Association for Computing Machinery

Advancing Computing as a Science & Profession

Comments on Voluntary Voting System Guidelines
U.S. Public Policy Committee of the Association for Computing Machinery (USACM)

May 5, 2008

On behalf of the U.S. Public Policy Committee of the Association for Computing Machinery (USACM), we are submitting the following comments on the Voluntary Voting System Guidelines (VVSG) released by the Election Assistance Commission (EAC).

With over 88,000 members, ACM is the world's largest educational and scientific computing society, uniting educators, researchers and professionals to inspire dialogue, share resources and address the field's challenges. USACM acts as the focal point for ACM's interaction with the U.S. Congress and government organizations. It seeks to educate and assist policy-makers on legislative and regulatory matters of concern to the computing community.

USACM is a standing committee of the ACM. It tracks US public policy initiatives that may impact the membership of ACM and the public at large, and provides expert input to policy-makers. This input is in the form of non-partisan scientific data, educational materials, and technical analyses that enable policy-makers to reach better decisions.

Members of USACM come from a wide-variety of backgrounds including industry, academia, government, and end users. Our goal in this review is to ensure technical feasibility and accuracy, best practices, and promotion of voter confidence in election

results while protecting potential vendors from unduly onerous or vague requirements, and also providing reasonable, actionable statements for local and Federal officials.

We have submitted our comments on specific recommendations through the online submission website. Those comments are also included here, following our general comments about the VVSG. We also include a glossary, and a proposed test for determining whether voting systems produce truly independent voter verifiable records. This test is meant as a hypothetical example, and is not an endorsement of a particular technology or technologies for use in voting systems.

We also note that the technology in this field is quite dynamic, and the issues quite complex. Even if all of our suggestions were accepted there will be issues yet to be addressed in the near future. We encourage the EAC to be proactive in anticipating changes that may present problems for accurate, safe voting, and to revisit these guidelines in a timely fashion.

Introduction

USACM strongly supports efforts to ensure that all voting systems — particularly computer-based electronic voting systems — embody careful engineering, strong safeguards, and rigorous testing in both their design and operation. The development and implementation of comprehensive voting system guidelines — including this effort by the Election Assistance Commission (EAC) and its Technical Guidelines Development Committee (TGDC) — are an important part of ensuring that elections are accurate, reliable, accessible, secure, and verifiable. We applaud the efforts of the EAC and TGDC to develop this edition of the VVSG. It represents a complete rewrite of previous editions, including some provisions that had not been reviewed since the 2002 Voting System Standards developed by the Federal Election Commission. The staffs of the EAC, the TGDC, and the National Institute of Standards and Technology are to be commended for their work.

We urge the EAC to adopt the TGDC recommended text with some modifications and clarifications, as described below and through the online comment system. With the passage of the Help America Vote Act (HAVA), systems were rushed to market without the benefit of a comprehensive set of Federal guidelines. This new edition of the VVSG is a welcome step forward.

Given the nature of current electronic voting systems, where security, accessibility, reliability, usability and privacy were not designed in from the beginning, some tensions in developing and finalizing this VVSG are unavoidable. A good guiding principle is to focus on desired election principles rather than desired election technologies. For instance, maximizing voter participation would be a good election principle, and strong usability and accessibility standards would be a means to support that principle. The goal is to respond to as many constituencies as possible. To show favoritism to one would preclude perfectly reasonable standards because a small percentage of voters are inconvenienced. To focus on specific technologies used in voting narrows the scope of this discussion — and this document — too much to be effective.

The concern over unverifiable voting machines is widespread throughout the computing community. In a 2004 poll that ACM conducted of its members, 95 percent of those responding indicated that they have serious concerns about electronic voting machines — concerns that should be addressed with specific safeguards. In an effort to bring the community's concerns to policymakers, ACM adopted a policy statement in support of physical audit trails, which a voter could inspect to verify their vote. The principle of software independence — as defined in the VVSG — is encouraging because it embraces the notion of being able to verify the results of an election independent of the machines used to cast the ballots. Another development in the VVSG — the innovation class — is also heartening. The VVSG should encourage technological innovation in voting systems, and welcome the innovation class as a means for ensuring that new devices and

new voting systems can be effectively tested to ensure that they can provide accurate, reliable, accessible, secure, usable, auditable, and verifiable elections.

While we appreciate the opportunity to provide comments on the VVSG, we are concerned about the timeliness of implementing this edition of the VVSG. As elections continue to be close in several jurisdictions, criticism — warranted and unwarranted — will be levied against voting systems. When problems or unusual results leave an election in doubt, conducting a transparent and credible recount becomes extremely difficult, leaving election officials with no choice but to conduct a revote or accept the existing results. Audit trails (if they exist) that are not physical may not accurately reflect the votes cast when undetected errors or tampering alter the outcomes of elections. The resulting lack of certainty in the results, especially in close races, not only undermines the accuracy of the vote, but also may serve to diminish citizen confidence in the fairness of the process. If this confidence is not strengthened by the time the next VVSG is implemented, we face the risk that the public will turn away from electronic voting systems — in whole or in part.

Software Independence

We have mentioned our support for the principle of software independence described in the VVSG. We include with our comments the letter[1] we sent to the then-Chairman of the TGDC, Dr. William Jeffrey, expressing our support for Software Independence and other recommendations made to the TGDC. Given the shortfalls of security testing, it is our long-standing belief that voting systems should also enable each voter to inspect a physical (e.g., paper) record to verify that his or her vote has been accurately cast and to serve as an independent check on the result produced and stored by the system. We are pleased that the TGDC recommends that voting systems must have an independent way of verifying a voter's selections.

[1] http://usacm.acm.org/usacm/PDF/USACMCommentsSTSPaper.pdf

An important part of ensuring a software independent system is developing both an effective test and definition for determining software independence. We find both lacking in this version of the VVSG. We recommend that you define software independence as meaning that an error or fault in the voting system's software is not capable of causing an undetectable change in election results. This will help provides state and local elections officials, as well as vendors, with the knowledge they need to help ensure that their systems are software independent. Without a specific test or a more specific definition, other groups will object to the principle on the grounds that the concept is too vague and indistinct to be effectively implemented. Given that many states currently do not conduct effective post-election audits, there is a need for software independence, together with clear guidance as to what makes a voting system software independent.. We recommend you include in the VVSG a process akin to the hypothetical example we outline in Appendix B — a process that demonstrates both the production of Independent Voter Verifiable Records and Software Independence.

Innovation Class

USACM supports the concept of the innovation class. However, we note that there has been a substantial amount of confusion about the scope of the innovation class, and the application and process associated with being certified under the innovation class. There is some question as to whether software dependent machines could be certified under the innovation class and whether the class could be applied to other voting devices not strictly related to achieving software independence. We recommend that the VVSG maintain a consistent strategy of only sanctioning voting systems in which a software fault cannot cause an undetectable error in election results, whether the system is evaluated under the well-defined software standard or the more progressive innovation standard. Put another way, innovation class systems should adhere to the software independence requirement.

