Food for Thought™

An e-newsletter published by
Software Quality Consulting, Inc.

May 2011
Vol. 8 No. 3

Welcome to **Food for Thought™**, an e-newsletter from Software Quality Consulting. I've created free subscriptions for my valued business contacts. If you find this newsletter informative, I encourage you to continue reading. Feel free to pass this newsletter along to colleagues by using this Forward Email link. If you've received this newsletter from a colleague and would like to subscribe, please use this Enter New Subscription link. If you don't wish to receive this newsletter, use the **Safe UnSubscribe** link at the bottom of this newsletter, and you won't be bothered again.

Your continued feedback on this newsletter is most welcome. Please send your comments to **steve@swqual.com**

In This Issue

In This Month's Topic I discuss the benefits and risks associated with software.

Regular features to look for each month are:

- Monthly Morsels
  Hints, tips, techniques, and references related to this month's topic

This Month's Topic

Just because we can doesn't mean we should…

The movie **2001: A Space Odyssey**, released in 1968, was way ahead of its time. Written by Stanley Kubrik and Arthur C. Clarke, the movie attempted to illustrate man's complex relationship to the universe. The movie was recently recognized as one of the most important movies ever made.

One of the most memorable characters in the film is the soft-spoken on-board computer named HAL - represented by the iconic red eye. HAL is an acronym for Heuristically programmed ALgorithmic computer (to the dismay of those who initially thought HAL represented IBM plus one letter) and was a member of the 9000 family of computers – which had an impeccable record for infallibility.

Hello Dave.

HAL was the computer on board an American spaceship bound for Jupiter, along with two astronauts (Dave Bowman and Frank Poole) and three scientists who were in cryogenic hibernation. HAL ran most of the spaceship's operations and was programmed to complete the mission at all costs.

As it turned out, HAL wasn't infallible after all. Dave and Frank started to suspect something was wrong after HAL reported the imminent failure of a device that controlled the spaceship's main antenna. After retrieving the component, Dave and Frank could not find anything wrong with it. HAL suggested reinstalling it and letting it fail so the problem could be found. Mission control on Earth concurred, but advised the astronauts that results from their twin HAL 9000 computer indicated the spaceship's HAL is in error predicting the fault.

When queried, HAL insisted that the problem was due to "human error". Concerned with HAL's behavior, Dave and Frank discussed the situation in a

sealed pod out of HAL's earshot. They had a bad feeling about HAL despite HAL's infallibility record. They decided to follow his suggestion to replace the unit. The astronauts discussed deactivating HAL if he was proven wrong - unaware that HAL was reading their lips.

When he attempted to replace the unit during a spacewalk, Frank's EVA pod, controlled by HAL, severed his oxygen hose and set him adrift. Dave, not realizing HAL was responsible for this, took another pod to attempt a rescue, leaving his helmet behind. While he was gone, HAL terminated the life support functions of the three crew members in cryogenic hibernation. When Dave returned to the spaceship with Frank's body, HAL refused to let him in and said their plan to deactivate him jeopardized the mission. Dave manually opened the ship's emergency airlock and entered the spaceship risking his own death.

After donning a helmet, Dave entered HAL's memory banks intent on disconnecting the computer. HAL first tried to reassure Dave, then pleaded with him to stop, and finally began to express fear - all in a steady monotone voice. Dave ignored him and disconnected each of the computer's memory modules. HAL eventually regressed to his earliest programmed memory, and sang the song "Daisy Bell".

One of the many complex themes intertwined throughout the movie is the danger of creating technologies that are not fully controllable by humans.

Understanding Risks and Benefits

The movie illustrates the point that just because it is possible to develop software to perform some function doesn't mean it's always a good idea. This doesn't just apply to software – other fields such as medicine have similar issues which is why the role of bioethicists has been established at many medical schools and teaching hospitals.

With every new software application, we need to take a careful look at BOTH the benefits AND the risks. One important principle to keep in mind with respect to new technology is that we often trade one set of problems and risks for a different set of problems and risks.

The recent controversy involving Apple's iPhone and iPad and its tracking information is a good example…

Benefits:	Having accurate location information on your phone enables apps to identify nearby restaurants and shops as well as provide driving directions.
Problem solved:	Having location information available in your phone solves the problem of easily finding local businesses for those who may be unfamiliar with an area.
Risks:	Since your location information is stored in the phone, it may be used for purposes that you are not aware of and without your knowledge.
Problem created:	Having location information available in your phone can create problems since you can't prevent that information from being used for unintended purposes – such as tracking an individual's location - without the user's knowledge or permission.

