Software Quality Journal **Vol. 16 No. 4, 2008**

Editorial: Election Software Exposed

James M. Bieman

Software quality problems are most exposed to the general public when failures occur in applications that are essential to social or economic processes. The problems are especially noticed when the processes are watched carefully by the news media. During the 2008 election process in the USA, media attention has focused on every election-related event, comment, nuance, and rumor. Problems in software that supports elections are most certainly noticed.

Each year software systems play a greater role in conducting elections. These systems are used to tally votes and track voters. They tend to be distributed and operate in real time, and their correct performance is critical. Failures of all kinds are reported as front page news, and rumors of failures spread widely. Election software demonstrates to the public the quality, or lack of quality of our software systems. And because democratic institutions now depend on election systems, the quality of these systems is an indicator of the strength of our society.

I am writing this editorial in late May 2008, which is far from the beginning of the election period in the USA, but the end is (hopefully) only four months away. This is a good time for me to make some predictions about how election software will perform in the November 2008 elections. By the time that you read this in print (the electronic publication will be posted well before the elections), you will know if I am a good prognosticator (Bieman, 2006).

To improve my odds, my predictions will be based on (almost) certainties. I will extrapolate from incidents that have already occurred. Because the responses to many of these incidents will not, in my judgment, solve the underlying problems, similar incidents will surely occur again. The consequences of these incidents are much more difficult to predict. Few people predicted ahead of time that bad user interfaces on touch screens made it impossible to know if the votes that were tallied in 2000 matched the intent of the voters. Because of the closeness of the election in Florida, these and other problems resulted in an election that is still contested by many.

So, here are my predictions of incidents that will occur in some precincts, districts, cities, counties, and/or states this year:

1. Registration systems will incorrectly determine who is a valid voter. Some of the databases will have inaccurate data due to data entry errors such as transposed data, problems with unusual names, and some of the systems will just fail (Hastings, 2008). Somewhere incorrect versions of a database will be used. Some valid votes will not be counted, and results will be delayed due to the evaluation of provisional ballots.
2. Large turnout will overwhelm the systems resulting in long lines and frustrated voters, and some of these voters will give up and not vote. Due to poor

software design and a lack of stress-testing systems will hang as happened in Denver in 2006, where "voting center delays -- with waits in some places of up to three hours -- forced an estimated 20,000 voters to abandon their efforts to vote on Election Day" (Weiss, 2008).

3. Some systems will generate inconsistent results. They will report more votes than voters as happened in a New Jersey primary where "tapes showed that the Republican ballot was activated 60 times, even though a total of 61 votes were cast for Republican candidates … It also said that the Democratic ballot was activated 362 times, yet a total of 361 votes were cast for Democratic candidates." Other systems will lose votes as apparently happened in Florida in 2006 when "a congressional race was decided by fewer than 500 votes even as the system recorded that some 18,000 voters had left the polls without casting a vote on the question" (Noyes, 2008). Some votes will be allocated to the wrong candidate or the wrong race as happened in Arkansas this year when election systems "allocated votes cast in one race to an entirely different race that wasn't even on the electronic ballot This problem resulted in the wrong candidate being declared the winner" (Zetter, 2008). Election officials noticed the inconsistencies, and the results were corrected after a recount. However, the incorrect results would have been certified if not for the vigilance of the officials and a paper audit trail generated by the system.
4. User interface problems will cause votes to be recorded for the wrong candidate. There were many reports of such problems related to the 2000 election in Florida precincts due to the layout of candidates on ballots that used touch screen interfaces. Other problems have been reported more recently in San Antonio due to overly sensitive touch screens where a voter discovered that the wrong "vote was cast because he inadvertently rested his hand on the screen of the voting kiosk while using his other hand to vote" (Anon, 2004). This voter caught the error on the review screen, but surely some voters missed similar errors.

