

(excerpts from: **Jonathan Zittrain The Future of the Internet—and How to Stop It.**

(http://www.technologyreview.com/printer_friendly_article.aspx?id=39163)

(Today) we're seeing an unprecedented shift of power from end users and software developers on the one hand, to operating system vendors on the other: The transformation is one from product to service.

(In the past) for decades we've enjoyed a simple way for people to create software and share or sell it to others. People bought general-purpose computers—PCs, including those that say Mac. Those computers came with operating systems that took care of the basics. Anyone could write and run software for an operating system, and up popped an endless assortment of spreadsheets, word processors, instant messengers, Web browsers, e-mail, and games. That software ranged from the sublime to the ridiculous to the dangerous—and there was no referee except the user's good taste and sense, with a little help from nearby nerds or antivirus software.

That was one reason we ended up with a single dominant OS for over two decades. People had Windows, which made software developers want to write for Windows, which made more people want to buy Windows, which made it even more appealing to software developers, and so on.

In the 1990s, both the U.S. and European governments went after Microsoft in a legendary and yet, today, easily forgettable antitrust battle. Their main complaint? That Microsoft had put a thumb on the scale in competition between its own Internet Explorer browser and its primary competitor, Netscape Navigator. Microsoft did this by telling PC makers that they had to ensure that Internet Explorer was ready and waiting on the user's Windows desktop when the user unpacked the computer and set it up, whether the PC makers wanted to or not. Netscape could still be prebundled with Windows, as far as Microsoft was concerned. Years of litigation and oceans of legal documents can thus be boiled down into an essential original sin: an OS maker had unduly favored its own applications.

(But) When the iPhone came out in 2007 (...) no outside code at all was allowed on the phone; all the software on it was Apple's. (...) In 2008, Apple announced a software development kit for the iPhone. Third-party developers would be welcome to write software for the phone, in just the way they'd done for years with Windows and Mac OS. With one epic exception: users could install software on a phone only if it was offered through Apple's iPhone App Store. Developers were to be accredited by Apple, and then each individual app was to be vetted.

The original sin behind the Microsoft case was made much worse (by Apple). The issue wasn't whether it would be possible to buy an iPhone without Apple's Safari browser. It was that no other browser would be permitted—or, if permitted, it would be only through Apple's ongoing sufferance. And every app sold for the iPhone would have 30 percent of its price (and later, that of its "in-app purchases") go to Apple. Famously proprietary Microsoft never dared to extract a tax on every piece of software written by others for Windows.

By late 2008, Google's Android Marketplace, created competition for the iPhone: Developers still registered in order to offer software through the Marketplace, but once they registered, they could put software up immediately, without review by Google. There was still a 30 percent tax on sales, and line-crossing apps could be retroactively pulled from the Marketplace. But there was and is a big safety valve: developers can simply give or sell their wares directly to Android handset owners without using the Marketplace at all. If they didn't like the Marketplace's policies, it didn't mean they had to forgo ever reaching Android users.

The most important reasons (why should we be worried at all) have to do with the snowballing replicability of the iPhone framework. The App Store model has boomeranged back to the PC. There's now an App Store for the Mac to match that of the iPhone and iPad, and it carries the same battery of restrictions.

The content restrictions are unexplored territory. At the height of Windows's market dominance,

Microsoft had no role in determining what software would and wouldn't run on its machines, much less whether the content inside that software was to be allowed to see the light of screen. Pulitzer Prize-winning editorial cartoonist Mark Fiore found his iPhone app rejected because it contained "content that ridicules public figures." Fiore was well-known enough that the rejection raised eyebrows, and Apple later reversed its decision. But the fact that apps must routinely face approval masks how extraordinary the situation is: tech companies are in the business of approving, one by one, the text, images, and sounds that we are permitted to find and experience on our most common portals to the networked world.

Governments have come to realize that this framework makes their own censorship vastly easier: what used to be a Sisyphean struggle to stanch the distribution of books, tracts, and then websites is becoming a few takedown notices to a handful of digital gatekeepers. Suddenly, objectionable content can be made to disappear by pressuring a technology company in the middle.

Given that outside apps can still run on a Mac and on Android, it's worth asking what makes the Stores and Marketplaces so dominant—compelling enough that developers are willing to run the gauntlet of approval and take a 30 percent hit on revenue instead of simply selling their apps directly.

The answer may lie in seemingly trivial places. Even one or two extra clicks can dissuade a user from consummating what he or she meant to do—a lesson emphasized in the Microsoft case, where the ready availability of IE on the desktop was seen as a signal advantage over users' having to download and install Netscape. The default is all-powerful, a notion confirmed by the value of deals to designate what search engine a browser will use when first installed. Such deals provided 97 percent of Firefox-maker Mozilla's revenue in 2010—\$121 million. The safety valve of "off-road" apps seems less helpful when people are steered so effortlessly to Stores and Marketplaces for their apps.

Today's developers are writing code with the notion not just of consumer acceptance, but also vendor acceptance.

The user gets put in the same situation: if I switch from iPhone to Android, I can't take my apps with me, and vice versa. And as content gets funneled through apps, it may mean I can't take my content, either—or, if I can, it's only because there's yet another gatekeeper like Amazon running an app on more than one platform, aggregating content. The potentially suffocating relationship with Apple or Google or Microsoft is freed only by a new suitor like Amazon, which is structurally positioned to do the same thing.

Innovation and communication was ignited by the rise of the PC and the Web and their generative characteristics. Software was installed one machine at a time, a relationship among myriad software makers and users. Sites could appear anywhere on the Web, a relationship among myriad webmasters and surfers. Now activity is clumping around a handful of portals: two or three OS makers that are in a position to manage all apps (and content within them) in an ongoing way, and a diminishing set of cloud hosting providers like Amazon that can provide the denial-of-service resistant places to put up a website or blog.

Both software developers and users should demand more. Developers should look for ways to reach their users unimpeded, through still-open platforms, or through pressure on the terms imposed by the closed ones. And users should be ready to try "off-roading" with the platforms that still allow. If we allow ourselves to be lulled into satisfaction with walled gardens, we'll miss out on innovations to which the gardeners object, and we'll set ourselves up for censorship of code and content that was previously impossible. We need some angry nerds.