www.computer.org/intelligent

The Procurement Woes Revisited

Kelly Neville, Embry-Riddle Aeronautical University
Robert R. Hoffman, Institute for Human and Machine Cognition
Charlotte Linde, NASA Ames Research Center
William C. Elm, Resilient Cognitive Solutions
Jennifer Fowlkes, CHI Systems

Vol. 23, No. 1
January/February 2008

The Procurement Woes Revisited

Kelly Neville, *Embry-Riddle Aeronautical University*
Robert R. Hoffman, *Institute for Human and Machine Cognition*
Charlotte Linde, *NASA Ames Research Center*
William C. Elm, *Resilient Cognitive Solutions*
Jennifer Fowlkes, *CHI Systems*

Our civilization needs to heal the wound between its social and the technical-scientific world views. —Joseph Goguen

The set of people who are frustrated every day by badly designed information technology is very large.[1] So is the set of people whose dollars pay for the badly designed technology. A conservative estimate ranges in the billions for the cost of large-scale information systems that end up collecting dust because they're not properly human-centered. Yes, billions and still counting—that's the scary part.

Within this large set of frustrated customers (see the sidebar "When Systems Development Neglects Human Considerations") is a subset whose job it is to do something about this situation. That subset includes policymakers, program managers, and systems engineers. It also includes a sub-subset comprising cognitive systems engineers, ethnographers, and many others who, in one vernacular or another, advocate human-centered computing. We must show that intelligent technologies—those designed to interact with humans or play a role in the cognitive work conducted in sociotechnical work systems—are usable, useful, and understandable.

Procurement woes

A review of government documents covering standards and requirements[2] shows that cost is always the horse that's pulling the cart: "The DoD components shall, as part of programs such as Human Systems Integration, minimize system support costs by addressing manpower affordability early in the acquisition process."[3] This emphasis on cost is understandable, and arguably necessary, but the way it's stated means that worker needs can always get jettisoned at the first sign of trouble. As the Penny Foolish Principle states, the true "human costs" always show up further along in development after human-centering considerations have been sacrificed.[4,5]

Other human-centering issues are more subtle. For instance, the DoD requires that software use necessitate minimal effort on the user's part: "Design-induced requirements for operator workload, accuracy, time constraint, mental processing, and communication shall not exceed operator capabilities."[6] Actually, this doesn't make good sense. We know from expertise studies that people achieve high levels of proficiency only after long hours of working hard, on hard problems. Software designed to always minimize difficulty, and not serve as Janus Machines[7] that support tough task training, wouldn't help workers progress along the path to expertise.

In a previous essay in this department,[8] two of us (Hoffman and Elm) discussed John MacNamara's notion of "desirements" as a way to shift terminology away from entrenched tradition and think about requirements differently. We especially challenged the notion that "requirements creep" is a nasty thing to be avoided or somehow done away with. We also suggested that a partial fix would be to train individuals versed in both systems engineering and cognitive systems engineering to be the future managers of large-scale procurements.

Joseph Goguen (1941–2006)

As we've learned, computer scientist Joseph Goguen anticipated these ideas and other HCC notions. He saw the challenge of anticipating and addressing technology's human impacts as tractable as well as ethically, methodologically, and economically necessary. This essay is a homage to him, so that we might highlight one key point: If the procurement woes that we've boldly offered in this HCC department were in fact anticipated a decade or more ago, then the situation now must *really* be bad. We find in Goguen's writings many discussion points and topics for elaboration. Our starting and ending points differ somewhat from his, but we have many interesting commonalities. If this essay stimulates any significant discussion, it will have done its job.

