

Collected Thoughts for a Project on Ethics and Software Engineering

by Vivian Weil

Computers are integral to most enterprises in the private sector as well as in government and the military. The quality and reliability of software is therefore important. Observers were quick to notice that software resembles engineering products, though it also has attributes of literary products. It has taken longer to recognize good reasons for trying to train and organize creators of software as engineers. Once it becomes clear that creating software requires a systems approach, coding (programming) becomes a component of a process that begins with discovering the requirements for a new product and includes putting together specifications for a system, designing the system, testing, measuring its characteristics, and maintaining it. This systems approach is characteristic of engineering, as is the development of standards and systematic practices to ensure quality and reliability. Practitioners trained to take this approach and oriented towards meeting an appropriate standard of care would properly be regarded as engineers. However, the creation of software has not yet in many places become the disciplined process just described. The practitioners are too heterogeneous in their training and too undisciplined in their practices.

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, 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, U of Illinois, Urbana

Helen Nissenbaum, Associate Director, Center of 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.