BRIEF COMMENTS ON:

1. EAC Unified Testing Initiative: Pilot Program & Discussion, January 2009, available at http://www.eac.gov/News/docs/presentation-day-1-unified-testing-initiative-overview.pdf/attachment_download/file
2. EAC Research Areas for the TGDC VVSG Recommendations – January 2009, available at http://www.eac.gov/program-areas/voting-systems/docs/nist-response.pdf/attachment_download/file
3. Applicability of VVSG Recommendations for Inclusion in an Amended VVSG 2005 -- DRAFT August 2008, ***not obviously available on the web.***
4. The EAC's Voting System Test Laboratory Program Manual, available at http://www.eac.gov/files/voting_systems/VotingSystemTestLabProMan.pdf
5. The EAC's Voting System Testing & Certification Program Manual, available at http://www.eac.gov/program-areas/voting-systems/voting-system-certification

SUBMITTED FEBRUARY 20, 2009 TO THE
UNITED STATES ELECTION ASSISTANCE COMMISSION

Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology
kaner@kaner.com
Member:
Election Assistance Commission
Technical Guidelines Development Committee

MY CREDENTIALS FOR COMMENTING ON THESE DOCUMENTS ARE AS FOLLOWS:

I am a member of the TGDC, nominated to the position by the Institute for Electrical & Electronics Engineers, the world's largest professional society involved in Electrical, Computer and Software Engineering.

> *NOTE: My analysis of these documents has been informed by discussions in IEEE's Standards Coordinating Committee 38 (Voting Standards). However the opinions expressed in this note are my own and have not been written to specifically reflect the views of SCC38, IEEE or Florida Institute of Technology.*

As Professor of Software Engineering at the Florida Institute of Technology, I head Florida Tech's Center for Software Testing Education & Research and am Principle Investigator on two projects funded by the National Science Foundation. I hold a doctorate in experimental psychology and have years of industrial work experience as a software human factors analyst, a software developer, tester, development manager and consultant, prior to my return to university to teach and research. I am the senior author of three books on software quality. One of them, *Testing Computer Software*, is (according to John Wiley & Sons, my publisher), the bestselling book on software testing in the history of the field.

I am an attorney whose work has focused on the law of software quality. I am a member of the American Law Institute's Members Consultative Group for the Principles of the Law of Software Contracts.

I am a senior member of the IEEE and of the American Society for Quality, executive vice president of the Association for Software Testing, and a member of the American Psychological Association, the American Psychological Society, the Human Factors and Ergonomics Society, the American Bar Association, the American Law Institute, the California Bar Association, the International Technology Law Association, and the Association for Computing Machinery.

OVERALL REACTION TO THE DOCUMENTS

Until this week, I was not aware of the existence of the first three documents and I expected testing and lab certification requirements (as documented in the manuals) to change substantially as VVSG-2007 was implemented.

The request from EAC to NIST to consider dropping Software Independence materially changes the VVSG-2007.

Without a clear requirement of Software Independence, the criticisms of VVSG-2007 from IEEE would have been sharper and more extensive. I also believe, based on discussions with the Voting Systems SIG chair, that the prominence, clarity and content of the requirement for Software Independence was very important to the Association for Software Testing's review of VVSG-2007. Many people believe this was the single most important enhancement to the VVSG.

An enormous amount of work went into VVSG-2007. It was far from a perfect document, but it was a substantial improvement over VVSG-2005. As far as I can tell, the core of that work is not reflected in the new plans. If this is what EAC plans to take forward, I think the work on VVSG-2007 was wasted. It is also not clear that there is any plan for actually improving the security, reliability or auditability of voting systems.

Given how little of TGDC's work is being proposed to be carried over into the new scheme for amending VVSG-2005, it is not clear why TGDC exists.