Throughout his distinguished career, Goguen worked to

Editors: Robert R. Hoffman, Patrick J. Hayes, and Kenneth M. Ford
Institute for Human and Machine Cognition
rhoffman@ihmc.us

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

When Systems Development Neglects Human Considerations

The *Wall Street Journal* reported that 50 percent of software projects fail to meet CEO expectations and 42 percent of corporate information technology projects are discontinued before completion.[1] A 1995 US Department of Defense study estimated that 46 percent of DoD-funded IT development efforts result in products that are delivered but not successfully used and 29 percent never even produce a product.[2] These statistics translate into workers who lose out because they don't have the technology they need to perform their work effectively, not to mention the billions of dollars squandered. For example, the US Internal Revenue Service spent $4 billion on a decision support system that, in the words of an IRS official, does "not work in the real world,"[3] and the US Federal Bureau of Investigation spent $170 million on a problem-riddled software development effort before abandoning it.[4] Other well-known system development disasters include the London emergency dispatch system released in 1993[5] and the US air traffic control system upgrade.[6] Each of these examples points to the neglect of human considerations during the development of sociotechnical systems.

References

1. P.E. Coyle, "Simulation Based Acquisition for Information Technology," paper presented at the *Academia, Industry, Government Crosstalk Conf.*, 1999; www.dote.osd.mil/presentations/Coyle051899/tsld001.htm
2. T.R. Leishman and D.A. Cook, "Requirements Risks Can Drown Software Projects," *Crosstalk*, vol. 15, 2002, pp. 4–8, www.stsc.hill.af.mil/crosstalk.
3. S. Gardner, *Marketplace*, news archives, Nat'l Public Radio, 31 Jan. 1997; http://marketplace.publicradio.org/shows/1997/01/31_mpp.html.
4. D. Eggen, "FBI Pushed Ahead with Troubled Software," *Washington Post*, 6 June 2005.
5. A. Finkelstein and J. Dowell, "A Comedy of Errors: The London Ambulance Service Case Study," *Proc. 8th Int'l Workshop Software Specification and Design* (IWSSD), IEEE CS Press, 1996, pp. 2–4.
6. D.F. Carr and E. Cone, "Can FAA Salvage Its IT Disaster?" *Baseline*, 8 Apr. 2002, www.baselinemag.com/article2/0,1540,656862,00.asp.

build bridges between rigorous formal systems and the messy, creative ways in which human minds and social groups actually operate. His early work was on fuzzy logic, then on the foundations of computer science, and then on the empirical study of group decision making. Examples include his analysis of the Watergate tapes and an analysis of the black-box recordings of cockpit conversations in aviation accidents. Goguen spanned boundaries. For instance, he considered lessons from jazz in developing a theory of consciousness.[9] His work, taken as a whole, was an attempt to include art, ethics, and group politics within the range of formal description—a mathematics of the complexity of human life.

This is evident in his statement of the procurement problem: "Experience shows that many failures are due to a mismatch between the social and technical aspects of a supposed solution."[10] (p. 97) In works spanning 1994–2000, Goguen argued that we need new thrusts in education and new approaches to system development that integrate the technical and social aspects of work, improve resilience, and enable people to cope with the complexity of the technology itself.[10–13]

A tipping point

Procurement woes bear repeating right now because we might be at a tipping point. The US Navy initiative for "human-system integration" (HSI) has gained voice and office in other branches of the military. With Steven Deal's indefatigable leadership, the Human-System Integration Working Group of the International Council on Systems Engineering has drafted language for a definition of HSI to be included in the *INCOSE Systems Engineering Handbook*.[14] There is wide agreement that cognitive systems engineering must (somehow) be integrated with systems engineering and not just "injected into" it.[15,16]

Quotables

Goguen's work is chock full of juicy quotables capturing procurement antipatterns. Antipatterns are patterns people follow regularly and with negative repercussions that limit the integration and even the consideration of human and social factors in technology development.[17]

In this essay, we hope to give you a sense of Goguen by sharing some of our favorite quotes. Some of these echo our own thoughts. Others contain insights that widened our eyes to the range of factors contributing to and affected by the current state of human-centering in systems development.

