Software Engineering Ethics

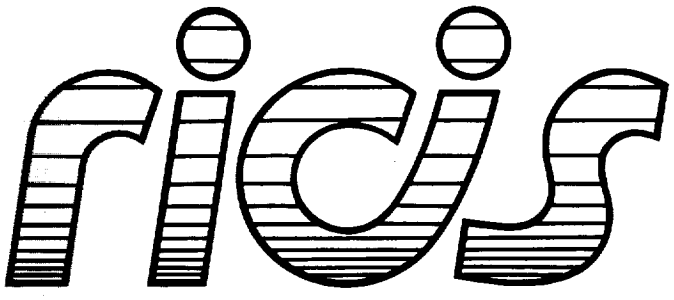
Rodney L. Bown

University of Houston-Clear Lake

April, 1991

Cooperative Agreement NCC 9-16
Research Activity No. SE.26

NASA Johnson Space Center
Engineering Directorate
Flight Data Systems Division



Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

Software Engineering Ethics

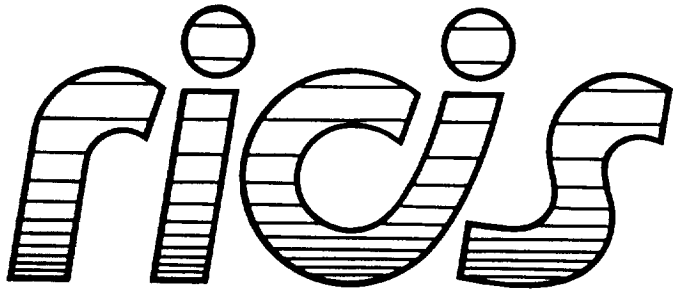
Rodney L. Bown

University of Houston-Clear Lake

April, 1991

Cooperative Agreement NCC 9-16
Research Activity No. SE.26

NASA Johnson Space Center
Engineering Directorate
Flight Data Systems Division



Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Dr. Rodney L. Bown, Associate Professor of Computer Systems Design at the University of Houston-Clear Lake. Dr. Bown also served as RICIS research coordinator.

Funding has been provided by the Engineering Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was William C. Young, of the Project Integration Office, Flight Data Systems Division, Engineering Directorate, NASA/JSC.

RICIS TECHNICAL NOTE

Software Engineering Ethics

Principal Investigator
Dr. Rodney L. Bown

Unsolicited white paper
in partial fulfillment of

RICIS Task SE. 26

April 1991

CONTENTS

Software Engineering Ethics

1. Introduction

This document is being submitted as an unsolicited position paper in partial fulfillment of RICIS Task SE. 26. A UHCL white paper on Software Engineering Ethics was written in July 1989 in response to a Software Engineering Institute project on software engineering ethics. The activities of RICIS Task SE.26 provided an opportunity to add some additional comments and an update to the 1989 paper. This document can be considered a set of notes and issues for further discussion by professional colleagues. The reader is free to use the cited examples and comments in support of computer system ethics.

2. Examples

Several public domain examples are reviewed with an associated discussion of software engineering ethics. When known, the references are cited. The examples include professional education and industry practices.

2.1 Lack of a System Viewpoint

Gibb has cited some of the Volvo's experiences in their attempt to produce a Corporate Information System [GILB88]. The estimated budget was eighty work-years. It had grown to 160 work-years by the time the President of Volvo canceled the project. When Gilb asked one of the programmers about the project, the reply was "I knew it wouldn't work - but my programs worked." The ethical point that should be observed is: the programmer did not feel the need to worry about whether the system was a failure, as long as his own component functioned.

A counter example that relates to the component versus systems perspective is the aerospace engineering culture. The success of an aerospace vehicle is dependent on the successful blend of robust components. The engineer that is responsible for the design of the landing gear has the aerospace corporate culture paradigm that the product is not the landing gear but is the total system. This system paradigm is foreign to most software practitioners.

2.2 Arrogance of PC DOS Software Vendors

This writer has the opinion that most MS-DOS software products do not recognize the difference between a public and private commode. The general public does not care if you flush your

commode in your private residence. The general public does care if you fail to flush the commode in a public rest room. Almost all PC software products do not flush the commode.

One example is GEM DRAW on the IBM PS/2. When the user exits GEM DRAW to return to DOS, some of the resident RAM will be lost. This will prevent an execution of JANUS Ada. The solution is to reboot in order to flush the "dodo" that GEM left in the system. These products exhibit the "me only" selfish attitude of the vendors. The vendors assume that the customer will purchase one product: "theirs" and never use the PC for multiple products.

The disclaimer is that the PC is a personal computer and not a work station. The argument is: "how many PC customers are sophisticated enough to understand that many of the PC MS-DOS software products create garbage and that a garbage collector is not installed in their system?".