SPECIFIC COMMENTS

1. The **EAC Research Areas for the TGDC VVSG Recommendations** asks NIST to develop alternatives to Software Independence. In particular, it suggests that NIST subordinate the principle of Software Independence (SI) to that of auditability. SI would be one way of achieving auditability, rather than a requirement in itself.

Software Independence is defined as that "quality of a voting system or device such that a previously undetected change or fault in software cannot cause an undetectable change or error in election outcome." (VVSG-2007 Appendix A). To achieve software independence, it is necessary to design the system in such a way as to ensure that:

(a) The system will not mis-present choices to the voter, inducing an incorrect vote from the voter, and
(b) if there were a complete audit, any error made by the voting system in receiving or processing the voter's votes would be detected.

In contrast, an audit is merely a "verification of statistical or exact agreement of records from different processes or subsystems of a voting system." In normal usage, a definition of auditability would be the quality of a product or system that enables persons to perform an audit effectively and efficiently. To achieve auditability, a system need only provide features that enable or support the task of auditing. This in itself provides no assurance that a complete audit would expose all errors. Audits are rarely exhaustive and as we have so often seen in the financial systems, significant numbers of errors or irregular practices can escape the notice of a diligent auditor.

My understanding is that the shift from software independence to the more vague notion of auditability was made at the behest of vendors and test labs who consider the standard of software independence too high or too difficult to verify.

The shift from the precisely-described *software independence* to the more vague notion of *auditability* will have an impact on software development. When a requirement is described specifically and clearly, a developer/designer/manager who understands that it is a fundamental requirement will ask at every point in design and implementation,

> *"If there is an error in this part of the product or process, how will an auditor be able to determine that the result is wrong?"*

This is the question we want developers to ask. It is the result the public wants, or should want, in every voting system.

The less specific the guidance, the more interpretable (vague) the requirement, the less the impact we should expect to see on the fine grain of the code.

It appears that NIST is attempting to salvage much of software independence via a redefinition of auditability. **EAC Research Areas for the TGDC VVSG Recommendations** defines auditability as the "quality of a voting system or device such that any error in its recording of votes or vote totals, whether randomly occurring or maliciously induced, is detectable."

I have not seen the new NIST definition in other voting-related documents and it does not have a close analog in discussions of audits and auditability that I have read in legal, accounting or quality-related auditing literatures.

It is hazardous to redefine a widely-understood term with an unusual and precise technical definition, *especially* when many of the key stakeholders and decisionmakers will be unfamiliar with whatever regulation or standard redefines it.

Except, perhaps, that the distinction between software independence and NIST-auditability might trim away a software-independence ban on user-interface errors that induce an error in *casting* the vote (which is then correctly counted), it appears to me that software independence and NIST-auditability are the same. If software independence is unachievable or incomprehensible, so is NIST-auditability.

The vendors and labs did not protest against "software independence" because they don't like those two words and would prefer to test against an equivalently defined "auditability." The pressure that encourages substitution of "auditability" for "software independence" will provide an ongoing demand to apply the common, and more forgiving, meaning of "auditability."

Let's look again at *software independence*:

> If there is a way that an error (or a malicious revision) in the code can yield an error in the outcome of an election, and we cannot tell that the election is wrong unless we know about the coding error:
> (a) the system is not software independent
> (b) we cannot be sure that the reported outcome of the election is the actual outcome of the election, and
> (c) we cannot provide voters with empirical proof that an election outcome is trustworthy, and
> (d) we cannot be sure that a hostile power (organized crime or another nation) has not tampered with our election results to suit their needs.

Whatever you call it, in any jurisdiction that uses software-controlled voting or vote processing, software independence is a fundamental requirement of a functioning democracy.

EAC should reaffirm the requirement of software independence in voting systems and reject NIST-auditability as a dangerously ambiguous alternative.

If the underlying problem is that the current generation of vendor code cannot meet this requirement, a reasonable solution would be to stage the introduction and enforcement of different requirements from the same standard. Thus, the new standard might require usability improvements by the end of 2010 but not software independence until 2011.

