For a Project on Ethics and Software Engineering

by Vivian Weil and Michael Davis

Because computers are integral to most businesses, governments, and nonprofit institutions, everything from ballistic missiles to paychecks depends on the quality of software. Yet stories of software failure of one sort or another are common. Some, like Therac-25, involve loss of life, but most--like those involving loss of data--are merely about undue expense, inconvenience, and other injuries to welfare. Our project would be a contribution to preventing such failures.

Although software is often viewed as a literary product because it consists of a series of symbols, it resembles traditional engineering products in both function and creation. Some software has in fact replaced complex mechanical devices (for example, mechanical timing gears). Much software is the outcome of systematic research, specification, design, testing, manufacture, measurement, and maintenance. All software can be evaluated in the way engineering products are, by their safety, reliability, durability, utility, cost, and so on. Recognizing that in these ways software creation resembles engineering, many observers for several decades now, have referred to the creation of software as "software engineering."

Yet, reality seems to fall short of the hope expressed in that term. "Software engineers" seldom have a degree in engineering. They in fact have a heterogeneous origin and training. Often, their work lacks the systematic approach of engineering. Neither they nor their employers are willing to vouch for the fitness for use of their work in the way engineers do.

Some observers have identified barriers that have kept creators of software from accepting responsibility for the safety, reliability, and utility of their products in the way expected of engineers. If this analysis is correct, there is a pressing need to remove the barriers that prevent software creators from becoming engineers in a full sense. In part the problem is intellectual: to conceptualize engineering aided by an understanding of how the practices of engineering develop. In part the problem is practical: to figure out how to support the development of systematic practices and appropriate standards for different domains, e.g. for life critical or mission critical software, and for different settings in companies and government agencies. Perhaps there should be a minimum standard for all software.

There is a need then to deal with the question of software engineering. This would mean at least delineating the appropriate scientific knowledge, the body of techniques of design, and the organizational and managerial aspects of the role. That conceptual exercise is essential to dealing with the practical aspects of bringing an appropriate safety and reliability orientation into the creation of software. A project to facilitate the transition from haphazard ways of creating computer software to more disciplined engineering practices, friendly to creativity but oriented toward safeguarding against risks, would have to include contributions from a number of different sources. These sources include, in addition to computer science: history and philosophy of engineering, ethics, law (with regard to liability), private (and government?) sector employers, and professional societies. The project would include an examination of models of a systems approach to developing software for their applicability in different

contexts and for the distinctive responsibilities they generate. The products of the project would be an exemplar of a code of professional responsibility designed for software engineers that includes its rationale, and a handbook of ethics problems and cases with commentary. These, together with initiatives the project would instigate in professional societies and in education, would be designed to accelerate the professionalization of software creators.

There follows a list of names and affiliations of likely people who might contribute to the project.

Vivian Weil, Director, Center for the Study of Ethics in the Professions (CSEP)

Michael Davis, Senior Research Associate, CSEP

Ilene Burnstein, Associate Professor of Computer Science, IIT

Michael S. Mahoney, History, Princeton University

Ronald Staudt, Professor of Law & Director Computer Development, Chicago-Kent College of Law, IIT

Michael Loui, Professor of Computer Theory, University of Illinois, Urbana

Helen Nissenbaum, Associate Director, Center for Human Values, Princeton University

Barry Weiss, Gordon & Glickson, P.C.

Vince Serritella, Software Engineering Initiative, Motorola

June Rokoff, Senior Vice President for Software Development, Lotus Development Co.

Donald Gotterbarn, IEEE/ACM Ethics Committees

Someone from the Carnegie Mellon University Center for Software Improvements.