CRT-DWF Draft 20060317

This document contains draft text for Core Requirements and Testing (CRT) for the following parts of the next iteration of the VVSG:

        - Consolidation: Volume III Section 4.8.5
    - Reporting: Volume III Section 4.9
        - General reporting functionality: Volume III Section 4.9.1
        - Audit, status, and readiness reports: Volume III Section 4.9.2
        - Vote data reports: Volume III Section 4.9.3
            - General functionality: Volume III Section 4.9.3.1
            - Ballot counts: Volume III Section 4.9.3.2
            - Vote totals: Volume III Section 4.9.3.3
    - Transmitting results: Volume III Section 4.10
    - Auditing and independent verification: Volume III Section 4.11
- Reference models: Volume III Chapter 5
    - Process model: Volume III Section 5.1
    - Vote-capture device state model: Volume III Section 5.2
    - Logic model: Volume III Section 5.3

Standards on data to be provided: Volume IV

- Technical Data Package: Volume IV Chapter 1
    - Scope: Volume IV Section 1.1
    - Implementation statement: Volume IV Section 1.2
    - System hardware specification: Volume IV Section 1.3
    - Software design and specification: Volume IV Section 1.4
    - System test and verification specification: Volume IV Section 1.5
    - Configuration management plan: Volume IV Section 1.6
    - Quality assurance program: Volume IV Section 1.7
    - System change notes: Volume IV Section 1.8
    - Configuration for testing: Volume IV Section 1.9
- Voting Equipment User Documentation: Volume IV Chapter 2
    - System overview: Volume IV Section 2.1
    - System functionality description: Volume IV Section 2.2
    - System security specification: Volume IV Section 2.3
    - System operations manual: Volume IV Section 2.4
    - System maintenance manual: Volume IV Section 2.5
    - Personnel deployment and training requirements: Volume IV Section 2.6
- Test Plan: Volume IV Chapter 3
- Test Report: Volume IV Chapter 4
- Public Information Package: Volume IV Chapter 5

Testing standard: Volume V

- Overview: Volume V Chapter 1
- Introduction to test methods: Volume V Chapter 2
- Documentation and design reviews (inspections): Volume V Chapter 3

- Test protocols: Volume V Chapter 4

Supplemental guidance (not part of the VVSG): Volume VI

- Procedures required for correct system functioning: Volume VI Chapter 1

Text for security requirements (STS), human factors and privacy requirements (HFP), hardware and software performance requirements and hardware workmanship requirements (AG) are distributed in separate documents. Where such requirements appear in this document, they are only notes on things that need to be coordinated.

The draft text for different sections is at differing levels of maturity. Notes on work that is yet to be done and problems that need to be fixed are shown highlighted like this. These notes are for our own use and will not appear in the final draft. Similarly, the Impact field of requirements is for our own use and will not appear in the final draft.

Notes applying to the entirety of the document follow.

Scoping of reviews (source code, design, what have you) and related requirements (coding conventions, QA & CM) with respect to COTS and non-COTS, previously reviewed vs. not. The important thing is not whether something is hardware, firmware, software, or even whether it is "commercial" or "off the shelf," but whether the objectives of the review that the test lab would otherwise be doing are sufficiently satisfied by scrutiny that the item has previously received. If something is WUUP, then it has received some testing through widespread use. If something is non-WUUP but has been certified under some other program and/or previously shown to be correct (e.g., an ASIC with a formally verified design), this too may suffice. Whether either of these cases do in fact suffice depends on the specifics. This needs to be discussed with STS before we come to a consensus on how to scope the various reviews. The current standard seems inconsistent in what is excluded from testing in different places, and borderline cases are not handled clearly. [2] refs: I.4.1.1, I.4.1.2, I.4.2.3, II.4.2.1, II.4.6.1, II.5.2. [5] refs: I.5.1.1, I.5.1.2, I.5.2.3, I.7.1, II.4.2.1, II.4.6.1, II.5.2. Issue #2685, need to verify that what vendor asserts is WUUP is actually WUUP. Affected sections in this draft: Volume III Section 3.2.1.1, Volume III Section 3.5.1.1, Volume IV Section 1.4, Requirement IV.1.4-23, Volume V Chapter 3.

Audit log and logging requirements need synchronization with STS. The following sections of [2] have not yet been incorporated: I.2.2.5, I.4.4.1, I.4.4.2, I.4.4.3. See also Volume III Section 3.2.2 and Volume III Section 4.11. It is unclear to what extent ballot accounting and retention of CVRs should be included under auditing or divided between STS and CRT.

All things "remote" (configuration using a network, remote voting, transmitting results—Volume III Section 4.10) need synchronization with STS as most of the issues there are security-related. [2] I.1.5.4 (Public Network DRE) might need incorporating somewhere; I.5 and I.6.5 are already under revision.

All things "integrity." Requirements for maintaining data integrity and detecting and reporting loss of integrity are scattered through the VVSG. Not much distinction was/is made between defending from attacks versus resistance to internally or externally induced data errors. The choices of which requirements to carry over here versus what I assumed STS would take over need to be reviewed and adjusted as needed to cover all bases.

All requirements get descriptive text IDs.

Delete all punchcard? (Never got definitive direction.)

General: after STS, HFP, AG text have been integrated, go through [5], ensure that all useful requirements and informative text are retained somewhere.

Global change: terminology to be resolved:

- Change qualification to certification, testing, or conformity assessment, as appropriate
- ITA versus VSTL versus test lab versus testing authority
- Ballot preparation, formatting, and production: poorly distinguished, may suffice to lump them all together without discriminating
- Ballot configuration versus ballot format versus ballot style: In 2002 VSS, ballot configuration is the abstract, the other two synonymously refer to the concrete. In VVSG'05, ballot configuration and ballot style synonymously refer to the abstract, and ballot format refers to the concrete. Per Murphy's Law, the term most used in practice is the ambiguous one (ballot style).

Harmonize stuff carried over from [2] with any changes made in [5].

Volume I Guidelines overview

Chapter 1 Description and rationale of significant changes vs. [2] or [5]

1.1 Document structure

The VVSG have been restructured to reduce redundancy in the requirements and to reduce fragmentation of requirements.

Figure 1 shows a conceptual model of requirements, including their relationships to activities in the voting process. To structure the VVSG entirely according to the voting process would make it more usable by some readers. Unfortunately, as shown in Figure 1, a given requirement may relate to many activities. To structure the VVSG entirely according to the voting process would create more redundancy rather than less.

The new structure compromises by handling the two most common cases in two separate sections. Those requirements that relate to most or all activities are included in Volume III Chapter 3, General Requirements, and are organized by subject area. Those requirements that relate to single activities are included in Volume III Chapter 4, Requirements by Voting Activity, and are organized by activity. The case that is not handled elegantly is where a given requirement relates to more than one activity, but less than most activities. This has not occurred often enough to present a problem.

A detailed model of the voting process with many activities is included in Volume III Section 5.1. For the purpose of structuring the document, the voting process has been simplified to the list of activities represented by subsections of Volume III Chapter 4.

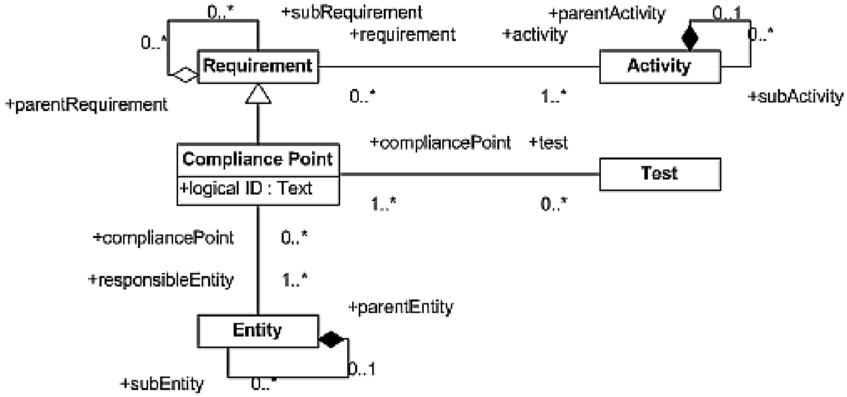


Figure 1 Conceptual model of requirements

Narration of Figure 1

This is a Unified Modeling Language (UML) class diagram with semantics as defined in [6].

Classes: Requirement, Activity, Compliance Point, Test, Entity.

Compliance Point is a subclass of Requirement. Compliance Points have an attribute logical ID of type Text. (A Compliance Point is an identified, testable Requirement.)

Zero to many Requirements in the role subRequirement relate to zero to many Requirements in the role parentRequirement through aggregation. (A requirement may have multiple parents.)

Zero to many Requirements in the role requirement relate to one to many Activities in the role activity.

Zero to many Activities in the role subActivity relate to zero or one Activities in the role parentActivity through composition.

One to many Compliance Points in the role compliancePoint relate to zero to many Tests in the role test.

Zero to many Compliance Points in the role compliancePoint relate to one to many Entities in the role responsibleEntity.

Zero to many Entities in the role subEntity relate to zero or one Entities in the role parentEntity through composition.

1.2 Precision and testability

In Resolution #25-05, the TGDC requested that NIST perform a complete review and revision of requirements in the Voting Systems Standards to ensure that they are sufficiently precise to enable meaningful testing and to expand the testing standards to specify test methods for those requirements.

For certification of voting systems to be consistent, fair, and meaningful, it is necessary to control variability in the conformity assessment system. Testing cannot be an afterthought to a standard: both the requirements to be tested and the methods by which they are to be tested must be specified with appropriate precision. The hypothetical example in Table 1 illustrates the codependence of requirements and test methods.

Example text	Impact on testing
The unit shall respond to all user input in a timely fashion.	Vague requirement leaves tester in the position of determining what is considered "timely," creates opportunities for inconsistent evaluation and challenges by vendors.
The unit shall respond to all user input in 3 seconds or less.	Good requirement leading to pass-fail verdict. However, the test method to verify the requirement is undefined. Different testing authorities using different test methods may get different results. The vendor could challenge that the set of user inputs chosen by a test lab is atypical of use in practice.
The test lab shall measure and report the mean response time and worst response time over the following set of user inputs, employing the test ballot configuration defined in Section XYZ: opening the ballot; voting for one candidate in each contest; [...]. Units with worst response time exceeding 3 seconds shall be disqualified.	In conjunction with the good requirement, this specified test method enables consistent, informative, and difficult-to-challenge results.

Table 1 Requirements and test methods example

In addition, it is necessary that the outputs of the conformity assessment system be consistently applied in the certification process. In response to TGDC Resolution #27-05, NIST has eliminated the provisions in [2] and [5] for certification of voting systems that do not conform to the requirements.[1] One member of the TGDC indicated that this provision was of historical origin and is of no further use.

1.3 Coding conventions

Volume 1, Section 4.2 and Volume 2, Section 5.4 of [2] define coding conventions and a source code review to be conducted by ITAs. Vendors are permitted to use current best practices in lieu of the coding conventions defined in the VSS; however, the coding conventions in the VSS are not aligned with the state of the practice, and if followed, could do more harm than good.

The misalignments are (1) that the conventions, some of which were carried over from [1], are out of date, and (2) that the conventions, limited by their language-independence, are variously incomplete and/or inappropriate in the context of different programming languages with their different idioms and practices.

While they address integrity and maintainability to an extent, the coding conventions are primarily a means to the end of facilitating ITA evaluation of the code's correctness to a level of assurance beyond that provided by black-box testing. That evaluation is underspecified in [2], yielding a cart-before-horse situation in which adherence to the coding conventions could be verified much more rigorously than the correctness of the software.

As part of Resolution #29-05, the TGDC requested that NIST evaluate the [2] software coding standards with respect to their applicability to the recommended standards, and either revise them, delete them, or recommend new software coding standards, as appropriate.

In response, NIST has made recommendations as follows. Coding conventions addressing the need for integrity in voting software have been retained, expanded, and made mandatory,[2] while stylistic conventions that are made redundant by more recent, publicly available coding conventions have been removed in favor of the published conventions. The requirement that these conventions address maintainability is made clear. Whether the coding conventions addressing integrity can also be replaced by recent, publicly available coding conventions for high-integrity software is yet to be determined.

One possibly controversial recommendation included in the changes is to require the use of a programming language that supports structured exception handling. This rules out the C language, which remains in wide use, and forces a migration to a descendant language, namely C++, C#[3] or Java. Similarly, older versions of Visual Basic that lacked structured exception handling are superseded by Visual Basic .NET.

This recommendation is induced by existing requirements in the VSS, specifically:

> I.2.2.5.2.2.g. Nested error conditions shall be corrected in a controlled sequence such that system status shall be restored to the initial state existing before the first error occurred.

> I.4.2.3.e. Each module shall have a single entry point, and a single exit point, for normal process flow. ... The exception for the exit point is where a problem is so severe that execution cannot be resumed. In this case, the design must explicitly protect all recorded votes and audit log information and must implement formal exception handlers provided by the language.

It appears to be the intent of these requirements that the voting system software should (A) exhibit behaviors that are representative of structured exception handling, and (B) accomplish these using "formal exception handlers provided by the language." In context, this is puzzling, since the VSS specifically allowed languages that did not support any semblance of formal exceptions. However, as of 2005, programming languages supporting structured exceptions are widely available and widely used, and they contain other refinements and evolutionary advances, relative to their exceptionless ancestors, that contribute to enhanced software integrity, maintainability, and understandability. To require the use of structured exceptions now is in the same spirit of best practices as the VSS' 1990 requirement for structured control constructs. The alternative is to accept less readable source code and a higher likelihood of masked failures.

It is possible to implement exception handling without use of formal exception handlers, just as it is possible to construct robust programs entirely in assembly language or using only GoTos for control flow. But these less structured techniques obfuscate the code and make logic verification more difficult. "One of the major difficulties of conventional defensive programming is that the fault tolerance actions are inseparably bound in with the normal processing which the design is to provide. This can significantly increase design complexity and, consequently, can compromise the reliability and maintainability of the software." [21]

Though potentially painful, the migration from languages not supporting structured exceptions is facilitated by closely related languages that evolved from one another: C and C++, C# or Java, Visual Basic and Visual Basic .NET.

1.4   Logic verification

This revision of the Voluntary Voting System Guidelines adds logic verification to the testing campaign to achieve a higher level of assurance that the system will count votes correctly.

Traditionally, testing methods have been divided into black-box and white-box test design. Neither method has universal applicability; they are useful in the testing of different items.

Black-box testing is usually described as focusing on testing functional requirements, these requirements being defined in an explicit specification. It treats the item being tested as a "black box," with no examination being made of the internal structure or workings of the item. Rather, the nature of black-box testing is to develop and utilize detailed scenarios, or test cases. These test cases include specific sets of input to be applied to the item being tested. The output produced by the given input is then compared to a previously defined set of expected results.

White-box testing (sometimes called clear-box testing to suggest a more accurate metaphor) allows one to peek inside the "box," and focuses specifically on using knowledge of the internals of the item being tested to guide the testing procedure and the selection of test data. White-box testing can discover extra non-specified functions that black-box testing wouldn't know to look for and can exercise data paths that would not have been exercised by a fixed test suite. Such extras can only be discovered by inspecting the internals.

Complementary to any kind of testing is logic verification, in which formal methods are used to prove that the logic of the system satisfies certain assertions. When it is impractical to test every case in which a failure might occur, formal methods can be used to prove the correctness of the logic generally. However, verification is not a substitute for testing because there can be faults in a formal proof just as surely as there can be faults in a system. Used together, testing and verification can provide a high level of assurance that a system's logic is correct.

A commonly raised objection to logic verification is the observation that, in the general case, it is exceedingly difficult and often impractical to verify any nontrivial property of software. This is not the general case. While these guidelines try to avoid constraining the design, all voting system designs should preserve the ability to demonstrate that votes will be counted correctly. If a voting system is designed in such a way that it *cannot* be shown to count votes correctly, then that voting system does not meet the requirements for certification.

1.5   Public Information Package (PIP)

Public assurance that the voting system is fit for use can occur vicariously, through trust in the test lab and election officials; indirectly, through verification that the certification process was responsibly executed; directly, through election verification; or through a combination of these.

In Resolution #28-05, the TGDC requested that NIST recommend standards on data to be provided, called a "Public Information Package," that must be publicly available and published as evidence that the certification process was responsibly executed. These requirements now appear in Volume IV Chapter 5.

1.6 Deletions

Requirements regarding the system's handling of unofficial data and reports have been deleted or converted to procedural requirements (Requirement VI.1.1-12) because the distinction between unofficial and official data is often outside the scope of the voting system. It is now assumed that any vote data present on a voting system and any reports that it generates are potentially official. Requirements on the reconciliation of provisional ballots and other activities involved in the creation of official data are unaffected by this change.

As discussed in Volume I Section 1.2, the provision for certification of voting systems that do not conform to the requirements has been deleted.

As discussed in Volume I Section 1.3, prescriptive coding conventions not directly related to system integrity have been deleted in favor of published, credible conventions.

Requirements on system and device availability have been deleted because they did not reflect the logistical overhead of repairing equipment on election day and because it is generally impossible to place precinct equipment back into service after it has been repaired on election day without raising concerns about possible tampering. The weight of reliability is now carried by the Mean Time Between Failure requirements, which aim to discourage equipment from failing in the first place.

Requirements to designate one set of redundant cast vote records in a DRE as the "primary" set have been deleted for compatibility with Independent Verification, which requires both sets of records to be usable for reconstructing vote totals.

Requirements that were redundant with the definitions of device classes (e.g., [2] I.2.4.3.2.1.b, all paper-based systems shall allow the voter to punch or mark the ballot to register a vote) have been deleted.

Requirements predicated on state law, local practices, software developed by the voting jurisdiction, and other variables that are indeterminate and untestable in the federal certification process have been deleted.

Requirements that were stated in terms of vague generalities, such as "appropriate" or "intended" options or behavior, for which no precise replacement could be determined and to which no testing value could be ascribed, have been deleted.

Vacuous requirements, such as "Be of any size and shape consistent with its intended use," have been deleted.

Redundant requirements, such as "Comply with the requirements of Section Y" when Section Y is already known to be applicable, have been deleted.

Informative text that was overtaken by changes in the requirements or the structure of the guidelines has been deleted.

Punchcards? Lever machines?

Chapter 2 Options not standardized

2.1 Merged ballot approach to open primaries

In paper-based systems, open primaries have sometimes been handled by printing a single ballot format that merges the contests from all parties and instructing the voter to vote only in the contests applicable to a single party. This approach requires additional logic in the tabulator to support the rejection or discarding of votes that violate these special instructions, while the approach of assigning different ballot formats to different parties does not.

Support for the merged ballot approach is not required for a tabulator to satisfy the requirements in these Guidelines for support of open primaries. Although the merged ballot approach does allow the selection of party to be made in private, the issues with usability and tabulation logic that it incurs raise doubt of whether the benefits of standardizing the approach would exceed the cost in added complexity. Voting systems may provide this option as an extension to the Guidelines without breaking conformance.

In systems affected by this issue, assigning different ballot configurations for different parties sacrifices the privacy of the party selection to avoid the issues with usability and tabulation logic. However, the conflict addressed in this trade-off exists *only* in paper-based systems where poll workers are responsible for giving voters the correct ballot format. DREs and EBPs can provide privacy for the selection of party and then activate a ballot that contains only the contests appropriate to that selection.

2.2 Recall candidacy linked to recall question

In some jurisdictions, a vote for a candidate to replace a recalled official is counted only if the recall question on the same ballot was voted, and sometimes only if it was voted in the affirmative. Like the merged ballot approach to open primaries, the issues with usability and tabulation logic that this approach incurs raise doubt of whether the benefits of standardizing the approach would exceed the cost in added complexity. Voting systems may provide this option as an extension to the Guidelines without breaking conformance.

Chapter 3 List of sources reviewed

Revise to cite all of the state laws and whatever else that have been reviewed.
Should we mention the test reports? (Do not contain requirements *per se*)

3.1 Review of existing standards, specifications, and related work

To ensure that previously written requirements would not be overlooked, NIST reviewed the following resources. The resulting guide to existing requirements has not been put into publishable form but is being utilized by project members as they develop new recommendations.

NIST also reviewed sample ballot formats, vote data reports and other materials from several states.

3.1.1 Standards, draft standards, regulations, and guidelines

Rev. all VSS/VVSG

[HAVA] Help America Vote Act of 2002, Public Law 107-252, 2002-10-29.

[2002VSS] 2002 Voting Systems Standards, available from http://www.fec.gov/pages/vssfinal/vss.html.

[P1583/D5.3.1] IEEE Draft Standard for the Evaluation of Voting Equipment, draft 5.3.1, 2004-10-08, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[CoE 2004-09-30] Council of Europe, Committee of Ministers to member states on legal, operational, and technical standards for e-voting, adopted by the Committee of Ministers on 2004-09-30 at the 898th meeting of the Minister's Deputies, e-mail from Lori Steele, 2004-11-10.

[EML4] Election Markup Language v4.0, 2005-01-24, available from http://www.oasis-open.org/committees/election/index.shtml.

[SP 500-256] Sharon J. Laskowski et al., "Improving the Usability and Accessibility of Voting Systems and Products," NIST SP 500-256, 2004-05, available from http://www.vote.nist.gov/Final%20Human%20Factors%20Report%20%205-04.pdf.

[508] Section 508 of the Rehabilitation Act: Electronic and Information Technology Accessibility Standards, 2000-12-21, available from http://www.access-board.gov/508.htm.

[ADA] ADA Checklist for Polling Places, 2004-02, available from http://www.usdoj.gov/crt/ada/votingchecklist.htm.

3.1.2 Issue lists

Update to 5.3.2b?

[D5.3.1 Comments 2004-10-19] Comments for d5-3-1 dated 10-19-2004 revC.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Software Comments 2004-09-01] Software comments 5.0 (9-01-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Security Comments 2004-08-18] Security extract V5 Comments - 2nd NJ Meeting.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Reliability Accuracy Comments 2004-09-06] 5.0 Comments Section 5.2 & 6.2 (9-6-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Accessibility Comments 2004-08-01] V5 Ballot Accessibility Comments - TG3 (8-1-04) .xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Environmental 2004-08-15] 5.0 Comments Section 5.4 & 6.4.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 EMC 2004-08-23] 5.0 Comments Section 5.5 (8-23-04).xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Provisional 2004-09-10] Gough-Provisional Ballot Comments.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 COTS 2004-06-18] Resolutions for COTS Comments for Draft 5.0 of IEEE P-1583, http://www.lipsio.com/COTS/docs/COTS.resolved.html.

[5.0 TDP 2004-04-23] 5.0 p1583 _TDP-Proposed resolution_Apr04.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

[5.0 Comments 2003-10-16] Ballot Comment Form 5-0 10-16-2003.xls, available from http://grouper.ieee.org/groups/scc38/1583/private/history_041019.html (password-protected).

3.1.3 Requests for proposals

[AR] "A request for proposals for a Help America Vote Act compliant voting system," Request for Proposals # ARSOS-HAVA--005, Arkansas, 2005-07-11. Available from http://www.sos.arkansas.gov/hava-rfp-vote-system.html, 2006-01-25.

[AZ] "OCR and DRE Voting Equipment - Statewide," Request for Proposal, Arizona, 2003. E-mail from Allan Eustis, 2004-10-12.

[CO-REG] "Statewide Voter Registration System," Request for Proposals # DOS-HAVA-0001, Colorado, 2004-01-16, formerly available from http://www.sos.state.co.us/pubs/hava/hava_main.htm (now gone).

[CO-IVV] "Independent Verification and Validation for SCORE Project," Request for Proposals # DOS-HAVA-0002, Colorado, 2004-06-03, formerly available from http://www.sos.state.co.us/pubs/hava/hava_main.htm (now gone).

[GA] Request for Proposal GTA000040, Georgia, 2001. Available from http://www.nass.org/Georgia RFP.pdf, 2006-01-25.

[KS] Request for Proposal # 08455, Kansas, 2005-05-16. Available from http://www.kssos.org/elections/05elec/Voting_Equipment_RFP.pdf, 2006-01-25.

[MD] "Direct Recording Electronic Voting System and Optical Scan Absentee Voting System for Four Counties," Project Number SBE-2002-01, Maryland, 2001-07-17, available from http://www.elections.state.md.us/citizens/voting_systems/voting_system_procurement.html.

[MI] Invitation To Bid # 071I4001011, Michigan, 2003, available from http://www.michigan.gov/sos/0,1607,7-127-1633_11619_27151-77943--,00.html.

[MS] Request For Proposal # 3443, Mississippi, 2005. Available from http://www.its.state.ms.us/rfps/3443.htm, 2006-01-25.

[ND] Request for Proposals # 108.6-03-001, North Dakota, 2003-10-31. Available from http://www.state.nd.us/hava/documents/docs/vsp-rfp-official.pdf, 2006-01-26.

[OH-VOT] "Statewide Voting System(s)," Request For Proposal # SOS0428365, Ohio, 2003-05-23, available from http://www.sos.state.oh.us/sos/hava/index.html.

[OH-REG] Request For Proposal # SOS032786279, Ohio, 2003-04-09, available from http://www.sos.state.oh.us/sos/hava/index.html.

[UT] Solicitation # DG5502, Utah, 2004-07-09, available from http://purchasing.utah.gov/BidHeaders/8750.pdf, 2006-01-27.

3.1.4 Testimony

[Coney 2004-09-22] Lillie Coney, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Conrad 2004-09-22] Frederick Conrad, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Deutsch 2004-09-21] Herb Deutsch, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Fischer 2004-09-20] Eric A. Fischer, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Gaston 2004-09-20] Charles A. Gaston, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Golden 2004-09-22] Diane Cordry Golden, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Jones 2004-09-20] Douglas W. Jones, testimony to EAC, available from http://www.cs.uiowa.edu/%7Ejones/voting/nist2004.shtml.

[Jones 2004-09-23] Douglas W. Jones, supplemental testimony to EAC, available from http://www.cs.uiowa.edu/%7Ejones/voting/nist2004supp.shtml.

[King 2004-09] Merle S. King, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Noren 2004-09] Wendy S. Noren, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Redish 2004-09-22] Janice Redish, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Relton 2004-09-21] Joy Relton, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Saltman 2004-09-20] Roy G. Saltman, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Shamos 2004-09-20] Michael I. Shamos, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

[Wallach 2004-09-20] Dan S. Wallach, testimony to EAC, available from http://vote.nist.gov/sept04hearings.htm.

Volume II Terminology standard

The Terminology Standard defines terms that are used in the Product Standard, Standards on Data to be Provided, and Testing Standard.

Terminology for standardization purposes must be sufficiently precise and formal to avoid ambiguity in the interpretation and testing of the standard. Terms must be defined to mean exactly what is intended in the requirements of the standard, no more and no less. Consequently, this terminology may be unsuitable for applications that are beyond the scope of the Guidelines. Readers are especially cautioned to avoid comparisons between this terminology and the terminology used in election law.

These terms are being defined as needed to disambiguate product requirements. They are not necessarily compatible with the glossary of [5], nor do they necessarily represent a NIST consensus.

Merge in relevant definitions from [5], [4], and misc. drafts as needed, with corrections per review.

Some comments in /home/dflater/HAVA/misc/Verified Voting Foundation Glossary Comments.pdf may still apply by the time VVSG2 glossary is nearing completion. Review at that time.

1-of-M voting: N-of-M voting where *N* = 1.

absentee ballot: Ballot resulting from absentee voting.

absentee voting: Voting that can occur unsupervised at a location chosen by the voter.

active period: Span of time during which a vote-capture device is either ready to begin a voting session or is in use in a voting session. See Volume III Section 5.2.

Acc-VS: (Accessible Voting Station) Voting station equipped for individuals with disabilities referred to in 42 USC 15481 (a)(3)(B).

administrator: Role defined in Dangling ref: STS Draft Access Control Section.

ALVS: (Alternative Language Voting Station) Voting station that provides alternative language accessibility pursuant to 42 USC 1973aa-1a.

archival: Able to survive for a period of time without significant loss. Note: In the context of voting, the relevant period of time is usually 22 months. See Volume III Section 3.6.2.

archivalness: Ability of a medium to preserve its content for a period of time without significant loss. Note: In the context of voting, the relevant period of time is usually 22 months. See Volume III Section 3.6.2.

ballot image: Electronically produced record of all votes cast by a single voter.

ballot rotation: Process of varying the order of the candidate names within a given contest.

benchmark: Quantitative point of reference to which the measured performance of a system or device may be compared.

callable unit: (Of a software program or analogous logical design) Function, method, operation, subroutine, procedure, or analogous structural unit that appears within a module.

card puncher: Vote-capture device that allows a voter to record votes by means of holes punched in designated voting response locations.

cast ballot: Ballot in which the voter has taken final action in the selection of candidates and measures and irrevocably confirmed his or her intent to vote as selected. See also read ballot and counted ballot.

cast vote record: Archival record of all votes produced by a single voter. Note: Cast vote records may be in electronic, paper, or other form. Electronic cast vote records are also called ballot images.

central election official: Role defined in Dangling ref: STS Draft Access Control Section.

central tabulator: Tabulator that counts votes from multiple precincts at a central location. Note: Voted ballots are typically placed into secure storage at the polling place and then transported or transmitted to a central tabulator. A tabulator that may be configured for use either in the precinct or in the central location may satisfy the requirements for both *Precinct tabulator* and *Central tabulator*.

challenged ballot: Ballot cast by a voter whose eligibility to vote is disputed by someone who is not an election official. See also provisional ballot.

class: (1) Identified set of requirements. (2) Voting systems or devices to which those requirements apply. See Volume III Section 2.6.

closed primary: Primary election in which the voter receives a ballot containing only those partisan contests pertaining to the political party with which the voter is affiliated, along with nonpartisan contests and ballot issues presented at the same election. Note: Usually, unaffiliated voters are permitted to vote only on nonpartisan contests and ballot issues.

combined precinct: Two or more precincts assigned the same polling place.

conformity assessment: Demonstration that specified requirements relating to a product, process, system, person or body are fulfilled. ([11])

counted ballot: Read ballot whose votes are included in the candidate and measure vote totals. See also cast ballot and read ballot.

crossover vote: Scratch vote. Note: The term scratch vote is preferred because crossover vote is more likely to be misinterpreted.

cross-party endorsement: Endorsement of a given candidate by two or more political parties.

cumulative voting: Voting variation in which the voter is entitled to allocate a fixed number of votes (*N*) over a list of *M* candidates or write-ins. Note: Unlike N-of-M voting, cumulative voting allows the voter to allocate more than one vote to a given candidate.

CVR: Cast vote record.

device: Functional unit that performs its assigned tasks as an integrated whole.

DRE: (Direct Record Electronic) Combination vote-capture device and tabulator that gathers votes via an electronic voter interface, records voting data and ballot images in memory components, and produces a tabulation of the voting data.

EBM: (Electronically-assisted Ballot Marker) Vote-capture device that gathers votes via an electronic voter interface and produces an executed paper ballot as a result. Note: The ballot output by an EBM may or may not include a bar code. An EBM may mark ballot positions on a pre-printed ballot or it may print an entire ballot. The latter kind are called EBPs.

EBP: (Electronic Ballot Printer) EBM that prints an entire ballot.

election district: Administrative division in which voters are entitled to vote in contests that are specific to that division, such as those for state senators and delegates. Note: An election district may overlap multiple precincts, and a precinct may overlap multiple election districts (see split precinct).

election judge: Role defined in Dangling ref: STS Draft Access Control Section.

election verification: Confirmation that all recorded votes were counted correctly. See also voter verification.

electronic device: Device that uses electricity.

electronic voter interface: Component of an electronic vote-capture device that communicates ballot information to the voter and accepts input from the voter.

EMPB: (EBM-Marked Paper Ballot) Ballot marked by an EBM.

EMS: (Election Management System) Tabulator used to prepare ballots and programs for use in casting and counting votes and to consolidate, report, and display election results. Note: The EMS produces a printed report of the vote count and may produce a report stored on electronic media.

error: (Voting system accuracy) Incorrect capturing, recording, storing, consolidation, or reporting of a vote or of data that denote ballot format or precinct. (Source: Derived from [2] I.3.2.1, I.3.2.6.1.1 and II.C.5.) Note: The word "error" is used in the general sense elsewhere, e.g., in Volume III Section 3.2.1.2.

error rate: Value of the ratio of the number of errors that occur to the volume of data processed. ([2] I.3.2.1)

failure: (Voting system reliability) Event that results in (a) loss of one or more functions, (b) degradation of performance such that the device is unable to perform its intended function for longer than 10 seconds, (c) automatic reset, restart or reboot of the voting system, operating system or application software, (d) a requirement for an unanticipated intervention by a person in the role of poll worker or technician before the test can continue, or (e) error messages and/or audit log entries indicating that a failure has occurred. (Source: Expanded from [2] I.3.4.3.)

find: Determine and deliver a finding. (Based on [20] definition #11.)

finding: Result of a formal evaluation by a test lab or accredited expert; verdict. (Based on [20] definition #6.)

firmware: Computer programs (software) stored in nonvolatile memory.

general election: Election in which there are no partisan contests.

hesitation mark: Small dot made by resting the point of a writing utensil on a ballot.

implementation statement: Statement by a vendor indicating the capabilities, features, and optional functions and extensions that have been implemented in a voting system. Note: Also known as implementation conformance statement.

in-person voting: Voting that occurs at a polling place under the supervision of poll workers.

inspection: Examination of a product design, product, process or installation and determination of its conformity with specific requirements or, on the basis of professional judgement, with general requirements. ([11])

instant runoff voting: Ranked order voting.