This is a short list, and I am sure that there will be some other incidents (another prediction!). For example, I have not included security incidents in my list of predictions. That is because, as far as we know, the prior snafus have not been caused by security beeches. However, numerous security flaws have been reported including election software vendors "parking files on an unprotected public Internet location" (Harris, 2003), and passwords that matched the name of the system vendor (Harris and Wynn, 2005). Code reviews of election software produced by three vendors revealed serious security risks caused by inadequate software designs. Only "radical changes to the software and architecture" can reduce these risks (Blaze, 2007). Security events can certainly occur as election software has captured the attention of hackers (McWilliams, 2003). However, my predictions of problems during this year's elections are not based on security risks.

The reported problems that have occurred during actual elections do not appear to be caused by malicious behavior. Rather, the apparent cause is poor software development practices including inadequate design and testing as well as inexperienced developers. I

suspect that the process maturity level of the current crop of election software is “ad hoc” at best. So, what could help to improve the quality of election software?

I do not advocate requiring process assessments, such as the CMMI, of all venders. Rather, I suggest that election software should be treated like other election processes. The software should be open for public inspection by everyone.

Traditionally, in the USA, political parties are allowed to have poll watchers, who represent political parties and candidates, to observe the voting process. In addition, observers are allowed to watch the vote counting process (Anon, 2008). In general, vote counting processes are well documented and visible. This openness provides greater confidence that an election is conducted fairly.

With computerized voting and vote tallying, the election processes are encoded in software. The only way to provide visibility of the process is to allow observers to examine the software, including source code, and supporting hardware mechanisms. In effect, I advocate an open-source software, and open-hardware policy.

Election system companies have blocked the inspection of their systems claiming that valuable trade secrets are at risk of exposure (Paul, 2007). The use of claims of “trade secrets” to hide election system behavior and design is rubbish. There should be no right to hide shoddy designs in public systems. The right to open up election software to inspection can and should be enacted into law. The law should state that officials could only install systems from companies willing to let the world see their designs. Proprietary rights can still be protected via copyrights and patents. It is hard to trust unexamined systems, especially systems that handle critical political processes.

The full exposure of election software to open inspections can only lead to more accurate and trustworthy elections. No system will be foolproof, especially if the elections are very close. However, I will be much happier if the software (and hardware) is kept exposed and naked. What do you think?

References

Anon. 2004. You touch it, you voted for it. *San Antonio Business Journal*, October 21, 2004.

Anon. 2008. Guiding Principles of Vote Counting. *ACE Encyclopedia. The Electoral Knowledge Network*, (http://aceproject.org/ace-en/topics/vc/vc20).

Bieman, J. 2004. The role of prognostication in software design. *Software Quality Journal*, 12:1(7--8).

Blaze, M. 2007. California voting systems code review now released. *Matt Blaze’s Exhaustive Search: Science, Security, Curiosity*, August 2, 2007 (/www.crypto.com/blog/ca_voting_report).

Harris, B. 2003. Voting system integrity flaw. *Scoop Independent News*, www.scoop.co.nz, February 5, 2003.

Harris, B. and Wynne, K. 2005. Hack of real-life voting system. *Black-BoxVoting.org*, March 9, 2005 (http://www.bbvforums.org/forums/messages/1954/3826.html)

Hastings, D. 2008. Inaccurate voter rolls could delay some results. *The Boston Globe*, March 3, 2008.

McWilliams, B. 2003. *New security woes for e-vote firm*. Wired, August 7, 2003.

Noyes, K. 2008. Voting 2.0, Part 1: The trouble with closed systems. *LinuxInsider*, May 23, 2008.

Paul, R. 2007. Court: Protecting trade secrets takes priority over election transparency. *ARS Technica* (http://arstechnica.com/news.ars/post/20070625-florida-appeals-court-says-trade-secret-protection-takes-priority-over-election-transparency.html), June 25, 2007.

Weiss, T. 2008. Report blames Denver election woes on flawed software: local election officials were also slammed for a `casual approach to technology'. *Computer World*, December 13, 2006.

Zetter, K. 2008. Arkansas election officials baffled by machines that flipped race. *Wired Blog Network*, (http://blog.wired.com/27bstroke6/2008/05/arkansas-voting.html), May 29, 2008.