2.3 Violation of Upward Compatibility

Intel has produced an upward compatible family of computers. Intel has very carefully stated the requirements for upward compatibility in their software and hardware technical manuals. Intel produced the 8086 with a documented instruction set. Some of the bit patterns have been reserved by Intel. In other words, Intel has said "Thou shalt not touch. If you do touch we can no longer guarantee the results." These stated requirements support an observation that Intel should receive a clean bill of health.

Now here come the software vendors. In order to save some clock cycles, many vendors used illegal operation codes on the 8086. Intermediate flags are stored in Intel reserved bit locations to save an instruction cycle. Software vendors are interested in an immediate return of investment and do not know how to spell or understand the life cycle issues of software engineering.

The result is the following scenario:

a. The software vendor uses some of the unused bit patterns to achieve a performance advantage over their competition.
b. The customer upgrades his 8086 PC to the 80286 or to the 80386. The customer believes Intel when he is told that the system is software upward compatible.
c. Unknown to the customer is the fact that Intel is now using the previously reserved bits to support new features of the 80286 or 80386.
d. The customer executes his "old" 8086 software and crashes his system.

This writer has discussed this issue with several sophisticated users of PC MS DOS tools and has been told the above scenario is not an isolated case.

2.4 Internet Worm

The infamous Internet Worm incident of November 2, 1989 has produced a wealth of technical and legal responses from the computing community [CACM89]. The initial public response produced approximately a zero response with respect to ethics. The point this writer wishes to discuss is the strange attitude that Robert Morris Jr. exhibited with respect to the computer system owned and operated by his organization: Cornell University. This writer is creating this document using a licensed copy of Word Perfect on a PC which is owned by the University of Houston-Clear Lake. The PC and its associated software are professional tools and are not toys or playgrounds. Has Robert Morris Jr. demonstrated to the computing community that many users treat their computer systems as toys and playgrounds for their personal pleasure?

This writer knows of office computers that are used as toys for computer games. There seems to be a lack of respect for computers. One does not play with someone else's automobile without permission. Why should some users believe that it is a right to play with their organization's computer system. The Internet Worm became famous because we are now dealing with networks and the potential for wide spread harm to innocent users.

In a machine shop, it may be appropriate to allow hobbyists to use certain machines after hours to produce personal items. The machines are robust and are in a stand alone domain. The machines will not be damaged by proper use. The users have probably satisfied some benign permission policy and abide by a set of procedures that are understood by all participants.

If there is a network connection, computers must be treated with great respect. Unauthorized use of a professional tool should not be tolerated. The computing community should be concerned that many "young" practitioners seem to treat computers as toys and not as tools.

2.5 Internet Worm Revisited

Dr. John McHugh presented a tutorial "Trusted Systems, Software Engineering, and the Internet Worm at HCL on 14 July 1989 [McHU89]. Chart 11 of the tutorial states:

> Much has been said in the media and technical press about the blame for the worm infestation. Much of the evidence points to a graduate student at Cornell as the culprit. The fact that the defects in UNIX were known has been used to describe the incident as "unsportsmanlike".

It has been claimed that none of the assurance methods used by the computer security community could have prevented this kind of attack.

Poor software engineering is nowhere mentioned.

A direct quote from that presentation: What are the ethics of pushing a technology beyond its intended limits and fundamental paradigms? UNIX and UNIX like products are being sold for application domains that may exceed the underlying paradigms of UNIX. This writer will let Dr. John McHugh stand on his own presentation and highly regarded credentials.

2.6 Student Cheating and Company Hiring Interviews

This writer has discussed job interviews with some of his former students. The students indicated that the potential employers were interested in two facts: What is your grade point average (GPA) and what programming languages/operating systems can you use?

The university is attempting to teach core knowledge about computing life cycle issues and the technical paradigms of software engineering. The industry should be concerned about application of core knowledge to design current and future computing systems. The design of future systems may require the use of future technology. The university must provide the fundamentals that will allow a practitioner to grow with his chosen discipline and technology.

The university is concerned about cheating. This writer does understand the difference between student peer exchange of information, reusable software, and cheating. If the company wants a high GPA, how can the teacher discuss the ethics of cheating? The student knows that the employer will provide a reward for a high GPA. The shortest path to a high GPA is to cheat. This teacher discusses life and property issues of software engineering but the student understands that he will receive a reward for cheating. What are the ethics? There are "C" students who understand the core knowledge of their courses, who have had deaths in their family, work part time, and are raising children. These students suffer because the employer does not have the sophistication to conduct a proper campus interview.

The employer should be asking questions during the interview that identifies the students understanding of core knowledge and the level of maturity of the student. Core knowledge supersedes a particular programming language.