On software engineering

Goguen's arguments about software engineering convey his passion about overcoming designer-centered and reductionist approaches to technology development:

> Experience with real projects shows that there is no such orderly progression from one phase to the next; instead, there is a continual projection forward and backward. … the nature and limitations of such models [e.g., waterfall and so-called process models] do not seem to have been widely appreciated.[12] (pp. 176–177)

> Indeed, the activities necessary for a successful system development project cannot always be expected to fit in a natural way into any system of pre-given categories.[11] (p. 35)

> Similarly, requirements documents must serve a number of different stakeholders.[11] (p. 37)

Goguen proposed an alternative way to conceive software development on the basis of Humberto Maturana and Francisco Varela's notion of self-organizing or "autopoietic" systems:[18]

> A software development project is not a formal mathematical entity. Perhaps it is usefully seen as an autopoietic process, an evolving organization of informational structures, continually recreating itself by building, modifying, and reusing its structures … Autopoietic systems are about as far as we know how to get from rigid top-down hierarchical goal-driven control systems; autopoietic systems thrive on error, and reconstruct themselves on the basis of what they learn from their mistakes. Autopoiesis can be considered an implementation technique for postmodernism.[10] (pp. 116–117)

Following this lead, the most recent discussions of alternative approaches to system development rely heavily on notions of evolutionary adaptive development[15] and resilience engineering.[19]

On requirements specification

Goguen recognized that just as controlling a complex sociotechnical system development process is impossible using regimented top-down controls, managing requirements about complex sociotechnical systems within such regimentation is also impossible:[10]

> The very rapid rate of change of requirements, which is so typical of large projects, implies an even more rapid rate of change for specifications. This makes many formal methods very difficult, perhaps even impossible, to apply in practice. (p. 115)

> It is *not* just as easy to find specifications and invariants for the flight control software of a real airplane as it is for a sorting algorithm; in fact, finding specifications and invariants is not an important activity in real industrial work. On the contrary, it turns out that finding requirements (i.e., determining what kind of system to build), structuring the system (modular design), understanding what has already been done (reading documentation and talking to others), and organizing the efforts of a large team, are all much more important for a large system development effort. (p. 101)

We see here that Goguen appreciated the idea that came to be known as the Envisioned World Problem.[20] Seeds for this idea emerged at about the same time in several scientists' writings.[21,22] As Goguen argued, this notion that "designs are hypotheses" calls into question the view that requirements are stable things that can be preformulated:[11]

> This explains why it can be so difficult to determine the requirements for a large system: it only becomes clear what the requirements are when the system is successfully operating in its social and organizational context; requirements evolve as system development proceeds, and a reasonably complete and consistent set of requirements for a large, complex system can only emerge from a retrospective reconstruction … Determining whether some system meets its requirements is the outcome of a complex social process that typically involves negotiation, and may involve legal action. Thus, it is usually entirely misleading to think of requirements as pre-given.[10] (p. 37)

On method

As the quotations show so far, Goguen was concerned with the methods of systems development and whether they provide a sound basis for a design:

> Moreover, the requirements phase of a large system development project … has the greatest economic leverage, … is also the least explored, and has the least satisfactory intellectual foundations.[12] (p. 166)

Goguen sought better and richer ways of measurement and evaluation, advocating the use of observational techniques for understanding the structure of sociotechnical systems and the ways in which people actually work. He acknowledged that using and developing methods that bridge the social and technical aspects of work is a difficult task. He also acknowledged that innovative and multidisciplinary work generally can be difficult and that some attempts are "greeted largely with incomprehension."[10] (p. 109) Particularly, he was concerned about the fact that commonly used techniques lead to a limited understanding of the problem space:

> This means that the needs of the user, both as individual and as organisation, are not addressed systematically; in general, they are only incompletely known to the development team, and there are often some serious misconceptions. [23] (p. 153)

Goguen was convinced that methods used by the social sciences were needed to understand and represent the complexity of cognitive work.

> The problems of requirements elicitation cannot be solved in a purely technological way, because social context is much more crucial.[23] (p. 153)

He argued that ethnographic methods (the observation of people in their natural environments) are genuinely scientific, appropriate for use in developing complex systems, and scalable to software development efforts' needs. He adopted a viewpoint called *ethnomethodology*, which

> tries to reconcile a radical empiricism with the situatedness of social data by looking closely at how competent members of a group actually organize their interactions.[11] (p. 40)

Although ethnomethodology relies on ethnographic methods, it differs from traditional sociology by focusing on the methods (hence, "-methodology") used by individuals and society (hence, "ethno-") to make sense of things and achieve social order. Ethnomethodology has had special impact in the sociology of scientific knowledge. For instance, the dependence of meaning on context has implications for the view that scientific knowledge is "objective."[24] Goguen applied ethnomethodology in his treatment of requirements analysis:

