Professional ViewPointz

While Nathan Slept, What Rich, Powerful Technologists Know About Their Customers

By

Charles L. Mauro
President
Mauro New Media

While Nathan slept, what rich, powerful technologists know about their customers.

Thomas J. Watson Sr., founder of IBM, once said, “I think there is a world market for maybe five computers.” No one denies Watson was a very smart man, the very essence of the modern capitalist, but his projected demand for computers was down right foolish. Sometimes rich, powerful technologists reveal staggering levels of stupidity when it comes to understanding the implications of their technology or, more frequently, the needs of their customers. Chairman Watson’s statement has now been officially pushed from the pedestal. Thank you, Nathan Myhrvold. It was a long time coming.

Technology Review (TR), the bi-monthly publication of MIT, is a terrific, pro-technology magazine. In columns that patently promote the work of MIT engineering departments, TR often stretches the envelope of what technology can and cannot do now and what technology will and will not do in the future. Occasionally TR strays from its standard fare and commissions a piece on the more “human” aspects of technology. These articles frequently produce amazing insights into how the world’s elite technologists think about those untidy little black boxes called “customers.”

The big party

When the July/August 2002 issue of TR slid from my pile of unread magazines I noticed immediately the ironic cover titled: “Why Software Is So Bad?” What could TR have to say about why software is so bad? This is no small question when considered in the context that a significant percentage of the world’s software is poked and prodded by graduates of MIT. I remember well the three absolutely critical requirements for obtaining VC funding during the Internet boom. It was a checklist with three big boxes which required: 1) a Harvard MBA for CEO, 2) a Stanford MBA to head Marketing and 3) an MIT Ph.D. in charge of Technology. It was assumed that world-class business models (really world-class software) were “baked” into such a mix. It’s no mystery that even if the boxes were all checked that robust, usable and profitable software-based

products and services never came out of the oven. We now know that the Internet boom was really a huge amateur software development party that rarely included a realistic understanding of the cognitive or motivational needs of those little black boxes called customers. What does this have to do with Nathan Myhrvold?

Getting to the point

The article "Why software is so bad and what to do about it," by Charles C. Mann, is an essentially straightforward exposition of what is already known about why software is unreliable and overly complex. If one is prone to sprinkling PowerPoint presentations with quotes on bad software design, the article is a virtual treasure trove. My favorite: "*Incredibly, software projects often devote 80 percent of their budgets to repairing flaws they themselves produce."* This is not the reason, however, why I found the TR article so amazing.

John Benditt, TR's editor-in-chief, has a feeling for the dramatic, even in arcane subjects. He always seems to save the best for last. And it is the closing statement made by Nathan Myhrvold, the Gulfstream-owning, Princeton educated, Ph.D.-wielding, ex-Microsoft Research-directing, Master of the Universe that stopped the clock. Here is what Dr. Myhrvold has to say about why software is so bad:

""The classic dilemma in software is that people continually want more and more and more stuff," says Nathan Myhrvold, former Chief Technology Officer of Microsoft. "Unfortunately", he notes, "the constant demand for novelty means that software is always "in the bleeding edge phase," when products are inherently less reliable." In 1983 he says, Microsoft Word had only 27,000 lines of code. "Trouble is, it didn't do very much"—which customers today wouldn't accept. If Microsoft had not kept pumping up Word with new features, the product would no longer exist."

"Users are tremendously non-self-aware," Myhrvold adds. At Microsoft, he says, corporate customers often demand that the company simultaneously add new features and stop adding new features. "Literally, I have heard it in the same breath, a single sentence. "Were not sure why we should up upgrade to this new release-it has all the stuff we don't want-and when are you going to put in these three things?" And you say, "Whaaat?". Myhrvold's sardonic summary: Software sucks because users demand it to.""

Well, there we have it. For the first time, Nathan Myhrvold gives us the real answer on why software is so bad. Thank you John Benditt and TR. This quote is a home run.

Clueless or maybe not?

We have in Dr. Myhrvold a man who probably controlled more programming resources than anyone in history. Yet in the end, he cannot see even the simplest point behind customer needs and software development. His customers were not asking for new features. They were asking for the right features.

It has always been curious to me that software has proceeded along its own trajectory with respect to product development methodology. In a very real sense, software has ignored some of the most fundamental rules of product development science. The first rule: never ask the customer to design your product. Product development experts identified the weakness in Dr. Myhvold's approach decades ago with the introduction of the Ford Edsel and other disasters of early market research. In almost any industry other than software, an executive employing Dr. Myhrvold's methodology would be considered to be asleep at the wheel. It seems as if basic management science was not part of Dr. Myhrvold's development model. Maybe he had a better model, but I can't find it. Over the past 25 years I have been involved in many mission-critical product development projects and conducted usability tests on hundreds of products, including a good amount of software. I have never seen "users tremendously non-self-aware." On the

other hand, I have seen many rich and powerful development executives “clueless” about why users cannot operate their products easily and without frustration. This leads me to the first paradox of software development: **users are not software designers, nor are software engineers users**. Dr. Myhrvold’s statements reveal a staggering lack of perspective in terms of what is important in software engineering and “why software is so bad.” He and many other technologists see software as an engineering problem. Ask the customer what they want and figure out how to build it. From an engineering point of view, this makes perfect sense. After all, if there are no new features, what is there to engineer?

Quick and dirty is not cheap

Asking the customer to design your product (the technical equivalent to asking them what features they want) is a quick and dirty method for defining the functional profile for software-based products. Clearly Microsoft spends staggering amounts of cash on software development, but where does the money go? The answer is it goes into stuffing fewer and fewer needed features into increasingly bloated software. One can only imagine the real complexity of this process, given the need to maintain dominance of the Windows Operating system.