Final point here about the ethics of hiring students. This writer has the personal opinion that companies do not want to

hire at the universities with a few exceptions. This writer has talked to many supervisors and the common theme is that three years of experience is desired. Where does the college graduate get the first three years of experience? Many professions recognize the need for a period of internship that follows the completion of formal education. Industry must accept the responsibility to support a transition period between an undergraduate education and full membership within the professional design community.

2.7 Computing Practitioners and the Commodity Market

The common folklore is that the software practitioner will stay with one organization for approximately three years. This interval reflects a common development cycle for many software projects. It also reflects that most organizations treat software and the software practitioners as commodities that can be brought when required. Some of the answers reside in the nature of the company's product and the nature of software. Software engineering and its products are part of a system product and remain hidden from view. This situation does not promote loyalty between the employer and the software employees. What are the life cycle problems of short term employment in the software discipline.

Much of the Boeing 757/767 software was developed by contract employees and not by Boeing employees. This was a management decision based on the paradigm that Boeing did not have a long term commitment to software production and software personnel. Boeing does have a long term commitment to the traditional aerospace disciplines such as aerodynamics, propulsion, structures, management, and finance. There is a traditional career path available for practitioners in these fields. The rest of the airframe was developed and produced by these traditional in house technical organizations and appropriate subcontractors.

Most organizations do not have a defined career path for the software practitioner. It is difficult to find a vice-president or director of software engineering in most organizations. What are the ethics of both the company and the software practitioners when they both have a short term perspective of software? The software practitioner is not motivated to exhibit loyalty to an organization that uses his skill as a commodity. This culture promotes distrust between software practitioners and the employers. What does this distrust do to the ideas of a total software life cycle perspective advocated by the software engineering community?

2.8 New Projects and Old Programming Languages

Folklore indicates that FORTRAN has been selected as the software language for the Texas super collider project because it is the language understood by the physicists. This writer is appalled by the arrogance exhibited towards software by experts in other disciplines. Since medical doctors may understand Basic, should Basic be the language of choice for running a hospital? The software engineering discipline has a difficult task in just trying to reach software practitioners. This writer asserts that a Fortran programmer has the ethical responsibility to recognize when a new project exceeds the basic paradigms of Fortran. The Fortran programmer may not know how to use Ada or some other modern language but he must recognize the existence of a modern alternative technology.

This writer observes that many software practitioners do not belong to professional societies or read the professional literature. If a Fortran programmer has worked in isolation for years, he should still be reading the literature. The existence of Ada should not be a surprise to the Fortran programmer. There is a fundamental flaw in the software work ethic. This flaw is exemplified by a Fortran, Cobol, of C expert that thinks his expertise is adequate for all time. Most other professions recognize the changing nature of their discipline and actively maintain their competitive edge. Even accountants learn new ideas due to changes in the tax laws.

This writer attended the 11th International Conference on Software Engineering in Pittsburgh during May 1989. The above discussion is one example of Human Hardware that was the topic of a plenary session. Professor Melvin Kranzberg of Georgia Tech University eloquently discussed the issues of human hardware. Many of us believe the human is flexible and adaptable. Professor Kranzberg presented many examples that indicated humans tend to be hard and inflexible. So the hardness of humans may not be an software engineering ethics issue but one of human nature.

2.9 Schedule and Budget

The Department of Defense and their contractors use the well know paradigm of "get it on contract now and we will change it later". This is partially caused by the need to convince Congress that the system is truly required now and must be in this year's budget. This paradigm provides some success when the item under development and production is in the form of a widget. A typical widget is an airframe that lacks its weapon system, spare parts, and training facilities. At least the airframe exists and can evolve into a usable system with subsequent financing for engineering changes, spares, and personnel training.

Software is inherently an intellectual activity which does not respond a cyclic on/off switch. Gilb in his book advocates the

paradigm of evolution [GILB88]. This paradigm says "deliver as little as possible". This will allow the customer to grow with the product. It is unethical to underbid a project to the detriment of all. However the evolution paradigm has at least provided some hint as how both the customer and the developer can start on a subset of the proposed final product.

3. Recent Public Domain Comments on Hacking

There is a software.risks bulletin board monitored by Dr. Peter G. Neumann of SRI International. Dr. Neumann publishes an edited version of the submissions in the Software Engineering Notes which is a publication of the Special Interest Group on Software Engineering (SIGSOFT) of the Association of Computing Machinery (ACM). Recently the RISKS bulletin board has provided an active dialogue about the ethical and legal issues related to the activities of computer hackers. In particular the debate is discussing hacking activities that leak information without damage to the computing system.

In addition the March 1991 issue of the Communications of the ACM provides a lively discussion about a recent criminal indictment of Craig Neidorf [CACM91]. Craig Neidorf was accused by the U.S. government of fraud and interstate transportation of stolen property regarding a document published in his electronic newsletter, Pharck. The trial began on July 23, 1990, and ended four days later when the government dropped the charges. ACM has published a debate on the case in the March issue of CACM.