Regarding whether the class applies to a broader array of voting devices, our understanding is that the innovation class would only be focused on specific devices meant to achieve software independence. If it is the TGDC's and the EAC's contention that the innovation class is broader than that, it should clarify the application of the innovation class and detail the process involved in being certified under it.

We also are concerned that the current testing process for the innovation class is vague. Without a more definitive testing procedure this process runs the risk of being an end-run around the standards. We recommend that the VVSG include a specific test or several kinds of tests to demonstrate that the innovation class submission can produce secure, accessible, reliable, and verifiable election results equal to or better than voting systems that can be tested to the VVSG without the innovation class. In addition to describing these tests, there must also be some description of the experts and process involved in judging two things: whether the device or system in question must apply for certification through the innovation class, and whether that device or system should be certified. To simply refer to a review panel process is insufficient.

Transparency and Feedback

Given the complex nature of voting systems — and computer systems in general — it is not uncommon for some problems to arise after the testing phase is over and the systems are operational. This is particularly true when systems are scaled up in size, such as the expansion from precinct systems to citywide voting centers that Denver attempted during the 2006 general elections. If voting standards are adjusted once every few years, they likely will not keep pace with changes in technology and new problems that appear after long hours of use or changes in scope and/or scale. There should be some means for addressing new problems or concerns with voting standards between iterations of a VVSG. While this is handled through the EAC's certification manuals and processes, it is important to include feedback from this certification process into the standards. There needs to be a process developed — as part of the standards — to incorporate this

feedback. We recommend that any problems found in the testing processes that are not covered by the standards be addressed quickly, prior to a subsequent iteration of the standards. For instance, if systems are consistently demonstrating a functional problem — one that could affect the election process and which are not covered by the standards, reporting this activity should result in corrective actions that are as binding as standards, approved by the EAC and appropriate advisory groups. With such a process, the decertification and recertification prompted by top-to-bottom reviews such as those held in California could be made less disruptive and more broadly applicable.

Accuracy and Usability

A critical factor in election accuracy is accurately capturing a voter's intent. The voting process always starts with the voter's intent, which must be converted from a selection on the machine through the user interface. Therefore, designing usable interfaces by building upon a body of best practices and knowledge for interface design is a critical first-step toward accuracy in elections.

The VVSG should reflect the above principle. We present some specific comments below to help clarify and increase focus on the importance of usability in its affect on accuracy.

Section 3.2 of Part 1 of the draft VVSG cites the basic usability standards of the HAVA. These requirements define important and fundamental functional capabilities of the voting system, but are incomplete in that they do not specify any goal or mechanism to achieve usability in the initial casting of the ballot. By omitting usability requirements for the primary vote-casting activity, an incorrect impression is given that this is not a point of emphasis. While the standards themselves should reflect the goal of designing usable interfaces to capture voter intent, the law should as well. If the EAC puts forward amendments to the law in the future, we suggest that it address this gap.

Specifically, we would recommend the following new clause be amended to HAVA:

"i. Have a vote-casting mechanism presented, following best practices for user interface design, to enhance the ability of voters to accurately make selections that represent their intent. The design approaches for reaching this goal may include such basic principles as consistency, visibility, feedback, mapping between selections and candidates, and clear visual design"

Accessibility and Usability

While these terms are used separately within the VVSG, accessibility and usability have the same goal — making the voting experience and the voting equipment as easy as possible for the voter (and in the case of setup, shutdown and auditing, the poll worker) to use. While it is important to make sure that those with disabilities are able to vote with privacy and the other election guarantees provided to all voters, it is a mistake to restrict accessibility and usability concerns to only those with disabilities. Limiting accessibility features to machines specifically designated for users with disabilities may limit the ability of other voters to benefit from technologies and innovations that could improve their voting experience. Similarly, by restricting features to a limited number of machines, costs for those machines will be greater. They will likely be used more often, and reach their mean time to failure faster than other machines. To the extent feasible, we recommend that accessibility features be included with as many voting machines as possible and practical.

Assistive devices that must be connected to voting systems may raise security concerns. Specifically, devices that must interface with the voting system software may introduce viruses, or the interaction of two disparate systems may prompt unintentional problems with the voting system. We recommend that jurisdictions should provide common assistive devices that can be connected via industry standard interfaces (such as USB) with voting systems. This would allow for testing of the interface as part of the certification process. Other assistive devices are either external to the voting system or

connect through some mechanism that does not require a software interface (such as the audio devices currently available with some voting systems); Voters who need such devices should be allowed to bring such a device with them to vote.

Testing and Certification

Testing to specific requirements, while necessary, is only one of the necessary steps to ensure that a voting system is worthy of certification. USACM recommends that the processes of testing and certification maximize the opportunities for independent review by qualified individuals prior to approval of any voting system. Review and testing by a range of qualified evaluators will increase the likelihood that systems will perform as needed. The transparency provided by such testing will strengthen the trust in the voting system — something that is process dependent, not technology dependent.

When we think about testing requirements, we should consider the overall testing strategy and how it fits in with the voting process. We start with development of voting equipment (hardware and software) by a vendor who may or may not be *trustworthy* (see Appendix A for a definition). There are a few different ways we can check the vendor's trustworthiness. These ways can include: process quality improvements — such as Capability Maturity Model Integration — as part of the certification process; the use of independent test labs (with a mix of testing techniques) for certifying software; holding the company to a higher liability standard; public or outside expert review of the software; or some combination of these and other methods (not all of which can be implemented through the VVSG). If there are concerns about feasibility, practicality or expense of particular methods, adjustments should be made with emphasis on preserving the process of demonstrating the trustworthiness of the voting systems — that the underlying systems are worthy of certification. The testing requirements and processes should always be focused on ensuring accurate, reliable, accessible, secure, and verifiable elections. To the extent that logistical concerns become blocks to effective testing and/or certification, the burden should be on the voting systems to demonstrate (much as it is in

the innovation class requirements) that they will not pose significant logistical burdens in the testing and voting processes.

Another important part of the testing process is to conduct tests that reflect the possible conditions voting systems will experience in the field including tests for accessibility, usability, security, and reliability. Systems that pass tests in idealized laboratory conditions may not fare as well in field conditions. Vendors and testing personnel may be too close to voting systems to understand how accessible or usable they may be for the average voter or poll worker. If tests are restricted to only lab conditions, or are narrowly constrained to focus on the guidelines and nothing else, testing authorities and test labs are risking the equivalent of teaching to the test — worried only about what is in the VVSG, regardless of the impact a flaw or error could have on elections. USACM recommends that voting systems should be tested, and benchmarks met, in conditions most likely to be found in polling places.