It is better to state a requirement explicitly today, with an enforcement date, than to abandon the requirement "for now" until vendor code gradually improves. If software independence is not made an requirement, there is no reason to expect it to be any easier to retrofit into systems in 2020 than in 2010.

2. The five documents do not discuss support for the election auditor in much detail, but the EAC should be intensely concerned with this, whether or not Software Independence is kept as a system requirement.

One of the flaws that I raised in my review of VVSG-2007 is that it does not treat the auditor as a distinct type of user of the system, for whom the system should be made usable.

It is easy to meet the letter of a task requirement (in this case, traditionally-defined auditability) while providing a feature set that is expensive to use, difficult to use, prone to human error, and unsupportive of public observation of the task being conducted.

Audits are the key empirical tool available to the State to demonstrate to its citizens that votes were collected and processed accurately and therefore that the result can be trusted. The ability to make such a demonstration is fundamental to a democracy. If the public cannot trust the results of an election, it has no reason to accept the legitimacy of the elected government. Down this road lies chaos.

The documents I am reviewing in this memo use the phrase "auditability" frequently but do not propose mechanisms for ensuring that the auditing capabilities are usable.

- ***Over the short term, EAC should require voting system assessments to include an assessment of the human factors associated with auditing, performed by competent human factors specialists.***
- ***For the longer term, EAC should commission research on usability of voting systems by auditors and should create a standard to guide implementation and assessment.***

3. The **EAC Research Areas for the TGDC VVSG Recommendations** asks NIST to consider ways to separately test and certify components of a voting system.

I agree with EAC that it is highly desirable to separately test and certify the components of any system, including a voting system. The more trustworthy and reliable we can make the individual parts, the easier and cheaper it is to thoroughly test the system as a whole.

I also agree that it is highly desirable to specify a common data format and other information needed to make components interoperable. This will reduce design and testing costs.

As the response from NIST notes, however, any voting system built from those components must be tested as a whole.

There is a risk that must be managed.

If a local election official believes that a reliable voting machine (or collection of machines) can be made by snapping together reliable components, what is to stop that official from creating a unique, untested combination, for example to jury-rig a functioning voting machine when one or more machines fail in a precinct on a busy election day? The incorrect but common expectation that reliable parts will snap together into a reliable combination is not an issue for current VVSG, which rejects certification of components.

This risk must be addressed with great care in any VVSG that enables certified components. That might be difficult because in practice, the risk might be an issue of local election-day management practices rather than voting system design.

If mitigating a clear risk arising out of component certification is outside of the scope of VVSG, should VVSG approve component certification and accept the unmitigated risk or should it bar component certification until processes are in place for managing the conduct of elections?

4. **The EAC Unified Testing Initiative** proposes the elimination of independent state certification and independent local acceptance testing of voting systems.

Software testing is unlike the testing of a series of identical manufactured units. It is more like empirical evaluation of a design than like manufacturing quality control.

In manufacturing QC, we have a finite pool of types of errors to test for. We can create a best method for testing for these errors and then we can repetitively apply the same test to each unit. For manufacturing QC, this is competent testing and it can be optimized via statistical control.

In contrast, in design assessment (and in software testing), a defect that is found in one unit will be found in all units designed the same way. Repetition of the same test is waste. The pool of potential tests for any nontrivial program is infinitely large because a failure might be triggered under one set of circumstances but not under similar ones. Variation of testing is essential.

Design assessment is often driven by failure mode and effects analysis (FMEA). That is, a collection of people rely on historical data, product specifications, and their own wits, to imagine "all" of the ways a product can fail. They consider the potential harm associated with each possible failure. They then create tests to check for those failures that can have serious consequences. FMEA is a widely used design assessment technique; it is standard practice for life-critical devices and systems. The software testing analog to FMEA is risk-based software testing. As with FMEA, different people will imagine different risks and create different tests.