MMPB: (Manually-Marked Paper Ballot) (1) Vote-capture device consisting of a paper ballot and a writing utensil. (2) Paper ballot that was marked by a person using a writing utensil.

module: Structural unit of software or analogous logical design, typically containing several callable units that are tightly coupled. Note: Modular design requires that inter-module coupling be loose and occur over defined interfaces. A module should contain all elements needed to compile or interpret successfully and have limited access to data in other modules. A module should be substitutable with another module whose interfaces match the original module. In software, a module typically corresponds to a single source code file or a source code / header file pair. In object-oriented languages, this typically corresponds to a single class of object.

MTBF: (Mean Time Between Failure) Value of the ratio of operating time to the number of failures that occur. ([2] I.3.4.3)

N-of-M voting: Voting variation in which the voter is entitled to allocate a fixed number of votes (*N*) over a list of *M* candidates or write-ins, with the constraint that at most 1 vote may be allocated to a given candidate. See also cumulative voting.

nonpartisan contest: Contest such that eligibility to vote in that contest is independent of political party affiliation or lack thereof.

nonvolatile memory: Memory in which information can be stored indefinitely with no power supplied. Note: Read-only memory (ROM), programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory are examples of nonvolatile memory.

open primary: Primary election in which the voter may choose a political party at the time of voting and vote in partisan contests associated with that party, along with nonpartisan contests and ballot issues presented at the same election. Note: Also known as pick-your-party primary. Some states require voters to publicly declare their choice of party at the polling place, after which the poll worker provides or activates the appropriate ballot. Other states allow the voters to make their choice of party within the privacy of the voting booth. Voters also are permitted to vote on nonpartisan contests and ballot issues that are presented at the same election.

operational test: Test conducted on voting equipment in an active (operational) state by a procedure in the form of a scientific experiment.

operational testing: Testing using operational tests.

optical scanner: Tabulator that counts votes that were recorded by means of marks made on the surface of a paper ballot.

paper-based device: Device that records votes, counts votes, and/or produces a report of the vote count from votes cast on paper cards or sheets.

partisan contest: Contest such that eligibility to vote in that contest is restricted based on political party affiliation or lack thereof. Note: The affiliation might be the registered affiliation of the voter or it might be an affiliation declared at the time of voting. See closed primary, open primary.

PCOS: (Precinct Count Optical Scanner) Optical scanner used as a precinct tabulator.

poll worker: Role defined in Dangling ref: STS Draft Access Control Section.

precinct: Administrative division in which voters cast ballots at the same polling place. Note: It is possible for two or more precincts to cast ballots at a given polling place. See combined precinct.

precinct tabulator: Tabulator that counts votes at the polling place. Note: These devices typically tabulate ballots as they are cast and print the results after the close of polls. For DREs and some paper-based systems, these devices provide electronic storage of the vote count and may transmit results to a central location over public telecommunication networks. A tabulator that may be configured for use either in the precinct or in the central location may satisfy the requirements for both *Precinct tabulator* and *Central tabulator*.

primary election: Election in which there are partisan contests. Note: Primary elections are held to determine which candidate will represent a political party in a subsequent general election.

profile: Subset of a standard for a particular constituency or purpose that defines the requirements, options, constraints, and extensions that are specific to that constituency or purpose.

programmed device: Electronic device that includes software or firmware installed or commissioned by the voting system vendor.

provisional ballot: Ballot cast by a voter whose eligibility to vote is disputed by an election official. See also challenged ballot.

punchcard reader: Tabulator that counts votes that were recorded by means of holes punched through a paper ballot.

ranked order voting: Voting variation in which voters express their intent by ordering candidates from strongest to weakest preference. Note: Implementations of ranked order voting differ in whether voters are required to rank every candidate and in the algorithm used to determine a winner or winners.

read ballot: Cast ballot that has been processed. Note: A read ballot may or may not be counted. For example, an optical scan cast ballot that has been scanned successfully is a read ballot. See also cast ballot and counted ballot.

record: Preserved evidence of activities performed or results achieved (e.g., forms, reports, test results).

report: Self-contained, timestamped, archival record, such as a printout or analogous electronic file, that is produced at a specific time and subsequently protected from modification.[4]

review-required ballot: Ballot that is flagged or separated for some form of manual processing.

scratch vote: Explicit vote that conflicts with the vote(s) implied by a straight party vote. ([15]) Note: Also called crossover vote.

split precinct: Precinct serving voters from two or more administrative divisions, such as election districts, that require different ballot configurations.

straight party voting: Voting variation in which the selection of a political party in a special contest implies votes for the candidates endorsed by that party in all straight-party-votable contests on the ballot.

tabulator: Device that counts votes.

testing: Determination of one or more characteristics of an object of conformity assessment, according to a procedure. Note: "Testing" typically applies to materials, products or processes. ([11])

thought mark: Hesitation mark.

vote-capture device: Device that is used directly by a voter to vote a ballot.

voter: Role defined in Dangling ref: STS Draft Access Control Section.

voter verification: Confirmation that all votes were recorded as the voter intended. See also election verification. Note: It is debatable whether an ambiguous record, such as an MMPB containing marginal marks or a punchcard containing dimpled or hanging chads, satisfies the intent of voter verification. On the one hand, the paper record was produced directly by the voter and deliberately cast, so arguably it represents the intent of the voter. On the other hand, a conscientious voter would never intentionally cast an ambiguous ballot.

voting device: Device that is part of the voting system.

voting process: Entire array of procedures, people, resources, equipment and locations associated with the conduct of elections.

voting session: (1) Span of time beginning when a ballot is enabled and ending when that ballot is printed, cast or spoiled (depending on the technology used). See Volume III Section 5.2. (2)

Interaction between the voter and vote-capture device that occurs during that span of time.

voting station: Vote-capture device with its privacy enclosure.

voting system: Integrated equipment that is used to define ballots, to cast and count votes, to report and/or display election results, and to maintain and produce audit trail information, plus associated documentation used to operate the system, maintain the system, identify system components and their versions, test the system during its development and maintenance, maintain records of system errors and defects, and determine specific changes made after system certification. Note: A voting system may also include the transmission of results over telecommunication networks.

VVPAT: (Voter-Verified Paper Audit Trail) Voting device that supports voter verification using a paper audit trail.

write-in: Vote for a candidate who is explicitly named by the voter in lieu of choosing a candidate who is already listed on the ballot. Note: This does not preclude writing in the name of a candidate who is already listed on the ballot.

WUUP: (Widely Used Uncustomized Package) Software, firmware, device or component that is used by many different people or organizations for many different applications and that is incorporated into the voting system with no vendor- or application-specific modification. Note: This term is an attempted generalization of the more familiar COTS (Commercial Off-The-Shelf) term to include non-commercial packages.

Volume III Product standard

Chapter 1 Introduction

1.1 Structure

NIST is recommending a reorganization of the VVSG to bring them in line with applicable standards practices that are abstracted from our years of association with ISO, W3C and other standards-creating organizations. This includes expanding the *conformance clause* that was added in [5], identifying testable requirements as *compliance points*, and defining *classes*, which allow requirements to vary as needed to accommodate variations in voting equipment.

Preferably, requirements should specify *what* (the desired performance), not *how* (a design to accomplish that). For example, a requirement that reads "single-bit errors shall be detected" is preferable to one that reads "products shall use memories with parity bits." Classes are created to resolve the conflict that occurs when the *what* depends on the *how*. For example, the unstated assumption that the voting equipment would have an electronic memory at all requires placing the preceding example in a subclass for electronic voting equipment.

Design-constraining requirements are controversial because vendors would like the freedom to provide the desired qualities / performance in different ways. However, in cases where vendors are unable to determine for themselves whether or not a given design is conforming, they may welcome design constraints as a way to avoid repeated failures and costly retesting of their products. Moreover, in cases where the desired quality is difficult to define abstractly, an enumeration of conforming cases may be the only practical alternative, particularly if there is only one design approach that is ever actually usable in practice. Some pragmatism is required.

A vendor who is submitting a system for testing must make an *implementation statement* that identifies exactly which classes the system is asserted to support. Conformity assessment activities are catalogued according to which compliance points they exercise. The set of conformity assessment activities appropriate to that system may then be determined automatically. Upon passing those tests and reviews, the system may be qualified for only the claimed classes. There is no provision for certification of voting systems that do not conform to the requirements (see Volume I Section 1.2).

Identified compliance points and a classification mechanism in the VVSG facilitate traceability from state standards to the VSS. States may define their own profiles over the VVSG, adding compliance points they deem necessary without excessive repetition and revision of VVSG text.

Chapter 2 Conformance Clause

2.1 Scope and applicability

The Voluntary Voting System Guidelines are intended primarily for use by:

- Designers and manufacturers of voting systems;
- Test labs performing the analysis and testing of voting systems in support of the EAC national certification process;
- Software repositories designated by the EAC or by a state; and
- Test labs and consultants performing the state certification of voting systems.

The Guidelines may also be of use to election officials in setting requirements for voting systems in requests for proposals.

The Guidelines include:

- A product standard (Volume III) defining requirements that apply to voting systems that

vendors produce;
- Standards on data to be provided (Volume IV) defining requirements that apply to documentation, reports, and other information that vendors and test labs deliver;
- A testing standard (Volume V) defining test methods and protocols that test labs implement; and
- A terminology standard (Volume II) defining terms used in the foregoing.

2.2 Structure of requirements

Each voting system requirement is identified according to a hierarchical scheme in which higher-level, "parent" requirements (such as "provide accessibility for visually impaired voters") are supported by lower-level subrequirements (e.g., "provide an audio-tactile interface"). Thus, requirements are nested.

Some requirements are directly testable and some are not. The latter tend to be higher-level and are included because (1) they are testable indirectly insofar as their lower-level requirements are testable, and (2) they often provide the structure and rationale for the lower-level requirements.

The applicability of a requirement is specified with the *Applies to:* field, which indicates the class(es) of voting systems or devices to which the requirement applies. Classes are defined in Volume III Section 2.6.

A requirement having *N* different classes separated by commas in its *Applies to:* field is equivalent to *N* different requirements that apply to each listed class individually.

The scope of a parent requirement is inherited by its subrequirements unless they explicitly specify a narrower scope. The scope may be narrowed through a generic relation (e.g., *DRE* is a subclass of *Vote-capture device*) or a partitive relation (e.g., a *DRE* is part of a *Voting system*).

2.3 Normative language

The following keywords are used to convey conformance requirements:

- **Shall** indicates a mandatory requirement to do something. Synonymous with "is required to."
- **Is prohibited** indicates a mandatory requirement *not* to do something. Synonymous with "shall not."
- **Should, Is encouraged** indicate an optional recommended action, one that is particularly suitable, without mentioning or excluding others. Synonymous with "is permitted and recommended."
- **May** indicates an optional, permissible action. Synonymous with "is permitted."

Text using these keywords is directly applicable to achieving conformance to the Guidelines.

Informative parts of this document include discussion, examples, extended explanations, and other matter that is necessary for proper understanding of the Guidelines and conformance to them.

2.4 Conformance designations

A voting system conforms to the product standard if all stated requirements that apply to the voting system and its constituent devices are fulfilled. The implementation statement (see Volume III Section 2.5) declares the capabilities, features and optional functions that have been implemented and are subject to conformance and certification testing.

There is no concept of partial conformance—neither that a voting system is *x* % conforming, nor that a device that is not a complete voting system by itself is conforming. Individual devices of voting systems are not tested or certified except as parts of complete systems.[5]

2.5 Implementation statement

An implementation statement documents the requirements that have been implemented by the voting system, the optional features and capabilities supported by the voting system, and any extensions (i.e., additional functionality beyond what is defined in the Guidelines) that it implements.

An implementation statement may take the form of a checklist to be completed for each voting system submitted for certification. It is used by test labs to identify the conformity assessment activities that are applicable.

Ⓡ 2.5-1 Implementation statement

An implementation statement shall include:

a. Full product identification of the voting system, including version number or timestamp;

b. Separate identification of each device (see below) that is part of the voting system;

c. Version of VVSG to which certification is desired;

d. Classes implemented (see Volume III Section 2.6.3);

e. Device capacities and limits (especially those appearing in Volume III Section 5.3.1); and

f. List of languages supported.

Source: New requirement.

DISCUSSION

A keyboard, mouse or printer connected to a programmed vote-capture device, as well as any optical drive, hard drive or similar component installed within it, are considered components of the vote-capture device, not separate devices. The vote-capture device is "responsible" for these components—e.g., a DRE must prevent unauthorized flashing of the firmware in its optical drive or other components that could be subverted to manipulate vote outcomes.

Vendors may wish to contact their intended testing labs in advance to determine if those labs can supply them with an implementation statement *pro forma* to facilitate meeting this requirement.

2.6 Classes

2.6.1 Voting device terminology

Voting device	Device that is part of the voting system. *Voting device* subsumes *Vote-capture device*, *Paper-based device*, *Electronic device*, *Tabulator*, *VVPAT*, and all device voting variations (*In-person voting device*, etc.).
Vote-capture device	Device that is used directly by a voter to vote a ballot. *Vote-capture device* subsumes *ALVS*, *Acc-VS*, *Card puncher*, *MMPB*, *EBM*, and *DRE*.
Paper-based device	Device that records votes, counts votes, and/or produces a report of the vote count from votes cast on paper cards or sheets. *Paper-based device* subsumes *Card puncher*, *MMPB*, *EBM*, *Punchcard reader*, and *Optical scanner*.
Electronic device	Device that uses electricity. *Electronic device* subsumes *Programmed device* and *Punchcard reader*.
Programmed device	Electronic device that includes software or firmware installed or commissioned by the voting system vendor. *Programmed device* subsumes *EBM*, *DRE*, *Optical scanner*, and *EMS*.
Tabulator	Device that counts votes. *Tabulator* subsumes *DRE, Punchcard reader*, *Optical scanner*, *EMS*, *Precinct tabulator* and *Central tabulator*.
Precinct tabulator	Tabulator that counts votes at the polling place. Note: These devices typically tabulate ballots as they are cast and print the results after the close of polls. For DREs and some paper-based systems, these devices provide electronic storage of the vote count

	and may transmit results to a central location over public telecommunication networks. A tabulator that may be configured for use either in the precinct or in the central location may satisfy the requirements for both *Precinct tabulator* and *Central tabulator*. *Precinct tabulator* subsumes *PCOS*.
Central tabulator	Tabulator that counts votes from multiple precincts at a central location. Note: Voted ballots are typically placed into secure storage at the polling place and then transported or transmitted to a central tabulator. A tabulator that may be configured for use either in the precinct or in the central location may satisfy the requirements for both *Precinct tabulator* and *Central tabulator*.
Card puncher	Vote-capture device that allows a voter to record votes by means of holes punched in designated voting response locations.
MMPB	(Manually-Marked Paper Ballot) Vote-capture device consisting of a paper ballot and a writing utensil.
EBM	(Electronically-assisted Ballot Marker) Vote-capture device that gathers votes via an electronic voter interface and produces an executed paper ballot as a result. Note: The ballot output by an EBM may or may not include a bar code. An EBM may mark ballot positions on a pre-printed ballot or it may print an entire ballot. The latter kind are called EBPs. *EBM* subsumes *EBP*.
EBP	(Electronic Ballot Printer) EBM that prints an entire ballot.
VVPAT	(Voter-Verified Paper Audit Trail) Voting device that supports voter verification using a paper audit trail.
DRE	(Direct Record Electronic) Combination vote-capture device and tabulator that gathers votes via an electronic voter interface, records voting data and ballot images in memory components, and produces a tabulation of the voting data.
Punchcard reader	Tabulator that counts votes that were recorded by means of holes punched through a paper ballot.
Optical scanner	Tabulator that counts votes that were recorded by means of marks made on the surface of a paper ballot. *Optical scanner* subsumes *PCOS*.
EMS	(Election Management System) Tabulator used to prepare ballots and programs for use in casting and counting votes and to consolidate, report, and display election results. Note: The EMS produces a printed report of the vote count and may produce a report stored on electronic media.
PCOS	(Precinct Count Optical Scanner) Optical scanner used as a precinct tabulator.
ALVS	(Alternative Language Voting Station) Voting station that provides alternative language accessibility pursuant to 42 USC 1973aa-1a.
Acc-VS	(Accessible Voting Station) Voting station equipped for individuals with disabilities referred to in 42 USC 15481 (a)(3)(B).

Table 2 Voting device terminology

2.6.2 Classes overview

A class simultaneously identifies a set of requirements and a set of voting systems or devices to which those requirements apply. The purpose of classes is to categorize requirements into related groups of functionality that apply to different types of voting systems and devices.

Classes may subsume other classes. For example, *Paper-based device* subsumes *Card puncher*, *Punchcard reader*, and *Optical scanner*. The subsuming class is called the superclass while the subsumed classes are called subclasses. A group of related classes forms a classification hierarchy or lattice.

Subclasses "inherit" the requirements of their superclasses. Additionally, a subclass may further constrain a class by adding new requirements. However, a subclass may not relax or remove requirements inherited from a superclass.

Classes may be declared to be *disjoint* (mutually exclusive), or not, as appropriate.

A voting system conforms to a class if all stated requirements identified by that class are fulfilled. Since subclasses may not relax or remove requirements inherited from a superclass, it is true in all cases that a voting system conforming to a subclass also conforms to all of its superclasses. For example, a voting system conforming to any subclass of *Voting system* fulfills the general requirements that apply to all voting systems.

The classification mechanism is useful in many different contexts when there is a need to identify specific portions of the VVSG. Table 3 provides several examples.

Context	Use
VVSG	Requirements applicable to a given class
Implementation statement	This system conforms to a specified class
Conformity assessment	Tests and reviews applicable to the specified class
Certification	Scope of certification is the specified class
Declaration of conformity	This product is certified to that class
Request for proposals	Seeking to procure a system conforming to a specified class

Table 3 Use of classes in different contexts

Figure 2 and Figure 3 repeat in pictorial form the classification hierarchies that are defined in the next section to illustrate their high-level structure. A class is represented by an oval containing the name of the class. When two classes are connected by a line, this indicates that the higher class subsumes the lower one.

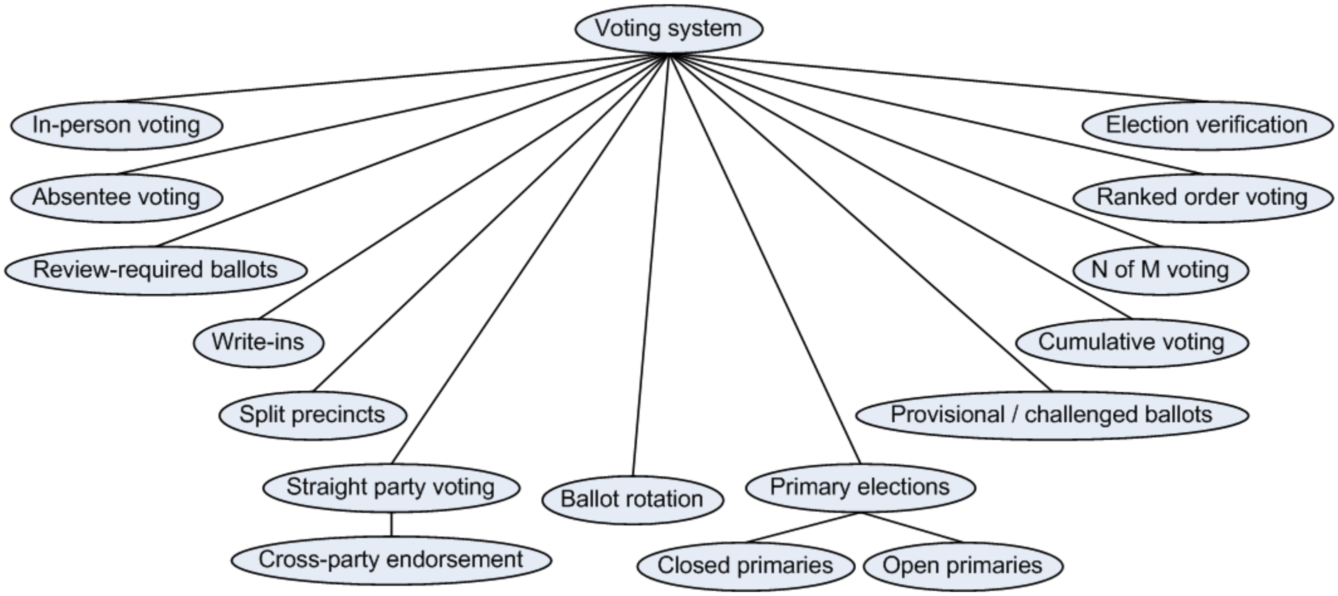


Figure 2 Voting system classes

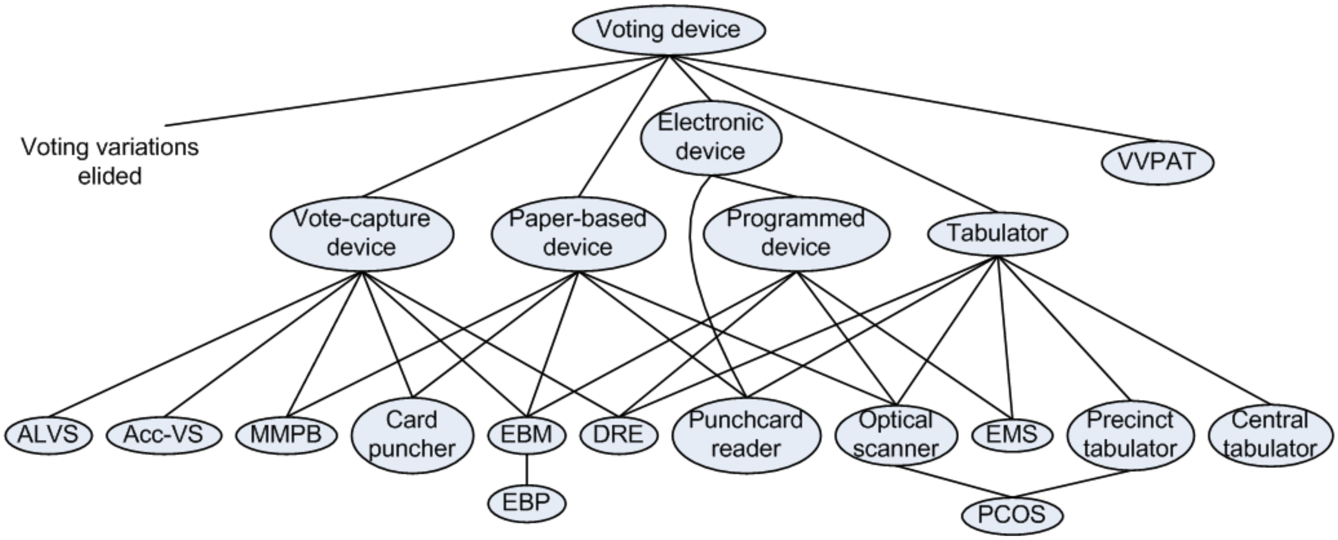


Figure 3 Voting device classes

2.6.3 Classes identified in implementation statement

2.6-1 Implementation statement, system classes

An implementation statement for a voting system shall identify all applicable classes from Volume III Section 2.6.3.1.

Source: New requirement.

2.6-2 Implementation statement, device classes

For each distinct device included in the system, an implementation statement for a voting system shall identify:

a. All applicable classes from Volume III Section 2.6.3.2; and

b. All applicable classes from Volume III Section 2.6.3.3.

Source: New requirement.

In the following subsections, references following the names of classes indicate the origin of those classes.

2.6.3.1 Supported voting variations (system-level)

The classes enumerated in this section identify voting variations supported by the voting system. Although the intent of most is apparent from the applicable requirements, the following may require additional explanation.

Conformance to the *Write-ins* class indicates that the voting system is capable of end-to-end processing of write-in votes. If the voting system requires that write-in votes be counted manually, then it does not satisfy Requirement III.4.8-1 and therefore does not conform to the *Write-ins* class. However, it may conform to the *Review-required ballots* class (see below).

The same principle applies to the *Absentee voting* class and the *Provisional / challenged ballots* class. If the counting of these ballots is external to the voting system, then the system does not satisfy Requirement III.4.8-2 or Requirement III.4.8-3 and therefore does not conform to the *Absentee voting* or *Provisional / challenged ballots* class, respectively.

Conformance to the *Review-required ballots* class indicates that the voting system is capable of flagging or separating ballots for later processing and including the results of that processing in the reported totals. If the consolidation of counts from review-required ballots with counts from other ballots is external to the voting system, then the system does not satisfy Requirement III.4.9-27 and therefore does not conform to the *Review-required ballots* class.

In some systems, write-in votes are counted as a single ballot position representing all write-ins, and these votes are assigned to candidates through manual post-processing if and only if the election is close enough to warrant the effort. Although this approach does not conform to the *Write-ins* class, the system's handling of write-in votes is identical to its handling of other ballot positions, so the behavior is verifiable.

Choose all that apply.

- In-person voting ([2] I.2.5.2)
- Absentee voting ([2] I.2.5.2)
- Provisional / challenged ballots ([2] I.2.5.2, I.2.2.8.2o)
- Review-required ballots ([2] I.2.5.2)
- Primary elections
    - Closed primaries ([2] I.2.2.8.2a)
    - Open primaries ([2] I.2.2.8.2b)
- Write-ins ([2] I.2.2.8.2e)
- Ballot rotation ([2] I.2.2.8.2g)
- Straight party voting ([2] I.2.2.8.2h)
    - Cross-party endorsement ([2] I.2.2.8.2i)
- Split precincts ([2] I.2.2.8.2j)
- N of M voting ([2] I.2.2.8.2k)
- Cumulative voting ([2] I.2.2.8.2m)
- Ranked order voting ([2] I.2.2.8.2n)
- Election verification ([5] I.C.1.2.2)

The following subclasses are tentatively deleted until it becomes more clear that they are needed and/or whether they go on the system or the voting device. See PREFACE.

- Remote configuration (incoming data) ([2] I.5)
    - Public network remote configuration
    - Wireless network remote configuration
- Remote data delivery (outgoing data) ([2] I.5)
    - Public network data delivery ([2] I.1.5.4, I.5)
    - Wireless network data delivery

The class *Remote data delivery* identifies all systems in which data are transmitted from individual voting machines to some other machine, regardless of whether or not the target machine is located within the same polling place.

2.6.3.2 Supported voting variations (device-level)

It is necessary to specify voting variations at the device level as well as the system level because a system may support a given voting variation without having that support in every device. For example, a system may support absentee voting by having absentee ballot support in one special tabulator and in the central EMS. However, for the most part, these should agree with the variations claimed at the system level.

Choose all that apply.

- In-person voting device ([2] I.2.5.2)

- Absentee voting device ([2] I.2.5.2)
- Provisional / challenged ballots device ([2] I.2.5.2, I.2.2.8.2o)
- Review-required ballots device ([2] I.2.5.2)
- Primary elections device
    - Closed primaries device ([2] I.2.2.8.2a)
    - Open primaries device ([2] I.2.2.8.2b)
- Write-ins device ([2] I.2.2.8.2e)
- Ballot rotation device ([2] I.2.2.8.2g)
- Straight party voting device ([2] I.2.2.8.2h)
    - Cross-party endorsement device ([2] I.2.2.8.2i)
- Split precincts device ([2] I.2.2.8.2j)
- N of M voting device ([2] I.2.2.8.2k)
- Cumulative voting device ([2] I.2.2.8.2m)
- Ranked order voting device ([2] I.2.2.8.2n)
- Election verification device ([5] I.C.1.2.2)

2.6.3.3 Voting device classes

The classes enumerated in this section identify different types of voting devices. Choose all that apply.

- Vote-capture device
- Paper-based device ([2] I.1.5.2)
- Electronic device
    - Programmed device
- Tabulator
    - Precinct tabulator ([2] I.1.5.5)
    - Central tabulator ([2] I.1.5.6)

- VVPAT (voter-verified paper audit trail) ([5] I.7.9)
- ALVS (alternative language voting station) ([4] I.2.2.7.2)
- Acc-VS (accessible voting station) ([5] I.3.2)

- Card puncher ([2] I.1.5.2, "punchcard")
- MMPB (Manually-Marked Paper Ballot)
- EBM (Electronically-assisted Ballot Marker)
    - EBP (Electronic Ballot Printer)
- DRE (Direct Record Electronic) ([2] I.1.5.3)
- Punchcard reader ([2] I.1.5.2, "punchcard")
- Optical scanner ([2] I.1.5.2, "marksense")
- EMS (Election Management System) ([2] I.2.2.6)

PCOS is implied if *Precinct tabulator* and *Optical scanner* are identified.

2.6.4 Semantics of classes

A class simultaneously identifies a set of requirements and a set of voting systems or devices to which those requirements apply.

For a class C, let S(C) represent the set of voting systems or devices identified by C and let R(C) represent the set of requirements applicable to those voting systems or devices.

A subclass identifies a superset of the requirements and a subset of the voting systems or devices identified by its superclass. A voting system that conforms to a subclass necessarily conforms to its superclass. The superclass is said to *subsume* the subclass.

If class C_1 subsumes C_2, then

$$R(C_2) \supseteq R(C_1)$$

$$S(C_2) \subseteq S(C_1)$$

A class may have multiple superclasses. Let P(C) represent the set of superclasses of C. Then

$$R(C) \supseteq \bigcup_{x \in P(C)} R(x)$$

$$S(C) \subseteq \bigcap_{x \in P(C)} S(x)$$

Given classes C_3 and C_4, one may derive a new subclass by combining C_3 and C_4. By default, this new class identifies the union of the requirements and the intersection of the voting systems or devices identified by C_3 and C_4. However, additional requirements that applied to neither superclass may apply specifically to the new subclass. The combining operation on classes is represented with a wedge (^).

$$R(C_3 \wedge C_4) \supseteq R(C_3) \cup R(C_4)$$

$$S(C_3 \wedge C_4) = S(C_3) \cap S(C_4)$$

A class that is derived by combining classes that are disjoint is said to be *incoherent* and identifies no voting systems or devices. The set of requirements identified by an incoherent class is likely to be self-contradictory.

2.7 Extensions

Extensions are additional functions, features, and/or capabilities included in a voting system that are not defined in the Guidelines. To accommodate the needs of states that may impose additional requirements and to accommodate changes in technology, these Guidelines allow extensions. However, as extensions are essentially subclasses of one or more classes defined in these Guidelines, they are subject to the integrity constraint that applies to all subclasses: an extension may not contradict nor relax requirements that would otherwise apply to the system and its constituent devices.

Chapter 3 General Requirements

3.1 General design requirements

Note: the ballot counter requirements from [2] have been converted into functional requirements (Requirement III.4.11-1.1 and Requirement III.4.11-1.2).

R 3.1-1 No cheating

Voting systems shall contain no logic or functionality for the purpose of producing fraudulent election results.

Source: New requirement.

Applies to: Voting system

Test reference: Volume V Section 3.7

R 3.1-2 GeneratedID482

All programmed devices shall include control logic and data processing methods incorporating error detection and correction methods.

Source: [2] I.2.2.2.1.d as modified by [3] 5.2.1.1.d.[7]

Applies to: Programmed device

R 3.1-3 GeneratedID486

All programmed devices shall provide software that monitors the overall quality of data read-write and transfer quality status, checking the number and types of errors that occur in any of the relevant operations on data and how they were corrected.

Source: [2] I.2.2.2.1.e.

Applies to: Programmed device

R 3.1-4 GeneratedID490

All voting devices shall:

a. Display a permanently affixed nameplate or label containing the name of the manufacturer or vendor, the name of the device, its part or model number, its revision identifier, its serial number, and if applicable, its power requirements;

b. Display a separate data plate containing a schedule for and list of operations required to service or to perform preventive maintenance; and

c. Display advisory caution and warning instructions to ensure safe operation of the equipment and to avoid exposure to hazardous electrical voltages and moving parts at all locations where operation or exposure may occur.

Source: [2] I.3.4.6.

Applies to: Voting device

Issue #1081 what if maintenance schedule won't fit?

AG has this in Maintainability. Resolve overlap.

R 3.1-5 Privacy enclosures

All vote-capture devices supporting in-person voting shall include a voting booth or enclosure for poll site use. Such booths or enclosures may be integral to the vote-capture devices or supplied as separate components, and shall:

a. Be integral to, or make provision for the installation of, the voting device;

b. Ensure by their structure stability against movement or overturning during entry, occupancy, and exit by the voter; and

c. Provide privacy for the voter, and be designed in such a way as to prevent observation of the ballot by any person other than the voter.

Source: [2] I.2.4.1.2.1.b and I.3.2.4.1.

Applies to: In-person voting device ^ Vote-capture device

HFP: This design requirement appears to be redundant with the performance requirement appearing in [5] I.3.1.7.1.a. OK to delete?

Impact: Deleted redundant requirement to meet accessibility requirements in Section X.

