

Things That Define Big Software Companies

Adrian Kosmaczewski

2022-02-11

Looking at the software industry, it appears that most big companies usually share more traits than they would like to admit. Take for example their products: at any given time, big software companies all had at least one product of various similar categories, roughly grouped in three big areas.

The companies fitting such a description are the usual suspects:

- Microsoft
- Apple
- Google
- Amazon
- Oracle
- IBM

Let's start with the foundational parts first.

- **Computers (duh!).** All of these companies sell their own hardware things running their own software inside. Yes, Amazon sells Kindles, IBM sells mainframes, and Oracle sells SPARC servers.
- **Operating systems.** Windows, z/OS, macOS, Solaris, Chrome OS, Amazon Linux, you name it. If you're a big company, you gotta have your own operating system, or a Linux distro with some added value to sell for big bucks to enterprise customers. Usually POSIX-compatible, whatever that is. You can even have many; nowadays Microsoft has two Linux distros of their own (one for Azure, another for WSL), and in the 1980s they had their own Unix. How about that. By the way, the rush for a "modern" operating system is exactly what almost made Apple bankrupt in the 1990s.
- **Office suites;** more or less compatible with the formats used by the biggest of all of them, Microsoft Office, even if lately the OpenDocument standard is making inroads. Interestingly, this format family originally debuted in what became Oracle's own StarOffice. Lately these office suites been transpiled to JavaScript or even more lately Wasm, like Google Docs, Office 365, and Apple iWork, for example. They all suck and developers went back to pure text, usually in the form of Markdown or AsciiDoc to get actual work done. And if all else fails, there's PDF.

- **Web browsers;** previously, companies would actually work on their own HTML & JavaScript engines, but these days even Microsoft uses Chromium. New browsers are great until you use them for real. I stick with Firefox or LibreWolf, thankyousomuch. Even Amazon have (had?) their own browser and lots of products with the “Fire” moniker, fitting many of the categories above and below. Even IBM used to have its own web browser for another of its operating systems, that marvel of the 90’s called OS/2. *Le sigh.*

The second area of interest are developer tools.

- **Cloud platforms or “IaaS”;** everybody has their own these days, starting with the big three, and behind them Oracle and IBM. As for Apple’s iCloud, more an SaaS than anything, it offers some shy PaaS features for developers to use in mobile apps. With one billion users, it’s hard not to count iCloud in this list.
- **Programming languages;** always proprietary, more or less cross-platform, and lately very open source: C#, Swift, Go, Java, COBOL. To gain the heart of all developers, they are all multi-paradigm with both lambdas and objects, so you can do functional and OOP all in the same app. They can all (sorta) do desktop, backend, frontend, cloud, mobile, and Hello World. They are all bloated in their own special ways and people love and hate them in various proportions. And Amazon supports them all.
- **IDEs,** usually serving as a support and glue for all the other items in this section. Visual Studio remains a strong contender, and will celebrate its 25th birthday on March 17th. On the other side, Apple should phase out that bad joke of Xcode, and instead ask somebody who actually knows a thing or two about IDEs to help, like Google did when it asked JetBrains to create a new Android Studio. We can mention Eclipse, which serves as the basis of whatever IBM offers, and NetBeans, which serves Oracle well. And Amazon has Cloud9.
- **Self-sufficient backend programming systems:** COBOL, ASP.NET, Amazon Lambda, J2EE... Even Apple had one, a long time ago, called WebObjects. If they wanted they could make a bigger PaaS out of iCloud, and use Swift all over their datacenters. They could; if they don’t, it’s not because a lack of cash. On the other hand, this is a domain where Google particularly shines: they started with GWT and later begat Kubernetes, the platform to rule them all. They also moved to the frontend with Angular, by the way. Even more: their work on the V8 JavaScript engine gave origin to Node.js, arguably the most important web technology of the 2010s.
- **Database engines.** That was mostly a 1990s race, though; these days there’s too many too good open source database engines out there, in the FOSS arena, to justify the craziness of coming up with a new one. But for a while, around 1990, there was not a lot of choice and the war was raging. DB/2, SQL Server, lots of Amazon purpose-built databases,

just like Google's, and the eponymous RDBMS that made Larry Ellison a billionaire. Even Apple has something that looks like a PaaS database. And let's also not forget that FileMaker is made by a subsidiary of Apple.

Finally, we have the consumer space. Only Microsoft, Apple, Amazon, and Google are active in this area. I don't expect to see an Oracle smartwatch or an IBM music store anytime soon, if you see what I mean, but I've often been wrong in the past.

- **Game consoles:** that was the craze at the end of the 1990s, when Apple thought the Pippin was a good idea. Actually the iPhone and Android became much bigger mobile gaming platforms than what their creators thought they would be. Gaming is a category where Microsoft actually did something relevant and quite ground breaking.
- **Smartphone platforms;** that's a 2000s fashion, and one category that Microsoft lost completely. Now it's over, Apple and Google are a *de facto* duopoly now. Amazon wanted to be in this category too, didn't work, in spite of all the Fire.
- **Retail and payments.** The big thing these days. Many of these "software" companies are into retail of mostly digital goods (usually apps, games, books, and music), with the obvious exception of Amazon, who also sells and ships physical goods. And if you sell stuff, you might as well bypass payment companies and offer your own solution there, too, like Apple and Google, based on the strong support of their smartphone businesses.

Upcoming battlefields:

- No Code / Low Code platforms (Amazon, Google and Microsoft are already present in that space)
- Quantum computing (IBM is investing heavily in it)
- Wearable computing (Apple and Google are there)
- Healthcare, fitness, wellness (Apple is investing a lot in the field)

Here's the gist: software companies became big *because* all of these bricks generate a positive feedback effect as they reinforce one another, orbiting around the needs of somewhat orthogonal groups of users.

Playing the **platform game** is an expensive, but hugely lucrative game in the long run. Yes, it is expensive; to reach a billion users, you have to literally spend tens of billions of your favorite non-devalued currency.

Hewlett-Packard, Xerox, and countless other technology companies could have been part of this group, had they embraced software as an industry and a potential, and not as an afterthought. Facebook could count but they are more of a media company than anything else.

Maybe the Alibaba Group or the Tata Group could follow these steps? Maybe they already do to a large degree.

On the other hand, here's things people thought all big software companies could/should/had to do, but outsiders did them much better:

- Chat.
- Search engines.
- Social media. Apple and Google, I'm looking at both of you.