Conclusion

We thank the EAC and the TGDC for developing this version of the VVSG, as well as the dedicated NIST staff that helped develop the proposed requirements. In 2002 Congress gave NIST significant new responsibilities and created the TGDC with the specific intent of building much-needed technical expertise into voting guidelines. At the same time, HAVA appropriated billions of Federal dollars for the purchase of new voting systems based on Federal guidelines that were woefully inadequate. NIST, the TGDC and the EAC were given few resources and little time to develop new standards for equipment that state and local jurisdictions were purchasing.

The computing community found that the 2002 standards (which HAVA deemed to be adopted based on previous Federal Election Commission standards) and the 2005 revision were lacking in scope, depth and technical rigor. The evidence for the inadequacy of the standards is clear: Numerous independent technical reviews of voting

equipment currently in service have found major security, accessibility, and reliability flaws.

This draft is a sweeping and fundamental change from the previous standards, and a welcome step forward toward making voting systems accurate, reliable, accessible, secure, usable, and auditable. These high-level objectives support the most critical factor in elections — that voters have confidence in the results.

There is a growing sense that it will be many years before there are any major revisions of these standards, once they are adopted. Therefore, we urge the EAC to resist weakening the critical concepts in the draft that provide for the development of more robust voting systems. Such a weakening would repeat the pattern of inadequate Federal standards. USACM has outlined its support for many of the important principles — such as software independence, independent voter-verified records, innovation class and vulnerability testing — in this document. Clearly these principles are only meaningful if the requirements behind them are detailed, clear and rigorous. We have recommended many specific improvements to the detailed requirements and urge the EAC to adopt the text approved by the TGDC, incorporating the comments we have submitted.

Additionally, we would like to make ourselves available to NIST, the TGDC, and the EAC to provide technical advice and expertise. Our members have contributed to EAC and TGDC meetings in the past, and they look forward to the opportunity to continue to contribute as the EAC deliberates these standards. Please contact our Public Policy Office — 202-659-9711 — with any questions you may have.

Acknowledgments

Finally, ACM wishes to thank the members of USACM for their dedication in drafting and vetting these comments. In particular, ACM thanks the Chair of USACM, Eugene Spafford (Purdue University); the Chairs of USACM's Voting Subcommittee, Alec

Yasinac (Florida State University) and Barbara Simons (retired, formerly with IBM) for their leadership on this project; David Bruggeman (ACM's Public Policy Analyst) for serving as editor; and all the members of USACM's voting subcommittee listed below:

Annie Anton (North Carolina State University)
Jean Camp (Indiana University)
Lillie Coney (Electronic Privacy Information Center)
David L. Dill (Stanford University)
Jeremy Epstein (Software AG)
Edward W. Felten (Princeton University)
Harry Hochheiser (Towson University)
Lance Hoffman (George Washington University)
Douglas W. Jones (University of Iowa)
Cem Kaner (Florida Institute of Technology)
Kim Lawson-Jenkins
Vince Lipsio
Peter Neumann (SRI)
Barbara Simons (retired, formerly with IBM)
Eugene Spafford (Purdue University)
David Wagner (University of California, Berkeley)
Alec Yasinsac (Florida State University)

Specific Section by Section Comments on Draft VVSG

0. Preface to Comments

In August 2007, the Technical Guidelines Development Committee (TGDC) submitted recommended guidelines to the Election Assistance Commission. This draft introduces several new concepts including a requirement for Software Independence and Open Ended Vulnerability Testing, resulting in significant debate and controversy.

To assist with further evaluation of the VVSG draft, the EAC established a public feedback forum to seek an analytical approach that can produce a threat model to assist in attributing risk, defining costs, and estimating return on investment for corresponding VVSG draft provisions.

In response to this call for comments, members of the U.S. Public Policy committee of the Association of Computing Machinery (USACM) reviewed the VVSG draft, discussed many of the pertinent issues, produced a set of comments, reviewed and revised those comments, and provide them to the Elections Assistance Commission for its use. Thus, the comments herein in total are those of the USACM.

Each comment is tagged in brackets with one or more descriptive terms (e.g. imprecise, incomplete, inaccurate, vague) describing a summarizing reason for the comment.

Italicized words throughout these comments are defined in Appendix A.

Part 1. Equipment Requirements.

Section 2.7.A. Software Independence

USACM Comment #1. Software Independence Definition [imprecise]

USACM strongly supports voting systems that demonstrate software independence — where the results of the election can be verified independently of the voting system software. It is important that for software independence to be effectively implemented that it is clearly defined and that tests for demonstrating software independence are also clearly defined and described. To that end, USACM recommends that the first use of the word "undetected" be removed from the Software Independence definition. The definition would then read:

> Voting systems SHALL be software independent, that is, an error or fault in the voting system's software SHALL NOT be capable of causing an undetectable change in election results, even if the software fault(s) are not detected.

DISCUSSION: As currently written, the software independence definition focuses on undetectable software faults. The more important problem is undetectable changes in election results, which could result from detectable or undetectable software faults.

It is clear to us that the intent of software independence is to comprehensively protect election results from software faults. Thus, deleting the qualifier "undetectable" related to software faults captures the true essence of software independence.

We suspect that the reference to "undetectable software faults" may have resulted from an intent to emphasize that even the onerous category of "undetectable software faults" must not be able to affect election results. That notwithstanding, the proposed definition ensures that no software fault — including those that may be undetectable — can affect election results in an undetectable manner.

USACM Comment #2. Overlap between Sections 2.7 and 4.4 [ambiguous]

USACM recommends replacing Section 2.7.1 with the following subsection added to section 2.7:

> 2.7-B. Achieving Software Independence.
> Voting systems that use Independent Voter-Verifiable Records (IVVR) as described in Part 1, Section 4.4 below can satisfy the software independence requirement and thus achieve conformance to the VVSG. IVVR is currently the only approach for achieving software independence in this VVSG without going through the innovation class process for approval.
> Applies to: Voting system
> Test Reference: Part 3:4.1 “Initial Review of Documentation”,
> Requirement Part 3:4.2-C
> Source: New requirement

DISCUSSION. Details specified in Section 2.7.1 overlap extensively with data in Section 4.4. In addition to the unnecessary forward reference, this overlap creates confusion within the document and the potential for conflicting requirements, particularly if revisions occur.

USACM Comment #3. Software Independence Demonstration [incomplete]

USACM recommends that the VVSG include a step-by-step demonstration of a software independence assessment. An example of such a demonstration is included as Appendix B to these comments.

DISCUSSION: The process for demonstrating Software Independence is not clearly delineated in the document, and it is unclear how such an assessment would proceed from the document as it is written. Without such precision, the guidelines are open for a wide variety of interpretation as to what would demonstrate software independence. This would allow for groups to argue the requirement is overly broad, or that it is already demonstrated by existing voting systems — which is currently only true for voting systems with IVVR.

USACM Comment #4. Section 2.7.2 Innovation Class Process [incomplete]

E-voting faces numerous challenges and is a field ripe for further research. Federal and private investments should continue to be made and new, innovative approaches should continue to be developed. However, until the fundamental constraints of security testing can be adequately addressed, these systems should have to meet a high bar for independent voter-verification before they are certified. Without a definitive testing procedure, this process runs the risk of being an end-run around existing principles such as software independence. Any system intended for certification in the innovation class must demonstrate that it is at least as good, if not better, than other election systems.

