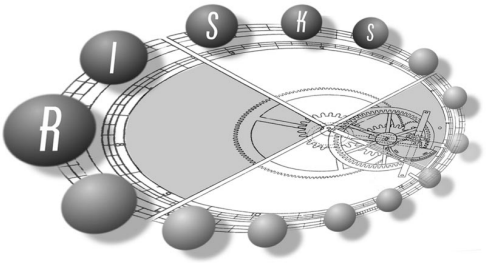

Peter G. Neumann

Integrity in Software Development

In the September 1997 "Inside Risks," David Parnas made a compelling case for defining the core body of software knowledge, evaluating the curricula, and licensing software practitioners the same way engineers are licensed. A desirable obligation he mentioned beyond basic competence is the ability to pass examinations relating to legal and ethical practice. Most professional organizations have well-documented codes of ethics. However, Parnas observes that whereas professional engineers who do not adhere to their oath can lose their licenses, software professionals have little to lose and may not have heard about such oaths.

Unfortunately, excuses such as short-term profits, preservation of corporate image, organizational and peer pressures, secrecy, and perhaps setting low standards in hopes of limiting liability suits for noncompliance are sometimes used as justifications for irresponsible behavior. On the other hand, real potential risks to individuals tend to discourage responsible behavior by people who might otherwise consider becoming whistle-blowers. The ethically inspired act of whistle-blowing might typically involve various aspects, such as being aware of and documenting a serious problem, informing one's superiors, having findings repeatedly rejected and systematically covered up, and being threatened with or subjected to retaliation.

When a possibly preventable disaster strikes, questions are raised, such as what was known beforehand and what actions had been taken. Many lessons can be learned from past experiences.

• Analyzing the Challenger disaster on January 28, 1986, the Presidential Commission noted other problems beside the previously identified risks of O-rings at low temperatures: inadequate spare parts; lack of training in maintenance; improper management planning; and safety problems with the main engine, brakes, flapper valves, and automatic landing system—plus a serious reduction in the quality-assurance effort.

• Union Carbide reportedly had prior warnings about inherent safety risks at Bhopal, but not to the database error implicated in the aldicarb oxime leak.

• For years tobacco companies have been aware of the addictiveness of nicotine and health risks associated with smoking. Their internal files are for the most part still not public.

Among the software development problems included in the RISKS archives, many cases are attributable to a lack of knowledge, training, and expertise on the part of requirements specifiers, designers, programmers, operators, and users. Some problems are attributable to the necessity of using systems, languages, and tools inherently risky. In some cases, serious problems were recognized in advance.

• The Aegis software and user interface involved in the Vincennes' shoot-down of the Iranian Airbus had some limitations (see *SIGSOFT Softw. Eng. Notes, 14*, 5, pp. 20–21, 1989, and *Computer-Related Risks*, ACM Press/Addison-Wesley, p. 35).

• The success rate of the Patriot missile was originally estimated at 80%, but subsequently downgraded to about 10% after Ted Postol's MIT analysis on the missile's ineffectiveness had been classified out from under him and Postol had been subjected to serious reprisals as a whistle-blower. (Also, a critical clock-drift problem had been discovered in the software, but the fix arrived only the day after the Dhahran barracks were hit by Iraqi Scud missiles.)

• Edward F. Wilson was responsible for developing government-required aerospace software quality-assurance programs for Amex Systems in 1986. He learned his employer would not implement those programs once submitted, and complained in writing. He was fired for "being a troublemaker" and subjected to anonymous death threats.

The U.S. offers whistle-blowers significant shares of funds recovered from lawsuits for fraud against the government. In the rather different context of software development, monetary gain should not be the primary incentive—although the absence of protection against serious monetary hardship can be an impediment to would-be whistle-blowers. The software profession needs a broadly accepted code of ethics that effectively penalizes flagrant abuses and provides some protection for high-integrity whistle-blowers. There are already several good starting points (for example, the IEEE provides formalized support for well-documented whistle-blowers through its ethics committee), but enforcement actions seem to be rare.

One of the repeated themes of "Inside Risks" is that no single would-be solution can eliminate computer-related risks. Ethical behavior is yet another prerequisite to acceptable-risk software development. C

The latest joint ACM/IEEE Code of Software Engineering Ethics will appear in the November 1997 issue of Communications. *The ACM's Code of Ethics and Professional Conduct is online (www.acm.org/constitution/ bylaw17.html), as is the IEEE code (www.ieee.org). Members of the ACM Committee on Computers and Public Policy contributed notably to this column.*

PAUL WATSON