Dr. Myhrvold’s has left us other brilliant insights about software development. In Wayt Gibbs’s 1997 article in Scientific American Magazine “Taking Computers to Task,” Dr. Myhrvold proffered his famous “Nathan’s First Law”: that says“ Software is a gas. It expands to fill its container.” In fact what he’s describing is more a misguided (?) policy than a natural law. “After all,” he observed later with a laugh, “if we hadn’t brought your processor to its knees, why else would you get a new one?” One can only imagine the implications of this “law” for companies like Intel, Dell and others whose upgrade cycle is tightly bound to Microsoft’s software functionality. When taken in this context, the real cost of Dr. Myhrvold's feature-driven development model for his users is staggering.

Having the last WORD

Dr. Myhrvold's point that Microsoft Word would cease to exist without the pounding and stuffing of new features into every upgrade reflects the market realities facing Microsoft. It does not reflect the functional needs of the customer. When one looks back at prior versions of MS Word, it is clear that even early versions possessed about 90% of the features that customers need for core task execution. Move to Word XP and those same core functions are present and used to execute the same core tasks in about the same relative percentage. So where is the real progress? It is at the fringes of functionality, not in the real meat of task execution. True enough, later generations of software added some important features but at the heart of core task execution, word-processing remains essentially unchanged. Think this is probably not what Dr. Myhrvold wants to hear? It is unlikely that his stock options would have made him rich beyond reason if Microsoft had not adopted a feature-centered development model. These insights are not lost on rich, powerful technologists. What Dr. Myhrvold's statement really reflects is the issue that keeps Bill Gates up at night (if anything keeps him up at night): what happens when new features no longer sell software? This is the real source of feature propagation in Microsoft Word and many other pieces of software, not only those produced by Microsoft. Software does not suck because users demand it to. It sucks because rich, powerful technologists demand it from their engineering teams.

The future is not what it used to be

In the coming century, software development will not be an engineering problem. It will be a cognitive engineering problem. Innovation based on customer-defined feature propagation eliminates the responsibility of the development team for **user-centered** product innovation. This means that software must be designed first, and then engineered. Not to put too fine a point on this, but this distinction was lost on Dr. Myhrvold. This is the new work for the future of software, and it is not easy work. What does this mean as we face yet more costly upgrades, aggressive licensing agreements, and finally more complex and generally unreliable software-based products and services?

Train tracks and boxcars

Eventually feature-dominated innovation leads to a crushing end. The United States Military found this out in the 1960s with weapons systems that rapidly became far too complex to operate and maintain. The only way for Microsoft to keep this feature-centered methodology profitable is to eliminate competition from core areas of software functionality. Shall I mention a few? Word-processing, spreadsheets, databases, presentation software, and finally, by maintaining control of the railroad tracks themselves, the operating system. If one controls the tracks, what goes in the boxcars is just more stuff, and, after all, who really cares what goes in the boxcars? We all have to buy (actually lease) the same boxcars, no matter that they are stuffed with junk we do not want or need just so we can get at the core functions we do need. Under this approach, you can ask as many customers as you like what they want in the boxcars. It keeps software engineers creating software that is increasingly complex and unreliable. Thank you, Dr. Myhrvold for your insight. And thank you TR for bringing it to light.

6/25/2002

Charles L. Mauro
President
MauroNewMedia.

References

Why Software Is So Bad; Technology Review, July/August 2002:
Charles C, Mann, Pages 33-38.
https://www.technologyreview.com/articles/mann0702.asp

Taking Computers to Task; Scientific American Magazine, July 1997; by Wayt Gibbs, Pages 82-89.
https://www.sciamarchive.com/html/ppv_frames.asp

About the author

Charles L. Mauro, President, MauroNewMedia, New York, New York

Mr. Mauro has received many citations and awards for product development and usability engineering including the Alexander C. Williams Award from the Human Factors and Ergonomics Society and citations from NASA and Association of Computing Machines (ACM). He has served on numerous national and international design review panels and has chaired 2 ANSI standards committees. Mr. Mauro also served on the Presidential Design Awards Program for the NEA and was a founding member of the Human Factors Society Special Interest Group on Consumer Products. Prior for founding MauroNewMedia in 1975 he held senior product design positions for Henry Dreyfus Associates and Raymond Loewy International. Mr. Mauro holds a BS in Industrial Design from Art Center College of Design and a Master's Degree in Human Factors Engineering from New York University. At NYU he was a NIOSH research fellow at the Rusk Institute of Rehabilitation medicine. He has received grants and fellowships from the Ford Foundation, National Institute Occupational Safety and Health, and The National Endowment for the Arts.

Over his 25 year career Mr. Mauro has managed many large-scale complex product development programs. He is responsible for software user interface solutions currently running at the heart of the world economy. Mr. Mauro is best known for integrating professional usability engineering science with user interface design in systems that demand documented usability performance. He consults on a regular basis with leading international corporations and startups on issues of product development, usability and user interface design.

Mr. Mauro has been accepted in federal court as an expert witness in design patents, trade dress, usability and other intellectual property issues related to user interface design. He holds numerous US and international patents. Mr. Mauro lectures at leading Graduate programs including: MIT Sloan School, Stanford, and other MBA programs.

About Taskz. com and MauroNewMedia

Taskz.com is a public interest site designed and maintained by MauroNewMedia. It is dedicated to increasing executive awareness of the benefits resulting from application of formal user-centered design methods in the solution of complex business and social problems. Founded in 1975 MauroNewMedia is a leading provider of professional usability research and high performance user interface design for traditional and screen-based products and services. For more information on MauroNewMedia please visit our web site at http://www.mauronewmedia.com