Software Quality Journal **Vol. 14 No. 2, 2006**

Editorial: What Makes a Software Failure a Page-One Story?

As I was pondering topics for an editorial, I began to scan some on-line newspapers for good stories that describe horrible software failures.

Alas, I found no good stories in that day's New York Times, Washington Post, Los Angeles Times, etc. Then I thought that surely the San Jose Mercury News would have news on some awful software glitch, but I had no such luck. Perhaps our software was not giving us problems this week, or perhaps the media is so used to hearing about software problems that they are no longer newsworthy.

The last headline story in my mind that day was the Sony music CD copy protection software that made serious modifications to the Windows operating system which include leaving security holes in the system. But that was last week's news (although it is a continuing story).

After the major newspapers failed to provide good ideas for an editorial, I visited the most reliable source for stories about unreliability, the Risks forum (www.risks.org). Virtually all software failures are described in the Risks forum before they reach the popular press. Let's take a look at some of the headlines from the forum on Thursday, 17 November 2005:

> "Software bug crashes Japanese stock exchange."
>
> "Flight Booking System Can't Recognise February 29."
>
> "Fun with Daylight Saving Time: … One wonders how well the embedded time-aware code in most electronic equipment will handle this."
>
> "Computer Glitch Lets Prisoners Out Early: … Some prisoners were also let out too late, which is just as bad."
>
> "Freddie Mac profits misstated due to software error."
>
> "Some Fast Lane accounts double-billed."
>
> "Sony CD DRM Blow-Up Continues -- Recalls Ordered, Lawsuits Possible": this story continues.

Stories on the Risks forum tend to be precursors and follow-ups to widely reported stories. These stories reach the mainstream press, since reporters also scan Risks forum stories.

What is it that these reported problems have in common? For one thing, the failures of these systems had enough impact, or potential impact, on people to be reported. What makes a failure of great enough importance to report?

A crashed stock exchange surely has enormous financial impact, since these systems handle a mind-boggling number of financial transactions every second with a value much greater than my lifetime income

A system that cannot book a flight on a particular date represents the kind of enormously frustrating computer interaction that we have all experienced in some way recently.

The daylight savings concern – it's not a failure yet – represents common worries that embedded computers are all around us, and are just waiting to fail in unison. This story reminds me of the year 2000 panic.

A glitch that opens up jail cells is frightening; it can lead to a scenario with entire prisons emptied out. It was lucky that only a few prisoners were released. In a science fiction story, the systems would have conspired, via the internet of course, to release all prisoners throughout the world, in an unintentional, or intentional, universal amnesty.

The Freddie Mac error may just surprisingly help corporate executives misstate profits. This failure needs further study to ensure that it is a computer software error, and not something else.

Fast Lane is a system to automatically bill driver's accounts when they pass a toll station on a toll road. Double billing sounds like an unintentional programming error, but why don't they ever under-bill?

The last story, the Sony copy-protection software is probably the most annoying error, at least to computer professionals. How dare Sony insert what turns out to be "malicious" software into the innards of our operating systems, without our permission? Now the problem forced Sony to recall all music CDs sold with the copy protection software, and there are a lot of these CDs.

Such failures show how vulnerable we are to computing systems. Our vulnerability is what makes them good stories. Of course, there are many other software problems that do not become "stories", they just frustrate and annoy us. Or perhaps, we just haven't realized the true risks.

These reported stories demonstrate the potential for software errors to result in large liability, repair costs, and embarrassment. The press represents the embarrassment, and the exposure that increases liability.

So what makes a software failure reach page one? I see several key factors that contribute to a failure story reaching the media:

- Significant financial loss.
- Human injuries and/or suffering caused by software.
- Mass frustration due to the failed software or unusable interfaces.
- Fear of technology, and failures that feed common fears of technology.
- Failure scenarios that would (if they were not true) make good fiction.
- Suspicions that the "failures" may mask fraud.
- Unexpected damage to our personal property, such as our computers, due to software "features".

The stories that reach page one are those that feed our imaginations.

To the software quality community these stories have one common lesson. Professional software assurance activities could have identified most of these faults before the systems failed. In most cases, either testing or analysis should have discovered the problems. In some cases, particularly the Sony music CD problem, an outside consultant could have identified the severe risk of this, now clearly insane, copy protection scheme.

We in the software quality research and practice community know that number and severity of software failures can be reduced. Independent assurance activities could have found most of these reported problems before they became press stories.

As computer systems become more and more interwoven with aspects of human life, publicity of software failures will also increase. The need for software assurance research and practice will only grow.

James Bieman
Fort Collins, Colorado
U.S.A