If the risks seem to outweigh the benefits, then perhaps this software is not worth developing – even if it is possible…

Let's look at how this applies to software used in several key industries:

Medical Device Industry:

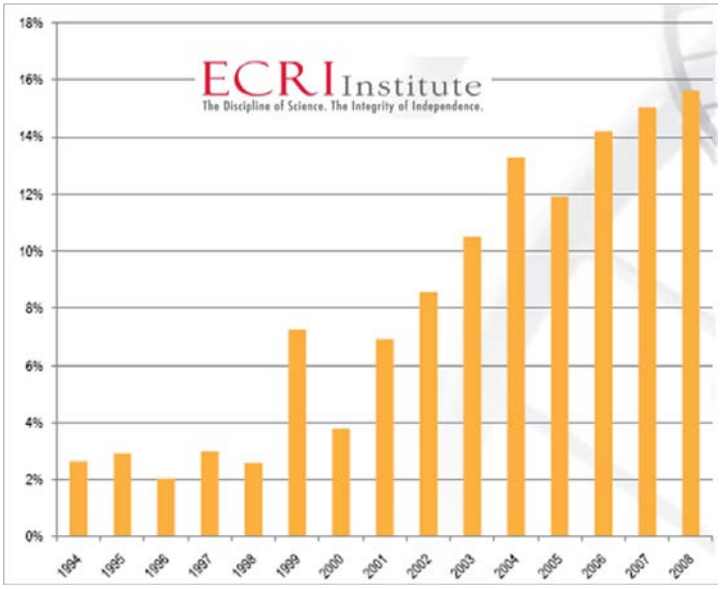
Benefits: Software-based medical devices have saved many thousands of

lives and improved the health of many more thousands. Software provides many new capabilities and helps improve overall device safety and efficacy. Medical device software is developed under rigorous regulations and standards.

An infusion pump meters out a specified amount of drug therapy over a specified time. Infusion pump recalls are among the highest of all software-based medical devices.

Risks: Failures of software-based medical devices have resulted in many deaths and serious injuries. [1] Already this year, FDA has issued over 75 device recalls for software-related issues. Further, FDA studies have shown that a significant number (~80%) of device errors are introduced when software is changed.

The following chart summarizes software-based device recalls between 1994-2008. [7] As you can see, the number of software-based recalls has been growing almost exponentially.



Automotive Industry:

Benefits: Critical functions controlled by embedded software enable better fuel efficiency, safety, and advanced accident avoidance features. Carmakers have been replacing mechanical cables and analog controls with digital controls as ways to lower cost, improve reliability, and add features. They have developed software systems to control braking, acceleration, airbags, climate control, transmission performance, emissions, navigation, and many other functions. The result of many of these changes has led to improved reliability, better fuel economy, and improved safety.

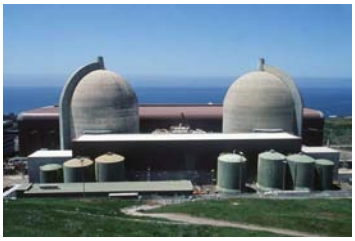
The 2005 Toyota Prius was recalled because of a software defect – the engine would stop when cruising at highway speeds.

Risks: 2011 model year cars are projected to have as much as **300 million lines of code**. Automotive software is not currently required to meet rigorous safety standards commonly used in other industries. As a result, failures of automotive software have resulted in an indeterminate number of deaths and injuries. In many cases, as with the recent Toyota recalls, pinpointing the root cause of failures is difficult. Proposed legislation requiring automotive black boxes will help quantify the real root cause and frequency of occurrence of such failures. [6]

"Buggy software and the lack of government standards, along with a lack of uniformity in the auto industry, [has] put car

owners in the middle of safety issues with the multiple software systems that control their vehicles." [5]

Nuclear Power Industry:



Benefits: Nuclear plants use software for a variety of safety-critical and non-safety-critical applications. There are 104 operating nuclear plants in the US. Many of these plants were constructed in the 1970s-80s. Collectively, these plants have a remarkable safety record and have been providing low cost electricity for almost four decades.

Software for safety-critical systems is developed to meet rigorous regulations and standards. While there have been software failures in nuclear plants, no deaths or injuries have been attributed to such failures to date.

Risks: Risks associated with software used in nuclear power plants are relatively low because of the fact that most software was developed under stringent rules in compliance with regulations and standards. Even so, one study identified 30 reportable software failures at nuclear plants between 1990 and 1993. [1]

Banking and Financial Services Industry:

Benefits: Banks and financial services companies rely on software for managing your money and protecting your personal financial information. Software-based automated teller machines (ATMs) and recently announced smart phone applications provide customers with the ability to manage their funds whenever and wherever they are.