USACM believes that the innovation class proposal and evaluation process is critical to effective implementation of new voting technologies and recommends that the document be expanded to include a detailed process description for determining what qualifies for the innovation class and how voting technologies can be effectively tested as part of the innovation class. This description should include who should decide whether a device or voting system qualifies for the innovation class and what criteria that device or system must meet to be certified as all or part of a voting system that meets the VVSG.

Part 1. Chapter 3. Usability, Accessibility, and Privacy.

USACM Comment #5. Section 3.1.3. Usability and Accessibility [imprecise]

The distinction between Acc-VS and VEBD sets up a false choice between accessible stations and other voting stations. This is problematic in several ways.

USACM recommends that the following text be added to section 1.3

> - All VEBD systems — even those that are not specifically designated Acc-VS systems — should implement all reasonable accessibility features from section 3.3. Reasonable accessibility features would include any that could benefit all voters, regardless of ability.

DISCUSSION: Relegating accessibility features to a completely separate class of machines can increase costs (as a result of relatively lower volume of production) and lead to reliability concerns (the added cost and difficulty of testing these special-purpose machines may make appropriate testing difficulty). Many accessibility features such as magnified text and speech output can be inexpensively supported on the same platforms that would likely be used as VEBD systems. Including these features will let many users with disabilities use VEBDs to vote.

USACM Comment #6. Section 3.2 General Usability Requirements [vague]

This section does not adequately describe why usability is an important part of voting system accuracy. As we described in the introduction, the voting process always starts with the machine capturing the voter's intent. Careful interface design is a critical first-step toward accuracy in elections. The specified usability requirements define important

and fundamental functional capabilities of the voting system, but are incomplete in that they do not specify any goal or mechanism to achieve usability in the initial casting of the ballot.

USACM recommends that the introductory paragraph be changed as follows:
“The voting system should support a process that provides a high-level of usability for all voters. The goal is for voters to be able to accurately cast their votes as intended while negotiating the process effectively, efficiently, and comfortably."

USACM Comment #7. Section 3.2.1.1 Overall Performance Metrics [incomplete]

This subsection provides suggested metrics for voting machine success, noting that "the tests associated with these requirements are designed as repeatable controlled experiments and not as ‘realistic’ measures of voting behavior, as might be found in a wide variety of voting contexts". In other words, they are providing benchmarks for system usability in the lab, not in real use. This is not very helpful if voting places are loud, poorly lit, etc. — in other words, not necessarily generalizable to real voting conditions.

USACM recommends that the discussion of metrics in section 3.2.1.1 be amended to include the following text:

> Voting systems should strive to meet these benchmarks in environments that closely simulate the conditions likely to be found in voting places. Systems should meet benchmarks when votes are cast in places that are crowded, noisy, poorly lit, overheated, and otherwise environmentally sub-optimal.

DISCUSSION: Systems that function well in laboratory settings may not function well in suboptimal environments. External factors that cause increased voter discomfort may lead to additional errors or other difficulties in using voting devices. As these difficulties may not be encountered in tests conducted in carefully controlled laboratory settings, usability tests should be conducted in the presence of potentially distracting

factors. Testing in an actual polling place with representative voters would be one way to introduce these potentially distracting factors.

USACM Comment #8. Section 3.2.2-C [Incomplete]

USACM recommends that the discussion of "Correction of ballot" should be amended to include the ability to modify a vote. Specifically, the text should be changed to read:

> The voting system SHALL provide the voter the opportunity to correct the ballot for either an undervote or overvote, or to change any votes, before the ballot is cast and counted.

DISCUSSION: Voter review of selections may lead to the identification of incorrectly cast votes, which would not necessarily be undervotes or overvotes. Voters should have the ability to change all such votes.

USACM Comment #9. 3.2.2.1-A Prevention of Overvotes [Imprecise]

USACM recommends that the requirements for changing overvotes should be changed to read as follows:

> The VEBD SHALL prevent voters from selecting more than the allowable number of choices for each contest. If this process causes the VEBD to make any changes to the selection of votes, the exact nature of any changes must be clearly presented to the user.

DISCUSSION: As currently stated, the discussion of this requirement reads as follows:

> This requirement does not specify exactly how the system must respond when a voter attempts to select an "extra" candidate. For instance, the system may prevent the selection and issue a warning, or, in the case of a single-choice contest, simply change the vote.

Unfortunately, simply ‘changing the vote’ may not match the voter’s goals and intentions. Voters must be given clear and appropriate feedback whenever such

changes are made. If the vote is changed the voter must have the opportunity to review that change.

USACM Comment #10. Section 3.2.3.1-A.1 — Visual Privacy

USACM recommends that the text of this requirement be amended to the following:

"The ballot, any other visible record, containing ballot information, and any input controls SHALL be visible only to the voter during the voting session and ballot submission, independent of whether the voter is seated or standing."

DISCUSSION: The additional clause, "independent of whether the voter is seated or standing," would clarify that appropriate shielding of the voting station must take into account variations in height of the voter, many of which may result from a voter who is seated while voting compared to one who is standing while voting.

USACM Comment #11. Section 3.2.4 Cognitive Issues [incomplete]

USACM recommends that the following requirement be added to the list of cognitive requirements:

> Non-threatening language: Warnings and alerts issued by the voting system SHALL use non-threatening language. Terms such as 'abort', 'die', or 'fatal error' should be avoided.
>
> "Clear and direct relationship between selection mechanism and candidate: Any mechanism that is used to select a candidate must be spatially near the candidate being selection, and positioned in such a way as to make a clear and unambiguous correspondence between the mechanism and candidate.

DISCUSSION: Threatening language may intimidate some voters and cause them to lose faith in their operation of the machine. Appropriately crafted messages can indicate the nature of problems without scaring voters. A clear and direct relationship between selection mechanism and candidates would prevent situations where, for example, the candidate name is on the far left side of the screen and the selection mechanism is on the far right side of the screen.

USACM Comment #12. 3.2.6.1-D System response indicator [incomplete]

USACM recommends that the requirement for system response be revised to read as follows:

> If the system has not completed its visual response within one second, it SHALL present to the voter, within 0.5 seconds of the voter's action, some indication that it is preparing its response. This indication should contain enough information to reassure the voter that the system is indeed functioning correctly.

DISCUSSION: For instance, the system might present an hourglass icon indicating that it is "busy" processing the voter's request. This requirement is intended to preclude the "frozen screen" effect, in which no detectable activity is taking place for several seconds. There need not be a specific "activity" icon, as long as some visual change is apparent (such as progressively "painting" a new screen). Progress bars or other indicators that provide concrete indication of progress towards task completion (in terms of percentage of task already completed) provide additional information and confidence, and should be used whenever possible.