3.1-6 GeneratedID507

Paper ballots used by paper-based voting devices shall meet the following standards:

a. Punches or marks that identify the unique ballot format shall be outside the area in which votes are recorded, so as to minimize the likelihood that these punches or marks will be mistaken for vote responses and the likelihood that recorded votes will obliterate these punches or marks;

b. If printed or punched alignment marks are used to locate the vote response fields on the ballot, these marks shall be outside the area in which votes are recorded, so as to minimize the likelihood that these marks will be mistaken for vote responses and the likelihood that recorded votes will obliterate these marks.

Source: [2] I.3.2.4.2.1.

Applies to: Paper-based device

DISCUSSION

See also Requirement IV.2.5-12.

3.1-7 GeneratedID514

Card punchers shall:

a. Be suitable for the type of ballot card specified;

b. Facilitate the clear and accurate recording of each vote intended by the voter;

c. Be designed to avoid excessive damage to vote recorder components; and

d. Incorporate features to ensure that the chad (debris) is completely removed, without damage to other parts of the ballot card.

Source: [2] I.3.2.4.2.2.

Applies to: Card puncher

3.1-8 GeneratedID522

The frame or fixture for punchcards shall:

a. Hold the ballot card securely in its proper location and orientation for voting;

b. When contests are not printed directly on the ballot card or sheet, incorporate an assembly of ballot label pages that identify the offices and issues corresponding to the proper ballot format for the polling place where it is used and that are aligned with the voting fields assigned to them; and

c. Incorporate a template to preclude perforation of the card except in the specified voting fields; a mask to allow punches only in fields designated by the format of the ballot; and a backing plate for the capture and removal of chad. This requirement may be satisfied by equipment of a different design as long it achieves the same result as the Standards with regard to (1) Positioning the card; (2) Associating ballot label information with corresponding punch fields; (3) Enabling of only those voting fields that correspond to the format of the ballot; and (4) Punching the fields and the positive removal of chad.

Source: [2] I.3.2.4.2.4.

Applies to: Card puncher

3.1-9 GeneratedID529

A frame or fixture for printed ballot cards is optional. However, if such a device is provided, it shall:

a. Position the card properly; and

b. Hold the ballot card securely in its proper location and orientation for voting.

Source: [2] I.3.2.4.2.5.

Applies to: MMPB

Impact: Deleted vacuous requirement to "Be of any size and shape consistent with its intended use" and redundant requirement to comply with design, construction, and maintainability requirements.

3.1-10 Ballot boxes

Ballot boxes and ballot transfer boxes, which serve as secure containers for the storage and transportation of voted ballots, shall:

a. Incorporate locks and/or seals;

b. Provide specific points where ballots are inserted, with all other points on the box constructed in a manner that prevents ballot insertion; and

c. If needed, contain separate compartments for the segregation of ballots that may require special handling or processing.

Source: [2] I.3.2.4.2.6.

Applies to: Paper-based device

Impact: Deleted vacuous requirement to "Be of any size, shape, and weight commensurate with their intended use."

DISCUSSION

Requirement III.3.1-10.c should be understood in the context of Requirement III.4.6-7.16, Requirement III.4.8-4.6, Requirement III.4.8-5 and Requirement III.4.8-7. The differing options in how to handle separable ballots mean that separate compartments might not be required. See also Requirement IV.2.3-1.

Ⓡ 3.1-11 GeneratedID543

Programmed vote-capture devices shall include an audible and/or visible activity indicator providing the status of each voting device. This indicator shall:

a. Indicate whether the device is in polls-opened or polls-closed state; and

b. Indicate whether a voting session is in progress.

Source: Clarified from [2] I.2.5.1.c and I.3.2.4.3.1.

DISCUSSION

Polls-closed could be broken down into pre-voting and post-voting states as in Volume III Section 5.2 or further divided into separate states for not-yet-tested, testing, ready/not ready (broken), and reporting.

Applies to: Vote-capture device ^ Programmed device

3.2 Security and System Integrity

STS: need security requirements for early voting (locking down equipment overnight). Equipment should support a locked or shutdown state that is suitable for the off-hours of early voting. This cannot require closing the polls as that can only happen once. As we have seen, some equipment can simply be turned off during this period with no harmful consequences. See Requirement VI.1.1-10 and Requirement VI.1.1-11. Also specify procedural / physical security as needed.

STS: [2] I.2.2.1, I.2.2.4 (including no single point of failure).

3.2.1 System Integrity Management

STS: I think we need a general requirement abstracting Requirement III.4.6-9, Requirement III.4.10-1 and the relevant parts of [2] I.3.2.3.1. The placement of these requirements on specific activities seems pretty arbitrary.

3.2.1.1 Executable code and data integrity[6,7]

Scope these requirements to non-WUUP software and firmware per resolution of COTS issue (see PREFACE).

R 3.2-1 GeneratedID554

Self-modifying code is prohibited.

Source: [2] I.4.2.2.

Applies to: Programmed device

Impact: The VSS text continues "except under the security provisions outlined in section 6.4.e" but there is no 6.4.e.

R 3.2-2 GeneratedID559

Remotely loaded code is prohibited.

Source: [3] Section 5.6.2.2.

Applies to: Programmed device

Impact: This IEEE-originated tightening of the restrictions in [2] I.4.2.2 makes explicit something that was implied in [2] (many requirements about what must be "resident").

R 3.2-3 GeneratedID564

Dynamically loaded code other than dynamically linked libraries that are a standard part of the platform is prohibited.

Source: [3] Section 5.6.2.2.

Applies to: Programmed device

Impact: This IEEE-originated loosening of the restriction in [2] I.4.2.2 is to avoid outlawing Windows, where there is no alternative to DLLs.

R 3.2-4 InterpreterVersionSpec

If interpreted code is used, it shall only be run under a specific, identified version of an industry standard runtime interpreter.

Source: [3] Section 5.6.2.2.

Applies to: Programmed device

DISCUSSION

This prohibition is to ensure that the software tested and approved during the certification process does not change behavior because of a change to the interpreter.

Impact: This IEEE-originated loosening of the restriction in [2] I.4.2.2 is to allow the use of interpreted Java. [2] mentions Java specifically in I.4.2.1 but then prohibits all interpreted code in I.4.2.2.

R 3.2-5 NoOverwriteCode

During an election, all programmed devices shall prevent replacement or modification of executable code (e.g., by other programs on the system, by people physically replacing the memory or medium containing the code, or by faulty code).

Source: Rewording/expansion of [2] I.4.2.2.

Applies to: Programmed device

DISCUSSION

This requirement may be partially satisfied through a combination of read-only memory (ROM), the memory protection implemented by most popular WUUP operating systems, error checking as described in Volume III Section 3.2.1.2, and access and integrity controls.

3.2-6 NoOverwriteData

All voting devices shall prevent access to or manipulation of vote data or audit records (e.g., by physical tampering with the medium or mechanism containing the data, by other programs on the system, or by faulty code) except where this access is necessary to conduct the voting process.

Source: Rewording/expansion of [2] I.4.2.2.

Applies to: Voting device

DISCUSSION

This requirement may be partially satisfied through a combination of the memory protection implemented by most popular WUUP operating systems, error checking as described in Volume III Section 3.2.1.2, and access and integrity controls. Systems using mechanical counters to store vote data must protect the counters from tampering. If vote data are stored on paper, the paper must be protected from tampering. Modification of audit records after they are created is *never* necessary.

Look for overlaps with STS requirements (e.g., shall not put code on removable modules).

3.2.1.2   Error checking[7,8]

3.2-7 GeneratedID583

All programmed devices shall check information inputs for accuracy, completeness, and validity.

Source: [7] [SI-10].

Applies to: Programmed device

3.2-7.1 GeneratedID587

All programmed devices shall ensure that inaccurate, incomplete, or invalid inputs do not lead to irreversible error.

Source: [2] I.2.2.5.2.2.f.

3.2-8 ErrCheckShalls

All programmed devices that are capable of the following types of errors shall check for these errors at run time and respond defensively when they occur.

a. Arrays or strings with unenforced bounds (includes buffers used to move data);

b. Stack overflow errors;

c. CPU-level exceptions such as address and bus errors, dividing by zero, and the like;

d. Variables that are not appropriately handled when out of expected boundaries;

e. Known programming language specific vulnerabilities.

Source: [3] Section 5.6.2.2 expansion of [2] I.4.2.2, modified.

Applies to: Programmed device

DISCUSSION

It is acceptable, even expected, that logic verification will show that some error checks cannot logically be triggered and some exception handlers cannot logically be invoked. These checks and exception handlers are not redundant—they provide defense-in-depth against faults that escape detection during logic verification.

See also Requirement III.4.6-17.

Impact: Removed the one about case statements, which is not necessarily an error.

TO DO: Determine whether existing coding conventions for high-integrity software can subsume the following.

Ⓡ 3.2-8.1 GeneratedID601

If the software or firmware uses arrays or any analogous data structures and the programming language does not provide automatic run-time range checking of the indices, the indices shall be ranged-checked on every access.

Source: Expansion of [2] I.4.2.2.

DISCUSSION

All accesses should occur via dedicated accessors (functions, methods, operations, subroutines, procedures, etc.) that range-check the indices, or an equivalent mechanism. Range checking code should not be duplicated before each access.

Impact: Expansion was to specify what constitutes an acceptable "control."

3.2-8.2 GeneratedID606

If stack overflow does not automatically result in an exception, the software shall explicitly check for and prevent stack overflow.

Source: Added precision.

DISCUSSION

Embedded system developers use a variety of techniques for avoiding stack overflow. Commonly, the stack is monitored and warnings and exceptions are thrown when thresholds are crossed. In non-embedded contexts, stack overflow often manifests as a CPU-level exception related to memory segmentation, in which case it can be handled pursuant to Requirement III.3.2-8.3 and Requirement III.3.2-18.2.

3.2-8.3 cpucheck

The software shall implement such handlers as are needed to detect and respond to CPU-level exceptions.

Source: Added precision.

DISCUSSION

For example, under Unix a CPU-level exception would manifest as a signal, so a signal handler is needed.

3.2-8.4 GeneratedID613

All scalar or enumerated type parameters whose valid ranges as used in a callable unit (function, method, operation, subroutine, procedure, etc.) do not cover the entire ranges of their declared data types shall be range-checked on entry to the unit.

Source: Elaboration on Requirement III.3.2-8.d, which is an expansion of [2] I.4.2.2.

DISCUSSION

This applies to parameters of numeric types, character types, temporal types, and any other types for which the concept of range is well-defined.[9]

3.2-9 ErrCheckShoulds

All programmed devices that are capable of the following types of errors should check for these errors at run time and respond defensively when they occur.

a. Pointer variable errors;

b. Dynamic memory allocation and management errors.

Source: [3] Section 5.6.2.2 expansion of [2] I.4.2.2, modified.

Applies to: Programmed device

TO DO: Determine whether existing coding conventions for high-integrity software can subsume the following.

3.2-9.1 GeneratedID623

For languages having pointers or otherwise providing for specifying absolute memory locations, the system should validate pointers or addresses before they are used.

Source: Slight revision of [3] 6.6.4.2.e.

DISCUSSION

Improper overwriting should be prevented in general as required by Requirement III.3.2-5 and Requirement III.3.2-6. Nevertheless, even if read-only memory would prevent the overwrite from succeeding, an attempted overwrite indicates a logic fault that must be corrected. Software design should ensure that the validity of the pointer is determinable, and determined, to a greater extent than merely checking that it is not null. Pointer usage that is fully encapsulated within a standard platform library is treated as WUUP software.

Impact: This is "should" not "shall" only because it is very difficult in the general case to validate a pointer. It is easier to design the system in such a way that pointers are not required. (Some would go so far as to make *that* a requirement.)

3.2-9.2 GeneratedID628

If dynamic memory allocation is performed, the software should be instrumented and/or routinely analyzed with an industry standard tool for detecting memory management errors.

Source: Added precision.

Impact: This is "should" not "shall" only because such tooling may not be available or applicable in all cases.

3.2-10 GeneratedID632

The detection of any of the errors enumerated in Requirement III.3.2-8 and Requirement III.3.2-9 shall be treated as a complete failure of the callable unit in which the error was detected. An appropriate exception shall be thrown and control shall pass out of the unit forthwith.

Applies to: Programmed device

Impact: This closes the loophole where a vendor might include the mandatory checks but then ignore the results.

3.2-11 NoDisableAsserts

Error checks detailed in Requirement III.3.2-8 and Requirement III.3.2-9 shall remain active in certified production code.

Applies to: Programmed device

DISCUSSION

These errors are incompatible with voting integrity, so masking them is unacceptable. Vendors should not implement error checks using the C++ assert() macro, which is often disabled, sometimes automatically, when software is compiled in production mode.

"Inevitably, the programmed validity checks of the defensive programming approach will result in run-time overheads and, where performance demands are critical, many checks are often removed from the operational software; their use is restricted to the testing phase where they can identify the misuse of components by faulty designs. In the context of producing complex systems which can never be fully tested, this tendency to remove the protection afforded by programmed validity checks is most regrettable and is not recommended here." [21]

Impact: This closes the loophole where a vendor might code in such a way that checks get disabled when the software is compiled or run in production mode (as opposed to debugging mode). [2] II.5.4.2.w (optional coding conventions) *required* asserts to be disabled in the production code. That was just wrong.

3.2-12 GeneratedID641

Exceptions resulting from failed error checks or CPU-level exceptions shall require intervention by an election judge or administrator.

Applies to: Programmed device

DISCUSSION

These errors are incompatible with voting integrity, so masking them is unacceptable.

Impact: This closes the loophole where a vendor might throw the required exceptions but then mask them in an exception handler.

3.2-13 GeneratedID646

Electronic devices shall include a means of identifying device failure and any corrective action needed.

Source: Generalized from [2] I.2.4.1.2.2.c and I.2.4.1.3.d.

Applies to: Electronic device

3.2-14 GeneratedID650

Electronic devices should proactively detect equipment failures and alert an election judge or administrator when they occur.

Source: Response to Issue #2147.

Applies to: Electronic device

Impact: Afraid to make this a "shall" because continual self-test could be too onerous for some kinds of equipment.

3.2-15 GeneratedID655

Electronic devices shall proactively detect or prevent basic violations of election integrity (e.g., stuffing of the ballot box or the accumulation of negative votes) and alert an election judge or administrator if they occur.

Source: Response to Issue #2147.

Applies to: Electronic device

DISCUSSION

Equipment can only verify those conditions that are within the scope of what the equipment does. However, insofar as the equipment can detect something that is blatantly wrong, it should do so and raise the alarm. This provides defense-in-depth to supplement procedural controls and auditing practices.

3.2.1.3 Exception handling and recovery

For specific requirements regarding misfed paper ballots or hangs during the vote-casting function, see Requirement III.4.6-13.3, Requirement III.4.8-12 and Requirement III.4.8-13.

Ⓡ 3.2-16 GeneratedID661

All systems shall be capable of resuming normal operation following the correction of a failure in any device.

Source: Extrapolated from [2] I.2.2.3.

Applies to: Voting system

Ⓡ 3.2-16.1 GeneratedID665

Exceptions and system recovery shall be handled in a manner that protects the integrity of all recorded votes and audit log information.

Source: Extracted and generalized from [2] I.4.2.3.e.

Ⓡ 3.2-17 GeneratedID668

All voting devices shall be capable of resuming normal operation following the correction of a failure in any component (e.g., memory, CPU, ballot reader, printer) provided that catastrophic electrical or mechanical damage has not occurred.

Source: Reworded from [2] I.2.2.3.b and c.

Applies to: Voting device

Ⓡ 3.2-18 GeneratedID672

Error conditions shall be corrected in a controlled fashion so that system status may be restored to the initial state existing before the error occurred.

Source: Generalization from [2] I.2.2.5.2.2.g.

Applies to: Programmed device

DISCUSSION

"Initial state" refers to the state existing at the start of a logical transaction or operation. Transaction boundaries must be defined in a conscientious fashion to minimize the damage. Language changed to "may" because election judges responding to the error condition might want the opportunity to select a different state (e.g., controlled shutdown with memory dump for later analysis).

Impact: This generalization from [2] I.2.2.5.2.2.g (from the nested case to the non-nested case) clarifies the reason we need exception handling.

3.2-18.1 GeneratedID677

Nested error conditions shall be corrected in a controlled sequence so that system status may be restored to the initial state existing before the first error occurred.

Source: Slight relaxation of [2] I.2.2.5.2.2.g.

Impact: Relaxation was the "shall" to "may" change mentioned in Requirement III.3.2-18 discussion.

3.2-18.2 cpux

CPU-level exceptions shall be handled in a manner that restores the CPU to a normal state and allows the system to log the event and recover as with a software-level exception.

Source: Added precision.

DISCUSSION

System developers should test to see how CPU-level exceptions are handled and make any changes necessary to ensure robust recovery. Invocation of any other error routine while the CPU is in an exception handling state is to be avoided—software error handlers often do not operate as intended when the CPU is in an exception handling state.

3.2-19 GeneratedID684

When recovering from non-catastrophic failure of a device or from any error or malfunction that is within the operator's ability to correct, the system shall restore the device to the operating condition existing immediately prior to the error or failure, without loss or corruption of voting data previously stored in the device.

Source: [2] I.2.2.3.a.

Applies to: Programmed device

DISCUSSION

If, as discussed in Requirement III.3.2-18, the system is left in something other than the last known good state for diagnostic reasons, this requirement clarifies that it must revert to the last known good state before being placed back into service.

3.2.2 System auditing and event logging

This section is to be provided by STS. The text here is only notes. See PREFACE.

3.2.2.1 Entry content requirement

See also [2] I.2.2.5.2.1

3.2.3 Hardware security

This section is to be provided by STS. The text here is only notes.

3.2.3.1 Memory protection requirements

Overlap with Volume III Section 3.2.1.1 and Volume III Section 3.2.1.2.

3.3 Accessibility, usability, and privacy, general requirements

This section is to be provided by HFP. The text here is only notes. See also Requirement III.3.1-5.

[2] I.2.2.5.2.2:

All voting systems shall meet the following requirements for error messages:

a. The system shall generate, store, and report to the user all error messages as they occur;

b. All error messages requiring intervention by an operator or precinct official shall be displayed or printed unambiguously in easily understood language text, or by means of other suitable visual indicators;

c. When the system uses numerical error codes for trained technician maintenance or repair, the text corresponding to the code shall be self-contained, or affixed inside the unit device. This is intended to reduce inappropriate reactions to error conditions, and to allow for ready and effective problem correction; Why should even trained technicians be expected to deal with "guru meditation numbers?" Just ban them.

d. All error messages for which correction impacts vote recording or vote processing shall be written in a manner that is understandable to an election official who possesses training on system use and operation, but does not possess technical training on system servicing and repair;

e. The message cue for all systems shall clearly state the action to be performed in the event that voter or operator response is required;

[... f and g were already handled in other sections.]

[7] [SI-11] Control: The information system identifies and handles error conditions in an expeditious manner.

Supplemental Guidance: The structure and content of error messages should be carefully considered by the organization. User error messages generated by the information system should provide timely and useful information to users without revealing information that could be exploited by adversaries. System error messages should be revealed only to authorized personnel (e.g., systems administrators, maintenance personnel). Sensitive information (e.g., account numbers, social security numbers, and credit card numbers) should not be listed in error logs or associated administrative messages. The extent to which the information system is able to identify and handle error conditions should be guided by organizational policy and operational requirements.

3.4 H/W and S/W performance, general requirements

Text for this section is provided separately (AG deliverable). Requirements appearing here are only to provide targets for cross-referencing.

Issue with humidity causing opscan ballots to expand or curl and jam the machine: should have been prevented by environmental requirements and testing. Review these requirements in light of the reported failures in Fairfield County, Ohio, 2004-12-15 (AG).

3.4.1 Reliability (MTBF)

The MTBF issue. Currently, [5] II.C.4 only establishes 90 % confidence that MTBF exceeds 45 hours. Many people have pointed out that this results in a high likelihood of failure during a typical election day, even higher with early voting. Issue #2681, low MTBF requirement creates opportunities for fraud: jurisdictions with blended systems can deploy the unreliable equipment to precincts with unwanted votes.

3.4.2 Accuracy/Error Rates

🅡 3.4-1 10general

All systems shall achieve an error rate of no more than one in 10,000,000 ballot positions.

Source: Extrapolated from [2] I.3.2.1.

Applies to: Voting system

DISCUSSION

This general requirement is elaborated by Requirement III.4.6-10, Requirement III.4.6-16, Requirement III.4.8-15, Requirement III.4.8-17, and Requirement III.4.10-2.3.

Test reference: Volume V Section 4.2.2.2

3.5 Workmanship

3.5.1 Engineering practices / Coding

This section describes essential design and performance characteristics of the software used in voting systems, addressing both system-level software, such as operating systems, and voting system application software, including firmware. The requirements of this section are intended to ensure that voting system software is reliable, robust, testable, and maintainable.

The general requirements of this section apply to software used to support the entire range of voting system activities. Although this section emphasizes software, the standards described also influence hardware design considerations.

While there is no best way to design software, the use of outdated and *ad hoc* practices is a risk factor for unreliability, unmaintainability, etc. Consequently, these guidelines require the use of modern programming practices. The use of widely recognized and proven software design methods will facilitate the analysis and testing of voting system software.

3.5.1.1 Scope

Scope these requirements to non-WUUP software and firmware per resolution of COTS issue (see PREFACE).

The requirements of this section apply to all software used in any manner to support any voting-related activities, regardless of the ownership of the software or the ownership and location of the hardware on which the software is installed or operates. These requirements apply to:

- Software that operates on voting devices and vote counting devices installed at polling places under the control of the voting jurisdiction;
- Software that operates on ballot printers, vote counting devices, and other hardware typically installed at central or precinct locations (including contractor facilities); and
- Election management software.

In addition to the requirements of this section, all software used in any manner to support any voting-related activities must meet the requirements for security described in Volume III Section 3.2.

Some voting systems use generic equipment such as personal computers that may be used for other purposes and have resident general purpose software such as operating systems, programming language compilers, database management systems, and web browsers. Such software is governed by these guidelines unless:

- The software provides no support of voting system capabilities;
- The software is removable, disconnectable, or switchable such that it cannot function while voting system functions are enabled; and
- Procedures are provided that confirm that the software has been removed, disconnected, or switched.

3.5.1.2 Selection of programming languages

R 3.5-1 GeneratedID735

Software and firmware associated with the logical, numeric, and interactive operations of voting shall be produced in a high-level programming language with support for structured exception handling, such as Java, C++, C#, Visual Basic .NET, or Ada.

Source: Rewrite of [2] I.4.2.1.

Applies to: Programmed device

DISCUSSION

"Structured exception handling" means try/throw/catch or equivalent statements. The requirement for the use of high-level language for logical, numeric, and interactive operations does not preclude the use of assembly language for hardware-related segments, such as device controllers and handler programs. Also, operating system software may be designed in assembly language.

Impact: See discussion in Volume I Section 1.3 regarding structured exceptions. Added "interactive operations" to the list because obfuscating the user interface is as dangerous as obfuscating the tally code. Added firmware to close loophole in hardware/software testing dichotomy.

3.5.1.3 Selection of general coding conventions

3.5-2 publishedcredible

Software and firmware associated with the logical, numeric, and interactive operations of voting shall consistently adhere to a published, credible set of coding rules, conventions or standards (herein simply called "coding conventions") intended to enhance the workmanship, security, integrity, testability, and maintainability of applications.

Source: Rewrite of [2] I.4.2.6.

Applies to: Programmed device

DISCUSSION

Coding conventions that are excessively specialized will not meet the criteria for intent.

Impact: Added "interactive operations" to the list because obfuscating the user interface is as dangerous as obfuscating the tally code. Added firmware to close loophole in hardware/software testing dichotomy.

Test reference: Volume V Section 3.6

3.5-2.1 GeneratedID748

Coding conventions shall be considered published if they appear in a publicly available book, magazine, journal, or new media with analogous circulation and availability, or if they are publicly available on the Internet.

Source: Clarification of [2] I.4.2.6.

DISCUSSION

Following are examples of coding conventions that are freely available on the Internet as of 2005-02-17. These are only examples and are not necessarily the best available for the purpose.

- Java: "Code Conventions for the Java$^{\text{TM}}$ Programming Language," Sun Microsystems. http://java.sun.com/docs/codeconv/.
- C++: "Programming in C++, Rules and Recommendations," Mats Henricson and Erik Nyquist. http://www.doc.ic.ac.uk/lab/cplus/c++.rules/. (A revised and expanded version was published in Industrial Strength C++, Prentice-Hall, 1996.)
- C#: "Design Guidelines for Class Library Developers," Microsoft. http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconnetframeworkdesignguidelines.asp.

Update refs before final version

Impact: Clarification of "published, reviewed, and industry-accepted" language from [2] I.4.2.6.

R 3.5-2.2 GeneratedID758

Coding conventions shall be considered credible if at least two different organizations with no ties to the creator of the rules or to the vendor seeking certification, and which are not themselves voting equipment vendors, independently decided to adopt them and made active use of them at some time within the three years before certification was first sought.

Source: Clarification of [2] I.4.2.6.

DISCUSSION

If the "three year rule" was satisfied at the time that a system was first submitted for certification, it is considered satisfied for the purpose of subsequent recertifications of that system. However, new systems must meet the three year rule as of the time that they are first submitted for certification, even if they reuse parts of older systems.

Impact: Clarification of "published, reviewed, and industry-accepted" language from [2] I.4.2.6.

200509 Boulder mtg, WQ posited that some authority would maintain a list of acceptable coding conventions for vendors to follow. While it is not clear who, exactly, would do this, it would solve the problem of defining what constitutes a credible coding convention in the standard.

3.5.1.4 Software modularity and programming

3.5-3 GeneratedID765

Voting system application software shall be designed in a modular fashion.

Source: Extracted and revised from [2] I.4.2.3.

Applies to: Programmed device

DISCUSSION

See module. The modularity rules described here apply to the component submodules of a library.

Impact: Removed unclear, untested requirement on COTS. This should not conflict with any coding conventions. In general, I have attempted to distinguish the class-level concept (module) from the method-level concept (for which I use the new term "callable unit"). [2] did not distinguish these adequately.

3.5-3.1 GeneratedID771

Each module shall have a specific function that can be tested and verified independently of the remainder of the code.

Source: Extracted and revised from [2] I.4.2.3.a.

DISCUSSION

In practice, some additional modules (such as library modules) may be needed to compile the module under test, but the modular construction allows the supporting modules to be replaced by special test versions that support test objectives.

3.5-4 Module size and grouping

Modules shall be small, easily identifiable, and constructed to be grouped according to functionality.

Source: Revision of [2] II.5.4.2.i, as revised by Section 6.6.4.2, Paragraph i of [3].[7]

3.5-4.1 Callable unit size limit

No more than 50 % of all callable units (functions, methods, operations, subroutines, procedures, etc.) should exceed 25 lines of code in length (excluding comments and blank lines), no more than 5 % of all callable units should exceed 60 lines in length, and no callable units should exceed 180 lines in length.

Source: Revision of [2] II.5.4.2.i, as revised by Section 6.6.4.2, Paragraph i of [3].[7]

DISCUSSION

"Lines," in this context, are defined as executable statements or flow control statements with suitable formatting.

Impact: Clarified and updated with "module" replaced by "callable unit" and new exclusion for blank lines. We need length limits to keep the logic verification tractable. The original limits, defined in terms of entire modules rather than callable units, missed the mark.

3.5-4.2 GeneratedID781

Initializations of read-only lookup tables shall be exempt from length limitations.

Source: Relaxation of [2] requirement.

Impact: Resolve unintended consequence of length limits.

3.5-4.3 GeneratedID785

Read-only lookup tables longer than 25 lines should be placed in separate files from other source code if the programming language permits it.

3.5.1.5 Control constructs

3.5-5 GeneratedID788

In software and firmware modules associated with the logical, numeric, and interactive operations of voting, unstructured programming is prohibited.

Source: Generalization and summary of [2] I.4.2.4 and II.5.4.1.

Applies to: Programmed device

3.5-5.1 GeneratedID792

Neither GoTo nor any equivalent construct shall be used.

Source: Generalization and summary of [2] I.4.2.4 and II.5.4.1.

DISCUSSION

"Equivalent constructs" include intentional exceptions, early exits from loops or other program blocks that serve no other purpose, and similar linguistic evasions. Early exits that conform to the chosen coding conventions are otherwise permitted.

Impact: Relaxed [2] stance on early exits. Although a ban on early exits was widely accepted at one time, such a ban is now controversial at best.

3.5-5.2 GeneratedID797

Unstructured exception handling (e.g., On Error GoTo, setjmp/longjmp, or explicit tests for exception conditions after every executable statement) is prohibited.

Source: Extension of [2] requirements for structured programming.

Impact: If structured exceptions were in wide use in 1990, this requirement probably would have been in the VSS already.

3.5-5.3 GeneratedID801

Exception handling shall be implemented using the formal exception handlers provided by the language.

Source: Extension of [2] requirements for structured programming, also I.4.2.3.e.

DISCUSSION

It is possible to implement exception handling without use of formal exception handlers, just as it is possible to construct robust programs entirely in assembly language or using only GoTos for control flow. But these less structured techniques obfuscate the code and make logic verification more difficult. "One of the major difficulties of conventional defensive programming is that the fault tolerance actions are inseparably bound in with the normal processing which the design is to provide. This can significantly increase design complexity and, consequently, can compromise the reliability and maintainability of the software." [21]

Impact: Close loophole of writing Fortran in C++.

3.5.1.6 Comments

3.5-6 GeneratedID807

All software and firmware modules associated with the logical, numeric, and interactive operations of voting should include header comments that provide at least the following information for each callable unit (function, method, operation, subroutine, procedure, etc.):

a. The purpose of the unit and how it works (if not obvious);

b. A description of input parameters, outputs and return values, exceptions thrown, and side-effects;

c. Any protocols that must be observed (e.g., unit calling sequences);

d. File references by name and method of access (read, write, modify, append, etc.);

e. Global variables used (if applicable);

f. Audit event generation;

g. Date of creation; and

h. Change log (revision record).

Source: Revised from [2] I.4.2.7.a.

Applies to: Programmed device

DISCUSSION

Header comments and other commenting conventions should be specified by the selected coding conventions in a manner consistent with the idiom of the programming language chosen. If the coding conventions specify a coding style and commenting convention that make header comments redundant, then they may be omitted. Otherwise, in the event that the coding conventions fail to specify the content of header comments, this generic guideline should be applied. Change logs need not cover the nascent period, but they must go back as far as the first baseline or release that is submitted for certification, and should go back as far as the first baseline or release that is deemed reasonably coherent.

Impact: Added exceptions and audit events, revised language, other nits. The discussion on change logs responds to a known controversy regarding how far back change logs must go.

3.5.1.7 Quality assurance

3.5.1.8 Configuration management

Text for this section is not yet available (AG task). The following text from earlier documents is retained only in case portions of it remain applicable.

Quality assurance and configuration management

Volume 1, Sections 7 and 8 and Volume 2, Section 7 of [2] require the vendor to follow certain quality assurance and configuration management practices and require the ITA to conduct several audits and documentation reviews to ensure that they were followed. The quality assurance and configuration management requirements in the VSS are a means to the end of ensuring that the vendor has followed responsible engineering practices in general, and are not necessarily the best or most up-to-date guidelines for that purpose.

In Resolution #30-05, the TGDC requested that NIST review and analyze quality assurance and configuration management standards and recommend changes to the VVSG based on that analysis.

Since the Voting Systems Standards were first issued, it has become possible for vendors to be certified under ISO 9000 and/or appraised under CMMI [18]. It is not clear whether a separate standard for voting system vendors, in lieu of requiring ISO 9000 certification to a scope of operations appropriate to the purpose of developing voting systems, is any longer necessary or desirable. However, at its January 2005 meeting, the TGDC expressed fear over the expense and administrative burden involved in ISO 9000 compliance. NIST has not yet completed the review and analysis of related standards to determine whether a less expensive alternative exists or whether the existing standards could be retained without sacrificing quality.

3.6 Archival requirements

3.6.1 Archivalness of media

Ⓡ 3.6-1 22monthsReq

All systems shall maintain the integrity of election management, voting and audit data, including cast vote records, during an election and for a period of at least 22 months afterward.

Source: Merged from [2] I.2.2.11 and I.3.2.3.2.

Applies to: Voting system

DISCUSSION

See also Requirement VI.1.1-2, Requirement IV.2.3-6 and Volume III Section 3.6.2.

3.6.2 Period of retention (informative)

This informative subsection provides extended discussion for Requirement III.3.6-1 and Requirement VI.1.1-2.

United States Code Title 42, Sections 1974 through 1974e, states that election administrators must preserve for 22 months "all records and paper that came into (their) possession relating to an application, registration, payment of poll tax, or other act requisite to voting." This retention requirement applies to systems that will be used at any time for voting of candidates for federal offices (e.g., Member of Congress, United States Senator, and/or Presidential Elector). Therefore, all systems must provide for maintaining the integrity of voting and audit data during an election and for a period of at least 22 months thereafter.

Because the purpose of this law is to assist the federal government in discharging its law enforcement responsibilities in connection with civil rights and elections crimes, its scope must be interpreted in keeping with that objective. The appropriate state or local authority must preserve all records that may be relevant to the detection and prosecution of federal civil rights or election crimes for the 22-month federal retention period, if the records were generated in connection with an election that was held in whole or in part to select federal candidates. It is important to note that Section 1974 does not require that election officials generate any specific type or classification of election record. However, if a record is generated, Section 1974 comes into force and the appropriate authority must retain the records for 22 months.

