Don’t Neglect Usability in the Total Cost of Ownership

Pearl Brereton neglected to include product usability in the software procurement paradigm, as she described it in “The Software Customer/Supplier Relationship” (Feb. 2004). Whether organizations are buying large amounts of commercial off-the-shelf, custom, or component software, all are concerned about productivity and total cost of ownership. But software product selection is multidimensional, including functionality, price, maintainability, reliability, architectural compliance, and usability. Though poor usability is a major factor in the total cost of ownership, it is typically not even viewed as a cost. Consumer organizations have virtually no understanding of a product’s usability before they buy it because software vendors don’t make the information available.

Usability problems manifest as uncontrolled overhead. Overhead costs increase significantly when end users find tools confusing, time-consuming, error-prone, inconsistent, require excessive training, or discourage exploration. Each of these factors undermines business benefits and the expected return on investment.

Supplier and consumer organizations alike have much to gain from peering into the black box of usability. To lend some guidance, the National Institute of Standards and Technology (NIST) Industry Usability Reporting (IUSR) project is developing a method to assist them in using usability data in software-procurement decisions.

The project aims to increase the visibility of software usability by encouraging software suppliers and consumer organizations to work together to understand user needs and develop a common usability reporting format for sharing usability data before users make their decisions.

The ANSI/INCITS-354 Common Industry Format (CIF) for Usability Test Reports is a standard method for reporting usability test findings developed by the project and coordinated by NIST to institutionalize the use of technical usability data for comparing products from different vendors. The format is a U.S. standard and is being submitted to the International Standards Organization. It specifies the format for reporting the results of usability evaluations and provides metrics for how usable a product is in a particular context of use. Using CIF indicates that good usability practices are being followed, that the results are reproducible, and that specific effectiveness, efficiency, and satisfaction metrics have been provided.

Industry Usability Reporting Project Steering Committee,
National Institute of Standards and Technology
Gaithersburg, MD

Where’s the Steak?

Peter J. Denning’s “The Profession of IT” column (“The Social Life of Innovation,” Apr. 2004) is the most pessimistic thing I’ve read in a long time. I can’t rebut what he says but, oh, how depressing.

It has long been obvious that schmoozing, advertising, and even dumb luck can accelerate pure hype to the speed of light and sell it to the masses in bulk, as much with software as with any other product.

But Denning tells us the converse is also true, that not only is promotion sufficient for selling vacuous ideas, it is necessary for selling even good ones. Good inventions therefore have no hope of adoption without the same techniques that successfully sell junk products and ideas. Nothing is judged on its merits. It’s all sizzle and no steak.

Poor wretches like me who

PAUL WATSON

value solving real problems to high standards are thus relegated to either being ignored or having to abandon what we value and become hypesters. If Denning's cynical analysis is reality, then, please, somebody help me repair my edifice of denial.

Rodney Bates
Wichita, KS

Author Responds:

There is no doubt that salesmanship has moved a lot of snake oil over the years. And that many healing remedies never saw the light of day because their inventors had no clue how to demonstrate their value. This is the way the world works. Impact does not happen spontaneously; someone must supply the energy. Those who spend no energy getting their inventions into practice or recruiting others to do it have no right to expect their inventions to generate that impact. It might happen anyway, but if it does, it's pure luck.

Am I cynical? The dictionary says cynicism is "believing the worst of human nature and motives; having a sneering sense of disbelief." Bates demonstrates this well. I am optimistic about human nature and offer a path for those who want their inventions and ideas to have impact.

Peter J. Denning
Monterey, CA

Discourage the Sale of Credentials in Anonymous Employment

John Gerdes, Jr. did not mention how to prohibit the selling of credentials in his "Technical Opinion" column ("The Viability of Supporting Anonymous Employees," Apr. 2004). Incorporating the credential owner's public key into the credential is not enough; the owner must be discouraged from selling the corresponding secret key together with the credential. This is partially achieved by requiring the secret key be extremely valuable to the credential owner. But, even then, the owner might give the key to someone he or she trusts not to use it for anything besides pretending to own the credential.

Lennart Meier
Zürich, Switzerland

For Voice Interfaces, Hold the SALT

Li Deng's and Xuedong Huang's "Challenges in Adopting Speech Recognition" (Jan. 2004) misrepresented the state of voice dialogue language standards. One recently established example is Speech Application Language Tags, or SALT, which extends existing Web markup languages to enable multimodal (speech plus other modalities) and telephony (speech-only) access to the Web. It is thus superior to the earlier standard—VoiceXML, a programming language supporting only telephony interaction for call-center speech applications.

The World Wide Web Consortium (W3C) advanced VoiceXML 2.0 to the status of a Recommendation (Mar. 2004) and is working on VoiceXML 3.0. Though SALT was submitted to the W3C, it is not on the standards track.