USACM Comment #13. Section 3.2.8.1-A Ease of Normal Operations [vague]

USACM recommends replacing the first sentence in this subsection as:

> "While a certain amount of complexity is unavoidable, setup, polling, and shutdown procedures should not require any special expertise. The procedures SHALL require no more than two hours of training for the typical poll worker to learn, understand, and perform."

DISCUSSION: This paragraph as written is unnecessarily vague in describing the complexity of the setup, polling, and shutdown procedures. The proposed change is a reasonable practical requirement, as a routine poll-worker training session duration is two

hours. The requirement is also testable by giving a two-hour training class, and then having the students conduct startup, polling, and shutdown procedures.

USACM Comment #14. Section 3.2.8.1-B Usability Testing by Manufacturer for Poll Workers [imprecise]

USACM recommends that the first sentence of this subsection be replaced with the following sentence:

> "The manufacturer SHALL conduct summative usability tests on the voting system using individuals who are representative of the age, education, and technological literacy of poll workers and SHALL report the test results, using the Common Industry Format, as part of the TDP."

DISCUSSION: This wording will guide the test population to more closely parallel the user population and is consistent with 3.2.8.1-C.1, which calls for documentation to be presented at "a level appropriate for non-expert poll workers." Typical poll workers are not technologically literate, nor are they typically experts in computers. They do not have specialized training.

USACM Comment #15. Section 3.2.8.1-C.3 "Enabling verification of correct operation" [incomplete]

This section should cover verification that the system has not been inappropriately tampered with. USACM recommends that the list of requirements in Section 3.2.8.1-C.3 be expanded to include the following:

- Has not been tampered with.

DISCUSSION: Poll workers need to have some unambiguous way of verifying that a VEBD system has not been modified or misused, as any inappropriate changes to system state might lead to election miscounts or fraud.

USACM Comment #16. 3.3.1-C No dependence on personal assistive technology [incomplete]

USACM recommends that the discussion on personal assistive technology be revised to add the following requirement:

> "Accessible voting stations SHOULD allow Voters the option of using personal assistive devices if they so desire. To alleviate security concerns, such devices cannot directly interface with the voting system software unless such a device is controlled by elections officials and provided with the voting system"

DISCUSSION: voters who have assistive devices or other controls that they prefer to use should be given the option of connecting these devices via common ports. Despite the comment in the existing discussion, the term "personal assistive device" does not appear to be defined in Appendix A of the VVSG. The limitation of personal assistive devices to those that cannot interface with the voting system software is to eliminate the possibility that a device brought in from the outside could interact with the voting system in such a way as to harm (unintentionally or not) the operation of the system software. Devices provided with the voting system that could interact with the software can be controlled and tested, reducing the risk of adverse consequences from software interaction between the assistive device and the voting system software. Although the inclusion of such devices will require additional testing for the voting system, it will increase the accessibility of the system while maintaining security.

USACM Comment #17. Section 4.2.2 Hand audit of IVVR record

USACM recommends that the first "of" in the first sentence of the discussion in this section should be removed. It does not refer to any specific item and is extraneous to the discussion.

USACM Comment #18. Section 4.4. Independent Voter Verifiable Record IVVR is unclear [vague]

USACM recommends that the second and third sentences in the first paragraph in Section 4.4 be replaced by:

> IVVR is a human-interpretable representation of a voter's choices. There are two categories of voter-verifiable paper records (VVPR) that may meet the IVVR standard:

DISCUSSION: Neither the term "*independent*" or the phrase "voter verifiable" are defined or inherently clear in the phrase "Independent Voter Verifiable Record". The proposed change simplifies the text and clarifies the meaning of these terms.

USACM Comment #19. Section 4.4 Circular Definitions [incorrect]

USACM recommends replacing the definitions of "Independent Voter Verifiable Record", "IVVR", and IVVR vote-capture device" as described below:

> Independent Voter Verifiable Record**:** This record (an IVVR) is a persistent, simple, human-interpretable representation of a voter's choices. It is independent in the sense that the voter's choices are semantically clear without electronic, electro-mechanical, mechanical, codebook, or other translation. It is voter verifiable in the sense that it is presented to voters for their review before they commit to their selections.
>
> IVVR: Independent Voter Verifiable Record.
>
> IVVR vote-capture device: A device that interacts with a voter to produce an independent voter-verifiable record.

DISCUSSION: The three definitions presently in the VVSG Draft are circular, i.e. each relies on reference to the other two to form their definition. This results in vague representations with no real definition at all. Moreover, the definitions do not capture the essential meaning necessary to understand these concepts. The proposed wording removes the circularity and clarifies the meanings of these critical terms.

USACM Comment #20. 4.4.1-A.5 — IVVR vote-capture device, IVVR durability

USACM recommends that the requirement text be modified to the following text:

"IVVR vote-capture devices SHALL create an IVVR that will remain unchanged for minimally 22 months unaffected by power failure, software failure, or other technology failure. The IVVR must also remain unaffected from conditions in which it is stored (such as temperature extremes or humidity)."

DISCUSSION: The added sentence addresses the fact that the durability of an IVVR is as dependent on the conditions under which it is stored as the conditions by which it is made. The requirement as written only addresses how the IVVR is produced; it does not speak to the durability of the IVVR after it is produced.

USACM Comment #21. 4.4.3 PCOS Systems Printing PCOS Exception [incomplete]

USACM recommends that subsection 4.4.3 be taken out of the VVPAT section and become its own section (e.g. Section 4.5) in the same VVSG part/chapter. Additionally, USACM recommends replacing the existing wording with the following:

> PCOS systems can provide the VVSG required recording mechanism independence. A PCOS voting system involves paper ballots marked in a way that is both human and machine-readable. These paper ballots are routinely marked legitimately at only two times:
>
> 1. When they are printed and
> 2. When the voter marks them.
>
> The following exception applies to optical scan ballots as required for supporting audit and recount.

DISCUSSION: This subsection needs to be taken out of the VVPAT section because the current organization would subsume PCOS systems under VVPATs, and that does not accurately reflect the nature of PCOS. PCOS systems are independent of VVPAT systems, particularly when considering tabulation of ballots. Printing scanners is a relatively novel voting system paradigm. There is little in the literature that investigates the threat that printing scanners pose to a voting system. Allowing printing during the scanning process can add integrity information to ballots, but also adds a non-trivial threat relative to marking undervoted ballots or by spoiling properly marked ballots either

programmatically or unintentionally. The requirements for this technology must ensure that this printing capability cannot be abused to ensure that some ballot types are always passed by without inclusion in the count or audit.