For 22-month document retention, the general rule is that all printed copy records produced by the election database and ballot processing systems must be so labeled and archived. Regardless of system type, all audit trail information spelled out in Volume III Section 3.2.2 must be retained in its original format, whether that be real-time logs generated by the system, or manual logs maintained by election personnel. The election audit trail includes not only in-process logs of election night (and subsequent processing of absentee or provisional ballots), but also time logs of baseline ballot definition formats, and system readiness and testing results.

In many voting systems, the source of election-specific data (and ballot formats) is a database or file. In precinct count systems, this data is used to program each machine, establish ballot layout, and generate tallying files. It is not necessary to retain this information on electronic media if there is an official, authenticatable printed copy of all final database information. However, it is recommended that the state or local jurisdiction also retain electronic records of the aggregate data for each device so that reconstruction of an election is possible without data re-entry. The same requirement and recommendation applies to vote results generated by each precinct device or system.

3.7 Interoperability

STS: need to harmonize with access control requirements, also the new text from RP.

Overlap with [5] I.C.3.2 (data formats for token objects).

Ⓡ 3.7-1 InteropReq

All systems shall maximize interoperability and integratability with other systems and/or devices of other systems.

Source: Generalized from database design requirements in [2] I.2.2.6, TGDC Resolution #23-05, and some state RFP(s).

Applies to: Voting system

Ⓡ 3.7-1.1 GeneratedID841

All Election Management Systems shall maximize interoperability and integratability with respect to election programming data and report data (the content of vote data reports, audit reports, etc.).

Source: Generalized from database design requirements in [2] I.2.2.6, TGDC Resolution #23-05, and some state RFP(s).

Applies to: EMS

Ⓡ 3.7-1.2 GeneratedID845

All DREs shall maximize interoperability and integratability with respect to ballot image data.

Source: Generalized from database design requirements in [2] I.2.2.6, TGDC Resolution #23-05, and some state RFP(s).

Applies to: DRE

Ⓡ 3.7-1.3 InteropDesignExport

The interoperability and integratability requirement may be met by providing the capability to export data in a non-proprietary, open standard format.

Source: Drill-down from TGDC Resolution #23-05.

Ⓡ 3.7-1.4 InteropDesignDB

The interoperability and integratability requirement may be met by storing data in a documented schema in a standards-conforming database in such a manner that other applications can read and interpret the data.

Source: Drill-down from [2] I.2.2.6.

DISCUSSION

"Standards-conforming" refers to support for a standard query language and standard API.

Chapter 4 Requirements by voting activity

4.1 Election programming

STS: Audit record stuff from [2] I.4.4.1 (pre-election audit records) deferred until STS and CRT can synchronize on audit records. See PREFACE.

Election programming is the process by which central election officials use election databases and vendor system software to logically define the voter choices associated with the contents of the ballots.

There are significant variations among the election laws of the 50 states with respect to permissible ballot contents, voting options, and the associated ballot counting logic.

Ⓡ 4.1-1 GeneratedID857

All systems shall include an EMS.

Source: Artifact of system/device clarification.

Applies to: Voting system

Ⓡ 4.1-2 GeneratedID861

The EMS shall provide for the logical definition of the ballot, including the definition of the number of allowable choices for each office and contest.

Source: [2] I.2.3.2.a.

Applies to: EMS

Ⓡ 4.1-2.1 GeneratedID865

The EMS shall be capable of collecting and maintaining

a. Offices and their associated labels and instructions;

b. Candidate names and their associated labels; and

c. Issues or measures and their associated text.

Source: [2] I.2.3.1.1.1.b.

Ⓡ 4.1-3 GeneratedID871

The EMS shall provide for the logical definition of political and administrative subdivisions, where the list of candidates or contests varies between precincts.

Source: [2] I.2.2.6.a and I.2.3.2.b.

Applies to: EMS

Ⓡ 4.1-4 GeneratedID875

The EMS shall enable central election officials to define multiple election districts.

Source: [2] I.2.2.6.a.

Applies to: EMS

Test reference: Test 4

Impact: Database references handled by Volume III Section 3.7.

Ⓡ 4.1-5 GeneratedID881

The EMS shall enable central election officials to define and identify contests, candidates, and issues using all voting variations indicated in the implementation statement.

Source: [2] I.2.2.6.b, I.2.2.8.2, I.2.3.2.d.

Applies to: EMS

Test reference: Need an "issues" test

Impact: Database references handled by Volume III Section 3.7.

4.1-5.1 GeneratedID887

In all systems, the Election Management System shall allow the definition of contests where the voter is allowed to choose at most one candidate from a list of candidates.

Source: Implicit in [2].

Test reference: Test 2, Test 3, Test 19, Test 22

4.1-5.2 GeneratedID891

In all systems, the Election Management System shall allow the definition of political parties and the indication of the political parties (if any) that endorsed each candidate.

Source: Implicit in [2].

Test reference: Test 2, Test 3, Test 19, Test 22

4.1-5.3 GeneratedID895

EMSs of the *Primary elections device* class shall support the definition of both partisan and nonpartisan contests.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: *EMS* ^ *Primary elections device*

Test reference: Test 7, Test 8

4.1-5.4 GeneratedID900

EMSs of the *Write-ins device* class shall support the definition of contests that include ballot positions for write-in opportunities.

Source: [2] I.2.4.3.1.d.

Applies to: *EMS* ^ *Write-ins device*

Test reference: Test 9, Test 15, Test 28, Test 29, Test 32, Test 33

Impact: Removed untestable reference to state law.

Ⓡ 4.1-5.5 GeneratedID906

EMSs of the *Straight party voting device* class shall be capable of defining the necessary straight party contest and associated metadata to support the gathering and recording of votes for the slate of candidates endorsed by a given political party.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: EMS ^ Straight party voting device

Test reference: Test 11, Test 30

Ⓡ 4.1-5.6 GeneratedID911

EMSs of the *Cross-party endorsement device* class shall be capable of defining the necessary straight party contest and associated metadata to support the gathering and recording of votes for the slate of candidates endorsed by a given political party when a given candidate is endorsed by two or more different political parties.

Source: Clarification or extension of existing requirements.

Applies to: EMS ^ Cross-party endorsement device

Test reference: Test 12

Ⓡ 4.1-5.7 GeneratedID916

EMSs of the *Split precincts device* class shall support the definition of election districts and precincts in such a way that a given polling place may serve two or more election districts.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: EMS ^ Split precincts device

Test reference: Test 13

Ⓡ 4.1-5.8 GeneratedID921

EMSs of the *N of M voting device* class shall be capable of defining contests where the voter is allowed to choose up to a specified number of candidates (N(*r*) > 1, per Volume III Section 5.3) from a list of candidates.

Source: Added precision, based on [2] I.2.2.8.2, I.2.3.2.a and glossary.

Applies to: EMS ^ N of M voting device

Test reference: Test 14, Test 15, Test 21, Test 31, Test 32, Test 33

4.1-5.9 GeneratedID926

EMSs of the *Cumulative voting device* class shall be capable of defining contests where the voter is allowed to allocate up to a specified number of votes (N(*r*) > 1, per Volume III Section 5.3) over a list of candidates however he or she chooses, possibly giving more than one vote to a given candidate.

Source: Added precision, based on [2] I.2.2.8.2, I.2.3.2.a and glossary.

Applies to: EMS ^ Cumulative voting device

Test reference: Test 16, Test 34

4.1-5.10 GeneratedID931

EMSs of the *Ranked order voting device* class shall be capable of defining contests where the voter is allowed to rank candidates in a contest in order of preference, as first choice, second choice, etc.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: EMS ^ Ranked order voting device

Test reference: Test 17

4.1-6 GeneratedID936

The EMS shall record the election contests, candidates, issues, and political and administrative subdivisions exactly as defined by central election officials.

Source: [2] I.2.2.2.1.a / [5] I.2.1.2.a.

Applies to: EMS

DISCUSSION

This is an accuracy requirement.

Impact: Added "political and administrative subdivisions."

4.1-7 GeneratedID942

The EMS shall record the options for casting and recording votes exactly as defined by central election officials.

Source: Reworded from [2] I.2.2.2.1.b / [5] I.2.1.2.b.

Applies to: EMS

DISCUSSION

This is an accuracy requirement.

4.1-8 GeneratedID947

The EMS shall provide for the generation of master and distributed copies of election definitions as needed to configure each voting device in the system.

Source: Reworded from [2] I.2.3.2.e.

Applies to: EMS

4.2 Ballot preparation, formatting, and production

STS: Audit record stuff from [2] I.4.4.1 (pre-election audit records) deferred until STS and CRT can synchronize on audit records. See PREFACE.

4.2-1 GeneratedID952

The EMS shall enable central election officials to define ballot formats and select voting options.

Source: [2] I.2.2.6.c.

Applies to: EMS

Impact: Database references handled by Volume III Section 3.7.

Test reference: Test 23 and all other tests.

4.2-1.1 GeneratedID958

The EMS shall be capable of automatically formatting ballots in accordance with the requirements for offices, candidates, and measures qualified to be placed on the ballot for each political subdivision and election district.

Source: [2] I.2.3.1.1.1.a.

4.2-1.2 EMSplus

The EMS shall provide for the inclusion in a given ballot format of any contest in which the voter would be entitled to vote.

Source: Extrapolated from relevant requirements in [2].

4.2-1.3 EMSminus

The EMS shall provide for the exclusion from a given ballot format of any contest in which the voter would be prohibited from voting because of place of residence or other such administrative or geographical criteria.

Source: [2] I.2.3.2.c.

DISCUSSION

In systems supporting primary elections, this would include the exclusion of partisan contestss that are not votable by the selected political party.

4.2-1.4 GeneratedID966

The EMS shall uniformly allocate space and fonts used for each office, candidate, and contest such that the voter perceives no active voting position to be preferred to any other.

Source: [2] I.2.3.1.2.c.

4.2-1.5 YourAdHere

The EMS shall enable central election officials to add jurisdiction-dependent text, line art, logos and images to ballot formats.

Source: Reworded from [2] I.3.2.3.1.d

4.2-1.6 GeneratedID971

EMSs of the *Primary elections device* class shall support the association of different ballot configurations with different political parties.

Source: Reworded from [2] I.2.3.1.1.1.d.

Applies to: EMS ^ Primary elections device

DISCUSSION

In paper-based systems, open primaries have sometimes been handled by printing a single ballot format that merges the contests from all parties, instructing the voter to vote only in the contests applicable to a single party, and rejecting or discarding votes that violate this instruction. To satisfy the requirements for *Primary elections device*, the EMS must be *capable* of associating different ballot configurations with different political parties.

Test reference: Test 7, Test 8

R 4.2-1.7 GeneratedID977

EMSs of the *Ballot rotation device* class shall support the production of rotated ballots and/or the activation of ballot rotation functions in vote-capture devices through the inclusion of relevant metadata in distributed election definitions and ballot formats.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: EMS ^ Ballot rotation device

Test reference: Test 10

R 4.2-1.8 GeneratedID982

EMSs of the *Split precincts device* class shall support the definition of distinct ballot configurations for voters from two or more election districts that are served by a given polling place.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: EMS ^ Split precincts device

Test reference: Test 13

R 4.2-2 GeneratedID987

The EMS shall provide for the generation of master and distributed copies of ballot formats as needed to configure each voting device in the system.

Source: Reworded from [2] I.2.2.6.d.

Applies to: EMS

Impact: Database references handled by Volume III Section 3.7.

4.2-2.1 Ballot format shall be identifiable

The EMS shall generate codes or marks as needed to uniquely identify the ballot format associated with any ballot.

Source: [2] I.2.3.1.1.1.e.

DISCUSSION

In paper-based systems, identifying marks would appear on the actual ballots. DREs would make internal use of unique identifiers for precincts and ballot formats but would not necessarily present these where the voter would see them.

4.2-2.2 Precinct should be identifiable

The EMS should be capable of generating codes or marks as needed to uniquely identify the precinct associated with any ballot.

Source: New requirement to support combined precincts.

DISCUSSION

Enabling this capability makes it possible to report by precinct even if combined precincts share a common ballot format and tabulator. However, in other cases, including a precinct identifier on the ballot would be a needless complication. Equivalent results are achievable by using different ballot formats or at least different tabulators for different precincts.

4.2-3 GeneratedID998

The EMS shall support retention and reuse of ballot formats from one election to the next.

Source: [2] I.2.3.1.2.e and g.

Applies to: EMS

4.2-4 GeneratedID1002

The EMS shall prevent unauthorized modification of any ballot formats.

Source: [2] I.2.3.1.2.f.

Applies to: EMS

4.3 Equipment preparation

4.3.1 Software installation

Unclear how responsibilities are divided between EMS and vote-capture devices, and between central election officials and election judges. We may have a case where the central election officials put the ballot formats on memory cards but they are "installed" by election judges. (Semantics of "install.")

Ⓡ 4.3-1 GeneratedID1008

The EMS shall enable central election officials to install ballot formats and election-specific programs.

Source: [2] I.2.2.6.e.

Applies to: EMS

Impact: Database references handled by Volume III Section 3.7.

Ⓡ 4.3-2 GeneratedID1013

The EMS shall enable central election officials to test that ballot formats and programs have been properly prepared and installed.

Source: [2] I.2.2.6.f, I.4.4.2.c.

Applies to: EMS

Impact: Database references handled by Volume III Section 3.7.

Ⓡ 4.3-3 GeneratedID1018

Programmed devices shall include a capability to automatically verify that the software and ballot formats have been properly selected and installed in the equipment and immediately notify an election official of any errors.

Source: [2] I.2.3.3.b, I.4.4.2.c.

Applies to: Programmed device

Impact: Deleted "or in a programmable memory devices," possibly redundant, possibly prohibited in the case of executable code (STS).

Ⓡ 4.3-4 GeneratedID1023

Programmed devices shall include a capability to automatically verify that software correctly matches the ballot formats that it is intended to process and immediately notify an election official of any errors.

Source: [2] I.2.3.3.c, I.4.4.2.c.

Applies to: Programmed device

4.4 Equipment setup for security and integrity

STS: Audit record stuff from [2] I.4.4.2 (system readiness audit records) deferred until STS and CRT can synchronize on audit records. See PREFACE.

4.4.1 *In situ* logic and accuracy testing

The purpose of *in situ* logic and accuracy testing is to detect malfunctioning and misconfigured devices before polls are opened. It is not a defense against fraud.[10]

Election personnel conduct equipment and system readiness tests prior to the start of an election to ensure that the voting system functions properly, to confirm that system equipment has been properly integrated, and to obtain equipment status and readiness reports. The content of those reports is defined in Volume III Section 4.9.

Ⓡ 4.4-1 GeneratedID1030

All systems shall provide the capabilities to:

a. Verify that all voting devices are properly prepared for an election and collect data that verifies equipment readiness;

b. Verify the correct installation and interface of all system equipment;

c. Verify that hardware and software function correctly; and

d. Segregate test data from actual voting data, either procedurally or by hardware/software features.

Source: [2] I.2.3.4.1, I.2.3.5.a2 and b2 (the second a and b, respectively), I.4.4.2.a.

Applies to: Voting system

Impact: [2] I.2.3.4.1.b moved to Requirement III.4.9-5. [2] I.2.3.4.1.e doesn't make sense / do not understand in this context (if you consolidate 10 readys and 1 not-ready, you get not-ready, right?). [2] I.2.3.4.1.a2 (the second a) moved to Requirement V.3.6-4. [2] I.2.3.4.1.b2 (the second b) moved to Requirement III.4.4-7.

4.4-2 GeneratedID1039

All programmed devices shall include built-in measurement, self-test, and diagnostic software and hardware for monitoring and reporting the system's status and degree of operability.

Source: [2] I.2.2.4.1.j, I.2.2.8.1.a.

Applies to: Programmed device

4.4-3 GeneratedID1043

Programmed tabulators shall provide the capability for election judges to submit test ballots for use in verifying the integrity of the system.

Source: [2] I.2.4.3.3.s, generalized from DREs; I.4.4.2.d and f.

Applies to: Programmed device ^ Tabulator

4.4-3.1 GeneratedID1047

Programmed tabulators shall isolate test ballots such that they are accounted for accurately in vote counts and are not reflected in official vote counts for specific candidates or measures.

Source: [2] I.2.4.3.3.t, generalized from DREs; I.4.4.2.e.

4.4-4 GeneratedID1050

Paper-based tabulators shall support conversion testing that uses all potential ballot positions as active positions.

Source: [2] I.2.3.4.2.a, I.4.4.2.f.

Applies to: Paper-based device ^ Tabulator

4.4-5 Paper-based tabulators, testing calibration

Paper-based tabulators shall support the use of test ballots to test the calibration of the paper-to-digital conversion (i.e. the calibration of optical sensors, the density threshold, and/or the logical reduction of scanned images to binary values, as applicable).

Source: Interpretation of [2] I.2.3.4.2.b.

Applies to: Paper-based device ^ Tabulator

Impact: Original language: Paper-based tabulators shall support conversion testing of ballots with active position density for systems without pre-designated ballot positions.

4.4-6 GeneratedID1058

Paper-based vote-capture devices shall include a means of verifying that the ballot punching or marking mechanism is properly prepared and ready to use.

Source: [2] I.2.4.1.2.1.a.

Applies to: Vote-capture device ^ Paper-based device

DISCUSSION

In the case of manually marked paper ballots this requirement is mostly moot. (Sharpen the pencils.)

4.4-7 LANoSideEffect

Logic and accuracy testing functions shall introduce no residual side-effects other than audit log entries and status changes to note that the tests have been run with a successful or failed result.

Source: [2] I.2.3.4.1.b2 (the second b), significantly revised.

Applies to: Voting device

DISCUSSION

Status changes required to satisfy Requirement III.4.5-1 and Requirement III.4.5-2.

Impact: As written the original requirement was unsatisfiable.

4.5 Opening polls

Ⓡ 4.5-1 MustTest1

Programmed devices shall provide an internal test or diagnostic capability to verify that all of the tests specified in Volume III Section 4.4 have been successfully completed.

Source: [2] I.2.4.1.1.a.

Applies to: Programmed device

Ⓡ 4.5-2 MustTest2

Programmed devices shall provide for automatic disabling of an untested device until it has been tested.

Source: [2] I.2.4.1.1.b.

Applies to: Programmed device

Ⓡ 4.5-3 GeneratedID1074

Paper-based tabulators shall include a means of activating the ballot counting device.

Source: [2] I.2.4.1.2.2.a.

Applies to: Paper-based device ^ Tabulator

Ⓡ 4.5-4 GeneratedID1078

Paper-based tabulators shall include a means of verifying that the ballot counting device has been correctly activated and is functioning properly.

Source: [2] I.2.4.1.2.2.b.

Applies to: Paper-based device ^ Tabulator

Ⓡ 4.5-5 GeneratedID1082

Programmed vote-capture devices shall provide designated functions for opening the poll.

Source: [2] I.2.4.1.3, generalized.

Applies to: Vote-capture device ^ Programmed device

Ⓡ 4.5-5.1 GeneratedID1086

Programmed vote-capture devices shall include a security seal, a password, or a data code recognition capability to prevent the inadvertent or unauthorized actuation of the poll-opening function.

Source: [2] I.2.4.1.3.a.

Ⓡ 4.5-5.2 GeneratedID1089

Programmed vote-capture devices shall include a means of enforcing the execution of poll-opening steps in the proper sequence if more than one step is required.

Source: [2] I.2.4.1.3.b.

Ⓡ 4.5-5.3 GeneratedID1092

Programmed vote-capture devices shall include a means of verifying that the system has been correctly activated.

Source: [2] I.2.4.1.3.c.

4.6 Casting

These functional capabilities include all operations conducted at the polling place by voters and officials while polls are open, including the generation of status messages.

STS: Audit record stuff from [2] I.4.4.3 (in-process audit records) deferred until STS and CRT can synchronize on audit records. See PREFACE.

4.6.1 Ballot activation

Ⓡ 4.6-1 DRE and EBP, ballot activation

DREs and EBPs shall support ballot activation.

Source: [2] I.2.4.

Applies to: DRE, EBP

Ⓡ 4.6-1.1 DRE and EBP, at most one cast ballot per session

DREs and EBPs shall enable poll workers either to initiate, or to provide the voter with the credentials necessary to initiate, a voting session in which the voter may cast at most one ballot.

Source: [2] I.2.4.2.d, rewritten to respect the limits of what the system can do.

DISCUSSION

See also Requirement VI.1.1-4.

4.6-2 DRE and EBP, control ballot format

DREs and EBPs shall enable poll workers to control the ballot format(s) made available to the voter, whether presented in printed form or electronic display, such that each voter is permitted to record votes only in contests in which that voter is authorized to vote.

Source: [2] I.2.4.2.a.

Applies to: DRE, EBP

DISCUSSION

See also Requirement III.4.2-1.2, Requirement III.4.2-1.3, and Requirement VI.1.1-5. More than one ballot format may be available in the case of open primaries (Requirement III.4.6-2.4).

4.6-2.1 DRE and EBP, enable all applicable contests

DREs and EBPs shall activate all portions of the ballot upon which the voter is entitled to vote.

Source: [2] I.2.4.2.g.

4.6-2.2 DRE and EBP, disable all non-applicable contests

DREs and EBPs shall disable all portions of the ballot upon which the voter is not entitled to vote.

Source: [2] I.2.4.2.h.

4.6-2.3 DRE and EBP, select ballot format for party in primary elections

DREs and EBPs of the *Primary elections device* class shall enable the selection of the ballot format that is appropriate to the party affiliation declared by the voter in a primary election.

Source: [2] I.2.4.2.f.

Applies to: DRE ^ Primary elections device, EBP ^ Primary elections device

DISCUSSION

In paper-based systems, open primaries have sometimes been handled by printing a single ballot format that merges the contests from all parties, instructing the voter to vote only in the contests applicable to a single party, and rejecting or discarding votes that violate this instruction. To use that approach on a DRE or EBP would violate Requirement III.4.6-2.2.

4.6-2.4 DRE and EBP, open primaries, party selection should be private

In an open primary on a DRE or EBP, the voter should be allowed to choose a party affiliation at the start of the voting session and vote the appropriate ballot format in privacy (i.e., the choice of affiliation should be private as well as the selection of votes on the ballot).

Source: New requirement.

Applies to: DRE ^ Open primaries device, EBP ^ Open primaries device

Test reference: Test 8

4.6.2   General voting functionality

4.6-3 Align voting targets with candidate names

All vote-capture devices shall ensure that vote response fields, selection buttons, or switches properly align with the specific candidate names and/or issues.

Source: [2] I.2.3.1.1.1.f.

Applies to: Vote-capture device

DISCUSSION

See also Requirement VI.1.1-1. Devices may be unable to meet this requirement if paper ballots are not produced correctly.

HFP: Does this belong in [5] I.3.1.6 (Interaction Issues)?

4.6-4 Support required languages

All vote-capture devices shall be capable of rendering an image of the ballot in any of the languages required by 42 USC 1973aa-1a.

Source: [2] I.2.3.1.3.1.a.

Applies to: Vote-capture device

DISCUSSION

42 USC 1973aa-1a is the Voting Rights Act of 1965, as amended.

HFP: Does this belong in [5] I.3.2.7 (English Proficiency)?

Ⓡ 4.6-5 No advertising

The ballot presented to the voter shall not display or link to any advertising or commercial logos of any kind, whether public service, commercial, or political, unless added by central election officials using the functionality described in Requirement III.4.2-1.5.

Source: Clarification of [2] I.2.3.1.3.1.b.

Applies to: Vote-capture device

Ⓡ 4.6-6 Capture votes

All vote-capture devices shall record the selection and non-selection of individual vote choices for each contest and ballot measure.

Source: [2] I.2.4.3.1.c.

Applies to: Vote-capture device

Ⓡ 4.6-6.1 Voter interaction with DREs and EBMs

DREs and EBMs shall

a. enable the voter to easily identify the selection button, switch, or active area of the ballot display that is associated with each candidate or ballot measure response;

b. allow the voter to select his or her preferences on the ballot in any legal number and combination;

c. indicate that a selection has been made or canceled;

d. indicate to the voter when no selection, or an insufficient number of selections, has been made in a contest;

e. notify the voter when the selection of candidates and measures is completed; and

f. provide responses to each voter entry in no more than three seconds.

Source: [2] I.2.4.3.3.b through e, g, and l.

Applies to: DRE, EBM

Impact: Deleted [2] I.2.4.3.3.a, redundant with STS requirements.

HFP: Some or all of these belong in [5] I.3.1? (Undervoting already in I.3.1.2.a.)

R 4.6-6.2 DRE and EBM, prevent overvoting

DREs and EBMs shall prevent the voter from overvoting.

Source: [2] I.2.4.3.3.f.

Applies to: DRE, EBM

Test reference: Test 39, Test 40

DISCUSSION

Preventing overvotes avoids the unintentional loss of votes from voters who overvote accidentally. For those who would overvote deliberately, a protest vote is more validly communicated through undervoting (vote for none). The effect on the candidate totals is identical.

Impact: Retained requirement per resolutions of Issue #2323 and #2715. [5] did the opposite and allowed overvoting on DREs—rationale unknown.

HFP: Overlap with [5] I.3.1.2.

4.6.3 Voting variations

R 4.6-7 Vote-capture device, voting variations

All vote-capture devices shall support the gathering of votes using all voting variations indicated in the implementation statement.

Source: Extrapolated from [2] I.2.2.8.2 and I.2.4.

Applies to: Vote-capture device

4.6-7.1 Vote-capture device, 1-of-M

All vote-capture devices shall be capable of gathering and recording votes in contests where the voter is allowed to choose at most one candidate from a list of candidates.

Source: [2] I.2.4. Extended [2] I.2.4.2.e to all systems.

Test reference: Test 2, Test 3, Test 19, Test 22

4.6-7.2 Vote-capture device, indicate party endorsements

All vote-capture devices shall be capable of indicating the political parties (if any) that endorsed each candidate.

Source: Added precision.

4.6-7.3 Vote-capture device, closed primaries

Vote-capture devices of the *Closed primaries device* class shall be capable of gathering and recording votes within a voting process that assigns different ballot formats depending on the registered political party affiliation of the voter and supports both partisan and nonpartisan contests.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ *Closed primaries device*

Test reference: Test 7

4.6-7.4 Vote-capture device, open primaries

Vote-capture devices of the *Open primaries device* class shall be capable of gathering and recording votes within a voting process that assigns different ballot formats depending on the political party chosen by the voter at the time of voting and supports both partisan and nonpartisan contests.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ *Open primaries device*

DISCUSSION

In paper-based systems, open primaries have sometimes been handled by printing a single ballot format that merges the contests from all parties, instructing the voter to vote only in the contests applicable to a single party, and rejecting or discarding votes that violate this instruction. To satisfy the requirements for *Open primaries device*, the vote-capture device must be *capable* of handling the case where different ballot configurations are associated with different political parties.

Test reference: Test 8

4.6-7.5 Vote-capture device, write-ins

Vote-capture devices of the *Write-ins device* class shall record the voter's selection of candidates whose names do not appear on the ballot and record as many write-in votes as the voter is allowed, per the definition of N(*r*) in Volume III Section 5.3.

Source: [2] I.2.4.3.1.d.

Applies to: *Vote-capture device ^ Write-ins device*

Test reference: Test 9, Test 15, Test 28, Test 29, Test 32, Test 33

Impact: Removed untestable reference to state law.

4.6-7.6 Vote-capture device, ballot rotation

Vote-capture devices of the *Ballot rotation device* class shall be capable of gathering and recording votes when the ordering of candidates in ballot positions within each contest is variable.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: *Vote-capture device ^ Ballot rotation device*

Test reference: Test 10

4.6-7.7 Ballot rotation, equal time for each candidate

Programmed vote-capture devices that enable ballot rotation in a given contest shall alter the ordering of candidates or choices in such a manner that no candidate or choice shall ever have appeared in any particular ballot position two or more times more often than any other.

Source: Clarification or extension of existing requirements.

Applies to: *Vote-capture device ^ Programmed device ^ Ballot rotation device*

DISCUSSION

This is less restrictive than requiring sequential rotation. For a contest of *M* candidates, the order may be shuffled randomly after each batch of *M* ballots and rotated sequentially within each batch.

Test reference: Test 10

4.6-7.8 Vote-capture device, straight party voting

Vote-capture devices of the *Straight party voting device* class shall be capable of gathering and recording votes for a special contest in which the selection of a political party implies votes for the candidates endorsed by that party in all straight-party-votable contests on the ballot.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ *Straight party voting device*

Test reference: Test 11, Test 30

4.6-7.9 Vote-capture device, cross-party endorsement

Vote-capture devices of the *Cross-party endorsement device* class shall be capable of gathering and recording straight-party votes when a given candidate is endorsed by two or more different political parties.

Source: Clarification or extension of existing requirements.

Applies to: Vote-capture device ^ *Cross-party endorsement device*

Test reference: Test 12

4.6-7.10 Vote-capture device, split precincts

Vote-capture devices of the *Split precincts device* class shall be capable of gathering and recording votes in a precinct where there are distinct ballot formats for voters from two or more election districts.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ *Split precincts device*

Test reference: Test 13

4.6-7.11 Vote-capture device, N of M voting

Vote-capture devices of the *N of M voting device* class shall be capable of gathering and recording votes in contests where the voter is allowed to choose up to a specified number of candidates (N(*r*) > 1, per Volume III Section 5.3) from a list of candidates.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ N of M voting device

Test reference: Test 14, Test 15, Test 21, Test 31, Test 32, Test 33

4.6-7.12 Vote-capture device, cumulative voting

Vote-capture devices of the *Cumulative voting device* class shall be capable of gathering and recording votes in contests where the voter is allowed to allocate up to a specified number of votes (N(*r*) > 1, per Volume III Section 5.3) over a list of candidates however he or she chooses, possibly giving more than one vote to a given candidate.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ Cumulative voting device

Test reference: Test 16, Test 34

4.6-7.13 Vote-capture device, ranked order voting

Vote-capture devices of the *Ranked order voting device* class shall be capable of gathering and recording votes in contests where the voter is allowed to rank candidates in a contest in order of preference, as first choice, second choice, etc.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ Ranked order voting device

Test reference: Test 17

4.6-7.14 Vote-capture device, provisional / challenged ballots

Vote-capture devices of the *Provisional / challenged ballots device* class shall be capable of gathering and recording votes within a voting process that allows the decision whether to count a particular ballot to be deferred until after election day.

Source: Added precision, based on [2] I.2.2.8.2 and glossary.

Applies to: Vote-capture device ^ Provisional / challenged ballots device

DISCUSSION

Unique identification of each provisional/challenged ballot is required. See Requirement III.4.8-4.4.

Test reference: Test 18, Test 35

Ⓡ 4.6-7.15 DRE, categorize provisional ballots

DREs of the *Provisional / challenged ballots device* class shall provide the capability to categorize each provisional/challenged ballot.

Source: [3] 5.6.5.2.s.2.[7]

Applies to: DRE ^ Provisional / challenged ballots device

DISCUSSION

Categories (e.g., "regular provisional," "extended hours provisional," "regular extended hours") would be jurisdiction-dependent.

Ⓡ 4.6-7.16 Vote-capture device, review-required ballots

Vote-capture devices of the *Review-required ballots device* class shall be capable of gathering and recording votes within a voting process that requires certain ballots to be flagged or separated for review.

Source: Extrapolated from [2] I.2.5.2.

Applies to: Vote-capture device ^ Review-required ballots device

DISCUSSION

In some systems and jurisdictions, all ballots containing write-in votes might require flagging or separation for review. Support for the class indicates that the system can flag or separate ballots in this manner and include the results of the review in the reported totals (see Volume III Section 2.6.3.1). The reasons for which ballots are flagged or separated are jurisdiction-dependent. It is assumed that ballot presentation is unchanged for review-required ballots.

STS and HFP: Consider fraud/privacy issues related to ballot separation.

4.6.4 Recording votes

Ⓡ 4.6-8 Record votes as voted

Vote-capture devices shall record each vote precisely as indicated by the voter.

Source: [2] I.2.2.2.1.c / [5] I.2.1.2.c.

Applies to: Vote-capture device

DISCUSSION

This is an accuracy requirement.

Ⓡ 4.6-9 DRE, confirm votes recorded

DREs shall verify (i.e., actively check and confirm) the correct addition of voter selections to the memory components or persistent storage of the device.

Source: [2] I.3.2.4.3.3.c, expanded to include persistent storage.

Applies to: DRE

DISCUSSION

"Memory components or persistent storage" includes on-board RAM, nonvolatile memory, hard disks, optical disks, etc. See also Requirement III.4.6-10 and Requirement III.4.6-16.

Ⓡ 4.6-10 DRE, recording accuracy

For DREs, the acceptable voting system error rate (Requirement III.3.4-1) applies to recording the voter selections of candidates and contests into voting data storage.

Source: [2] I.3.2.1.b.1.

Applies to: DRE

Test reference: Volume V Section 4.2.2.2

Ⓡ 4.6-11 Power supply failure, retain no half-finished ballots

In the event of a failure of both main and backup power supply, any stored data associated with a ballot in progress, other than audit log data, shall not be retained.