> To sum up, we recommend a "*zooming*" method of requirements elicitation, whereby the more expensive but detailed methods are only employed selectively for problems that have been determined by other techniques to be especially important. From this point of view, the various techniques based on ethnomethodology can be seen as analogous to an electron microscope: they provide an instrument that is very accurate and powerful, but that is also expensive, and requires careful preparation to ensure that the right thing is examined. [23] (p. 162)

Bridging the technological, social, and ethical

Goguen found it astonishing that in the information age,

> there is no adequate theory of information, nor even any adequate definition of information.[10] (p. 112)

He tried to extend traditional information theory to human situations but found it didn't apply:

> Data can only become *information* when people care about it for some reason and are able to interpret it. This means that information technology … is bound up with the social at a very basic level having to do with the nature of information itself.[10] (p. 94)

> Meaning is an ongoing achievement of some social group; it takes *work* to interpret configurations of signs, and this work necessarily occurs in some particular context.[11] (p. 34)

> Information has an intrinsic ethical dimension.[10] (p. 112)

He then tried general systems theory for a complexity-based information theory,

> but again it became clear that no purely formal approach, however abstract and general, could deal with human meaning in any deep sense.[10] (p. 112)

> It follows that a suitable theory of information must be a *social theory of information*, rather than a *statistical theory of information*.[11] (p. 29)

> *An item of information is an interpretation of a configuration of signs for which members of some social group are accountable.* The goal is to get a theory of information adequate for understanding and designing systems that process information.[10] (p. 112)

Goguen considered information to be either "dry" or "humid," a distinction we find both useful and entertaining:

> Formalization is the process of making information drier (i.e., less situated) by using a more explicit and precise metalanguage. (p. 32)

Dry, formalized information is represented by formalized modeling languages. Information can be humid, too—that is, "situated." In the same 1999 article, Goguen cited recipes as an example of humid information. His point was quite serious:

> But we now know that ignoring the situated, social aspect of information can be fatal in designing and building software systems.[10] (p. 97)

> Western civilization is fundamentally entangled with a separation of technology from ethics, based on an untenable instrumental conception of technology (i.e., viewing technologies simply as tools we create rather than as a force shaping our view of the world). (p. 16)

Goguen proposed that developers must learn to take information's qualities into account. Specifically, information is situated, local, emergent, contingent, embodied, vague, and open.

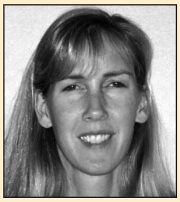
> [Therefore we can] understand why it is not possible to completely formalize requirements: They cannot be fully separated from the social context. More specifically, the qualities explain why so-called lifecycle phases cannot be fully formalized.[11] (p. 35)

An epitaph

The illumination that we see at the end of the tunnel actually comes from behind, not from ahead. It comes from the lanterns of those from whom we've learned, lighting a way forward for us. ■

