The Uniqueness of Software Errors and Their Impact on Global Policy*

Donald Gotterbarn
Software Engineering Ethics Research Institute, East Tennessee State University, USA

Keywords: software error, quality, global policy, SoDIS

ABSTRACT: The types of errors that emerge in the development and maintenance of software are essentially different from the types of errors that emerge in the development and maintenance of engineered hardware products. There is a set of standard responses to actual and potential hardware errors, including: engineering ethics codes, engineering practices, corporate policies and laws. The essential characteristics of software errors require new ethical, policy, and legal approaches to the development of software in the global arena.

INTRODUCTION

Given the emergence of the global economy and the rapid increase in the number of multinational corporations, it is important to develop effective global policies establishing the responsibility and liability for defects in software products.

The policies, if properly developed will provide uniform standards and encourage all multinational organizations to meet these standards or only use software which meets these standards. Many nations have well established legal remedies for defects in hardware which are based on reasonable expectations of the functions and the reliability of the hardware artifact. Given the short history of software in international commerce, no similar consistent set of standards has been developed.

* An earlier version of this paper was presented at the Engineering Foundation Conference on “Ethics for Science and Engineering Based International Industries”, Durham, NC, USA, September 1997.
Address for correspondence: Don Gotterbarn, Professor, Computer and Information Sciences, Software Engineering Ethics Research Institute, East Tennessee State University, Box 70711, Johnson City, TN 37614-0711, USA, www.cs.etsu.edu/gotterbarn
D. Goterbarn

The difficulty in formulating adequate policies related to software is not merely due to the newness of software in engineered products but the difficulty also is based on some essential differences between hardware and software defects. By identifying and highlighting these differences, we move toward identifying adequate software policies and practices.

SOFTWARE ERRORS: DIFFERENT FROM AND HARDER THAN HARDWARE ERRORS

This claim about software errors can be approached in different ways. Specialists in software testing take a bottom up approach, making fine distinctions between types of software errors such as faults, errors, and defects. To understand the differences between hardware and software errors it is better to use a top down approach, and distinguish two sorts of software system. There is, what has been described as, shrink wrapped software which is designed to accomplish a focused set of tasks like those in an accounting system—accounts receivable and accounts payable—or tasks in a word processing system—entering and deleting text. These shrink-wrapped software systems are designed independent from any engineered hardware system. On the other hand, there is software which is tightly coupled to an engineered hardware system. This type of software is a critical part of the functioning of the hardware. Examples of this are the software which is embedded into a pacemaker or the carburetor control of an automobile. In addition to embedded tightly coupled software, there is also tightly coupled software which is used to control a mechanical process such as the software which manages an assembly line.

Both the shrink wrapped and the hardware dependent software can have errors. There are some subtle differences made between errors for these two types of systems. People are more tolerant of errors in shrink-wrapped systems, than they are of errors in software tightly coupled to hardware related systems. People are dissatisfied if a word processor fails four times in one day, but not nearly as dissatisfied as they would be if a pacemaker failed once a month.

HARDWARE AND SOFTWARE QUALITY

A brief look at quality criteria for hardware and software products will help us understand the causes for some of the differences between hardware and software errors. The concept of quality in software is similar in many respects to the concept of quality in an engineered hardware product. For both, quality means that the product meets the user's specifications and satisfies the user's needs. Defects, which are not related to satisfaction of the requirements, do not reduce quality. Software and hardware also must be usable and safe.

Despite these surface similarities, there is a very significant difference in quality criteria for software and hardware. In general, hardware is engineered to perform a