Meanwhile VoiceXML has huge industry momentum. Thousands of commercial VoiceXML applications worldwide run on platforms from nearly 100 vendors. VoiceXML applications serve all industries, not just call centers, and scale up to the massive North American 1-800-555-1212 business-directory service. There are at most a handful of commercial SALT applications, and four SALT Forum companies recently joined the board of the VoiceXML Forum, while, 73% of the 59 SALT Forum companies making recent commercial announcements have invested substantially in VoiceXML.

VoiceXML is suitable for multimodal interactions. The X+V language (see www.w3.org/TR/ xhtml+voice) uses standard W3C mechanisms to compose XTHML and VoiceXML into a multimodal markup language. X+V is based on the elegant model-view-controller paradigm, making it well suited to a broad range of architectures and the seamless inclusion of other modalities, including pen input.

SALT bundles everything into a mass of complex JavaScript. SALT's heavy dependence on JavaScript makes even small examples from the SALT 1.0 specification up to five times larger than the corresponding VoiceXML (see www.voicexml.org/faqs.html). SALT developers need to worry about low-level concerns (such as "hanging" dialogues, as in SALT 1.0, Section 2.6.5) that can't happen in VoiceXML. All of SALT's extra programming increases development time and cost, even when authored indirectly through such tools as Java servlets and Java Server Pages.

VoiceXML's wide acceptance as a standard, huge industry uptake, suitability for multimodal interaction, and increased developer productivity clearly demonstrate its superiority.

Jim Farrans
Piscataway, NJ

Authors Respond:
Our response consists of four points. First, the drawbacks of VoiceXML 2.0 are well known, some already reflected in the VoiceXML modularization in X+V. The urgency to remedy these drawbacks prompted W3C to begin its VoiceXML 3.0 effort based on the modality interface document (MID) embracing SALT's design principles. Because both MID and SALT enable object-oriented programming, this type of language design is more mainstream and superior.

Second, VoiceXML 2.0 is narrowly designed for telephony applications, but SALT demonstrates that a well-designed speech interface can do more. While VoiceXML needs major modifications to work with XHTML, SALT can encompass telephony, as well as multimodal applications, not only in XHTML but in SVG, SMIL, and any other XML application. VoiceXML cannot match this extensibility.

Third, just because VoiceXML arrived years earlier than SALT and hence has wide industrial recognition does not mean VoiceXML is a technically superior approach.

And finally, viewing the SALT model as heavily dependent on JavaScript is a common misunderstanding in the VoiceXML Forum; for detailed clarification, see Sec. 2.6 and 2.8 of the SALT 1.0 specification and the SALT SVG profile contributed to W3C in June 2002.

Li Deng and Xuedong Huang
Redmond, WA

Look Deeper for Markup Roots

I found the title "Tracing the Roots of Markup Languages" of the article by Rishi Toshniwal and Dharma P. Agrawal (May 2004) misleading and the brief article itself disappointing. As a former software developer in the typesetting industry, I can say the article did not begin to tell the whole story of markup languages or to get at their real roots.

The authors took the story back no further than the development of SGML in the late 1980s and gave scant attention to this important work. Manual typographic markup systems are almost as old as printing itself. Markup languages for computer-generated text documents are generally acknowledged as having their origins in two formatting systems from the 1960s—RUNOFF for MIT's Compatible Timesharing System, and to a lesser extent PAGE-1 for the IBM S/360. These systems were aimed at documents printed on early computer line printers. At the same time, proprietary markup languages were evolving for the first generation of computerized phototypesetters (such as those from Compugraphics and Dymo Graphic Systems).

Many RUNOFF derivatives permeated the minicomputer world in the 1970s. That influence is still seen in the Unix utilities roff, nroff, and troff.

The early 1980s saw major developments in formatting languages for increasingly sophisticated output devices, led by Donald Knuth's TeX and Brian Reid's SCRIBE. In 1982, I published a proposal for yet another markup system—MFS: A Modular Text Formatting System—referencing several of these systems, along with a brief history. MFS sought to combine the concepts of a markup language and what is today called a "page description language." This role is now filled by the Adobe Postscript and Portable Document Format (PDF) languages, both far too complex for human creation or editing.

Interestingly, many of the early systems, especially SCRIBE, involved the idea of separating markup based on logical structure or meaning from markup defining a specific appearance. To those experienced in such systems, the central ideas of XML seem obvious and not at all new.

Toshniwal and Agrawal provided an interesting but brief summary of only the most recent Web-centric markup systems. A full history would be a valuable project that is yet to be produced.

James D. Mooney
Morgantown, WV

Please address all Forum correspondence to the Editor, *Communications,* 1515 Broadway, New York, NY 10036; email: crawfordd@acm.org.