References

1. K.J. Vicente, "Crazy Clocks: Counterintuitive Consequences of 'Intelligent' Automation," *IEEE Intelligent Systems*, Nov./Dec. 2001, pp. 73–75.
2. L. Lormann et al., *A Review of Government Requirements and Standards with Regard to Human Factors and Human-Centering of Technology*, to be published; available from Jennifer Fowlkes on request.
3. *Directive No. 1100.4, Guidance for Manpower Management*, US Dept. of Defense, 2005.
4. P. Koopman and R.R. Hoffman, "Work-arounds, Make-work, and Kludges," *IEEE Intelligent Systems*, Nov./Dec. 2003, pp. 70–75.
5. W. Zachary et al., "Human Total Cost of Ownership: The Penny Foolish Principle at Work," *IEEE Intelligent Systems*, Mar./Apr. 2007, pp. 22–26.
6. *DoD Design Criteria Standard: Human Engineering, MIL-STD-1472F*, US Dept. of Defense, 1999.
7. R.R. Hoffman, G. Lintern, and S. Eitelman, "The Janus Principle," *IEEE Intelligent Systems*, Mar./Apr. 2004, pp. 78–80.
8. R.R. Hoffman and W.C. Elm, "HCC Implications for the Procurement Process," *IEEE Intelligent Systems*, Jan./Feb. 2006, pp. 74–81.
9. D. Borgo and J. Goguen, "Rivers of Consciousness: The Nonlinear Dynamics of Free Jazz," L. Fisher, ed., *Jazz Research Proc. Yearbook*, vol. 25, 2005, pp. 46–58; www.iaje.org/iaje.aspx?pid=68.
10. J. Goguen, "Tossing Algebraic Flowers down the Great Divide," *People and Ideas in Theoretical Computer Science*, C.S. Calude, ed., Springer, 1999, pp. 93–129.
11. J. Goguen, "Toward a Social, Ethical Theory of Information," *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide*, G. Bowker et al., eds., Lawrence Erlbaum, 1997, pp. 27–56.
12. J. Goguen, "Requirements Engineering as the Reconciliation of Social and Technical Issues," *Requirements Engineering: Social and Technical Issues*, M. Jirotka and J. Goguen, eds., Elsevier, 1994, pp. 165–200.
13. J. Goguen et al., "An Overview of the TATAMI Project," *Cafe: An Industrial-Strength Algebraic Formal Method*, K. Futatsugi, A.K. Nakagawa, and T. Tamai, eds., Elsevier, 2000, pp. 61–78.
14. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,* Int'l Council on Systems Eng. (INCOSE), 2006.
15. R.W. Pew and R.S. Mavor, "Human-System Integration in the System Development Process: A New Look," *Report of the Committee on Human-System Design Support Changing Technology*, US Nat'l Research Council, 2007.
16. R.G. Eggleston et al., "Tightening the Linkage of CSE and Software Systems Engineering," *Proc. Human Factors and Ergonomics Soc. 48th Ann. Meeting* (HFES), 2004.
17. P. Laplante, R. Hoffman, and G. Klein, "Antipatterns in the Creation of Intelligent Systems," *IEEE Intelligent Systems*, Jan./Feb. 2007, pp. 91–95.
18. H.R. Maturana and F.J. Varela, *Autopoiesis and Cognition: The Realization of the Living*, Kluwer Academic, 1991.
19. E. Hollnagel, D.D. Woods, and N. Leveson, *Resilience Engineering: Concepts and Precepts*, Ashgate, 2007.
20. D.D. Woods and S.W.A. Dekker, "Anticipating the Effects of Technology Change: A New Era of Dynamics for Human Factors," *Theoretical Issues in Ergonomics Science*, vol. 1, no. 3, 2001, pp. 272–282.
21. J.M. Carroll and R.L. Campbell, *Artifacts as Psychological Theories: The Case of Human-Computer Interaction*, research report RC 13454, IBM T.J. Watson Research Center, 1988.
22. T. Winograd and F. Flores, *Understanding Computers and Cognition*, Ablex, 1986.
23. J. Goguen and C. Linde, "Techniques for Requirements Elicitation," *Proc. Requirements Eng. '93*, S. Fickas and A. Finkelstein, eds., IEEE CS Press, 1993, pp. 152–164; reprinted in *Software Requirements Engineering*, 2nd ed., R. Thayer and M. Dorfman, eds., IEEE CS Press, 1997.
24. R.R. Hoffman and L. Militello, *Perspectives on Cognitive Task Analysis*, Taylor and Francis, 2008.



Kelly Neville is a visiting associate professor in human factors at Embry-Riddle Aeronautical University. Contact her at nevillek@erau.edu.

Robert R. Hoffman is a research scientist at the Institute for Human and Machine Cognition. Contact him at rhoffman@ihmc.us.

Charlotte Linde is a senior research scientist in the Computational Sciences Division of the NASA Ames Research Center. Contact her at charlotte.linde@nasa.gov.

William C. Elm is president of Resilient Cognitive Solutions. Contact him at welm@resilientcognitivesolutions.com.

Jennifer Fowlkes is a managing cognitive engineer at CHI Systems. Contact her at jfowlkes@chisystems.com.