The debate on the bulletin board and in the CACM has ranged over many analogies of unauthorized activities such as picking of a neighbors flowers, walking through an unlocked house, joy riding in a car, and walking through an office building after hours. The debate indicates that ethics and law are ill defined related to unauthorized use of a resource that does not cause damage to the resource.

There is a rich source of publications on computer ethics. Donn Parker of SRI, International published an early study on computer ethics in [PARK79]. Parker has presented papers at various conferences and is one of the debaters in [CACM91]. Charles Pfleeger has included a chapter in his book on Security in Computing [PFLE89]. Dr. Maureen Campbell of Murray State University has provided dynamic discussions of ethics at various computing conferences including the joint JSC/CSC/UHCL ISIS Symposium in May 1990 [CAMP89], [CAMP90].

The national concern about computer ethics can be demonstrated by the number of papers, panels, and summaries that have been scheduled at technical meetings. The 12th National Computer Security Conference (1989) dedicated one of four tracks to Education and Ethics. Again in 1990, the 13th National Computer

Security Conference contained several papers on ethics. Some of the papers and executive summaries are listed in the Bibliograhpy.

It is obvious that computing ethics are not well defined by either the practitioners or by society. This writer's interpretation of the debate is that society has not recognized that the majority of computing systems should be defined as industrial grade tools. For the protection of society, industrial grade tools must be operated within authorized policies and procedures.

4. Final Comments

The world does not reward the individual that blows the whistle on unethical behavior. Software engineering is no different from many other professions. The individual needs a job in order to pay the bills. The companies need the contracts to stay in business. In many cases it will be the market place and reputation that determines success. The engineering profession has examples of every known variation of delivered and canceled systems. Unnecessary systems are delivered to unsophisticated customers. Good systems are canceled due to strange reasons. The sale of a high performance automobile to an 18 year youngster on a six year payment plan is no different from selling inadequate software to an unsophisticated customer. Do you tell the 18 year old that the high performance automobile is inappropriate? Do you tell your software customer that the software is inadequate?

References and Bibliography

[CACM89]
Communications of the Association for Computing Machinery (CACM) June 1989.
Theme issue on the"infamous" Internet Worm

[CACM91]
"Debate: The United States vs. Craig Neidorf." Communications of the ACM. March 1991. pp.22-43.

[CAMP89]
Campbell, Marlene. "Ethics: Mandate vs. Choice." Plenary Session, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland.

[CAMP90]
Campbell, Marlene. "Ethics: Mandate vs. Choice, an update." Plenary session, Information Security and Integrity Systems Symposium. 15-16 May 1990. University of Houston-Clear Lake, Houston, Texas.

[CORD90]
Cordani, John. "Virus Ethics: Concerns and Responsibilities of Individuals and Institutions." Proceedings, 13th National Computer Security Conference. 1-4 October 1989 Washington, DC. pp. 647-652.

[DeMA89]
DeMaio, Harry B. "Information Ethics, A Practical Approach." Proceedings, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland. pp. 603-633.

[DENN90]
Denning, Dorothy. "Concerning Hackers Who Break into Computer systems. Proceedings, 13th National Computer Security Conference. 1-4 October 1989 Washington, DC. pp. 653-664.

[FORC89]
Forcht, Karen A. "Ethical Use of Computers." Proceedings, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland. pp. 624-626.

[GILB88]
Gilb, Tom. Software Engineering Management Practical Principles. Wokingham: Addison-Wesley 1988.

[LEIG90]
Leighninger, Eric. "Discerning an Ethos for INFOSEC Community: What Ought We Do?." Proceedings, 13th National Computer Security Conference. 1-4 October 1989 Washington, DC. pp. 641-646.

[McHU89]
McHugh, John. "Trusted Systems, Software Engineering, and the Internet Worm." Software Engineering Research Center (SERC) presentation on July 14, 1989 at UHCL

[MART89]
Martin, Larry. "Unethical "Computer" Behavior: Who is Responsible." Proceedings, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland. pp. 531-540.

[PARK79]
Parker, Donn. "Ethical Conflicts in Computer Science & Technology." AFIPS Press 1979

[PARK89]
Parker, Donn. "Ethical Conflicts in Computer Science and Technology." Overture session, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland.

[PFLE89]
Pfleeger, Charles. Security in Computing. Englewood Cliffs: Prentice Hall. 1989.

[PREI90]
Preiss, Ralph J. "Society Runs on Trust." Proceedings, 13th National Computer Security Conference. 1-4 October 1989 Washington, DC. Executive Summary pp. 632

[WATT89]
Watt, Glenn D. "Malicious Code: An Ethical Dilemma." Proceedings, 12th National Computer Security Conference. 10-13 October 1989 Baltimore, Maryland. pp.542-552.