

Regulatory Compliance Software Development: Inevitability, Methods, Hopes and Concerns

Peter J. Knoke
University of Alaska Fairbanks
pjknoke@alaska.edu

Abstract

In the future, technology will continue to advance rapidly because there seem to be no likely mechanisms or desires for preventing that. The associated need for new technology regulations will continue to increase because there seem to be no proven mechanisms for predicting and avoiding abuses of the new technology, and yet those abuses often occur.

The development regulatory software for new technology (e.g., software for trustworthy intelligent robots) will be increasingly difficult because it is complex and because it deals with “wicked problems” (a recent example is software for healthcare.gov website). The most appropriate methods to use for regulatory compliance software for many new technologies are often not clear. Classical software development methods are unlikely to succeed for “wicked problems” generally (the term “wicked problems” has been used for many years, e.g. see Wikipedia (wicked problems), but the currently popular “agile” software development methods (e.g. Scrum, Extreme Programming, etc.) might also be unsuccessful absent significant modifications. Such needed modifications are currently the subject of considerable software engineering research.

It is usually generally hoped that the evolving new technologies will prove to be socially beneficial. However, while there are concerns are that the societal side effects of these technologies could range from undesirable to disastrous. A major problem is that such side effects are often unpredictable. Some well-known side effects of the Internet are an example of this (e.g., hacking, lack of security and loss of privacy).

The predicted “technological singularity” (cf Ray Kurzweil) is discussed in the context of the above issues. Some current active approaches to relevant problem solutions are discussed. However it is not known which of these approaches are most promising.

Biography

PETER J. KNOKE is a recently (7/14) retired Associate Professor of Software Engineering from the Computer Science Department (College of Engineering and Mining) at the University of Alaska Fairbanks. He had 30 years experience with multiple complex computer and software developments at GE, Syracuse University Research Corporation, NCR and Harris Corporation prior to joining the University of Alaska Fairbanks in 1988. He taught numerous UAF undergraduate and graduate courses in both Computer

Science and Software Engineering. He was a principal developer of a new Master of Software Engineering Program there. He is a registered professional engineer in Ohio and Florida. He received his BA and MS (Electrical Engineering) degrees from Dartmouth (NH) in 1955 and 1956, and his PhD (Electrical Engineering) from Syracuse University (NY) in 1968. Dr Knoke may be reached at pjknoke@alaska.edu.