USACM Comment #22. 4.4.3-A.1 Printing PCOS Print Area Restriction [incomplete]

USACM recommends replacing the existing wording in subsection 4.4.3-A.1 with the following:

> Optical scanners with printing capabilities that add markings to paper ballots as they are scanned SHALL ONLY permit printing in spaces designated on the ballots for that purpose. They SHALL NOT be capable of altering the contents of the human-readable CVR on the ballot. Specifically, optical scanners capable of adding markings to the scanned ballots SHALL NOT permit:
>
> a. Marking in the regions of the ballot that indicate voter choices;
> b. Marking in the regions of the ballot that contain the human-readable description of the marked choice; and
> c. Marking in regions reserved for timing marks.
> d. Marking in regions reserved for any other purpose.
> e. Marking in regions not designated for any purpose.

DISCUSSION: The present verbiage may allow stray marks or marking in areas designated for other purposes. The proposed wording clarifies and strengthens protection against overwriting by optical scanner/printer markings.

Part 3. Testing Requirements

Part 3. Chapter 1. Introduction

USACM Comment #23. Lack of Accessibility Testing Specification [incomplete]

In our review of the testing specifications, we have found the accessibility and testing requirements either lacking or in need of development to ensure conformance with accessibility and usability standards. USACM recommends that TGDC develop testing requirements and procedures for accessibility and usability for inclusion in the VVSG.

Part 3. Chapter 4. Documentation and Design Review

USACM Comment #24. Section 4.5.1. Software Review, Security [incomplete]

USACM Recommends that the following subsection be inserted before the current subsection 4.5.1-D (which would be renumbered 4.5.1-E):

4.5.1-D Required Use of Automated Static Analysis Tools
The test lab SHALL exercise automated static analysis tools against all software under review. At least one run SHALL be made as a baseline with the strongest security setting available and this report SHALL be produced as part of the review work product.

Thereafter, tool settings will be optimized by the OEVT team and team members must rigorously assess the appropriately tuned reports to determine if flagged events indicate dangerous security faults or not.
Applies to: Voting system

DISCUSSION: The software security review section is woefully lacking in description of the tools available to conduct security reviews. Static analysis tools have advanced dramatically in the past two years and can be important aids to assist reviewers in identifying vulnerabilities, particularly the most widespread vulnerability that includes known weaknesses and exploits through buffer overflow faults. Automated static analysis tools such as FLAWFINDER, Rough Auditing Tool for Security (RATS), and numerous commercial tools offer a unique opportunity for establishing an objective, consistent software metric. These tools can detect common developer errors including dead code

segments, memory leaks, memory overruns, race conditions, and several other common maladies. These tools are not comprehensive and cannot replace other testing. However, they can establish an objective baseline for absence of known vulnerabilities that cannot be duplicated by standards, rules, or open-ended tests.

There is now a plethora of reasonably priced static analysis tools on the market, embedded in development environments, and in open source products. Their use should be mandated in the OEVT process.

Section 5.3. Benchmarks.

USACM Comment #25. Section 5.3.1 General method [incorrect]

USACM recommends correcting the factual errors present in this subsection, as minimally enumerated below:

1. The system submitted for testing might be a representative sample of the pool of identical hardware/software systems, but the pool of tests should not be a representative sample of the events that happen during an election.
2. There is no reason to expect software reliability, software accuracy, and hardware misfeed rate to follow the same distribution.
3. The Poisson distribution is discrete, not continuous.
4. The Poisson process typically assumes a stationary underlying exponential distribution. The idea that software reliability, software accuracy, and hardware misfeed rates follow the same underlying distribution, or that the concatenation of these three (if there are only three) distributions would be anything like exponential is remarkable in its unlikelihood.
5. The observed event rate ("events" divided by "volume" over the course of a test campaign) is a highly biased measure.
    a. The first problem is that a regression test suite repeats the same tests from build to build. This gives rise to the classic problem of the "pesticide paradox" [1]. The test suite is a tiny sample of the collection of possible tests. When the suite reveals bugs, they are fixed. Ultimately, the test suite becomes a

collection of tests that have one thing in common: the software has passed all of them at least once. This differs from almost every other possible test (all of the ones that have not been run). Therefore, the reliability of the software is probably vastly overestimated.

b. The second problem is that the pool of tests is structured to cover a specification. It does not necessarily target vulnerabilities of the software. Nor is it designed to reflect usage frequencies in the field [2].

6. Determining the length of testing in advance by an approved test plan sounds scientific, but many practitioners consider this software testing malpractice. There is substantial evidence that bugs cluster. Given a failure, there is reason to do follow-up testing to study this area of the product in more detail. Rigidly adhering to a plan created in the absence of failure data is to rigidly reject the idea of follow-up testing. This underestimates the number of problems in the code. Worse, this testing method reduces the chance that defects will be found and fixed because it reduces — essentially bans — the follow-up testing that would expose those additional defects.

USACM Comment #26. Section 5.3.2. Critical Values [incorrect]

This section should be adjusted according to factual corrections made in the previous comment.

USACM Comment #27. 5.3.3 Reliability [incorrect]

USACM notes the apparent self-contradictions in this sub-section, enumerated in the discussion below.

DISCUSSION:

1. Failure rate data are not relevant to prediction of reliability in the field unless we assume that the failure rate in the lab is representative of the failure rate that will be found in the field. This might be rational for hardware, but unless we structure the software tests to map to usage in the field, there is no rational basis for this assumption vis-à-vis the software.

2. Pass/fail criteria are based on the concatenation of hardware and software failures. A paper jam rates the same as miscount of votes.
3. Counting all "failures" for statistical purposes creates an adversarial dynamic around the classification of anomalous behaviors. To the extent that an apparently-incorrect behavior is arguably not inconsistent with the specification, there is an incentive to class it as a non-bug and therefore not fix it. The incentives should favor improving the software, not classifying problems as non-problems

USACM Comment #28. Section 5.3.4. Accuracy [incorrect]

USACM notes the apparent self-contradictions in this sub-section as described in the discussion below.

DISCUSSION: Accuracy is operationalized (not formalized) as a ratio of errors found to volume of data processed. One may assume that the word "error" is tied tightly to events that yield a miscount of the votes, allow someone to cast extra votes, or cause someone to be unable to cast a vote. If "error" includes anything in the behavior of the program that would not create an error in election result, it is difficult to understand what this operationalization has to do with the naturalistic concept of "accuracy" in a system that collects and counts votes.

The operationalization is defective as an estimator unless the pool of tests is designed so as to be representative of the pool of behaviors in the field. If some aspect of the system causes a small mistake (e.g. 1-vote miscount), but is only tested once, that might be a major source of inaccuracy if everyone encounters it while voting, and it might be a trivial source if almost no one encounters it. For example, imagine a system that allowed ballots that could accept write-in votes for up to 100 candidates. Imagine an error in which 1 vote in 10 is lost in the 100th race that includes a write-in candidate. As a boundary case, this error might show up in several tests. However, it might never show up in an election. What is the accuracy using the described metric?

Without a mapping from the estimator to the construct being estimated, the metric is worthless. This is a fundamental issue in measurement. We normally call it construct

validity. The argument that this measure of accuracy estimates underlying system accuracy lacks even face validity.