In practice, many serious errors in voting systems have been revealed in state-level tests. This might not be the result of incompetent federal-level testing that can be fixed by replacing federal tests with more rigorous state tests. (This appears to be the thesis of **The EAC Unified Testing Initiative.**) It is a natural outcome of variation. The testers in different states imagine the problem space, the set of potential risks, differently from each other and from the federal testers and so they find different problems.

Merging the state and federal testing to create a single unified testing series eliminates the variation across different test labs. The predictable result is that more defects will be undetected by testing. Unless we find some other way to replace the variation, this is a serious error.

5. **The EAC Unified Testing Initiative** and the **Applicability of VVSG Recommendations for Inclusion in an Amended VVSG 2005** eliminate the requirement for exploratory security testing.

VVSG-2007 proposed that security testing include exploratory testing (coining a new name for exploration, "open-ended vulnerability testing.")

Exploratory testing is often risk-focused. When done by knowledgeable testers, it is more efficient than traditional testing methods because the testers:

- spend much more time pursuing risks, modifying what they do as they interpret the results of the tests so far,
- spend much less time documenting their work and more time looking for problems
- spend almost no time repeating tests they have run before (regression testing is a weak way to expose new problems)
- abandon lines of attack that appear unproductive, in favor of attacks that seem more likely to yield results.

On a critical project like a voting system, exploratory testing is extremely useful, but I do not suggest that it is sufficient. Well-documented tests that demonstrate coverage provide a historical record of *some* of the testing. It is useful for project control and for failure analysis in the event of bad events in the field.

The new documents appear to eliminate most of the security-focused testing that was written into VVSG-2007 including exploratory testing.

The demand for a rich set of security tests and other security-protective features came from the perception that it is important to manage security-related risks associated with voting systems. As far as I can tell, those risks have not recently diminished. Nor have the skill sets of programmers who might be employed by criminal organizations or foreign agencies to hack our elections gotten recently less sophisticated. Security of our elections remains a gravely serious problem. Security-related concerns have led to skepticism among many voters that reported election results do not actually reflect the votes cast by the voters. Mitigating these risks, in a way that gives voters reasonable confidence that the risks have in fact been mitigated, is a vital need.

My impression is that exploratory testing was removed at the behest of vendors and test labs who protested that it is too hard and too unpredictable.

Vendors who are used to highly structured, lavishly documented tests are likely to protest that exploratory testing is too unpredictable. When different tests are run every day, including many tests the system has never seen before, who knows what can happen? The system might fail in response to any of the new tests, at any time. Thus, the schedule might become unpredictable.

Unfortunately, when the product goes into use in the field, the election officials will create their own ballots--not the ballots described in the well-documented lab tests. They will present their own voters to the voting system--not the lab's testers who simulate voters while following a carefully written script. Thus, every system will face new test conditions as soon as it is put into production.

Which do we prefer?

- ***Find the bugs while running an actual election?***
- ***Or find them beforehand, in the lab?***

If schedule uncertainty exists because tests are exposing bugs in the code, managing this uncertainty by eliminating the testing doesn't eliminate the bugs. It leaves them in the system, to wreak their havoc on a real election instead of a simulated one.

The set of software tests that we run against a product is a tiny sample from the near-infinite pool of different tests that (in theory, with unlimited time and money) *could be* run. The goal of test design is to help select the most effective sample (the best group of tests) from the pool.

Formalized testing, of the kind done by test labs, involves formalized test documentation (produced at high cost) which has to be revised (at cost) every time the product changes enough to require a change in the test. With so much money spent on the many components of the test document set (test plan, test designs, etc.), less money is available for the actual task of running the tests and finding and reporting the errors. The result is a relatively small number of thoroughly documented tests.

Exploratory testing reflects a fundamentally different sampling strategy. It dispenses with the time-consuming formalities, using a human-guided adaptive strategy to go as directly as possible for the software's virtual throat.

