

The Consumerization of Software and How it Will Never be the Same

In a marketplace that's rapidly demanding convergence, and the best of all possible worlds, it's up to today's software vendors to answer the call.

Back when pocket protectors were both functional and fashionable (an engineer can dream, right?), engineering software and consumer software were radically different—not just because of the people who used them but also because of the inherently unique user experiences they provided.

But what happens when an unstoppable force (consumer software) meets an immovable object (engineering software)? In a marketplace that's rapidly demanding convergence and the best of all possible worlds, it's up to today's software vendors to answer the call.

The Evolution of Consumer Software

From right-click functionality to Microsoft's introduction of the ribbon, consumer software has long defined the user experience standards and expectations that most vendors follow. But with the recent explosion of beauty and simplicity across the general consumer market, expectations have changed.

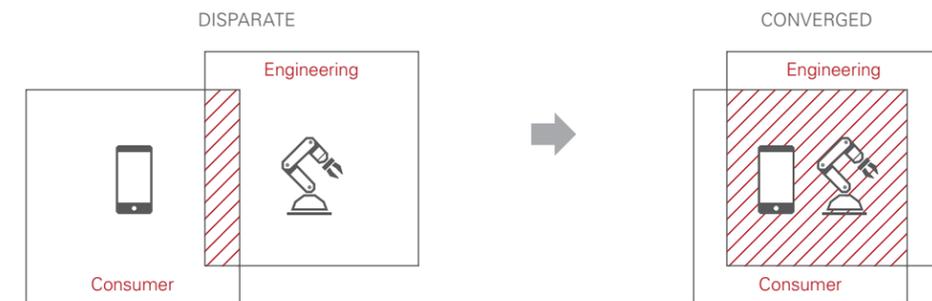
Consider the iPhone, first introduced in 2007 and unanimously hailed as an inflection point in the software

market for its seamless integration of gestures, single-button interfaces, aesthetically pleasing graphics, and fluidly simple community and purchasing experiences. Its sleek, modern graphics introduced animations, fades, and zooms that replaced the floating windows and unorganized tiling of traditional single screens. Its connected community and purchasing experience exposed a new world of extensibility by providing an interface to additional functionality not offered by the primary vendor. And its handprint on society became so indelible that even our youngest generations walk up to desktop monitors and expect to interact with it at a tactile level.

The Evolution of Demands on Engineering Software

Engineering software users of the past typically graduated from a university with an understanding of one programming language. Some were even experienced users who demanded exposure to the darkest corners of programming, with custom memory calls, from-scratch multithreading commands, and hand-optimized performance demands. Software was

THE CONVERGENCE OF ENGINEERING SOFTWARE AND CONSUMER SOFTWARE



hard and unapproachable to those who dared to enter without proper training and previous exposure.

And just like that, things started to change. First, graduating engineers were required to have a broad skillset of programming languages to tackle the challenges handed to them in the workplace. Like a trained warrior switching from sword to axe to bow, today's engineer can jump in and out of Python, C#, HTML, JavaScript, LabVIEW, and Swift. This puts an unprecedented demand on approachability and removes the expectation of expertise. Today's engineer expects to use multiple tools for any given application.

Secondly, the cost of accessing and acquiring data has rapidly decreased while the need for data has risen. And as technology has become more connected, the cost of processors has declined. According to DataBeans, the price of a processor decreased over 30 percent between 2011 and 2015. This has accelerated the need for highly approachable software by introducing more "nontraditional" programmers to the worlds of robotics, home automation, and even general data acquisition and analysis. Likewise, cultural trends like the Maker Movement and the emergence of consumer product start-ups being acquired for unreal amounts of money further illustrate this shift.

An Inevitable Convergence

For engineers, who are defined by the pride of conquering complex challenges, this confluence of usability and technical sophistication couldn't come at a better time. No longer tethered and bogged down by the intricate details of multiple languages, tools, and approaches (like writing an actor framework), they can now refocus on engineering's most grand and impactful challenges (like 5G research and the IoT). In this new tightly integrated future, engineers can find better, faster ways from point A to point B instead of spending their time making better maps.

Likewise, this convergence means that engineers can embrace a future in which they aren't sole proprietors of innovation. With software that is (gasp) easy to use, the rest of the world is catching up. And by acknowledging the prevalence of simplicity and beauty in software of all stripes, more and more smart people will play meaningful roles in major problem solving.

There's no denying that our world is getting more complicated and that our challenges are becoming harder to solve. Being asked to do more with less money, fewer people, and less time only heightens the tension. The sooner we embrace the convergence of engineering software and consumer software, the sooner we'll realize that "simple" is the best way to solve "complex."

"The same concepts that are driving the Maker Movement have rejuvenated engineering. Rather than working with complex protocols that only engineers can understand, today's software must provide an interface that's approachable and understandable to non-engineers on a conceptual level."

—Bob O'Donnell, TECHanalysis Research Chief Analyst