Section 5.4. Open Ended Vulnerability Testing

USACM Comment #29. OEVT Goal [imprecise]

USACM recommends that the present stated goal for OEVT (Sect 5.4, par 2, first sentence) be modified to read as follows:

> "The goal of OEVT is to test and analyze the target system to discover flaws that could adversely affect the election process and that may reflect systemic development process weaknesses."

DISCUSSION: The current text is focused on discovering flaws that could invalidate election results. The proposed language would allow OEVT to be used in checking for flaws in other aspects of election operations, including accessibility and usability. OEVT is not meant as a replacement for quality design. The use of OEVT must be carefully described. It is a process that is difficult to replicate (if it were easy to replicate, it would not really be open-ended), so any requirements on OEVT in this VVSG should focus on the process — how the team is selected, the scope of work — of OEVT rather than specific steps. As with other parts of the testing process, this must be open to review by qualified scholars.

USACM Comment #30. Section 5.4.1-C General Priorities [inaccurate]

USACM recommends that item #1 in this subsection be deleted.

DISCUSSION: A threat scenario need not be plausible if it is designed to rapidly evaluate the possibility of a failure under a plausible scenario. This is a nontrivial issue--extreme cases are efficient indicators of possible problems with less extreme values. A competent time-pressured team will test first with unrealistically harsh cases (the harshest reasonably creatable under the circumstances) and follow up with more realistic scenarios if and only if there is a failure with the harsh cases. Remember: In exploratory testing (OEVT is an example of exploratory testing), the testers are not required to turn themselves into mindless automata following a script. If they see a hint of a problem,

they are not required to move on to the next item in the script — they can follow it up. If there is a failure, they do not have to focus their reporting on this one failure. In a scripted case, an unrealistic case yields a bug report of a failure under unrealistic circumstances--an unsatisfactory report. In exploratory testing, the activities of learning, test design and test execution are parallel and mutually supportive. Therefore, indefensibly unrealistic tests are entirely proper if they are designed to yield a line of inquiry that leads to useful data.

Additionally, the existing text of this section makes a large assumption — that all of the requirements stipulated in that section will be met. As there have been documented cases of voting systems that have been certified but do not meet requirements (see the systems recently reviewed by the State of California), this is a risky assumption to make.

USACM Comment #31. 5.4.2-E OEVT team knowledge [incorrect]

USACM recommends that the word "Complete" in items numbered a and b in this subsection be replaced by the word "Expert".

DISCUSSION: No one, and no small team, can have "complete knowledge." A patently impossible requirement offers no guidance as to the expected level of knowledge and competence. On the other hand, "expert knowledge", while subjective, is a recognized standard.

USACM Comment #32. Section 5.4.4-A OEVT Fail Criteria — Failure Interpretation [incomplete]

USACM recommends that the following new subsection, with the title above, be added to Section 5.4 as follows:

> Software testing, including open-ended testing, cannot demonstrate the absence of flaws. Thus, its contribution to the certification process is twofold:
>
> a. A final filter to prevent faulty voting system software from achieving certification
>
> b. Detect vendors whose development processes are not sufficiently

mature to consistently produce high assurance products.

The OEVT team should consider a final finding of "failure" to indicate a need to redesign the system or the system testing strategy.

DISCUSSION: There is no software testing regimen that can claim comprehensive fault detection. Thus, the best that an OEVT team can hope to do is (1) Detect well-known faults left as a result of immature development processes and (2) Detect subtle faults that the team's specific skill sets enable them to find and that routine or even mature development processes may not prevent or detect.

During deliberations, the OEVT team must assess the vulnerabilities as they apply relative to vendor prescribed procedures. Fail criteria must reflect that an attack based on whether the identified vulnerability would be likely to occur, succeed, and escape detection.

USACM Comment #33. Section 5.4.4-C OEVT Fail Criteria - Critical Flaws [incorrect]

USACM recommends that subsection 5.4.4-C be modified as follows:

The voting device shall fail open-ended vulnerability testing if the OEVT team demonstrates one or more critical flaws that allow an attacker to violate VVSG requirements as specified in paragraph 5.4.4-A above, under a plausible description of how vulnerabilities or errors found in a voting device or the implementation of its security features are used to:

a. Change the outcome of an election;

b. Interfere with voters' ability to cast ballots or have their votes counted during an election; or

c. Compromise the secrecy of vote

without having to demonstrate a successful exploitation of said vulnerabilities or errors.

Potential vulnerabilities for which no exploit is demonstrated may be noted as observations, but may not rise to the level of findings.

DISCUSSION: OEVT failure is a serious event that may have severe financial

ramifications. Thus, it cannot be justified by hypothetical attacks. OEVT testers must be held to high scientific standards that can only be reflected by the three level process of:

a. Detecting vulnerability
b. Envisioning an exploit for each identified instance and by
c. Demonstrating each envisioned attack under plausible conditions.

USACM Comment #34. Structured Note-taking [incomplete]

USACM recommends adding a paragraph 5.4.5-B as follows:

> 5.4.5-B. OEVT team process documentation requirement.
> Each OEVT team will conduct structured note-taking during the analysis. Where possible, all notes will be shared among team members during the entire review period, but must be shared by all members during deliberations, before the final report is prepared.
> These structured notes become part of the team product and must be delivered along with the OEVT final report.

DISCUSSION: It is difficult to overstate the value of structured note-taking during the review process and making the notes database a work-product of each review. The level of continuity it provides between reviews justifies including it as a VVSG requirement. There are also two other benefits that may be equally as important:

1. Process Improvement. Understanding the details of the process that each team goes through can be a gold mine of best practices.
2. Accountability. OEVT is critically dependent on the skill and knowledge of the investigators. Structured note taking provides an avenue to analyze the team's effort.

USACM Comment #35. Section 5.4.6. VSTL Response to OEVT [incomplete]

USACM recommends changing the first full sentence in Section 5.4.6-A to read:

"The VSTL SHALL:

1. Forward the OEVT results to the VSTL licensing authority for their use in assessing vendor development process maturity and to assess potential corrective action; and
2. Examine the OEVT results in the context of all other security, usability, and core function test results and update their compliance assessment of the voting system based on the OEVT."

DISCUSSION: The addition of requirement one will encourage feedback to testing lab authorities and the Election Assistance Commission about issues, errors and anomalies uncovered during the testing process that are not connected to specific requirements of the VVSG. Without a feedback process for problems outside the terms of the VVSG, the testing process would be subject to the voting systems equivalent of teaching to the test — covering only those items outlined in the test, and ignoring anything else — regardless of how it could influence voting, voting administration and elections.

Reference

1 Boris Beizer, Software Testing Techniques, Second Edition, 1990
2 Musa, Software Reliability Engineering (http://members.aol.com/JohnDMusa/book.htm

Appendix A. Definitions

Capability Maturity Model Integrated (CMMI) — CMMI is a scientifically developed set of process management processes that allow system developers to establish, demonstrate, and achieve certification for mature development processes. CMMI is a product of the Software Engineering Institute.