In comments on VVSG-2007, I proposed that there should be exploratory testing for ***all types*** of software errors, not just security flaws. Unfortunately, the **EAC Unified Testing Initiative** goes in exactly the opposite direction, sacrificing variation (in security testing as well as traditional testing) in order to save costs and achieve a more predictable schedule. This is a serious error.

6. **A better way to introduce variation**. In my comments on VVSG-2007, I conveyed the views of IEEE SCC38's members that the artifacts associated with the voting system, including source code, tests and test results, should be made public and that members of the public should be allowed to obtain systems for testing (at their expense).

Given access to equipment and development artifacts, several universities would set up laboratories to test voting systems. Several foundations would donate funds to help set up such labs. In addition, based on my experience reviewing proposals for the National Science Foundation, I believe that there are several initiatives within NSF that would provide funding to establish such labs. Other organizations might establish labs independent of universities.

Making code visible to the public does not mean surrendering copyright in the code. Open Access licenses allow a person to read the code, test the code, even modify the code for their own use, but not distribute the code to others.

Given an Open Access regime, public testing can be peer reviewed and improved. Given the socially critical nature of the voting system, extensive public testing is likely. Therefore, given an Open Access regime, the amount and cost of formal lab testing could be reduced significantly, even in the way proposed in the **EAC Unified Testing Initiative**, because substantial additional variation would come from competent testing groups at universities and private labs.

7. An additional note about **The EAC Unified Testing Initiative.**

It is unfortunate that the only representation of voters in **The EAC Unified Testing Initiative** featured two statements from the voters: (1) "It's my choice to be a stupid voter" and (2) "Shiny objects distracted me and I voted Buchanan." The first quote needs no direct comment. As to the second, my understanding is that the butterfly ballots were sometimes misaligned, making it difficult for any voter to determine which hole to punch for which candidate. There are certainly plenty of photos on the web that demonstrate how confusing a misaligned ballot would be to any person and that appear to demonstrate misalignment on some of these voting machines.

This picture could convey, to a reasonable reader, a sense that the authors of this proposal hold in contempt both the population of voters and the research on the human factors of voting. Whether or not such an interpretation would be correct, perhaps this is not an impression for the United States Election Assistance Commission to convey.

8. The **Voting System Test Laboratory Program Manual, Version 1.0** includes a long section on **Conflict of Interest and Prohibited Practices Program** and another section on **Laboratory Independence**.

I submitted the following comment as part of my comments on VVSG-2007:

> VVSG specifies the testing that an independent test lab will perform. The first problem with this process is that the equipment vendor picks the test lab. This creates a strong incentive for the test lab to please the vendor, in order to obtain the vendor's repeat business and the business of this vendor's competitors. The labs therefore have a disincentive against creating tests that are more harsh or more extensive (more time consuming and more expensive) than the bare minimum specified in VVSG. This is not a genuinely independent set of tests and it is a poor way to engender public trust in the test results.
>
> VVSG must address and eliminate the problem of test lab conflict of interest.

Unless I misunderstand the **Voting System Test Laboratory Program Manual, Version 1.0**, this problem has not been addressed. The equipment vendor still chooses and pays the "independent" test lab. This creates an interest, on the lab's part, in pleasing the vendor. The duty to test thoroughly and competently, backed by NIST Certification, creates a conflicting interest.

The vendor should not be allowed to select or pay the independent test lab. Payment for lab services should instead be directed to an intermediary (such as EAC) who in turn pays the lab and bills the vendor on the lab's behalf.

Genuinely-independent lab testing does not eliminate the need for exploratory testing. It still focuses on a relatively small number of tests that are thoroughly planned and documented.

IN CLOSING

I hope these unsolicited comments are helpful. I would be glad to answer any questions or provide additional examples or details.

Cordially

Cem Kaner