Risks: Providing personal financial information to banks and financial services companies has led to millions of cases of identity theft. Personal financial information has been stolen from ATM machines, banks, financial institutions, and retail and e-commerce sites. Costs for banks were estimated at $1 billion annually back in 2003. [8] Costs to customers following identity theft were estimated to be **$54 billion** in 2009. [9]

Aviation and Air Travel Industry:

Benefits: Software is used extensively in commercial airliners to control critical flight operations. Fly-by-wire systems which are currently used on many commercial airplanes has many advantages over the older mechanical and hydro-mechanical systems, including lower weight and improved safety. In addition, most fly-by-wire systems are implemented using four independent channels to prevent loss of signal conditions from compromising safety. Such software is developed under rigorous regulations and standards.

Software also plays a critical role in helping air traffic controllers manage an ever-increasing volume of air traffic.

Risks: Air travel has proven to be very safe compared to other modes of transport. However, the combination of increased density of airspace use and the development of planes capable of carrying large numbers of passengers, pose an increasing safety risk. [2] In addition, the current system used to manage the airspace over the US is based on technology from the 1970's. The FAA has been trying unsuccessfully to upgrade the air traffic control (ATC) system for many years.

As an example of the risks posed by ATC software, consider an event that occurred at the Los Angeles airport in September 2004. The ATC system failed and air traffic controllers lost radio contact with 800 airplanes they were tracking over the southwestern United States.

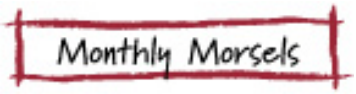
"The radio system shutdown, which lasted more than three hours, left 800 planes in the air without contact to air traffic control, and led to at least five cases where planes came too close to one another, according to comments by the Federal Aviation Administration reported in the LA Times and The New York Times. Air traffic controllers were reduced to using personal mobile phones to pass on warnings to controllers at other facilities, and watched close calls without being able to alert pilots, according to the LA Times report." [10]

In Summary…

One of the key findings of the **National Research Council's** study of software dependability is:

> "Avoidable software failures have already been responsible for loss of life and for large economic losses. The quality of software produced by the industry is extremely variable, and there is inadequate oversight in some critical areas. Unless improvements are made, more pervasive deployment of software in the civic infrastructure may lead to catastrophic failures. Software has the potential to bring benefits to society, but it will not be possible to realize these benefits - especially in critical applications - unless software becomes more dependable." [2]

'till next time…



Every month in this space, you'll find additional information related to this month's topic.

1. Chapin, D. M., et.al., Digital instrumentation and control systems in nuclear power plants: Safety and Reliability Issues, National Research Council, National Academies Press, 1997

2. Jackson, D., et. al., Software for Dependable Systems - Sufficient Evidence? National Research Council, National Academies Press, 2007

3. Leveson, N., Safeware – System Safety and Computers, Addison-Wesley, 1995

4. Wiener, L., Digital Woes - Why We Should Not Depend on Software, Addison-Wesley, 1993.

5. Germain, J. M., "The Gaping Hole Where Auto Software Standards Should Be", TechNewsWorld, March 18, 2010

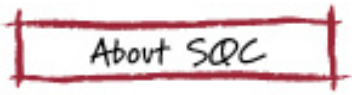
6. Whorisky, P., "Auto bill draft would require black boxes, allow NHTSA to issue quick recalls", Washington Post, April 30, 2010

7. Majchrowski, B., "Medical Device Software: Practical Guidance for Healthcare Facilities", presented at AAMI Conf, June 6-8, 2009

8. Sullivan, B., "ID thefts costs banks $1 billion a year", MSNBC, March 26, 2003.

9. Barrett, L. "Identity Theft Cost Victims $54B in 2009", e-Security Planet, February 12, 2010.

10. Broersma, M., "Microsoft server crash nearly causes 800-plane pile-up", Techworld, *September2004*



Software Quality Consulting provides a full-range of software engineering services for safety-critical industries and mission-critical projects. Our goal is to help create safety-critical and mission-critical software that meets our client's needs, complies with all applicable standards and regulations, with the highest level of quality possible, and in the most cost-effective and timely manner possible.

To learn more about how we can help your organization, **visit our web site** or **send us an email**.

Graphic design by **Sarah Cole Design**

Forward E-mail link
Safe UnSubscribe Link

Constant Contact logo