Source: [3] 5.4.4.b,[7] adjusted to resolve conflicts.

Applies to: Vote-capture device ^ *Programmed device*

DISCUSSION

The goals are to preserve voter secrecy and prevent tabulation of a duplicate ballot. See also Requirement III.4.6-13.3.

4.6-12 EBM, review before print

EBMs shall

a. allow the voter, before the ballot is marked, to review his or her choices and, if the voter desires, to delete or change his or her choices before the ballot is marked;

b. prompt the voter to confirm the voter's choices before marking his or her ballot; and

c. notify the voter after the ballot has been marked successfully that the ballot is ready to be cast.

Source: [2] I.2.4.3.3.h through j, modified for EBMs.

Applies to: EBM

HFP: Some or all of these belong in [5] I.3.1?

4.6-13 Casting

All systems shall support the casting of a ballot.

Source: [2] I.2.4. Extended [2] I.2.4.2.e to all systems.

Applies to: Voting system

DISCUSSION

This does not entail retaining a ballot image. DREs are required to retain ballot images (see Requirement III.4.11-1.4) but other devices might not.

4.6-13.1 Every voter gets to vote

All systems shall make it possible for each eligible voter to cast a ballot, provided that the limits declared in the implementation statement for each device are not exceeded.

Source: [2] I.2.4.2.b, generalized to all systems.

DISCUSSION

See also Requirement VI.1.1-3, Requirement VI.1.1-4 and Requirement VI.1.1-5.

Ⓡ 4.6-13.2 Paper-based, must have secure ballot boxes

Systems that include paper-based vote-capture devices shall include secure receptacles for holding voted ballots.

Source: [2] I.2.4.1.2.1.c.

Applies to: Paper-based device ^ Vote-capture device

Ⓡ 4.6-13.3 DRE, review and cast ballot

DREs shall

a. allow the voter, before the ballot is cast, to review his or her choices and, if the voter desires, to delete or change his or her choices before the ballot is cast;

b. prompt the voter to confirm the voter's choices before casting his or her ballot, signifying to the voter that casting the ballot is irrevocable and directing the voter to confirm the voter's intention to cast the ballot;

c. notify the voter after the vote has been stored successfully that the ballot has been cast; and

d. notify the voter that the ballot has not been cast successfully if it is not stored successfully, including storage of the ballot image, and provide clear instruction as to the steps the voter should take to cast his or her ballot should this event occur.

Source: [2] I.2.4.3.3.h through k.

Applies to: DRE

HFP: Some or all of these belong in [5] I.3.1?

DISCUSSION

If a DRE fails at the point of casting a ballot, it must clearly indicate to the voter and to election judges responding to the failure whether or not the ballot was cast. The following behavior would be non-conforming: "106 voting units experienced screen freezes. In staff opinion this is the most serious of errors. Election judges and technical staff reported that many of these units froze when the voter pressed the Cast Ballot button. This leads to great confusion for judges and voters. The voter leaves the polling place with little or no confidence that their vote was counted. In many cases, the election judges are unable to provide substantial confirmation that the vote was, in fact, counted." [16]

R 4.6-14 DRE, cast is committed

DREs shall prevent modification of the voter's vote after the ballot is cast.

Source: [2] I.2.4.3.3.n.

Applies to: DRE

DISCUSSION

See also Requirement VI.1.1-6, cast ballot.

4.6.5   Redundant records

This section contains design requirements to enhance the recoverability of DRE devices. This is usually separable from auditability, which is addressed by Independent Verification ([5] I.C / Dangling ref: FutureIV). However, in some systems, the same records might satisfy both these requirements and Independent Verification.

R 4.6-15 DRE, at least two separate copies of CVR

DREs shall record and retain at least two machine-countable copies of each cast vote record.

Source: [2] I.2.2.2.2, I.2.2.4.2 and I.3.2.4.3.2.c.

Applies to: DRE

DISCUSSION

Besides data stored in electronic memory, a paper record with barcodes or EBM-style markings would qualify as machine-countable.

R 4.6-15.1 DRE, redundant CVRs on physically separate media

These redundant records shall be written to media that are physically separate from one another (e.g., two separate memory cards or one electronic record and one paper record).

Source: [2] I.2.2.4.2 and I.3.2.4.3.2.c.

Impact: Ambiguous requirements [5] I.4.1.4.3.b.iii and iv punted to IV, where the question of what constitutes a "separate path" or "separate process" has been taken to its logical conclusion. See also Volume I Section 1.6.

Ⓡ 4.6-16 DRE, redundant CVRs, accuracy

For DREs, the acceptable voting system error rate (Requirement III.3.4-1) applies to recording voter selections of candidates and contests into each of these records.

Source: [2] I.3.2.1.b.2.

Applies to: DRE

Test reference: Volume V Section 4.2.2.2

4.6.6 Respecting limits

Ⓡ 4.6-17 Tabulator, prevent counter overflow

When a tabulator can no longer accept another ballot without the potential of overflowing a vote counter or otherwise compromising the integrity of the counts, it shall notify the user or operator and cease to accept new ballots.

Source: Clarification of [2] II.5.4.2.g.

Applies to: Tabulator

DISCUSSION

Assuming that the counter size is large enough such that the value will never be reached is not adequate. Vendors are required to state specific limits, and systems are required to react when those limits are reached. Even if the system could fit in more ballots than the documented limit, it is more important that the behavior of the system agree with the documentation and be predictable.

Impact: This closes the loophole where a vendor might include such controls but leave them in a disabled or inactive state.

Test reference: Test 36, Test 37, Test 38, Test 41

4.6-17.1 DRE, stop when full

When a DRE can no longer accept another ballot without the potential of overflowing a vote counter or otherwise compromising the integrity of the counts, it shall emit appropriate warnings and audit events and cease to enable new ballots.

Source: Clarification of [2] II.5.4.2.g.

Applies to: DRE

DISCUSSION

A DRE must not initiate a voting session if there is the possibility that the next ballot could not be properly cast and recorded. If there exists a way of voting the ballot that would exceed one of the limits, then the ballot must not be enabled.

Test reference: Test 36, Test 37, Test 41

4.7 Closing polls

4.7-1 DRE, no CVRs before close of polls

DREs shall prevent access to cast vote records until after the close of polls.

Source: [2] I.2.4.3.3.r.

Applies to: DRE

DISCUSSION

This does not apply to paper-based devices because the ballot is subject to handling beyond their control; however, a locked ballot box (per Requirement III.4.6-13.2 and Requirement III.3.1-10) serves the same purpose. See also Requirement VI.1.1-7.

4.7-2 Programmed vote-capture devices, poll-closing function

Programmed vote-capture devices shall provide designated functions for closing the polls.

Source: Reworded from [2] I.2.5.

Applies to: Vote-capture device ^ Programmed device

4.7-2.1 Programmed vote-capture devices, no voting when polls are closed

Programmed vote-capture devices shall prevent the further enabling or marking of ballots once the polls have closed.

Source: Reworded from [2] I.2.5.1.a.

4.7-2.2 DRE, no ballot casting when polls are closed

DREs shall prevent the further casting of ballots once the polls have closed.

Source: Reworded from [2] I.2.5.1.a.

Applies to: DRE

4.7-2.3 Programmed vote-capture devices, poll closing integrity check

Programmed vote-capture devices shall provide an internal test that verifies that the prescribed closing procedure has been followed and that the device status is normal.

Source: Reworded from [2] I.2.5.1.b.

4.7-2.4 Programmed vote-capture devices, report on poll closing process

Programmed vote-capture devices shall provide a means to produce a diagnostic test record that verifies the sequence of events and indicates that the poll closing process has been activated.

Source: Reworded from [2] I.2.5.1.d.

4.7-2.5 Programmed vote-capture devices, prevent reopening polls

Programmed vote-capture devices shall prevent reopening of the polls once the poll closing has been completed for that election.

Source: Revised from [2] I.2.5.1.e.

Impact: Changed from "preclude the unauthorized reopening of polls."

4.7-3 Precinct EMS, post-election reports

Precinct EMSs shall provide designated functions for generating precinct post-election reports.

Source: Reworded from [2] I.2.5.

Applies to: Precinct tabulator ^ EMS

4.8 Counting

4.8.1 Voting variations

R 4.8-1 Write-ins, system must include supporting tabulators

Voting systems conforming to the *Write-ins* class shall count all write-in votes using tabulators of the *Write-ins device* class.

Source: Added precision.

DISCUSSION

If the voting system requires that write-in votes be counted manually, then it does not conform to the *Write-ins* class. However, it may conform to the *Review-required ballots* class.

R 4.8-2 Absentee, system must include supporting tabulators

Voting systems conforming to the *Absentee voting* class shall count all absentee ballots using tabulators of the *Absentee voting device* class.

Source: Added precision.

DISCUSSION

If the voting system requires that absentee ballots be counted manually, then it does not conform to the *Absentee voting* class. However, it may conform to the *Review-required ballots* class.

R 4.8-3 Provisional, system must include supporting tabulators

Voting systems conforming to the *Provisional / challenged ballots* class shall count all provisional / challenged ballots using tabulators of the *Provisional / challenged ballots device* class.

Source: Added precision.

DISCUSSION

If the voting system requires that provisional/challenged ballots be counted manually, then it does not conform to the *Provisional / challenged ballots* class. However, it may conform to the *Review-required ballots* class.

4.8-4 Tabulator, voting variations

All tabulators shall support all voting variations indicated in the implementation statement.

Source: [2] I.2.2.8.1 plus I.2.2.8.2.

Applies to: Tabulator

4.8-4.1 Tabulator, 1-of-M

All tabulators shall be capable of tabulating votes, overvotes, and undervotes in contests where the voter is allowed to choose at most one candidate from a list of candidates.

Source: Implicit in [2].

Test reference: Test 2, Test 3, Test 19, Test 22

4.8-4.2 Tabulator, absentee voting

Tabulators of the *Absentee voting device* class shall be capable of tabulating votes, overvotes, and undervotes from absentee ballots.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Absentee voting device

4.8-4.3 Tabulator, provisional / challenged ballots

Tabulators of the *Provisional / challenged ballots device* class shall be capable of tabulating votes, overvotes, and undervotes in contests where the decision whether to count a particular ballot is deferred until after election day.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Provisional / challenged ballots device

Test reference: Test 18, Test 35

4.8-4.4 Tabulator, accept or reject provisional / challenged ballots individually

Tabulators of the *Provisional / challenged ballots device* class shall support the independent acceptance and rejection of individual provisional/challenged ballots.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Provisional / challenged ballots device

DISCUSSION

This is meant to rule out the mode of failure in which the IDs assigned to provisional ballots fail to be unique, rendering the system incapable of accepting one without also accepting the others with the same ID.

Test reference: Test 18, Test 35

4.8-4.5 Tabulator, accept or reject provisional / challenged ballots by category

Tabulators of the *Provisional / challenged ballots device* class shall support the acceptance and rejection of provisional/challenged ballots by category.

Source: [3] 5.6.5.2.s.3.[7]

Applies to: Tabulator ^ Provisional / challenged ballots device

DISCUSSION

For "category," see Requirement III.4.6-7.15. The behavior when an individual acceptance/rejection conflicts with a categorical acceptance/rejection is system-dependent and should be documented by the vendor.

4.8-4.6 Tabulator, review-required ballots

Tabulators of the *Review-required ballots device* class shall be capable of tabulating votes, overvotes, and undervotes from ballots that were flagged or separated for review.

Source: Extrapolated from [2] I.2.5.2.

Applies to: Tabulator ^ Review-required ballots device

DISCUSSION

In some systems and jurisdictions, all ballots containing write-in votes might require flagging or separation for review. Support for the class indicates that the system can flag or separate ballots in this manner and include the results of the review in the reported totals (see Volume III Section 2.6.3.1). The reasons for which ballots are flagged or separated are jurisdiction-dependent. It is assumed that ballot presentation is unchanged for review-required ballots.

4.8-4.7 Tabulator, primary elections

Tabulators of the *Primary elections device* class shall be capable of keeping separate totals for each political party for the number of ballots read and counted.

Source: Added precision, based on [2] reporting requirements.

Applies to: Tabulator ^ *Primary elections device*

DISCUSSION

In paper-based systems, open primaries have sometimes been handled by printing a single ballot format that merges the contests from all parties and instructing the voter to vote only in the contests applicable to a single party. This approach requires additional logic in the tabulator to support the rejection or discarding of votes that violate these special instructions, while the approach of assigning different ballot formats to different parties does not. Support for the merged ballot approach is not required for a tabulator to satisfy the requirements for *Primary elections device*. See Volume I Section 2.1.

This requirement to separate by party applies only to the number of read ballots and counted ballots. It does not apply to candidate and measure vote totals.

Test reference: Test 7, Test 8

4.8-4.8 Tabulator, write-ins

Tabulators of the *Write-ins device* class shall be capable of tabulating votes for write-in candidates, with separate totals for each candidate.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ *Write-ins device*

Test reference: Test 9, Test 15, Test 28, Test 29, Test 32, Test 33

4.8-4.9 Tabulator, ballot rotation

Tabulators of the *Ballot rotation device* class shall be capable of tabulating votes when the ordering of candidates in ballot positions within each contest is variable.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Ballot rotation device

DISCUSSION

This just means that ballot rotation must not impact the correctness of the count. A mode of failure would be getting confused about the mapping from ballot positions to candidates.

Test reference: Test 10

4.8-4.10 Tabulator, straight party voting

Tabulators of the *Straight party voting device* class shall be capable of tabulating straight party votes.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Straight party voting device

Test reference: Test 11, Test 30

4.8-4.11 Tabulating straight party votes

A straight party vote shall be counted as a vote in favor of all candidates endorsed by the chosen party in each straight-party-votable contest in which the voter does not cast an explicit vote.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Straight party voting device

DISCUSSION

This requirement intentionally says nothing about what happens when there is both a straight party endorsed candidate and an explicit vote in a given contest (a scratch vote).

Although it seems obvious that a scratch vote in a 1-of-M race should take precedence over a straight party vote, it is less obvious after considering the generalized case of an N-of-M race in which the number of candidates endorsed by the selected party might be less than *N*. Approaches supported by commercially available technology include (1) all straight party selections are cancelled when an explicit selection exists; (2) both straight party and explicit selections are counted; (3) both straight party and explicit selections are counted unless this exceeds *N*, in which case only the explicit selections are counted; (4) both straight party and explicit selections are counted unless this exceeds *N*, in which case straight party selections from the bottom of the list are dropped until the number of selections is reduced to *N*.

These Guidelines do not specify any particular approach to resolving scratch votes, but the approach(es) supported are required to be described in the Voting Equipment User Documentation. See Requirement IV.2.4-12.

Fix tests containing scratch votes

Test reference: Test 11, Test 30

4.8-4.12 Tabulator, cross-party endorsement

Tabulators of the *Cross-party endorsement device* class shall be capable of tabulating straight-party votes when a given candidate is endorsed by two or more different political parties.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Cross-party endorsement device

Test reference: Test 12

4.8-4.13 Tabulator, split precincts

Tabulators of the *Split precincts device* class shall be capable of tabulating votes for two or more election districts within the same precinct.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Split precincts device

Test reference: Test 13

4.8-4.14 Tabulator, N of M voting

Tabulators of the *N of M voting device* class shall be capable of tabulating votes, overvotes, and undervotes in contests where the voter is allowed to choose up to a specified number of candidates (N(*r*) > 1, per Volume III Section 5.3) from a list of candidates.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ N of M voting device

Test reference: Test 14, Test 15, Test 21, Test 31, Test 32, Test 33

4.8-4.15 Tabulator, cumulative voting

Tabulators of the *Cumulative voting device* class shall be capable of tabulating votes, overvotes, and undervotes in contests where the voter is allowed to allocate up to a specified number of votes (N(*r*) > 1, per Volume III Section 5.3) over a list of candidates however he or she chooses, possibly giving more than one vote to a given candidate.

Source: Added precision, based on [2] I.2.2.8.1, I.2.2.8.2 and glossary.

Applies to: Tabulator ^ Cumulative voting device

Test reference: Test 16, Test 34

4.8-4.16 Tabulator, ranked order voting

Tabulators of the *Ranked order voting device* class shall be capable of determining the results of a ranked order contest for each round of voting.

Source: [2] I.2.2.8.1 plus I.2.2.8.2.

Applies to: Tabulator ^ Ranked order voting device

DISCUSSION

This requirement is minimal. Since ranked order voting is not currently in wide use, it is not clear what, other than the final result, must be computed.

Test reference: Test 17

4.8.2 Ballot separation and rejection

4.8-5 Central paper tabulator, ballot separation or rejection

In response to designated conditions, central paper-based tabulators shall (a) outstack the ballot, (b) stop the ballot reader and display a message prompting the election official or designee to remove the ballot, or (c) mark the ballot with an identifying mark to facilitate its later identification.

Source: [2] I.3.2.5.1.2.

Applies to: Central tabulator ^ Paper-based device

4.8-5.1 Central paper tabulator, unreadable ballots and write-ins

All paper-based central tabulators shall perform this action in response to an unreadable ballot or (if applicable) a manually-marked paper ballot containing write-in votes.

Source: [2] I.3.2.5.1.2.

DISCUSSION

An EBM-produced ballot might encode write-in text in machine-readable form, obviating the need to segregate such ballots.

4.8-5.2 Central paper tabulator, overvotes, undervotes, blank ballots

All paper-based central tabulators shall provide a capability that can be activated by central election officials to perform this action in response to ballots containing overvotes, blank ballots, and ballots containing undervotes in a designated race.

Source: [2] I.3.2.5.1.2.

4.8-6 Precinct paper tabulator, reject unreadable ballots

In response to an unreadable ballot, all paper-based precinct tabulators shall return the ballot and provide a message prompting the voter to examine the ballot.

Source: [2] I.3.2.5.1.3.a.

Applies to: Precinct tabulator ^ Paper-based device

DISCUSSION

The option to submit the ballot as-is is not required for unreadable ballots because a damaged ballot might be impossible to handle mechanically. See also Requirement III.4.8-11.

Impact: Blank ballots moved to Requirement III.4.8-9.

4.8-7 Precinct tabulator, separate or mark write-ins

All precinct tabulators that process manually-marked paper ballots shall, in response to a manually-marked paper ballot with a write-in vote, segregate the ballot or mark the ballot with an identifying mark to facilitate its later identification.

Source: [2] I.3.2.5.1.3.b.

Applies to: Precinct tabulator ^ Paper-based device

DISCUSSION

An EBM-produced ballot might encode write-in text in machine-readable form, obviating the need to segregate such ballots.

4.8-8 Precinct tabulator, overvotes and undervotes

All precinct tabulators shall provide a capability to

a. Identify an over- or undervoted ballot;

b. Return the ballot to the voter (if paper ballots are used);

c. Provide feedback to the voter that identifies specific contests or ballot issues for which an over- or undervote is detected;

d. Allow the voter, at the voter's choice, to correct the ballot or submit the ballot "as is" without correction.

Source: Merged overlapping requirements in [2] I.2.4.3.2.2 (paper), I.2.4.3.3.e (DRE), I.3.2.5.1.3.c and d (paper). In [5], the analogous requirements are I.2.3.3.2.e through h (paper), I.2.3.3.3.e through h (DRE), I.3.1.2.a through e (all systems), I.4.1.5.1.d.iii and iv (paper). Significant differences in [5] are (1) added requirement for the *system* to "Notify the voter before the ballot is cast and counted of the effect of making more than the allowable number of selections for a contest" (42 USC 15481 a.1.A.iii.II) and (2) DREs do not prevent overvoting. N.B., 42 USC 15481 a.1.B states that the notification need not be done by the *system* in all cases.

Applies to: Precinct tabulator

DISCUSSION

In paper-based systems, correcting an overvoted ballot means spoiling it and voting a new one. Erasures are to be avoided. Overvotes do not apply in the cases of DREs and EBMs (Requirement III.4.6-6.2).

"Voter's choice" is a topic needing discussion. The language of the requirement derives from [2] I.2.4.3.2.2.b (there applying to overvotes and undervotes). There are deployed systems that require poll worker intervention to override a ballot rejection. BW says in practice the voter is startled by a ballot rejection and needs the poll worker to explain what happened. OTOH perhaps this is a usability issue that can be solved by providing better feedback to the voter. (HFP)

4.8-8.1 Turn off second chance voting for undervotes only

It shall be possible for central election officials to turn off this function entirely or by contest for undervotes while leaving it enabled for overvotes.

Source: Clarification of [2] requirements per CRT advice.

Impact: [5] I.2.3.3.2 removed the requirement for the ability to turn off second-chance voting, perhaps because HAVA requires second-chance voting to be enabled, in some systems, for overvotes. However, parallel changes were *not* made in I.4.1.5.1.d.iii and iv. Advice from CRT is that enabling second-chance voting for ordinary undervotes causes a train wreck in the precincts as nearly every ballot deliberately undervotes some contest in which the voter had no interest. Resolved consistent with CRT advice.

4.8-9 Precinct paper tabulators, reject blank ballots

All paper-based precinct tabulators shall provide a capability to

a. Identify a blank ballot;

b. Return the ballot to the voter;

c. Provide feedback to the voter that the ballot appears to be blank;

d. Allow the voter, at the voter's choice, to correct the ballot or submit the ballot "as is" without correction.

Source: [2] I.3.2.5.1.3.a and d, clarifying for the case of blank ballots.

Applies to: Precinct tabulator ^ *Paper-based device*

DISCUSSION

Special case of Requirement III.4.8-8 for blank paper ballots.

"Voter's choice" issue from Requirement III.4.8-8 repeated here (HFP).

4.8-9.1 Can reject blank ballots without rejecting all undervotes

It shall be possible for central election officials to enable this function for blank ballots without enabling it for ballots that only undervote some contests.

Source: Clarification.

Ⓡ 4.8-10 Ballots only blank on one side

Paper-based precinct tabulators should provide a capability analogous to that of Requirement III.4.8-9 to reject two-sided ballots that are blank on one side.

Source: New requirement.

Applies to: Precinct tabulator ^ *Paper-based device*

DISCUSSION

Failing to notice the second side of a two-sided ballot is reportedly a common cause of unintentional undervotes.

Ⓡ 4.8-11 Precinct paper tabulator, capability to reject marginal marks

All paper-based precinct tabulators should provide a capability to

a. Identify a ballot containing marks or punches that do not conform to vendor specifications;

b. Return the ballot to the voter;

c. Provide feedback to the voter that identifies specific contests or ballot issues for which a marginal mark or hanging chad is detected;

d. Allow the voter, at the voter's choice, to correct the ballot or submit the ballot "as is" without correction.

Source: New requirement.

Applies to: Precinct tabulator ^ *Paper-based device*

DISCUSSION

This capability would be useful even when EBMs are used as it could assist in detecting a malfunctioning EBM. In many cases, correcting a ballot means spoiling it and voting a new one. Erasures are to be avoided.

"Voter's choice" issue from Requirement III.4.8-8 repeated here (HFP).

4.8.3 Paper jams

4.8-12 Paper-based tabulator, ability to clear misfeed

If multiple feed or misfeed (jamming) occurs, a paper-based tabulator shall halt in a manner that permits the operator to remove the ballot(s) causing the error and reinsert them in the input hopper (if unread) or insert them in the ballot box (if read).

Source: [2] I.3.2.5.1.4.a, expanded to include jamming and ballots that were read.

Applies to: Paper-based device ^ Tabulator

DISCUSSION

See also Requirement III.4.8-13 and Requirement VI.1.1-8.

Impact: Tightened language from "if multiple feed is detected" to "if multiple feed occurs." Failure to detect is still a failure. Changed "card" to "ballot."

4.8-13 Paper-based tabulator, indicate status of misfed ballot

If multiple feed or misfeed (jamming) occurs, a paper-based tabulator shall clearly indicate whether or not the ballot(s) causing the error have been read.

Source: [13] 14.2.5.3 (page 46).

Applies to: Paper-based device ^ Tabulator

DISCUSSION

A similar issue arises with DREs that hang just as the voter presses the "cast ballot" button. See Requirement III.4.6-13.3. See also Requirement III.4.8-12 and Requirement VI.1.1-8.

4.8-14 Paper-based tabulators, rate of misfeeds

The rate of multiple feeds, misfeeds (jamming), and rejection of ballots that meet all vendor specifications shall not exceed 1 ballot in 10,000.

Source: Merge of [2] I.3.2.5.1.4.b and I.3.2.5.2.c, harmonized to 1 in 10,000 benchmark.

Applies to: Paper-based device ^ Tabulator

Impact: Original requirement in I.3.2.5.2.c: Paper-based tabulators shall reject ballots that meet all vendor specifications at a rate not to exceed 2 %.

4.8.4 Accuracy

Requirement III.3.4-1 applies to all voting systems and need not be repeated here. The following requirements elaborate the general requirement with respect to issues that are unique to paper-based systems.

R 4.8-15 Paper-based tabulator accuracy

For paper-based tabulators, the acceptable voting system error rate (Requirement III.3.4-1) applies to scanning paper ballots to detect selections for individual candidates and contests and converting them into digital data.

Source: From [2] I.3.2.1.a.1, I.3.2.1.a.2 and I.3.2.6.1.1.

Applies to: Paper-based device ^ Tabulator

Test reference: Volume V Section 4.2.2.2

R 4.8-15.1 Punchcard reader accuracy

Punchcard readers shall detect punches that conform to vendor specifications with an error rate satisfying Requirement III.3.4-1.

Source: Narrowed from [2] I.3.2.5.2.a and I.3.2.6.1.1.

Applies to: Punchcard reader

Test reference: Volume V Section 4.2.2.2

R 4.8-15.2 Optical scanner, EBM, accuracy

Optical scanners that read EMPBs shall detect EBM-generated vote indications with an error rate satisfying Requirement III.3.4-1.

Source: Narrowed from [2] I.3.2.5.2.a and I.3.2.6.1.1.

Applies to: Optical scanner

Test reference: Volume V Section 4.2.2.2

DISCUSSION

Reading of marginal marks should be a non-issue if EBMs are used. The requirement applies equally regardless of whether the EMPB contains a bar code, traditional marksense ovals, or what have you.

4.8-15.3 Optical scanner, MMPB, accurately detect perfect marks

Optical scanners that read manually-marked paper ballots shall detect marks that conform to vendor specifications with an error rate satisfying Requirement III.3.4-1.

Source: [2] I.3.2.5.2.a and I.3.2.6.1.1.

Applies to: Optical scanner

4.8-15.4 Optical scanner, MMPB, accurately detect imperfect marks

Optical scanners that read manually-marked paper ballots shall detect a 1 mm thick line that is made with a #2 pencil, that crosses the entirety of the voting target on its long axis, that is centered on the voting target, and that is as dark as can practically be made with a #2 pencil, with an error rate satisfying Requirement III.3.4-1.

Source: Many issues and public comments. Specification of mark originated with recommendation in Issue #1322, changed to reduce ambiguity.

Applies to: Optical scanner

DISCUSSION

Different optical scanning technologies will register imperfect marks in different ways. Variables include the size, shape, orientation, and darkness of the mark, the location of the mark within the voting target, the wavelength of light used by the scanner, the size and shape of the scanner's aperture, the color of the ink, the sensed background-white and maximum-dark levels, and of course the calibration of the scanner. The mark specified in this requirement is intended to be less than 100 % perfect, but reliably detectable, i.e., not so marginal as to bring the uncontrolled variables to the forefront. In plain English: scanning technologies may vary, but as a minimum requirement, all of them should be capable of reliably reading *this* mark.

4.8-15.5 Paper-based tabulators, ignore extraneous outside voting targets

Paper-based tabulators shall not record as votes any marks, perforations, smudges, or folds appearing outside the boundaries of voting targets.

Source: Clarified from [2] I.3.2.5.2.b.

DISCUSSION

In previous iterations of these Guidelines it was unclear whether "extraneous perforations, smudges, and folds" included perforations, smudges and folds appearing *within* voting targets. Those appearing within voting targets are now discussed in Requirement III.4.8-15.6 and Requirement III.4.8-15.7. Those other requirements are "should" not "shall"—technology in wide use as of 2006 cannot reliably distinguish extraneous marks within voting targets from deliberate marks.

Ⓡ 4.8-15.6 Optical scanner, ignore extraneous inside voting targets

Optical scanners should not record as votes imperfections in the ballot stock and similar insignificant marks appearing inside voting targets.

Source: Clarified from [2] I.3.2.5.2.b.

Applies to: Optical scanner

DISCUSSION

With technology that is in wide use as of 2006, insignificant marks appearing inside voting targets can be detected as votes. This problem should be minimized as much as possible.

Ⓡ 4.8-15.7 Optical scanner, MMPB, ignore hesitation marks

Optical scanners that read manually-marked paper ballots should not record as votes hesitation marks and similar insignificant marks.

Source: Clarified from [2] I.3.2.5.2.b.

Applies to: Optical scanner

DISCUSSION

With technology that is in wide use as of 2006, it may be possible to reliably detect reasonable marks and reliably ignore hesitation marks if the scanner is calibrated to a specific marking utensil. Unfortunately, in practice, optical scanners are required to tolerate the variations caused by the use of unapproved marking utensils. Thus, lighter marks of a significant size are detected at the cost of possibly detecting especially dark hesitation marks. Emerging technologies for context-sensitive ballot scanning may solve this problem. It is also solvable through procedures that ensure that all voters use only the approved marking utensil.

Ⓡ 4.8-15.8 Optical scanner, marginal marks, not position-dependent

The detection of marginal marks from manually-marked paper ballots shall be independent of the ballot position in which those marks occur.

Source: New requirement.

Applies to: *Optical scanner*

DISCUSSION

The behavior on marginal marks is generally indeterminate, but if marginal marks in position 1 are more likely to count as votes than equivalent marginal marks in position 2, then the election is skewed in favor of the candidate in position 1.

Ⓡ 4.8-15.9 Optical scanner, marginal marks, repeatability

The detection of marginal marks from manually-marked paper ballots should be repeatable.

Source: New requirement.

Applies to: *Optical scanner*

DISCUSSION

It is difficult to have confidence in the equipment if consecutive readings of the same ballots on the same equipment yield dramatically different results. However, it is technically impossible to achieve repeatable reading of ballots containing many marks that fall precisely on the sensing threshold. This requirement cannot be made mandatory unless and until a testable and fair benchmark for repeatability of optical scanning is determined.

4.8.5 Consolidation

Ⓡ 4.8-16 Precinct EMS consolidation

Precinct EMSs shall consolidate the data contained in each unit into a single report for the polling place when more than one vote-capture device or precinct tabulator is used.

Source: Reworded from [2] I.2.5.3.2.

Applies to: *Precinct tabulator* ^ *EMS*

DISCUSSION

For requirements on report content see Volume III Section 4.9.

Ⓡ 4.8-16.1 DRE, consolidate in 5 minutes

DREs shall, if the consolidation of polling place data is done locally, perform this consolidation in a time not to exceed 5 minutes per DRE.

Source: Reworded from [2] I.3.2.6.2.1.

Applies to: Precinct tabulator ^ *EMS* ^ *DRE*

DISCUSSION

This requirement assumes that the precinct is operating using DREs exclusively and that one of those DREs fills the role of EMS.

🅡 4.8-17 Consolidation accuracy

The acceptable voting system error rate (Requirement III.3.4-1) applies to the consolidation of vote selection data from multiple tabulators to generate jurisdiction-wide vote counts, including storage and reporting of the consolidated vote data.

Source: Reworded from [2] I.3.2.1.

Applies to: Voting system

Test reference: Volume V Section 4.2.2.2

4.9 Reporting

Although reporting is typically an EMS function, most of the requirements in this section are scoped to the entire system because any given EMS might not generate all of the specified information. For example, the precinct- and jurisdiction-level reports are likely to be generated by different EMSs located in the precinct and central location, respectively. The precinct EMSs need not have the capability to generate jurisdiction-level reports and vice-versa.

4.9.1 General reporting functionality

🅡 4.9-1 Reports are timestamped

All reports shall include the date and time of the report's generation, including hours, minutes, and seconds.

Source: New requirement.

Applies to: Voting system

DISCUSSION

Even if the clock's accuracy leaves something to be desired, second precision is useful to have if two reports are generated in quick succession.

4.9-2 Timestamps should be ISO 8601 compliant

Timestamps in reports should comply with ISO 8601 [10], provide all four digits of the year and include the time zone.

Source: Recommendation to avoid ambiguous timestamps.

Applies to: Voting system

4.9-3 Reporting is non-destructive

All programmed devices shall prevent data, including data in transportable memory, from being altered or destroyed by report generation.

Source: From [2] I.2.2.6.h, I.2.5.3.1.g, and I.2.5.3.2.d.

Applies to: Programmed device

DISCUSSION

The appending of an audit record reflecting the fact that a report has been generated is not considered an alteration.

4.9.2 Audit, status, and readiness reports

4.9-4 Audit reports

All systems shall be capable of producing reports of all of the pre-election audit records, system readiness audit records, and in-process audit records defined in Volume III Section 3.2.2.

Applies to: Voting system

Source: From [2] I.2.2.6.i, I.2.3.6 and I.2.5.3.1.f.

4.9-5 Status reports

All programmed devices shall provide the capabilities to obtain status and equipment readiness reports.

Source: Reworded from [2] I.2.3.4.1.b.

