including the phase of getting the product on the market. This is what Google offers its employees. We call this a “tightly coupled extended sandbox.” “Tightly coupled” because the entire process is internal to the company, and “extended” because the market is a part of the sandbox and entrepreneurs are taking advantage of the infrastructure already in place for the company itself.

For Apple, the sandbox innovation environment is the App Store, together with the developer network and all the tools that go with it. The App Store mechanism controls the quality of published apps and also provides a platform for reaching out to the end user. Moreover, Apple provides easy integration with iAds, an advertising framework that includes a payment mechanism for iPhone and iPad apps. This is a “loosely coupled sandbox”—Apple owns the infrastructure, but the entrepreneurs can be you or me or anyone else. The marketplace for Android apps offers similar services.

For all these models, the short time from idea, via implementation and quality control, to the market is crucial. This could explain why some feel that “everything has been invented.” Many of us might previously have had ideas for new apps, but they usually stay in the inventor’s head or never come out of the lab. Even when a new invention reaches the market, only a fraction of the potential user community notices most of them.

Today, with all the available tools, development is simpler, and nearly any app can be put on the market immediately. Marketing is an important factor. Through centralized channels, such as iAds or Android...
Market, it’s easy to find what we’re looking for. Thus, it’s not only a matter of the time to market, but also the time before the world at large knows what we’ve invented.

Clearly, the sandbox environment has much to offer developers. The tools, APIs, and other resources simplify the development. The marketing support and predefined tools for downloading and installation help with the last and most crucial part of the development process—reaching out to the customer.

The models also may offer an opportunity for generating revenue. But, if we exclude the few success stories, making money on apps might not be so easy in the long run. There are already half a million apps in the App Store and 200,000 on the Android market. Newcomers to this market could be forced to offer apps for free to be recognized, thus reducing the potential for revenue.

FROM THE USER’S PERSPECTIVE

This development environment also might not be sustainable in the long run from the user’s perspective.

For a given function, the user must choose from that overwhelming number of apps. For example, there are about 400 wakeup apps and 2,500 calendar apps in the App Store, with around the same number of similar apps on the Android market. Although the most popular apps are presented first, popularity might not be a very good indicator of quality.

After choosing an app, the user must agree to let it have access to phone resources. Few users will be able to evaluate the security risks, and most hit the okay button without thinking. Then the user must download the app, perhaps pay for it, and install it on the device. When moving to a new device, especially if the native OS is changed, the user must locate the apps, make selections, pay for them, and download them again.

On a particular smartphone or tablet, each of the various apps is represented as an icon in the apps window. It’s now up to the user to remember which app did what, locate the icon, and remember how to use it. Updating apps is the user’s task. Even if they’re developed under the same guidelines, the apps could have different user interfaces.

Donald Norman and Jakob Nielsen complain about “the misguided insistence by companies (e.g., Apple and Google) to ignore established conventions and establish ill-conceived new ones” (tinyurl.com/37yyrtv). When replacing an app, the user might therefore find that the new version offers a different user interface and different functionality.

Thus, while many enthusiastic users have welcomed the app idea, it might not be a good solution for the less technologically oriented. What might suit these users much better is a device in which all the necessary apps are embedded in the OS. We can expect this to happen in the long run.

THE FUTURE

The app market is arguably only a testbed for the major companies, an implementation of a digital ecosystem in which the apps can compete, using a survival-of-the-fittest scheme to find the most interesting functions and the most popular approaches to them.

By letting third-party developers create the initial versions, and offering them to the marketplace, the major companies have a very simple and profitable way to determine the functionality to include in the next version of the basic software. They can then use their resources to ensure that each of these apps, or each function, is offered in a high-quality version. As an example, the new iOS 5, Apple’s mobile OS, will clearly replace several of the more popular apps.

An even more competitive approach than including app functionality as a part of the native OSs is offering this functionality through browser-based systems (T. Mikkonen and A. Taivalsaari, “Reports of the Web’s Death Are Greatly Exaggerated,” Computer, May 2011, pp. 30-36). The advantages of centralized dynamic applications, true platform-independence, ubiquitous access, and no local installation or update of software could lead to the demise of both apps and large native OSs.

In the software world, will the browser, the dinosaur that can do everything, be the survivor?

Looking at a clear, dark, night sky, we see myriad stars. There are as yet not as many apps as there are stars, but each one might shine as brilliantly, showing the extent of human invention and covering every “dark spot,” every possible function. But in the long run, numbers and brilliance might not be enough. In our everyday lives, we seldom take the time to look at the stars. Instead we focus on functionality, practicality, and effectiveness. While the stars will always be there, we aren’t so sure that apps will survive in a practical world.

Alessio Malizia is an associate professor in the Computer Science Department at the Universidad Carlos III de Madrid, Spain. Contact him at alessio.malizia@uc3m.es.

Kai A. Olsen is a professor at the University of Bergen and Molde University College in Norway and an adjunct professor at the School of Information Services, University of Pittsburgh. Contact him at kai.olsen@himolde.no.

Editor: Neville Holmes, School of Computing and Information Systems, University of Tasmania; neville.holmes@utas.edu.au

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.