CMMI - Capability Maturity Model Integrated

High Assurance — *Assurance* is a term of art that typically refers to the rightful trust or confidence that a system user can have in the performance of the described system. In systems development theory, system *assurance* costs increase linearly until they approach an asymptotic turn, where the cost to increase assurance becomes exponential. *High assurance* systems generally demand *assurance* levels beyond the asymptotic turn by requiring *redundancy* and *independent* mechanisms. See also: *trustworthiness*.

Independent — Two events or items are *independent* if they are not causally correlated. In its purest sense, *independence* is Boolean and two events are either *independent* or they are not. Generally within the VVSG Draft context, events and processes may be evaluated on a continuum where the level of *independence* is determined either by the strength of the causal relationship or the impact of the existing causation.

Independent Recording Mechanisms (IRM) — Two mechanisms are independent if they are not controlled by a common mechanism and if they do not operate on the same data. A system employs IRMs if and only if it registers, marks, copies, renders or enters the voters selections in two forms that each are:

1. Caused by voter actions or are reviewed and verified by the voter and
2. Independent in the sense that modifying one form cannot impact the other.

Independent Record — In the VVSG Draft context, two *records* are *independent* if manipulation of one cannot impact the other. A single record is *independent* if it cannot be manipulated.

Independent System - The term *independent systems* in the VVSG Draft context, refers to *systems* that protect one another against concurrent failure. Thus, purely *independent* systems' failure modes are unrelated.

Mission Critical Systems (MCS) — MCS is a self-defining phrase that refers to systems that hold a particularly high failure cost, thus can endure only a very low failure rate. The connotation is that if an MCS system fails, the overall mission it supports is likely to fail as

well. MCS are often developed using *high assurance* processes.

Redundancy - Redundant systems provide identical or similar functionality from more than one source. In the VVSG Draft context, system *redundancy* protects against concurrent, thus against overall, system failure.

Reliability — *Reliability* refers to a systems overall ability to complete its intended purpose even in the face of unexpected natural events. A de facto interpretation of *reliability* does include protection against malicious intruders, though systems that are subject to malicious manipulation negatively impacts system *reliability* in its purest sense.

Static Analysis — *Static Analysis* is used to refer to any approach to software review or assessment that does not require the software's execution. As opposed to testing techniques that observe the program's execution, *Static Analysis* considers the code's structure, logic, data, and semantics based exclusively on the code's syntactic representation.

Trustworthiness — *Trustworthiness* refers to the rightful trust or confidence that a person can have in process or person. See also: *High Assurance*

Verification — *Verification* refers to a process of checking a result for correctness. It naturally requires *redundancy* and is best accomplished via *independent mechanisms*.

Appendix B. SAMPLE Demonstration Procedure for Software Independence

This test is meant as a hypothetical example, is for illustrative purposes only, and is not an endorsement of a particular process or technology for use in voting systems.

The following example illustrates how IVVR may be determined. We are assuming a single-issue election and a unique Compact Disk (CD) for each voter. By "visible surface," we mean the outside of the CD where, for example, movie titles currently are written or where someone could write information about a CD he or she has recorded.

In this voting system, the vote-capture device is an electronic writing pad. A voter display screen lists the candidates and the voter writes his or her selection on the writing pad. The vote capture device captures each vote (recognizing the script) and translates it into an electronic ballot that is written to a CD. The system also prints the written vote in the voter's handwriting on the visible surface of the CD in view of the voter. The vote is recorded from the signature pad and the CD is retained for audit purposes.

The following details apply:

1. The voter is asked to verify and approve what is printed on the outside of the CD before casting the ballot.
2. The device marks the CD as "accepted", in view of the voter, when the voter approves it and the device marks the CD as "rejected", in view of the voter, if the voter rejects it.
3. If the handwriting is not legible, the system will reject the vote and prompt the voter to try again.
4. The printed vote record is durable enough to remain unchanged and legible for a period of 22 months.
5. The CD does not contain any information about the time at which the vote was cast or the ordering of this vote compared to all other votes cast on this voting machine.

6. Information printed on the CD also reveals the polling place, precinct, set of contests the voter was eligible to vote on, and the date of the election.
7. The format of the data stored on the CD is fully disclosed to the public.

We start with the IVVR requirements (cf. Section 4.4 of the VVSG). We list the example assessment of whether this system (with the clarifications above) meets that requirement, and some discussion explaining the conclusion.

Here the IVVR is the printed copy of the vote, as printed on the outside of the CD.

4.4.1-A.1: Complies.

4.4.1-A.2: Complies. Here we assume that if the device can interpret the voter's handwriting, then so can an auditor. Alternatively, if the device will accept records that will not be legible to election officials and auditors, then such a system would not comply with 4.4.1-A.2.

4.4.1-A.3: Complies.

4.4.1-A.4: Complies.

4.4.1-A.5: Complies. See durability assumption above.

4.4.1-A.6: Complies.

4.4.1-A.7: Complies. Same issues as VVPAT.

4.4.1-A.8: Complies. Handwriting is a publicly available format.

4.4.1-A.9: Complies, under the assumption that the device prints the additional information listed above. If the CD does not show that additional information in human-readable form, then the device may not comply.

4.4.1-A.10: Complies.

4.4.1-A.11: Complies. IVVRs do not span multiple media.

4.4.1-A.12: Complies.

4.4.1-A.13: Complies.

4.4.1-A.14: Complies.

4.4.1-A.15: Complies. Depending upon how we interpret this requirement, compliance may require the device to include an electronic bitmap image of the scanned handwriting as part of the data stored electronically on the CD, but that should be straightforward to arrange.

4.4.1-A.16: Complies.

4.4.1-A.17: Complies.

Based on this analysis, we can conclude that the example voting system satisfies requirement 4.4.1-A (the primary requirement that is specific to IVVR vote-capture devices). Note that there are some additional requirements that must also be met if the device is submitted for approval as an accessible voting system (Acc-VS), e.g., 4.2.4-A, 4.2.4-B.

Finally, we can ask whether this system meets the SI requirement. In this case, there is a shortcut: IVVR systems in general qualify as SI (Sections 2.7, 4.1), so as we have determined that the system is an IVVR system it meets the definition of SI.

If the system did not meet the requirements for IVVR, we would have to separately determine whether it meets the SI requirement. Here we look to requirement 2.7-A. To determine whether the system complies with 2.7-A, we would have to consider all possible changes or faults in the software to see whether there are any that could cause an undetectable change or error in the election outcome. In this case, all such errors can be detected, via observational testing, post-election statistical audits, recounts, pollbook reconciliation, and/or the official canvass.

Note also that the voting system vendor, as part of the submission of the system for certification, must declare what conformance classes to which the vendor wants to claim the system complies. For instance, the vendor must decide whether to claim that the device is an "IVVR vote-capture device", whether to claim that it is an "Acc-VS", etc. The testing that is done is determined by what claims the vendor makes. (See Sections 2.3, 2.4 of the VVSG II.)