Applies to: Programmed device

DISCUSSION

These reports typically are generated during pre-voting logic and accuracy testing; see Volume III Section 4.4.1.

4.9-6 Readiness reports, per polling place

Readiness reports shall include at least the following information for each polling place:

a. The election's identification data;

b. The identification of the precinct and polling place;

c. The identification of all voting devices deployed in the precinct;

d. The identification of all ballot formats used in that precinct;

e. Confirmation that no hardware or software failures were detected during setup and testing, or a record of those that occurred; and

f. Confirmation that all vote-capture devices are ready for the opening of polls, or identification of those that are not.

Source: [2] I.2.3.5, separated generic precinct vs. precinct tabulator reqs, modified to deal with failures.

Applies to: In-person voting

DISCUSSION

In jurisdictions where there are no programmed devices in the precincts, confirmation of equipment readiness could occur through a manual check and signoff by election judges. These readiness reports could take the form of checklists, fill-in forms and signature sheets supplied to the precincts by a central authority.

4.9-7 Readiness reports, precinct tabulator

Readiness reports shall include the following information for each precinct tabulator:

a. The election's identification data;

b. The identification of the precinct and polling place;

c. The identification of the tabulator;

d. The contents of each active candidate register by office and of each active measure register at all storage locations;

e. Confirmation that no hardware or software failures were detected during setup and testing, or a record of those that occurred; and

f. Any other information needed to confirm the readiness of the equipment and to accommodate administrative reporting requirements.

Source: [2] I.2.3.5, separated generic precinct vs. precinct tabulator reqs, harmonized with Requirement III.4.9-8, modified to deal with failures, deleted "special voting options."

Applies to: Precinct tabulator

4.9-8 Readiness reports, central tabulator

Readiness reports shall include the following information for each central tabulator:

a. The election's identification data;

b. The identification of the tabulator;

c. The identification of all ballot formats used in the jurisdiction;

d. The contents of each active candidate register by office and of each active measure register at all storage locations;

e. Confirmation that no hardware or software failures were detected during setup and testing, or a record of those that occurred; and

f. Any other information needed to confirm the readiness of the equipment and to accommodate administrative reporting requirements.

Source: [2] I.2.3.6, harmonized with Requirement III.4.9-7, modified to deal with failures, deleted "special voting options."

Applies to: Central tabulator

4.9.3 Vote data reports

4.9.3.1 General functionality

Ⓡ 4.9-9 Reporting, ability to produce text

All devices used to produce reports of the vote count shall be capable of producing:

a. Alphanumeric headers;

b. Election, office and issue labels; and

c. Alphanumeric entries generated as part of the audit record.

Source: [2] I.3.2.7.2 / [5] I.4.1.7.2

Applies to: Voting system

Impact: Original requirement was scoped to printers. Generalized to allow for paperless reporting.

Ⓡ 4.9-10 Report all votes cast

All systems shall be able to produce an accurate, human-readable report of all votes cast.

Source: [2] I.2.2.2.1.c as expanded by [3] 5.2.1.1.c.[7]

Applies to: Voting system

AG: need HFP input (human-readable)

Ⓡ 4.9-11 Account for all cast ballots and all valid votes

All systems shall produce vote data reports that account for all cast ballots and all valid votes.

Applies to: Voting system

Ⓡ 4.9-12 Vote data reports, discrepancies can't happen

Vote data reports shall be completely consistent, with no discrepancy among reports of voting device data at any level.

Source: Reworded from [2] I.3.2.6.2.2, extended to all systems.

Applies to: Voting system

Test reference: Test 1, Test 24 and all other tests.

Impact: Removed "error-free" language, which has caused confusion with respect to apparent conflict with general accuracy requirements. [2] I.3.2.6.2.2 is restricted to DREs and talks about consolidation and reporting. In Issue #2349, EAC interpretation was "3.2.1 refers to ballot position accuracy and 3.2.6.2.2 refers to accuracy of tabulation." Error-freeness is still the standard in logic verification and in the handling of votes that are attributed to the wrong contests or candidates (see Requirement V.4.2-8).

4.9-12.1 Discrepancies that happen anyway must be flagged

Any discrepancy that is detectable by the system shall be flagged by the system by an annotation or error message in the affected report(s) and/or a separate discrepancy report.

Source: New requirement in response to Issue #1366.

DISCUSSION

If this requirement is applicable, then the system has failed to satisfy Requirement III.4.9-12 and is therefore non-conforming. Nevertheless, in practice it is essential that discrepancies be flagged by the system as much as possible so that they are not overlooked by election judges. The system cannot detect discrepancies if no single voting device is ever in possession of a sufficient set of data.

4.9-12.2 Discrepancies that happen anyway must be explainable

Any discrepancy in reports, regardless of source, shall be resolvable to a specific cause.

Source: Reworded and generalized from [2] I.3.2.6.2.2.

DISCUSSION

If this requirement is applicable, then the system has failed to satisfy Requirement III.4.9-12 and is therefore non-conforming. Nevertheless, in practice it is essential that a specific cause be determinable.

4.9-13 Reporting, combined precincts

All systems should be capable of generating reports that consolidate vote data from selected precincts.

Source: Derived from [13] 14.3.2.3, [14] 5.04.05.g and [15] Requirement 23.

Applies to: *Voting system*

DISCUSSION

Jurisdictions in which more than one precinct may vote at the same location on either the same ballot format or a different ballot format may desire reports that consolidate the voting location.

Ⓡ 4.9-14 Precinct tabulators, no tallies before close of polls

Precinct tabulators shall prevent the printing of vote data reports and the extraction of vote tally data prior to the official close of polls.

Source: Revised from [2] I.2.5.3.2.

Applies to: *Precinct tabulator*

DISCUSSION

Providing ballot counts does not violate this requirement. The prohibition is against providing vote totals. Ballot counts are required for ballot accounting, but early extraction of vote totals is an enabler of election fraud.

Impact: Changed from "prevent the printing of reports and the unauthorized extraction of data."

4.9.3.2 Ballot counts

General statement for Requirement III.4.9-15 through Requirement III.4.9-27

The following compliance points were distilled, refactored, and clarified from overlapping, subtly differing requirements appearing several places in Chapters 2 and 4 of [2], including: I.2.2.2.1.c (produce an accurate report of all votes cast), I.2.2.6.h (printed report of everything in I.2.5), I.2.2.9 (ballot counter), I.2.5.2 (means to consolidate vote data), I.2.5.3.1.a (geographic reporting), I.2.5.3.1.b (printed report of number of ballots counted by each tabulator), I.2.5.3.1.c (contest results, overvotes, and undervotes for each tabulator), I.2.5.3.1.d (consolidated reports including other data sources), I.4.4.4.a (number of ballots cast, using each ballot configuration, by tabulator, precinct, and political subdivision), I.4.4.4.b (candidate and measure totals for each contest, by tabulator), I.4.4.4.c (number of ballots read within each precinct and for additional jurisdictional levels, by configuration, including separate totals for each party in primary elections), I.4.4.4.d (separate accumulation of overvotes and undervotes for each contest, by tabulator, precinct, and additional jurisdictional levels), and I.4.4.4.e (for paper-based systems, the total number of ballots both processed and unprocessable, and the total number of cards read).

🅡 4.9-15 Report cast ballots

All systems shall report the total number of cast ballots at the precinct, election district, and jurisdiction reporting levels, by configuration.

Applies to: Voting system

DISCUSSION

In the case of 100 % DRE systems, it would suffice to provide a single total that is noted to represent both the number of cast ballots and the number of read ballots, since these are necessarily equal. Only when there is a tangible (paper) ballot is it possible to cast a ballot that is never read. There is no sub-requirement for separate reporting of provisional cast ballots because the system is unlikely to know whether a ballot is provisional until it is successfully read.

🅡 4.9-16 Report read ballots

All systems shall report the total number of read ballots at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

Applies to: Voting system

🅡 4.9-16.1 Report read ballots, multi-page

Systems that include paper-based devices shall, if there are multiple card/page ballots, report the total number of cards/pages read at the precinct, election district, and jurisdiction reporting levels, by configuration.

🅡 4.9-16.2 Report read ballots by party

Systems conforming to the *Primary elections* class shall report separate totals for each party in primary elections.

Applies to: Primary elections

Test reference: Test 7, Test 8

DISCUSSION

This requirement to report by party applies only to the number of read ballots. It does not apply to candidate and measure vote totals.

4.9-16.3 Report read provisional ballots

Systems conforming to the *Provisional / challenged ballots* class shall report the total number of provisional/challenged read ballots at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

Applies to: Provisional / challenged ballots

Test reference: Test 18, Test 35

4.9-17 Report counted ballots

All systems shall report the total number of counted ballots at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

Applies to: Voting system

DISCUSSION

See also Requirement III.4.9-18, which breaks down counted ballots by contest.

4.9-17.1 Report counted ballots by party

Systems conforming to the *Primary elections* class shall report separate ballot counts for each party in primary elections.

Applies to: Primary elections

Test reference: Test 7, Test 8

DISCUSSION

This requirement to report by party applies only to the number of counted ballots. It does not apply to candidate and measure vote totals.

4.9-17.2 Report counted provisional ballots

Systems conforming to the *Provisional / challenged ballots* class shall report the total number of provisional/challenged counted ballots at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

Applies to: Provisional / challenged ballots

Test reference: Test 18, Test 35

4.9-17.3 Report blank ballots

All systems should report the number of blank ballots (ballots containing no votes) that were counted at each reporting level (tabulator, precinct, election district, and jurisdiction), by configuration.

DISCUSSION

Some jurisdictions find this information to be useful. Blank ballots sometimes represent a protest vote.

4.9-18 Report counted ballots by contest

All systems shall report the number of counted ballots for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $\mathrm{K}(j,r,t_{\mathrm{E}})$ in Volume III Section 5.3.

Applies to: Voting system

DISCUSSION

This is by contest, while Requirement III.4.9-17 is the overall count. N-of-M in this context includes the most common type of contest, 1-of-M.

4.9.3.3 Vote totals

4.9-19 Report votes for each contest

All systems shall report the candidate and measure vote totals for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $\mathrm{T}(c,j,r,t_{\mathrm{E}})$ in Volume III Section 5.3.

Applies to: Voting system

DISCUSSION

N-of-M in this context includes the most common type of contest, 1-of-M.

Test reference: Test 24 and all other tests.

4.9-20 Report overvotes for each contest

All systems shall report the number of overvotes for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $\mathrm{O}(j,r,t_{\mathrm{E}})$ in Table 4.

Applies to: Voting system

DISCUSSION

N-of-M in this context includes the most common type of contest, 1-of-M. [2] required the reporting of overvotes even on 100 % DRE systems where overvoting is prevented (Requirement III.4.6-6.2); that requirement is retained here, though it may be redundant.

Overvotes are defined in Table 4. Consistent with the definition of undervotes (see Requirement III.4.9-21), the count is of votes lost to overvoting, not of ballots containing overvotes. This means that a ballot that overvotes an N-of-M contest would contribute *N* to the count of overvotes for that contest.

Test reference: Test 6, Test 20, Test 27

4.9-20.1 Reporting overvotes, ad hoc queries

All systems shall be capable of producing a consolidated report of the combination of overvotes for any contest that is selected by an authorized official (e.g.; the number of overvotes in a given contest combining candidate A and candidate B, combining candidate A and candidate C, etc.).

Source: From [2] I.2.2.6.h and I.2.5.3.1.e.

Test reference: Test 6

4.9-21 Report undervotes for each contest

All systems shall report the number of undervotes for each N-of-M or cumulative voting contest, at each reporting level (tabulator, precinct, election district, and jurisdiction), per the definition of $\mathrm{U}(j,r,t_{\mathrm{E}})$ in Table 4.

Applies to: Voting system

DISCUSSION

N-of-M in this context includes the most common type of contest, 1-of-M.

Undervotes are defined in Table 4 as needed to enable accounting for every vote as described in Volume III Section 5.3.3. Counting ballots containing undervotes instead of votes lost to undervoting is insufficient.

Test reference: Test 25, Test 26 and other tests with undervotes.

4.9-22 Ranked order voting, report results

Systems conforming to the *Ranked order voting* class shall report the candidate and measure vote totals for each ranked order contest for each round of voting/counting at the jurisdiction level.

Applies to: Ranked order voting

DISCUSSION

This requirement is minimal. Since ranked order voting is not currently in wide use, it is not clear whether a count must be reported for each permutation of choices, how bogus orderings are reported, or how it would be done at multiple reporting levels.

Test reference: Test 17

4.9-23 Include in-person votes

Systems conforming to the *In-person voting* class shall include votes collected from in-person voting in the consolidated reports.

Applies to: In-person voting

DISCUSSION

"Include" simply means that the final totals must reflect them. It does not entail separate totals for the different kinds of votes.

4.9-24 Include absentee votes

Systems conforming to the *Absentee voting* class shall include votes from absentee ballots in the consolidated reports.

Applies to: Absentee voting

DISCUSSION

"Include" simply means that the final totals must reflect them. It does not entail separate totals for the different kinds of votes.

4.9-25 Include write-in votes

Systems conforming to the *Write-ins* class shall include write-in votes in the consolidated reports.

Applies to: Write-ins

DISCUSSION

"Include" simply means that the final totals must reflect them. It does not entail separate totals for the different kinds of votes.

4.9-26 Include accepted provisional / challenged votes

Systems conforming to the *Provisional / challenged ballots* class shall include votes from accepted provisional/challenged ballots in the consolidated reports.

Applies to: Provisional / challenged ballots

DISCUSSION

"Include" simply means that the final totals must reflect them. It does not entail separate totals for the different kinds of votes. See also Requirement III.4.8-4.3, Requirement III.4.9-16.3 and Requirement III.4.9-17.2.

Test reference: Test 18, Test 35

4.9-27 Include accepted reviewed votes

Systems conforming to the *Review-required ballots* class shall include votes from accepted reviewed ballots in the consolidated reports.

Applies to: Review-required ballots

DISCUSSION

"Include" simply means that the final totals must reflect them. It does not entail separate totals for the different kinds of votes.

4.10 Transmitting results

STS: I think most of the requirements that go here are STS requirements. See PREFACE.

I assume that [2] I.5 (mostly informative, but lots of it) has been taken over by STS. Also I.6.5.

Requirements on *Remote data delivery* may apply even if results are electronically transmitted only within the polling place (see Volume III Section 2.6.3.1).

4.10-1 ConfirmExtractionReq

All systems shall ensure that extracted or duplicated information, including cast vote records extracted from DRE machines, is identical to that on the original storage medium.

Source: Reworded from Section 5.6.9.2, Paragraph k of [3].[7] Also, [2] I.2.3.5, final paragraph.

Applies to: Voting system

4.10-1.1 GeneratedID1667

All programmed devices shall verify (i.e., actively check and confirm) that information as extracted or duplicated to machine-readable media is identical to that on the original storage medium.

Applies to: Programmed device

The following requirements are tentative, awaiting resolution of general networking issue (see PREFACE).

4.10-2 GeneratedID1671

Systems conforming to the *Remote data delivery* class shall prevent data, including data in transportable memory, from being altered or destroyed by the transmission of results over telecommunications lines, local networks, etc.

Source: Reworded from [2] I.2.5.3.1 and I.2.5.3.2.d.

Applies to: Remote data delivery

4.10-2.1 HarmlessXmitReq

Transmitting results over telecommunications lines, local networks, etc. shall not result in modifications to any vote data in the sending system.

Source: Added precision.

Ⓡ 4.10-2.2 GeneratedID1677

Systems conforming to the *Remote data delivery* class shall provide a means to verify the successful transmission of data over communication links.

Source: [2] I.2.3.5, final paragraph.

Ⓡ 4.10-2.3 10xmit

For systems conforming to the *Remote data delivery* class, the acceptable voting system error rate (Requirement III.3.4-1) applies to the transmission of data over telecommunications lines, local networks, etc.

Source: [2] I.5.2.1.

DISCUSSION

This requirement will typically be met through use of a standard error-correcting protocol that meets Requirement III.3.4-1.

Test reference: Volume V Section 4.2.2.2

4.11 Auditing and independent verification

This section is to be provided by STS. The text here is only notes. See PREFACE.

Ⓡ 4.11-1 GeneratedID1685

All systems shall be auditable by central election officials.

Source: Generalized from many [2] requirements.

Applies to: *Voting system*

Include stuff from [2] I.2.2.5. Make sure got all the audit records.

Ⓡ 4.11-1.1 BallotCounter1

All tabulators shall maintain a count of the number of ballots read so far during a particular test cycle or election.

Source: Phrasing the functional requirement that was implied by design requirements in [2]

I.2.2.9.

Applies to: Tabulator

DISCUSSION

For auditability, the ballot count must be *maintained* (incremented each time a ballot is read) rather than calculated on demand (by counting the ballots currently in storage).

R 4.11-1.2 BallotCounter2

All tabulators shall enable election judges to determine the number of ballots read so far during a particular test cycle or election at any time during the test cycle or election without disrupting any operations in progress.

Source: Phrasing the functional requirement that was implied by design requirements in [2] I.2.2.9.

Applies to: Tabulator

DISCUSSION

[2] I.2.4 refers to separate "election counter" and "life-cycle counter;" the latter was an error (intended to delete).

Confirm tie-in with general security, integrity, and auditability requirements after STS sections are written. See other sub-requirements in 2.2.9.

R 4.11-1.3 AccurateCVRReq

DREs shall maintain an accurate cast vote record of each ballot cast.

Source: Reworded from [2] I.2.2.4.2.

Applies to: DRE

Test reference: Test 5

JW talk about expanding this to include optical scanners.

R 4.11-1.4 ReadableCVRReq

DREs shall provide a capability to retrieve ballot images in human-readable form.

Source: [2] I.2.2.4.2.b and I.3.2.4.3.2.d.

Applies to: DRE

Test reference: Test 5

HFP: define human-readable

4.11-2 GeneratedID1709

All programmed devices shall provide software that monitors the overall quality of data read-write and transfer quality status, checking the number and types of errors that occur in any of the relevant operations on data and how they were corrected.

Source: [2] I.2.2.2.1.e.

Applies to: Programmed device

Chapter 5 Reference models

5.1 Process model (informative)

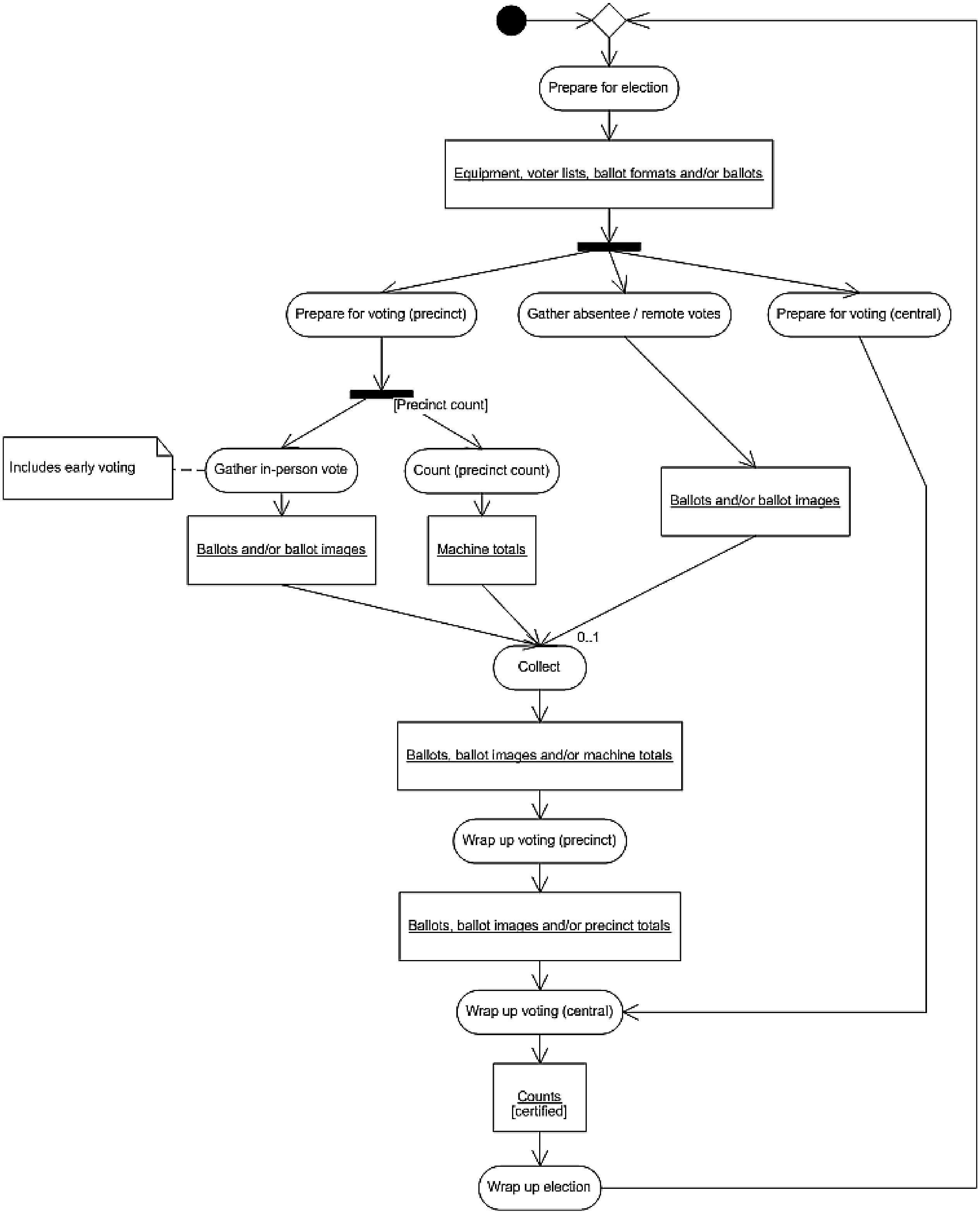
5.1.1 Introduction

This section contains 16 diagrams describing the elections and voting process. The diagrams are expressed in Unified Modeling Language (UML) version 2.0 [6].

To simplify the diagrams, the following shortcuts have been taken.

- The expansion regions around activities that are performed for every precinct or every voter are not shown.
- When a particular object may or may not exist depending on system and jurisdiction-specific factors (e.g., paper-based vs. DRE), that object is modelled as an optional parameter to an activity. This does not capture the constraint that subsequent activities must wait on this object in those jurisdictions where it applies (i.e., in some jurisdictions it is mandatory).
- Objects that flow downstream in an obvious manner through many activities are not shown as inputs/outputs of all of those activities.
- The propagation of the registration database from one election cycle to the next is not shown. The database appears as an input to the Register voters activity with no indication of its origin.
- Many activities produce reports and other objects that eventually flow into the Archive activity. These flows into the archive are not shown.

5.1.2 Diagrams



Prepare for election
Equipment, voter lists, ballot formats and/or ballots
Prepare for voting (precinct)
Gather absentee / remote votes
Prepare for voting (central)
[Precinct count]
Includes early voting
Gather in-person vote
Count (precinct count)
Ballots and/or ballot images
Ballots and/or ballot images
Machine totals
0..1
Collect
Ballots, ballot images and/or machine totals
Wrap up voting (precinct)
Ballots, ballot images and/or precinct totals
Wrap up voting (central)
Counts
[certified]
Wrap up election

Figure 4 Administer elections

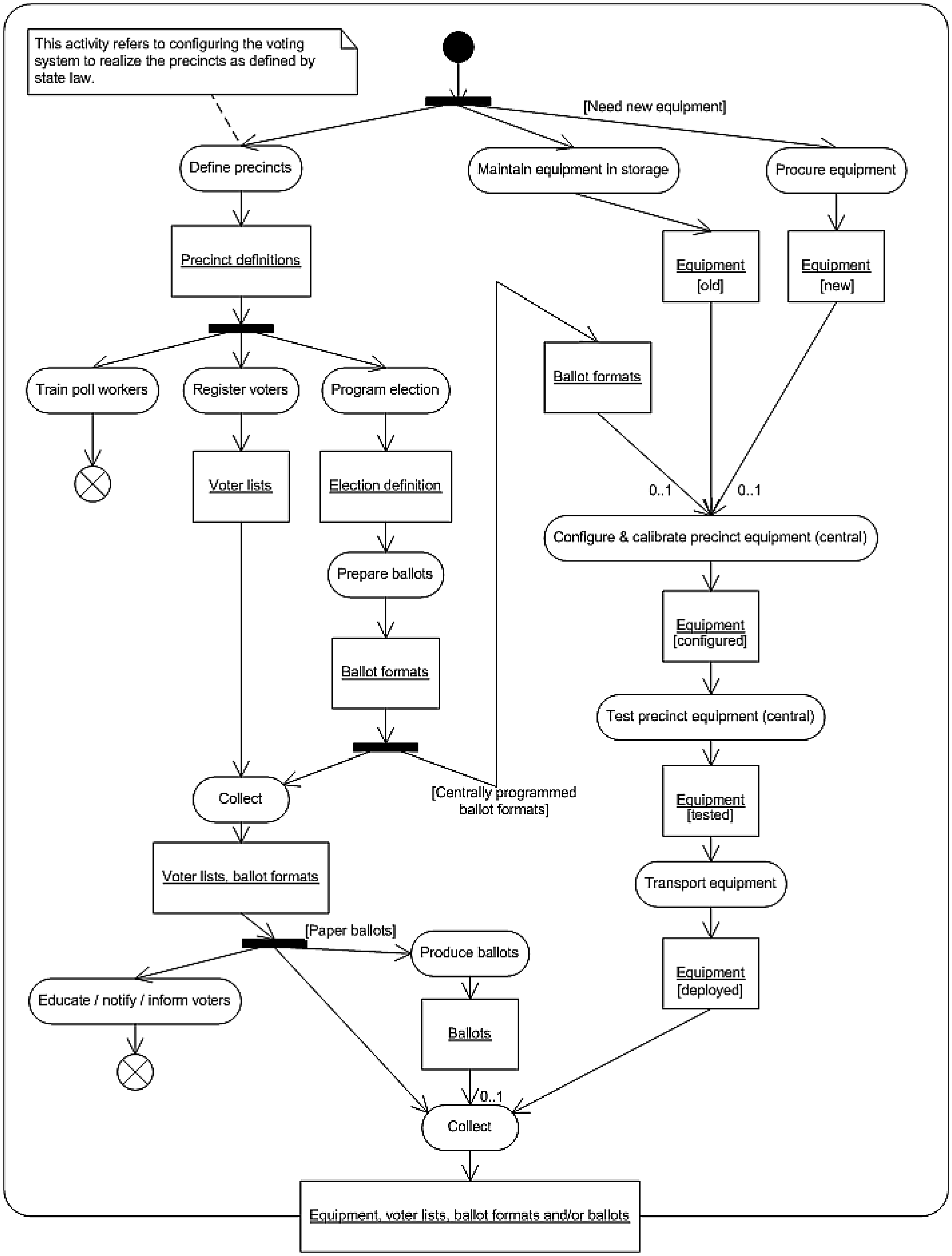


Figure 5 Prepare for election

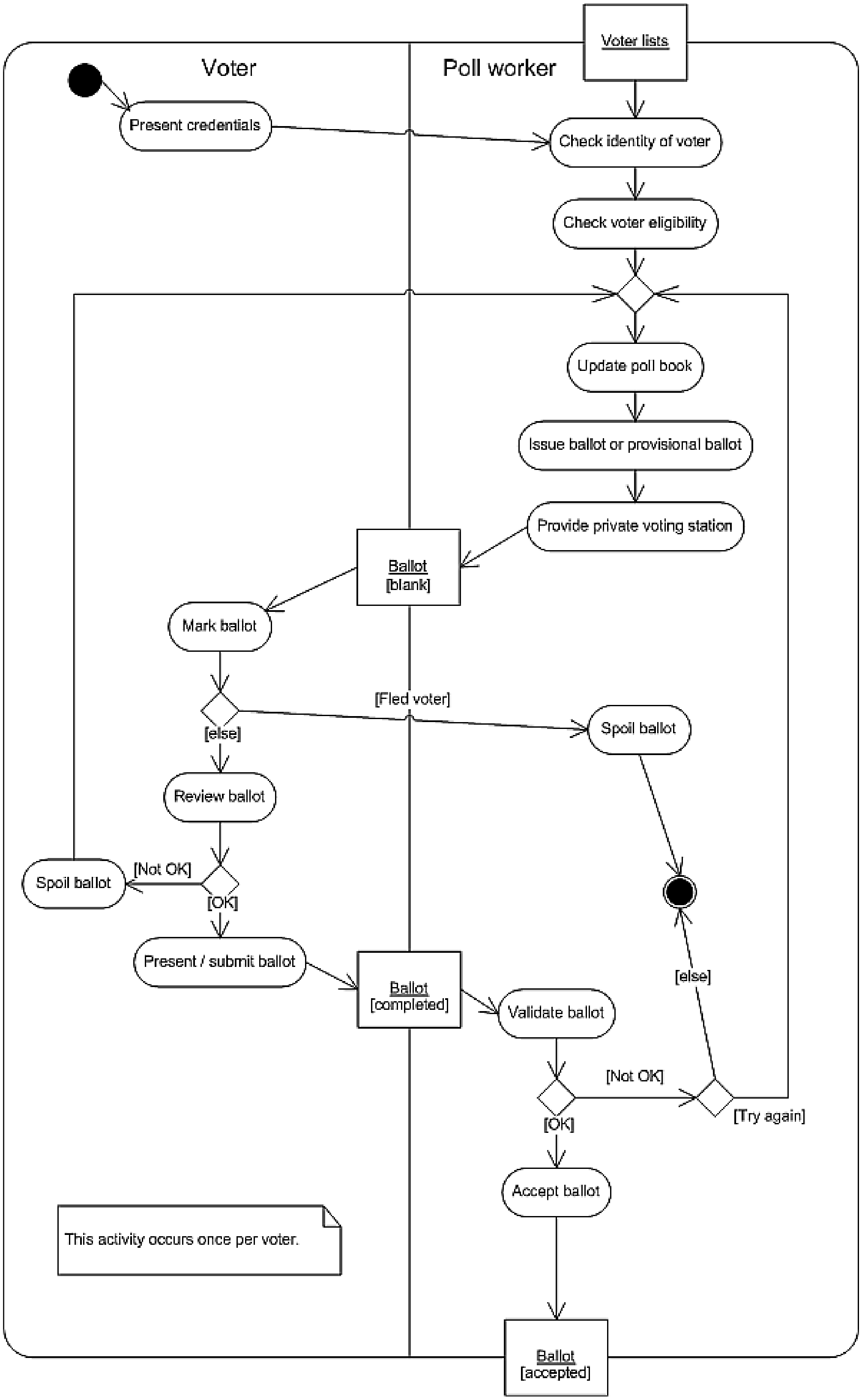


Figure 6 Gather in-person vote (paper-based)

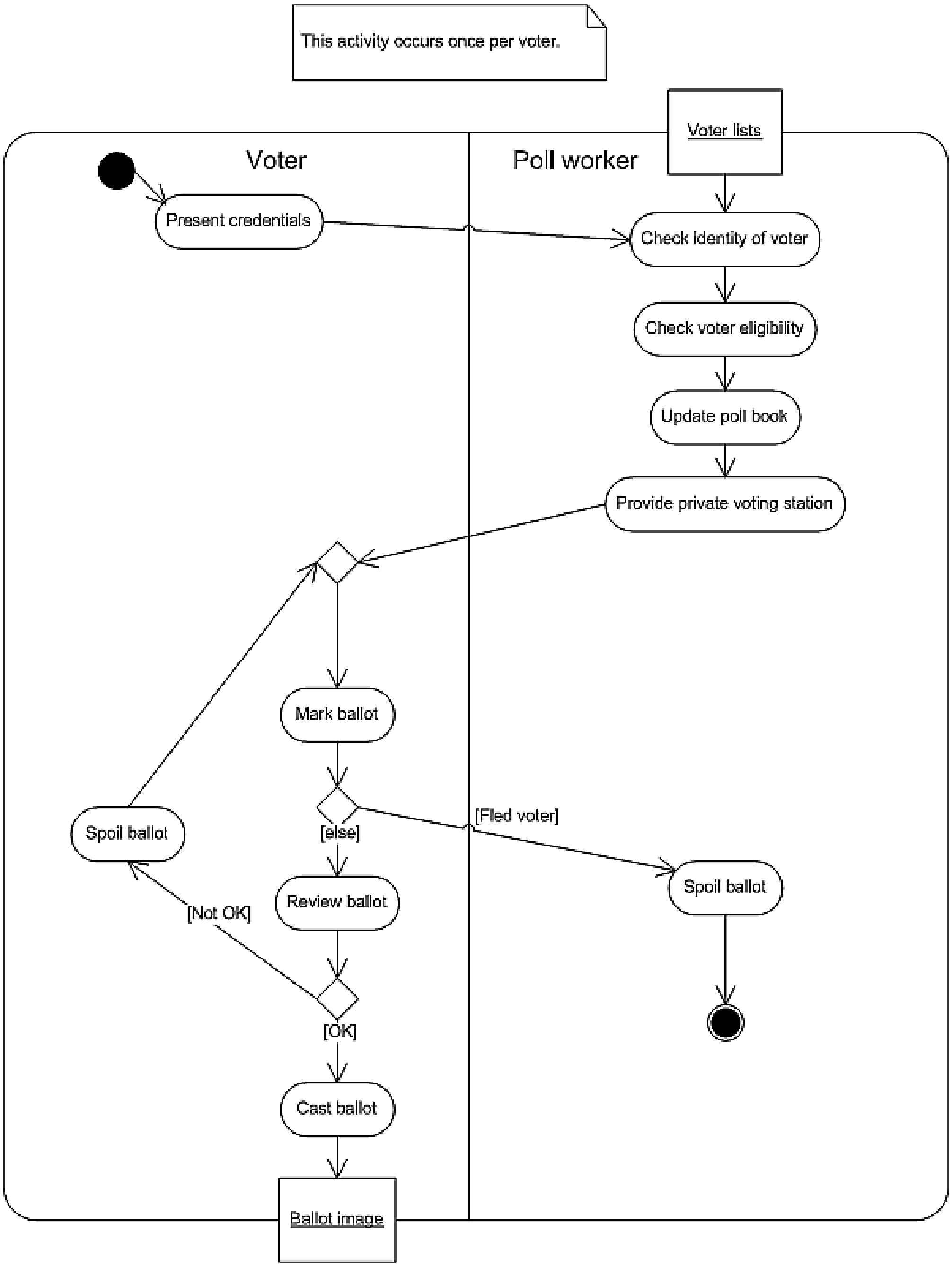


Figure 7 Gather in-person vote (DRE)

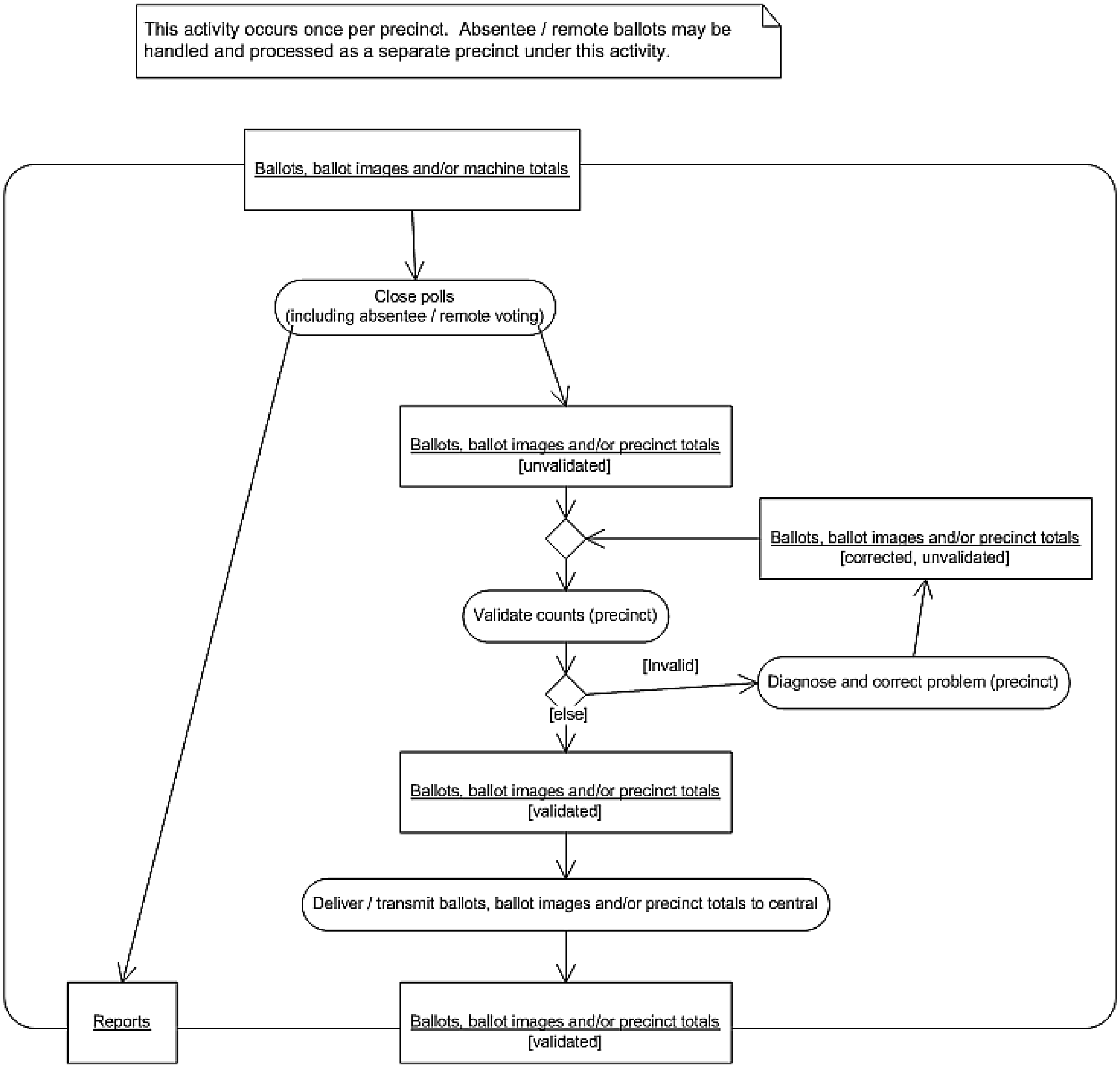

This activity occurs once per precinct. Absentee / remote ballots may be handled and processed as a separate precinct under this activity.
Ballots, ballot images and/or machine totals
Close polls
(including absentee / remote voting)
Ballots, ballot images and/or precinct totals
[unvalidated]
Ballots, ballot images and/or precinct totals
[corrected, unvalidated]
Validate counts (precinct)
[Invalid]
Diagnose and correct problem (precinct)
[else]
Ballots, ballot images and/or precinct totals
[validated]
Deliver / transmit ballots, ballot images and/or precinct totals to central
Reports
Ballots, ballot images and/or precinct totals
[validated]

Figure 8 Wrap up voting (precinct)

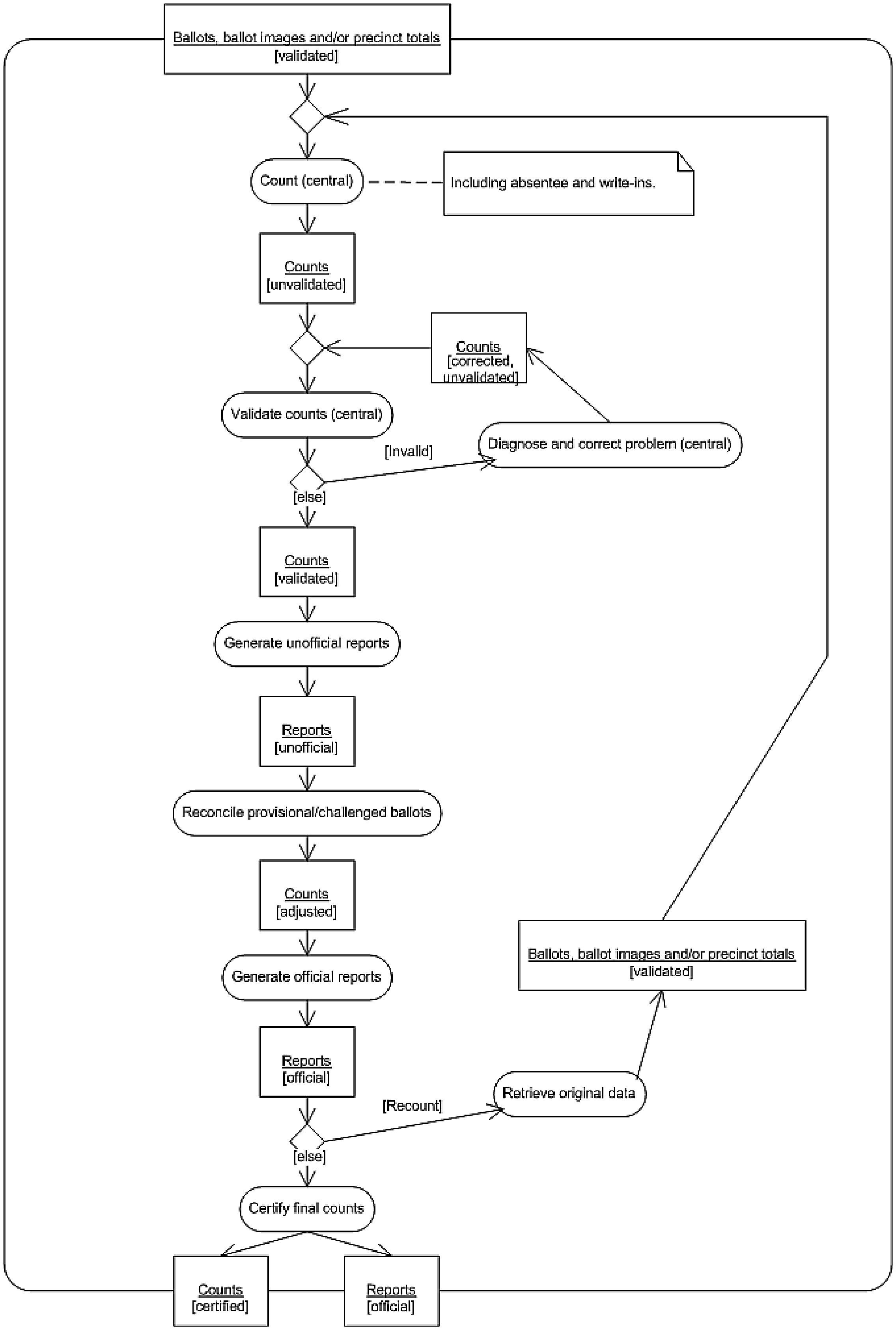

Ballots, ballot images and/or precinct totals [validated]
Count (central)
Including absentee and write-ins.
Counts [unvalidated]
Counts [corrected, unvalidated]
Validate counts (central)
Diagnose and correct problem (central)
[Invalid]
[else]
Counts [validated]
Generate unofficial reports
Reports [unofficial]
Reconcile provisional/challenged ballots
Counts [adjusted]
Generate official reports
Reports [official]
Ballots, ballot images and/or precinct totals [validated]
Retrieve original data
[Recount]
[else]
Certify final counts
Counts [certified]
Reports [official]

Figure 9 Wrap up voting (central)

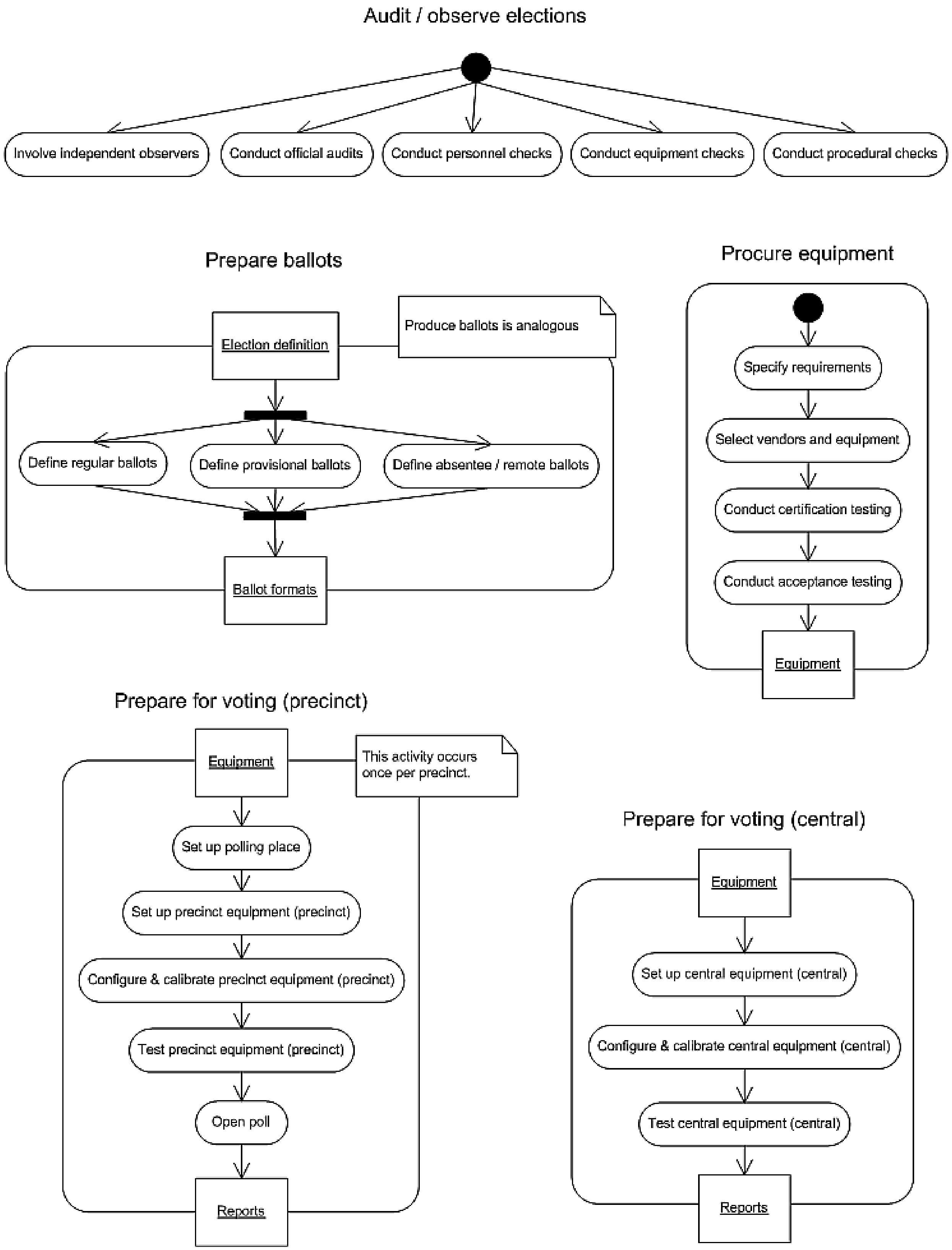

Audit / observe elections
Involve independent observers
Conduct official audits
Conduct personnel checks
Conduct equipment checks
Conduct procedural checks
Prepare ballots
Election definition
Produce ballots is analogous
Define regular ballots
Define provisional ballots
Define absentee / remote ballots
Ballot formats
Procure equipment
Specify requirements
Select vendors and equipment
Conduct certification testing
Conduct acceptance testing
Equipment
Prepare for voting (precinct)
Equipment
This activity occurs once per precinct.
Set up polling place
Set up precinct equipment (precinct)
Configure & calibrate precinct equipment (precinct)
Test precinct equipment (precinct)
Open poll
Reports
Prepare for voting (central)
Equipment
Set up central equipment (central)
Configure & calibrate central equipment (central)
Test central equipment (central)
Reports

Figure 10 Miscellaneous activities (1)

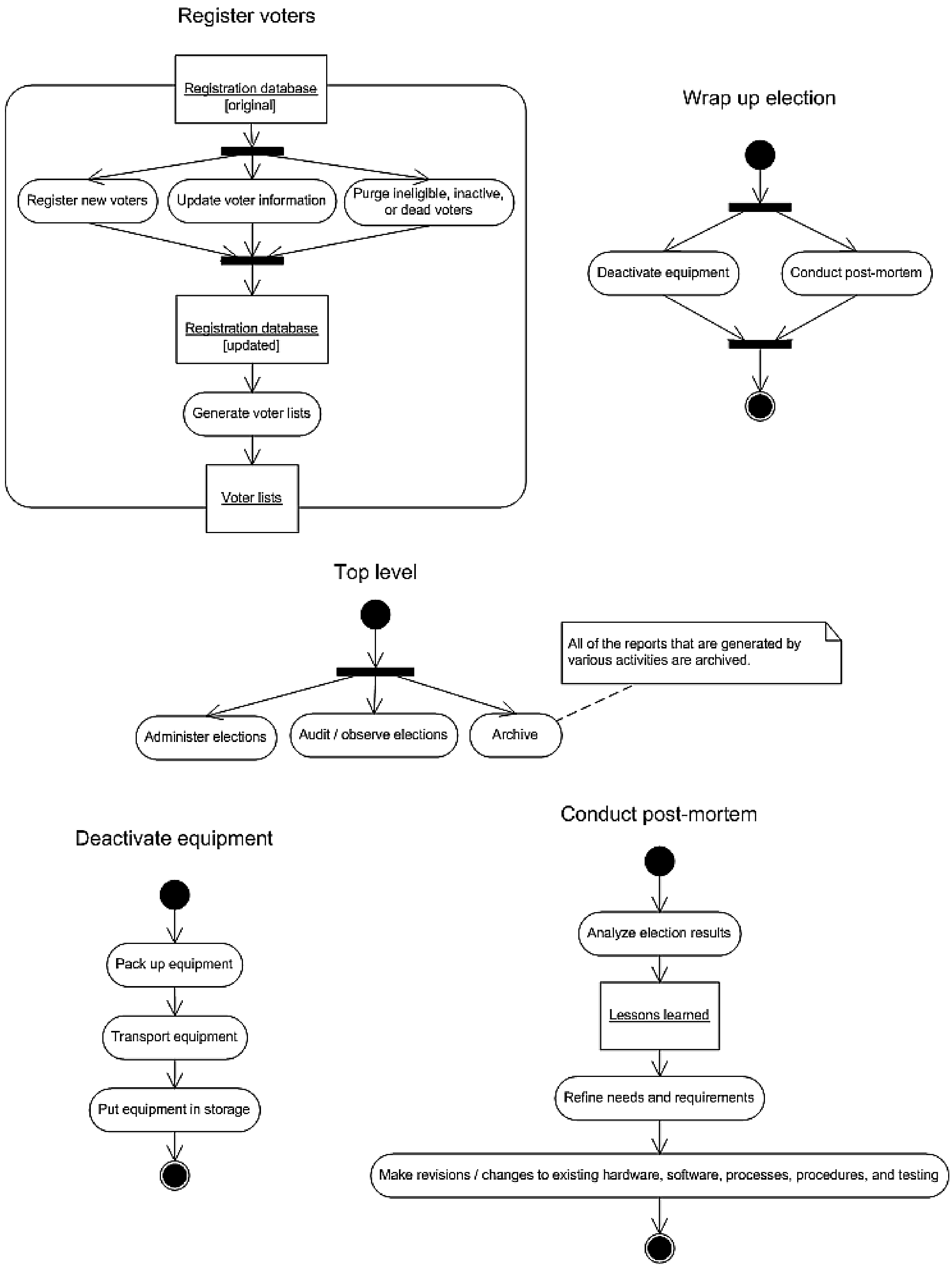

Register voters
Registration database [original]
Register new voters
Update voter information
Purge ineligible, inactive, or dead voters
Registration database [updated]
Generate voter lists
Voter lists
Wrap up election
Deactivate equipment
Conduct post-mortem
Top level
All of the reports that are generated by various activities are archived.
Administer elections
Audit / observe elections
Archive
Deactivate equipment
Pack up equipment
Transport equipment
Put equipment in storage
Conduct post-mortem
Analyze election results
Lessons learned
Refine needs and requirements
Make revisions / changes to existing hardware, software, processes, procedures, and testing

Figure 11 Miscellaneous activities (2)

5.1.3 Translation of diagrams

This subsection contains a rendering of the process model into text. The rendering is based on Petri Net Linear Form [19].

Although the form of the diagrams is being changed from drawings to text, the meanings of the diagram elements—activities, objects, etc.—continue to be as in UML 2.0 [6].

Activities are represented in this translation by the activity name in parenthesis. Objects are represented in this translation by the object name in square brackets. Sometimes the names of activities and objects will themselves be qualified by parenthetical phrases or object states in square brackets. These have been retained as-is, nesting the parenthesis or brackets as needed.

Sequential control and object flows are indicated with ->.

A flow may be qualified by a guard condition and/or a multiplicity such as 0..1. These notations are inserted immediately before and after the affected flow. For example, Daytime->0..1(Drink coffee) denotes an optional flow into the "drink coffee" activity that can only occur if the condition Daytime is true.

A node may be assigned an identifier that may be used as the target of flows from elsewhere in the diagram. The identifier is prefixed by an asterisk and is introduced by including it after the first occurrence of the node name. For example, (Do something *s) denotes an activity "do something" with the identifier *s. The node name may be omitted in subsequent references that include only the identifier.

The following special nodes appear with semantics as in UML 2.0. They are distinguished from objects and activities by being enclosed between < and >.

- <InitialNode>
- <ForkNode>
- <JoinNode>
- <DecisionNode>
- <MergeNode>
- <ActivityFinal>
- <FlowFinal>

When multiple flows follow from a node, they are listed between curly braces {} and separated by commas.

A semicolon indicates that the description is about to continue at a different node. A period indicates that the description of the diagram is complete.

Translation of the diagrams follows.

```
Diagram:  Administer elections

<InitialNode>
  -><MergeNode *merge>
  ->(Prepare for election)
  ->[Equipment, voter lists, ballot formats and/or ballots]
  -><ForkNode>{
    ->(Prepare for voting (precinct))
      -><ForkNode>{
        ->(Gather in-person vote)
          ->[Ballots and/or ballot images]
          ->(Collect *c),
        Precinct count
          ->(Count (precinct count))
          ->[Machine totals]
          ->0..1(*c)
      },
    ->(Gather absentee / remote votes)
      ->[Ballots and/or ballot images]
      ->(*c),
    ->(Prepare for voting (central))
      ->(Wrap up voting (central) *w)
  };
(*c)
  ->[Ballots, ballot images and/or machine totals]
  ->(Wrap up voting (precinct))
  ->[Ballots, ballot images and/or precinct totals]
  ->(Wrap up voting (central) *w)
  ->[Counts [certified]]
  ->(Wrap up election)
  -><*merge>.

Note (on Gather in-person vote):  Includes early voting.



Diagram:  Prepare for election

Output:  [Equipment, voter lists, ballot formats and/or ballots]
```

```
<InitialNode>
  -><ForkNode>{
    ->(Define precincts)
      ->[Precinct definitions]
      -><ForkNode>{
        ->(Train poll workers)
          -><FlowFinal>,
        ->(Register voters)
          ->[Voter lists]
          ->(Collect *c1),
        ->(Program election)
          ->[Election definition]
          ->(Prepare ballots)
          ->[Ballot formats]
          -><ForkNode>{
            ->(*c1),
            Centrally programmed ballot formats
              ->[Ballot formats]
              ->0..1(Configure & calibrate precinct equipment (central) *cc)
          }
      },
    ->(Maintain equipment in storage)
      ->[Equipment [old]]
      ->(*cc),
    Need new equipment
      ->(Procure equipment)
      ->[Equipment [new]]
      ->0..1(*cc)
  };
(*c1)
  ->[Voter lists, ballot formats]
  -><ForkNode>{
    ->(Educate / notify / inform voters)
      -><FlowFinal>,
    ->(Collect *c2),
    Paper ballots
      ->(Produce ballots)
      ->[Ballots]
      ->0..1(*c2)
  };
(*cc)
  ->[Equipment [configured]]
  ->(Test precinct equipment (central))
  ->[Equipment [tested]]
  ->(Transport equipment)
  ->[Equipment [deployed]]
  ->(Collect *c2)
  ->[Equipment, voter lists, ballot formats and/or ballots].
```

Note (on Define precincts): This activity refers to configuring the voting system to realize the precincts as defined by state law.

Diagram: Gather in-person vote (paper-based).

This diagram is divided to show which activities are done by the voter and which are done by the poll worker. The activity Spoil ballot may be done by either. Present credentials, Mark ballot, Review ballot, and Present / submit ballot are done by the voter. All others are done by the poll worker.

Note: This activity occurs once per voter.

Input: [Voter lists]
Output: [Ballot [accepted]]

```
[Voter lists]
  ->(Check identity of voter *check);
<InitialNode>
  ->(Present credentials)
  ->(Check identity of voter *check)
  ->(Check voter eligibility)
  -><MergeNode *merge>
  ->(Update poll book)
  ->(Issue ballot or provisional ballot)
  ->(Provide private voting station)
  ->[Ballot [blank]]
  ->(Mark ballot)
  -><DecisionNode>{
    Fled voter
      ->(Spoil ballot)
      -><ActivityFinal>,
    else
      ->(Review ballot)
      -><DecisionNode>{
        Not OK
          ->(Spoil ballot)
          -><*merge>,
        OK
          ->(Present / submit ballot)
          ->[Ballot [completed]]
          ->(Validate ballot)
          -><DecisionNode>{
            OK
              ->(Accept ballot)
              ->[Ballot [accepted]],
            Not OK
              -><DecisionNode>{
                Try again
                  -><*merge>,
                else
                  -><ActivityFinal>
              }
          }
      }
  }.
```

Diagram: Gather in-person vote (DRE).

This diagram is divided to show which activities are done by the voter and which are done by the poll worker. The activity Spoil ballot may be done by either. Present credentials, Mark ballot, Review ballot, and Cast ballot are done by the voter. All others are done by the poll worker.

Note: This activity occurs once per voter.

Input: [Voter lists]
Output: [Ballot image]

```
[Voter lists]
  ->(Check identity of voter *check);
<InitialNode>
  ->(Present credentials)
  ->(Check identity of voter *check)
  ->(Check voter eligibility)
  ->(Update poll book)
  ->(Provide private voting station)
  -><MergeNode *merge>
  ->(Mark ballot)
  -><DecisionNode>{
    Fled voter
      ->(Spoil ballot)
      -><ActivityFinal>,
    else
      ->(Review ballot)
      -><DecisionNode>{
        Not OK
          ->(Spoil ballot)
          -><*merge>,
        OK
          ->(Cast ballot)
          ->[Ballot image]
      }
  }.
```

Diagram: Wrap up voting (precinct)

Note: This activity occurs once per precinct. Absentee / remote ballots may be handled and processed as a separate precinct under this activity.

Input: [Ballots, ballot images and/or machine totals]

Outputs: [Reports], [Ballots, ballot images and/or precinct totals [validated]]

```
[Ballots, ballot images and/or machine totals]
  ->(Close polls (including absentee / remote voting)){
    ->[Reports],
    ->[Ballots, ballot images and/or precinct totals [unvalidated]]
    -><MergeNode *merge>
    ->(Validate counts (precinct))
    -><DecisionNode>{
      Invalid
        ->(Diagnose and correct problem (precinct))
        ->[Ballots, ballot images and/or precinct totals [corrected, unvalidated]]
        -><*merge>,
      else
        ->[Ballots, ballot images and/or precinct totals [validated]]
        ->(Deliver / transmit ballots, ballot images and/or precinct totals to central)
        ->[Ballots, ballot images and/or precinct totals [validated]]
    }
  }.
```

Diagram: Wrap up voting (central)

Input: [Ballots, ballot images and/or precinct totals [validated]]
Outputs: [Counts [certified]], [Reports [official]]

```
[Ballots, ballot images and/or precinct totals [validated]]
  -><MergeNode *merge1>
  ->(Count (central))
  ->[Counts [unvalidated]]
  -><MergeNode *merge2>
  ->(Validate counts (central))
  -><DecisionNode>{
    Invalid
      ->(Diagnose and correct problem (central))
      ->[Counts [corrected, unvalidated]]
      -><*merge2>,
    else
```

```
      ->[Counts [validated]]
      ->(Generate unofficial reports)
      ->[Reports [unofficial]]
      ->(Reconcile provisional/challenged ballots)
      ->[Counts [adjusted]]
      ->(Generate official reports)
      ->[Reports [official]]
      -><DecisionNode>{
        Recount
          ->(Retrieve original data)
          ->[Ballots, ballot images and/or precinct totals [validated]]
          -><*merge1>,
        else
          ->(Certify final counts){
            ->[Counts [certified]],
            ->[Reports [official]]
          }
      }
  }.
```

Note (on Count (central)): Including absentee and write-ins.

Diagram: Audit / observe elections

```
<InitialNode>{
  ->(Involve independent observers),
  ->(Conduct official audits),
  ->(Conduct personnel checks),
  ->(Conduct equipment checks),
  ->(Conduct procedural checks)
}.
```

Diagram: Prepare ballots

Note: Produce ballots is analogous.

Input: [Election definition]

```
Output:  [Ballot formats]

[Election definition]
  -><ForkNode>{
    ->(Define regular ballots)
      -><JoinNode *j>,
    ->(Define provisional ballots)
      -><*j>,
    ->(Define absentee / remote ballots)
      -><*j>
  };
<*j>
  ->[Ballot formats].


Diagram:  Procure equipment

Output:  [Equipment]

<InitialNode>
  ->(Specify requirements)
  ->(Select vendors and equipment)
  ->(Conduct certification testing)
  ->(Conduct acceptance testing)
  ->[Equipment].


Diagram:  Prepare for voting (precinct)

Note:  This activity occurs once per precinct.

Input:  [Equipment]
Output:  [Reports]

[Equipment]
  ->(Set up polling place)
  ->(Set up precinct equipment (precinct))
  ->(Configure & calibrate precinct equipment (precinct))
  ->(Test precinct equipment (precinct))
  ->(Open poll)
  ->[Reports].


Diagram:  Prepare for voting (central)

Input:  [Equipment]
Output:  [Reports]

[Equipment]
  ->(Set up central equipment (central))
  ->(Configure & calibrate central equipment (central))
  ->(Test central equipment (central))
  ->[Reports].


Diagram:  Register voters

Input:  [Registration database [original]]
Output:  [Voter lists]

[Registration database [original]]
  -><ForkNode>{
    ->(Register new voters)
      -><JoinNode *j>,
    ->(Update voter information)
      -><*j>,
    ->(Purge ineligible, inactive, or dead voters)
      -><*j>
  };
<*j>
  ->[Registration database [updated]]
  ->(Generate voter lists)
  ->[Voter lists].


Diagram:  Wrap up election

<InitialNode>
```

```
  -><ForkNode>{
    ->(Deactivate equipment)
      -><JoinNode *j>,
    ->(Conduct post-mortem)
      -><*j>
  };
<*j>
  -><ActivityFinal>.
```

Diagram: Top level

```
<InitialNode>
  -><ForkNode>{
    ->(Administer elections),
    ->(Audit / observe elections),
    ->(Archive)
  }.
```

Note (on Archive): All of the reports that are generated by various activities are archived.

Diagram: Deactivate equipment

```
<InitialNode>
  ->(Pack up equipment)
  ->(Transport equipment)
  ->(Put equipment in storage)
  -><ActivityFinal>.
```

Diagram: Conduct post-mortem

```
<InitialNode>
  ->(Analyze election results)
  ->[Lessons learned]
  ->(Refine needs and requirements)
  ->(Make revisions / changes to existing hardware, software, processes,
procedures, and testing)
  -><ActivityFinal>.
```

5.2 Vote-capture device state model (informative)

The state model shown in Figure 12 clarifies the relationship between the different equipment states that result from the opening and closing of polls and the suspension and resumption of voting in jurisdictions that allow early voting.

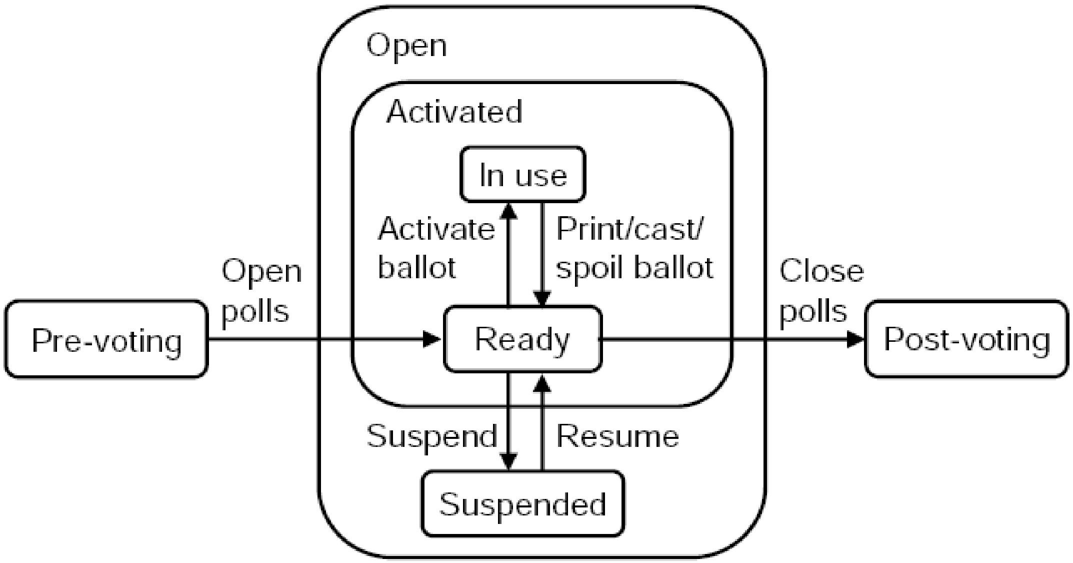


Figure 12 Vote-capture device states

The many steps that occur prior to the opening of polls are abstracted by the **Pre-voting** state. The many steps that occur after the close of polls are abstracted by the **Post-voting** state. Between these is a composite state **Open**, which contains the simple state **Suspended** and the composite state **Activated**. **Activated** in turn contains the simple states **Ready** and **In use**.

Upon the opening of polls, the vote-capture device transitions from the **Pre-voting** state to the **Ready** state (and, consequently, also to the **Open** and **Activated** composite states that contain it). From **Ready** it can transition to the **In use** state upon the activation of a ballot and return to the **Ready** state when that ballot is printed, cast or spoiled (the details depend on the technology in use). From **Ready** it can also transition to the **Suspended** state when an election official suspends voting and return to the **Ready** state when voting is resumed. Finally, from **Ready** it can transition to the **Post-voting** state when polls are closed.

In conformance with Requirement III.4.7-2.5, there is no transition from **Post-voting** back to **Open** except by beginning an entirely new election cycle, which is not modelled here.

A voting session lasts while the device is in the **In use** state. An active period lasts while the device is in the **Activated** state.

5.3 Logic model (normative)

This model defines the results that must appear in vote data reports and is used in verification of voting system logic. It does not address ranked order voting and does not attempt to define every voting variation that jurisdictions may use. It suffices for N of M (including 1 of M) and cumulative voting.

Approach for ranked order voting: IRV (instant runoff voting) is an iterative algorithm. If we want to verify IRV implementations, then we get a standard reference algorithm and we make them prove equivalence. But currently there is no standard reference IRV algorithm and no de facto standard. There are nearly as many algorithm variants as there are places that use IRV. Defer to 2010?

5.3.1 Domain of discourse

A noteworthy bound on the scope of the voting system, and hence the logic model, is that, as of the state of the practice in 2005, voting systems do not identify voters. Poll workers are responsible for maintaining the one voter, one ballot parity. The voting system is limited to handling ballots. Consequently, logic verification is limited to showing that those ballots are counted correctly.

Term	**Definition**
A(*t*,*v*)	Boolean function, returns true if and only if ballot *v* conforms to jurisdiction-dependent criteria for accepting or rejecting entire ballots, such as stray marks policies and voter eligibility criteria, as of time *t*. This value is false for provisional, challenged, and review-required ballots that are not [yet] validated. The system may not be able to determine the value of A(*t*,*v*) without human input; however, it may assign tentative values according to local procedures and state law, to be corrected later if necessary by input from election workers. The value of A(*t*,*v*) may change over time as a result of court decisions, registrar review of voter eligibility, etc. In a paper-based system, A(*t*,*v*) will be false if ballot *v* is unprocessable.
B(*v*)	The time at which ballot *v* is "started" (distributed to a voter, enabled or issued).
C(*r*,*t*)	The set of all candidates or choices for a contest *r*, including any write-ins appearing on ballots cast as of time *t*. In systems conforming to the *Write-ins* class, each distinct write-in candidate appears separately in C(*r*,*t*). Systems not conforming to the *Write-ins* class may nevertheless offer a ballot position for write-ins to be processed manually; in that case, C(*r*,*t*)

	contains a single entry representing all write-ins.
c, c_n, etc.	Individual candidates or choices.
D(v)	The time at which ballot v is "done" (either cast or spoiled).
J	The set of reporting contexts (including tabulators, precincts, election districts, and jurisdiction).
j, j_n, etc.	Individual reporting contexts.
K(j,r,t)	For a given contest and reporting context, the number of read ballots for which A(t,v) is true as of time t (i.e., the number of ballots that should be counted). Ballot formats that do not include contest r do not contribute to this total.
$\mathrm{L_B}$	A limit on the number of ballots or ballot images that a voting device is claimed to be capable of processing correctly.
$\mathrm{L_C}$	A limit on the number of ballot positions per contest that a voting device is claimed to be capable of processing correctly. (See also $\mathrm{L_W}$)
$\mathrm{L_F}$	A limit on the number of ballot formats that a voting device is claimed to be capable of processing correctly.
$\mathrm{L_P}$	For paper-based tabulators, a limit on the number of ballots per minute that the tabulator is claimed to be capable of tabulating correctly.
$\mathrm{L_R}$	A limit on the number of contests that a voting device is claimed to be capable of processing correctly.
$\mathrm{L_T}$	A numerical limit on vote totals that a voting device is claimed to be capable of processing correctly.
$\mathrm{L_V}$	A limit on the number of provisional, challenged, or review-required ballots that a voting device is claimed to be capable of processing correctly.
$\mathrm{L_W}$	A limit on the total number of distinct candidates or choices per contest, including write-ins, that a voting device is claimed to be capable of processing correctly. $L_W \geq L_C$. (See also $\mathrm{L_C}$)
N(r)	The maximum number of votes that may be cast by a given voter in contest r, pursuant to the definition of the contest. For N of M contests, this is the value N.
O(j,r,t)	For a given contest and reporting context, the number of overvotes in read ballots for which A(t,v) is true as of time t. Each ballot in which contest r is overvoted contributes N(r) to O(j,r,t).
R	The set of all contests.
r, r_n,	Individual contests in R.

etc.	
S(*c*,*r*,*t*,*v*)	Ballot *v*'s vote with respect to candidate or choice *c* in contest *r* as of time *t*. For checkboxes and the like, the value is 1 (selected) or 0 (not selected). For cumulative voting, the value is the number of votes that *v* gives to candidate or choice *c* in contest *r*. If the applicable ballot format does not include contest *r*, S(*c*,*r*,*t*,*v*) = 0.
S'(*c*,*r*,*t*,*v*)	Ballot *v*'s vote with respect to candidate or choice *c* in contest *r* as accepted for counting purposes (i.e., valid votes only), as of time *t*.
S(*r*,*t*,*v*)	The total number of votes that ballot *v* has in contest *r* as of time *t*. $S(r,t,v) = \sum_{c \in C(r,t)} S(c,r,t,v)$
T(*c*,*j*,*r*,*t*)	The vote total for candidate or choice *c* in contest *r* and reporting context *j* as of time *t*. This does not include votes that are invalid due to overvoting or votes from ballots for which A(*t*,*v*) is false.
t, t_n, etc.	Individual time points.
t_{O}	The time at which polls are opened.
t_{C}	The time at which polls are closed.
t_{E}	The time at which the value of A(*t*,*v*) is frozen for all ballots, the counting is complete, and final vote totals are required ("end").
U(*j*,*r*,*t*)	For a given contest and reporting context, the number of undervotes in read ballots for which A(*t*,*v*) is true as of time *t*. A given ballot contributes at most N(*r*) to U(*j*,*r*,*t*). Ballot formats that do not include contest *r* do not contribute to this total.
V(*j*,*t*)	The set of all ballots that have been "started" (distributed to voters, enabled or issued) within reporting context *j* by time *t*, including any that are presently being voted. Absentee ballots, provisional/challenged ballots, and review-required ballots are included in V if and only if the system claims conformance to the relevant classes. Ballots containing write-in votes may be included for systems not conforming to the *Write-ins* class if the system reports all write-in votes as a single ballot position. For more information on this exception see C(*r*,*t*) and Volume III Section 2.6.3.1.
v, v_n, etc.	Individual ballots in V(*j*,*t*).

Table 4 Terms used in logic verification

5.3.2 General assertions

Invariants:

$$t_O < t_C \le t_E$$

$$v \in V(j,t) \rightarrow B(v) \leq t$$

$$B(v) < D(v)$$

$$S(c,r,t,v) \geq 0$$

$$S'(c,r,t,v) \geq 0$$

The following assertions formalize several basic integrity constraints. Each textual assertion is intended to elucidate the formal assertion(s) that follow it. In case of discrepancy or confusion, the formal assertions are normative.

No one may vote before polls are opened or after polls have closed, or during the process of opening or closing the polls.

$$B(v) > t_O$$

$$D(v) < t_C$$

A ballot cannot have votes before it is voted.

$$t < B(v) \rightarrow S(r,t,v) = 0$$

A ballot cannot change once it is cast or spoiled.

$$t \geq D(v) \rightarrow S(c,r,t,v) = S(c,r,D(v),v)$$

5.3.3 Cumulative voting

All valid votes must be counted.[11]

$$t \geq t_E \wedge S(r,D(v),v) \leq N(r) \wedge A(t,v) \rightarrow S'(c,r,t,v) = S(c,r,D(v),v)$$

No invalid votes may be counted.

$$t \geq t_E \wedge \left(S(r,D(v),v) > N(r) \vee \overline{A(t,v)}\right) \rightarrow S'(c,r,t,v) = 0$$

The final vote totals must accurately reflect all valid votes and only valid votes.

$$t \geq t_E \rightarrow T(c,j,r,t) = \sum_{v \in V(j,t_E)} S'(c,r,t_E,v)$$

Every vote must be accounted for.

$$t \geq t_E \rightarrow \sum_{c \in C(r,t)} T(c,j,r,t) + O(j,r,t) + U(j,r,t) = K(j,r,t) \times N(r)$$

Note that all of the above assertions are predicated by $t \geq t_E$. No assertion has been made regarding the correctness of pre-final reports. Since the transmission and processing of vote data are not instantaneous, the correctness of a pre-final report can only be judged relative to some viewpoint (e.g., a central counting site, using whatever vote data they happen to have received and processed).

5.3.4 N of M contests (including 1-of-M)

N of M is identical to cumulative voting but for the addition of the following invariant, which reflects the design of a ballot format that allows only one vote in each ballot position (equivalent to a checkbox).

$S(c, r, t, v) \leq 1$

Volume IV Standards on data to be provided

Chapter 1 Technical data package (vendor)

1.1 Scope

This section contains a description of vendor documentation relating to the voting system that must be submitted with the system as a precondition of national certification testing. These items are necessary to define the product and its method of operation; to provide technical and test data supporting the vendor's claims of the system's functional capabilities and performance levels; and to document instructions and procedures governing system operation and field maintenance. Any other items relevant to the system evaluation, such as disks, tapes, source code, object code, and sample output report formats, must be submitted along with this documentation.

This documentation is used by the ITA in constructing the certification testing plan and is particularly important in constructing plans for the re-testing of systems that have been qualified previously. Re-testing of systems submitted by vendors that consistently adhere to particularly strong and well documented quality assurance and configuration management practices will generally be more efficient than for systems developed and maintained using less rigorous or less well documented practices.

Both formal documentation and notes of the vendor's system development process must be submitted for certification tests. Documentation describing the system development process permits assessment of the vendor's systematic efforts to develop and test the system and correct defects. Inspection of this process also enables the design of a more precise test plan. If the vendor's developmental test data are incomplete, the accredited test lab must design and conduct the appropriate tests to cover all elements of the system and to ensure conformance with all system requirements.

1.1.1 Content and format

The content of the Technical Data Package (TDP) is intended to provide clear, complete descriptions of the following information about the system:

a. Overall system design, including subsystems, modules and the interfaces among them;

b. Specific functional capabilities provided by the system;

c. Performance and design specifications;

d. Design constraints, applicable standards, and compatibility requirements;

e. Personnel, equipment, and facility requirements for system operation, maintenance, and logistical support;

f. Vendor practices for assuring system quality during the system's development and subsequent maintenance; and

g. Vendor practices for managing the configuration of the system during development and for modifications to the system throughout its life cycle.

1.1.1.1 Required content for initial certification

R 1.1-1 GeneratedID1896

The vendor shall submit to the ITA documentation necessary for the identification of the full system configuration submitted for evaluation and for the development of an appropriate test plan by the ITA for system certification testing.

Source: [2] I.9.2.

R 1.1-2 GeneratedID1899

The vendor shall provide a list of all documents submitted controlling the design, construction, operation, and maintenance of the system.

Source: [2] II.2.1.1.

Impact: Deleted subrequirement "in order of precedence" because nobody knew what it meant.

1.1-3 TDPContents

At minimum, the TDP shall contain the following documentation:

a. Implementation statement;

b. The voting equipment user documentation (Volume IV Chapter 2);

c. System hardware specifications;

d. Logic design and specifications;

e. System test and verification specifications;

f. Configuration management plan;

g. Quality assurance program;

h. System change notes; and

i. Configuration for testing.

Source: [2] II.2.1.1.1.

Impact: Added implementation statement, user documentation, configuration for testing; removed all that which was moved into the user documentation.

1.1.1.2 Required content for system changes and recertification

1.1-4 GeneratedID1916

For systems seeking recertification, vendors shall submit system change notes as described in Volume IV Section 1.8, as well as current versions of all documents that have been updated to reflect system changes.

Source: [2] II.2.1.1.2.

DISCUSSION

Vendors may also submit other information relevant to the evaluation of the system, such as test documentation, and records of the system's performance history, failure analysis and corrective actions.

1.1.1.3 Format

The requirements for formatting the TDP are general in nature; specific format details are of the vendor's choosing.

Ⓡ 1.1-5 GeneratedID1922

The TDP shall include a detailed table of contents for the required documents, an abstract of each document and a listing of each of the informational sections and appendices presented.

Source: [2] II.2.1.1.3.

Ⓡ 1.1-6 GeneratedID1925

A cross-index shall be provided indicating the portions of the documents that are responsive to documentation requirements enumerated in Requirement IV.1.1-3.

Source: [2] II.2.1.1.3.

1.1.2 Other uses for documentation

Although all of the TDP documentation is required for national certification testing, some of these same items may also be required during the state certification process and local level acceptance testing. Therefore, it is recommended that the technical documentation required for certification and acceptance testing be deposited in escrow.

1.1.3 Protection of proprietary information

Ⓡ 1.1-7 GeneratedID1931

The vendor shall identify all documents, or portions of documents, containing proprietary information not approved for public release.

Source: [2] II.2.1.3.

DISCUSSION

Any person or accredited test lab accepting proprietary information must agree to use it solely for the purpose of analyzing and testing the system, and must agree to refrain from otherwise using the proprietary information or disclosing it to any other person or agency without the prior written consent of the vendor, unless disclosure is legally compelled.

An accredited test lab may reject a Technical Data Package if it is so encumbered by intellectual property claims as to obstruct the lab's delivery of the Test Plan (Volume IV Chapter 3), Test Report (Volume IV Chapter 4) or Public Information Package (Volume IV Chapter 5).

An overuse of trade secret and patent protection may prevent certification by the EAC or by individual states. E.g., [12] 3.42: "The Vendor's entire proposal response package shall not be considered proprietary."

Votes are required to be counted in compliance with obvious algorithms that are published in state law and in these Guidelines. Consequently, opportunities for genuine trade secrets and upholdable patents are limited.

1.2 Implementation statement

Ⓡ 1.2-1 GeneratedID1938

The TDP shall include an implementation statement as defined in Volume III Section 2.5.

Source: New requirement.

DISCUSSION

Vendors may wish to contact their intended testing labs in advance to determine if those labs can supply them with an implementation statement *pro forma* to facilitate meeting this requirement.

1.3 System hardware specification

Ⓡ 1.3-1 GeneratedID1942

The vendor shall expand on the system overview included in the user documentation by providing detailed specifications of the hardware components of the system, including specifications of hardware used to support the telecommunications capabilities of the system, if applicable.

Source: [2] II.2.4.

1.3.1 System hardware characteristics

R 1.3-2 GeneratedID1946

The vendor shall provide a detailed discussion of the characteristics of the system, indicating how the hardware meets individual requirements defined in Volume III, including:

a. **Performance characteristics:** This discussion addresses basic system performance attributes and operational scenarios that describe the manner in which system functions are invoked, describe environmental capabilities, describe life expectancy, and describe any other essential aspects of system performance;

b. **Physical characteristics:** This discussion addresses suitability for intended use, requirements for transportation and storage, health and safety criteria, security criteria, and vulnerability to adverse environmental factors;

c. **Reliability:** This discussion addresses system and component reliability stated in terms of the system's operating functions, and identification of items that require special handling or operation to sustain system reliability;

d. **Maintainability:** Maintainability represents the ease with which maintenance actions can be performed based on the design characteristics of equipment and software and the processes the vendor and election officials have in place for preventing failures and for reacting to failures. Maintainability includes the ability of equipment and software to self-diagnose problems and make non-technical election workers aware of a problem. Maintainability also addresses a range of scheduled and unscheduled events; and

e. **Environmental conditions:** This discussion addresses the ability of the system to withstand natural environments, and operational constraints in normal and test environments, including all requirements and restrictions regarding electrical service, telecommunications services, environmental protection, and any additional facilities or resources required to install and operate the system.

Source: [2] II.2.4.1.

1.3.2 Design and construction

R 1.3-3 GeneratedID1955

The vendor shall provide sufficient data, or references to data, to identify unequivocally the details of the system configuration submitted for testing.

Source: [2] II.2.4.2.

Ⓡ 1.3-4 GeneratedID1958

The vendor shall provide a list of materials and components used in the system and a description of their assembly into major system components and the system as a whole.

Source: [2] II.2.4.2.

Ⓡ 1.3-5 GeneratedID1961

Text and diagrams shall be provided that describe:

> a. Materials, processes, and parts used in the system, their assembly, and the configuration control measures to ensure compliance with the system specification;
>
> b. The electromagnetic environment generated by the system;
>
> c. Operator and voter safety considerations, and any constraints on system operations or the use environment; and
>
> d. Human factors considerations, including provisions for access by disabled voters.

Source: [2] II.2.4.2.

1.4 Logic design and specification

The carryover requirements and informative text only say "software" or possibly "software and firmware." Adapt as appropriate to permit suitable review of vendor-specific logic implemented in software, firmware or hardware. See PREFACE (COTS issue).

The requirements of this section apply to all software or firmware installed or commissioned by the voting system vendor.

Ⓡ 1.4-1 GeneratedID1970

The vendor shall expand on the system overview included in the user documentation by providing detailed specifications of the software components of the system, including software used to support the telecommunications capabilities of the system, if applicable.

Source: [2] II.2.5.

1.4.1 Purpose and scope

Ⓡ 1.4-2 GeneratedID1974

The vendor shall describe the function or functions that are performed by the software programs that comprise the system, including software used to support the telecommunications capabilities of the system, if applicable.

Source: [2] II.2.5.1.

1.4.2 Applicable documents

Ⓡ 1.4-3 GeneratedID1978

The vendor shall list all documents controlling the development of the software and its specifications.

Source: [2] II.2.5.2.

Impact: Deleted subrequirement "in order of precedence" because nobody knew what it meant.

1.4.3 Software overview

Ⓡ 1.4-4 GeneratedID1983

The vendor shall provide an overview of the software.

Source: [2] II.2.5.3.

Ⓡ 1.4-4.1 GeneratedID1986

The software overview shall include a description of the software architecture, the design objectives, and the logic structure and algorithms used to accomplish those objectives.

Source: [2] II.2.5.3.a, reworded.

Ⓡ 1.4-4.2 GeneratedID1989

The software overview shall include the general design, operational considerations, and constraints influencing the design of the software.

Source: [2] II.2.5.3.b.

Ⓡ 1.4-4.3 GeneratedID1992

The software overview shall include the identification of all software items, indicating items that were:

a. Written in-house;

b. Procured and not modified; and

c. Procured and modified, including descriptions of the modifications to the software and to the default configuration options.

Source: [2] II.2.5.3.c.

Ⓡ 1.4-4.4 GeneratedID1998

The software overview shall include the following additional information for each item:

a. Item identification;

b. General description;

c. Software requirements performed by the item;

d. Identification of interfaces with other items that provide data to, or receive data from, the item; and

e. Concept of execution for the item.

Source: [2] II.2.5.3.d.

Ⓡ 1.4-4.5 GeneratedID2006

The software overview shall include a certification that procured software items were obtained directly from the manufacturer or a licensed dealer or distributor.

Source: [2] II.2.5.3.

1.4.4 Software standards and conventions

Ⓡ 1.4-5 GeneratedID2010

The vendor shall provide information that can be used by an accredited test lab or state certification board to support software analysis and test design. The information shall address standards and conventions developed internally by the vendor as well as published industry standards that have been applied by the vendor.

Source: [2] II.2.5.4.

Ⓡ 1.4-6 GeneratedID2013

The vendor shall provide information that addresses the following standards and conventions:

a. Software system development methodology;

b. Software design standards, including internal vendor procedures;

c. Software specification standards, including internal vendor procedures;

d. Software coding conventions, including internal vendor procedures;

e. Testing and verification standards, including internal vendor procedures, that can assist in determining the program's correctness and ACCEPT/REJECT criteria; and

f. Quality assurance standards or other documents that can be used to examine and test the software. These documents include standards for program flow and control charts, program documentation, test planning, and test data acquisition and reporting.

Source: [2] II.2.5.4.

Ⓡ 1.4-7 GeneratedID2022

The vendor shall furnish evidence that the selected coding conventions are "published" and "credible" as specified in Requirement III.3.5-2.

Source: New requirement.

1.4.5 Software operating environment

Ⓡ 1.4-8 GeneratedID2026

The vendor shall describe or make reference to all operating environment factors that influence the software design.

Source: [2] II.2.5.5.

1.4.5.1 Hardware environment and constraints

Ⓡ 1.4-9 GeneratedID2030

The vendor shall identify and describe the hardware characteristics that influence the design of the software, such as:

a. The logic and arithmetic capability of the processor;

b. Memory read-write characteristics;

c. External memory device characteristics;

d. Peripheral device interface hardware;

e. Data input/output device protocols; and

f. Operator controls, indicators, and displays.

Source: [2] II.2.5.5.1.

1.4.5.2 Software environment

Ⓡ 1.4-10 GeneratedID2040

The vendor shall identify the operating system and the specific version thereof.

Source: [2] II.2.5.5.2.

Ⓡ 1.4-11 GeneratedID2043

For systems containing compiled or assembled code, the vendor shall identify the compilers or assemblers used in the generation of executable code, and the specific versions thereof.

Source: [2] II.2.5.5.2.

DISCUSSION

Although compiled code should not be very sensitive to the versioning of the compiler, this information should be documented in case complications arise.

R 1.4-12 GeneratedID2047

For systems containing interpreted code, the vendor shall specify the industry standard runtime interpreter that shall be used to run this code, and the specific version thereof.

Source: New requirement.

DISCUSSION

See Requirement III.3.2-4.

1.4.6 Software functional specification

R 1.4-13 GeneratedID2052

The vendor shall provide a description of the operating modes of the system and of software capabilities to perform specific functions.

Source: [2] II.2.5.6.

1.4.6.1 Configurations and operating modes

R 1.4-14 GeneratedID2056

The vendor shall describe all software configurations and operating modes of the system, such as ballot preparation, election programming, preparation for opening the polls, recording votes and/or counting ballots, closing the polls, and generating reports.

Source: [2] II.2.5.6.1.

R 1.4-15 GeneratedID2059

For each software function or operating mode, the vendor shall provide:

a. A definition of the inputs to the function or mode (with characteristics, tolerances or acceptable ranges, as applicable);

b. An explanation of how the inputs are processed; and

c. A definition of the outputs produced (again, with characteristics, tolerances, or acceptable ranges, as applicable).

Source: [2] II.2.5.6.1.

FIXME: Narrow down what is meant by function here so that we don't get all this for every callable unit!

1.4.6.2 Software functions

Ⓡ 1.4-16 GeneratedID2067

The vendor shall describe the software's capabilities or methods for detecting or handling:

a. Exception conditions;

b. System failures;

c. Data input/output errors;

d. Error logging for audit record generation;

e. Production of statistical ballot data;

f. Data quality assessment; and

g. Security monitoring and control.

Source: [2] II.2.5.6.2.

1.4.7 Programming specifications

Ⓡ 1.4-17 GeneratedID2078

The vendor shall provide in this section an overview of the software design, its structure, and implementation algorithms and detailed specifications for individual software modules.

Source: [2] II.2.5.7.

1.4.7.1 Programming specifications overview

Ⓡ 1.4-18 GeneratedID2082

The programming specifications overview shall document the architecture of the software.

Source: Summary of [2] II.2.5.7.1.

Ⓡ 1.4-18.1 GeneratedID2085

This overview shall include such items as UML diagrams, flowcharts, data flow diagrams, and/or other graphical techniques that facilitate understanding of the programming specifications.

Source: [2] II.2.5.7.1.

Ⓡ 1.4-18.2 GeneratedID2088

This section shall be prepared to facilitate understanding of the internal functioning of the individual software modules.

Source: [2] II.2.5.7.1.

Ⓡ 1.4-18.3 GeneratedID2091

Implementation of the functions shall be described in terms of the software architecture, algorithms, and data structures.

Source: [2] II.2.5.7.1.

1.4.7.2 Programming specifications details

Ⓡ 1.4-19 GeneratedID2095

The programming specifications shall describe individual software modules and their component units, if applicable.

Source: [2] II.2.5.7.2.

Ⓡ 1.4-20 GeneratedID2098

For each module and callable unit, the vendor shall document:

a. Significant module and unit design decisions, if any, such as algorithms used;

b. Any constraints, limitations, or unusual features in the design of the software module or callable unit;

c. A description of its inputs, outputs, and other data elements as applicable with respect to communication over system interfaces (see Volume IV Section 1.4.9).

Source: [2] II.2.5.7.2.a, b, and e.

Impact: Deleted subrequirement f ("If the software module or unit contains logic...") and g ("If the software module is a database..."). Both are apparently redundant, though it is less clear for f, which is strangely written.

1.4-21 GeneratedID2105

If a module is written in a programming language other than that generally used within the system, the specification for the module shall indicate the programming language used and the reason for the difference.

Source: [2] II.2.5.7.2.c.

1.4-22 GeneratedID2108

If a module contains embedded commands for an external library or package (such as menu selections in a database management system for defining forms and reports, on-line queries for database access and manipulation, input to a graphical user interface builder for automated code generation, commands to the operating system, or shell scripts), the specification for the module shall contain a reference to user manuals or other documents that explain them.

Source: [2] II.2.5.7.2.d.

Impact: Removed requirement to list the commands. Should be obvious from the sources.

1.4-23 Callable unit specifications

For each callable unit (function, method, operation, subroutine, procedure, etc.) in source code or analogous logic design, the vendor shall specify:

a. The source code, for systems using software; analogous formal logic designs, for systems not using software;

b. The preconditions and postconditions, formally stated using the terms defined in Volume III Section 5.3.1, including any assumptions about capacities and limits within which the system is expected to operate;[12] and

c. A convincing argument (possibly, but not necessarily, a formal proof) that the preconditions and postconditions accurately represent the behavior of the callable unit.[13]

Source: a is from [2] II.2.1; b and c are new requirements.

DISCUSSION

WUUP software is not subject to code examination and therefore need not be provided in source form to the test lab. However, source code generated by a WUUP package and embedded in software modules for compilation or interpretation must be provided in human readable form to the test lab. The test lab may inspect these units to determine testing requirements or to verify that the code is unmodified and that the default configuration options have not been changed.

See PREFACE.

It is at the test lab's discretion to accept an argument as convincing. Thus, in cases where the relationship between preconditions and postconditions and the behavior of the callable unit is completely obvious or trivial, it may suffice to state as much.

Postconditions that impact something outside the domain of discourse are not of interest unless that thing impacts the behavior of some function with respect to the domain of discourse. The vendor must define such terms as are necessary to state any and all dependencies and assumptions that may impact the behavior of some function with respect to the domain of discourse and use them consistently in all affected preconditions and postconditions. *An excess of extraneous dependencies may negatively impact the test lab's ability to verify the system's correctness and thereby prevent certification.*

A callable unit might have no impact on anything in the domain of discourse and no dependency on anything in the domain of discourse. Such a unit must have a true precondition and a postcondition that states that nothing in the domain of discourse is changed.

1.4-24 assertionsproof

The vendor shall specify a formal proof, using the preconditions and postconditions, that the software or logic design as a whole satisfies each of the invariants and assertions indicated in Volume III Section 5.3 for the classes claimed in the implementation statement, for all cases within the aforementioned capacities and limits.

Source: New requirement.

1.4-25 TDP, justify long units

The vendor shall justify any callable unit lengths that violate Requirement III.3.5-4.1.

Source: [2] II.5.4.2.i.

1.4.8   System database

1.4-26 GeneratedID2125

The vendor shall identify and provide a diagram and narrative description of the system's databases and any external files used for data input or output.

Source: [2] II.2.5.8.

1.4-27 GeneratedID2128

For each database or external file, the vendor shall specify number of levels of design and the names of those levels (such as conceptual, internal, logical, and physical).

Source: [2] II.2.5.8.a.

1.4-28 GeneratedID2131

For each database or external file, the vendor shall specify any design conventions and standards (which may be incorporated by reference) needed to understand the design.

Source: [2] II.2.5.8.b.

1.4-29 GeneratedID2134

For each database or external file, the vendor shall identify and describe all logical entities and relationships and how these are implemented physically (e.g., tables, files).

Source: [2] II.2.5.8.c and d.

DISCUSSION

This requirement calls for a data model but the modelling language is no longer mandated.

1.4-30 GeneratedID2138

The vendor shall document the details of table, record or file contents (as applicable), individual data elements and their specifications, including:

a. Names/identifiers;

b. Data type (alphanumeric, integer, etc.);

c. Size and format (such as length and punctuation of a character string);

d. Units of measurement (such as meters, dollars, nanoseconds);

e. Range or enumeration of possible values (such as 0-99);

f. Accuracy (how correct) and precision (number of significant digits);

g. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply;

h. Security and privacy constraints; and

i. Sources (setting/sending entities) and recipients (using/receiving entities).

Source: [2] II.2.5.8.e.

DISCUSSION

The majority of this requirement may be satisfied by supplying the source of the database schema if it is in a widely recognized and standardized language.

Ⓡ 1.4-31 GeneratedID2151

For external files, vendors shall document the procedures for file maintenance, management of access privileges, and security.

Source: [2] II.2.5.8.f.

1.4.9 Interfaces

Ⓡ 1.4-32 GeneratedID2154

The vendor shall identify and provide a complete description of all internal and external interfaces, using a combination of text and diagrams.

Source: [2] II.2.5.9.

FIXME: Narrow down what is meant by interface here so that we don't get all this for every callable unit!

1.4.9.1 Interface identification

Ⓡ 1.4-33 GeneratedID2159

For each interface identified in the system overview, the vendor shall:

a. Provide a unique identifier assigned to the interface;

b. Identify the interfacing entities (systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable; and

c. Identify which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them).

Source: [2] II.2.5.9.1.

1.4.9.2 Interface description

1.4-34 GeneratedID2166

For each interface identified in the system overview, the vendor shall describe the type of interface (such as real-time data transfer, storage-and-retrieval of data) to be implemented.

Source: [2] II.2.5.9.2.a.

1.4-35 GeneratedID2169

For each interface identified in the system overview, the vendor shall describe characteristics of individual data elements that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:

a. Names/identifiers;

b. Data type (alphanumeric, integer, etc.);

c. Size and format (such as length and punctuation of a character string);

d. Units of measurement (such as meters, dollars, nanoseconds);

e. Range or enumeration of possible values (such as 0-99);

f. Accuracy (how correct) and precision (number of significant digits);

g. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply;

h. Security and privacy constraints; and

i. Sources (setting/sending entities) and recipients (using/receiving entities).

Source: [2] II.2.5.9.2.b.

R 1.4-36 GeneratedID2181

For each interface identified in the system overview, the vendor shall describe characteristics of communication methods that the interfacing entity(ies) will use for the interface, such as:

a. Communication links/bands/frequencies/media and their characteristics;

b. Message formatting;

c. Flow control (such as sequence numbering and buffer allocation);

d. Data transfer rate, whether periodic/aperiodic, and interval between transfers;

e. Routing, addressing, and naming conventions;

f. Transmission services, including priority and grade; and

g. Safety/security/privacy considerations, such as encryption, user authentication, compartmentalization, and auditing.

Source: [2] II.2.5.9.2.c.

R 1.4-37 GeneratedID2191

For each interface identified in the system overview, the vendor shall describe characteristics of protocols the interfacing entity(ies) will use for the interface, such as:

a. Priority/layer of the protocol;

b. Packeting, including fragmentation and reassembly, routing, and addressing;

c. Legality checks, error control, and recovery procedures;

d. Synchronization, including connection establishment, maintenance, termination; and

e. Status, identification, and any other reporting features.

Source: [2] II.2.5.9.2.d.

FIXME: Obsolescent viewpoint. Use standard protocols. See AG spreadsheet.

Ⓡ 1.4-38 GeneratedID2200

For each interface identified in the system overview, the vendor shall describe any other pertinent characteristics, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.).

Source: [2] II.2.5.9.2.e.

1.4.10   Appendices

The vendor may provide descriptive material and data supplementing the various sections of the body of the software specifications. The content and arrangement of appendices are at the discretion of the vendor. Topics recommended for amplification or treatment in appendix form include:

a. **Glossary:** A listing and brief definition of all software module names and variable names, with reference to their locations in the software structure. Abbreviations, acronyms, and terms should be included, if they are either uncommon in data processing and software development or are used in an unorthodox semantic;

b. **References:** A list of references to all related vendor documents, data, standards, and technical sources used in software development and testing; and

c. **Program Analysis:** The results of software configuration analysis algorithm analysis and selection, timing studies, and hardware interface studies that are reflected in the final software design and coding.

1.5   System test and verification specification

Ⓡ 1.5-1 GeneratedID2208

The vendor shall provide test and verification specifications for:

a. Development test specifications; and

b. National certification test specifications.

Source: [2] II.2.7.

1.5.1   Development test specifications

Ⓡ 1.5-2 GeneratedID2214

The vendor shall describe the plans, procedures, and data used during software development and system integration to verify system logic correctness, data quality, and security. This description shall include:

a. Test identification and design, including test structure, test sequence or progression, and test conditions;

b. Standard test procedures, including any assumptions or constraints;

c. Special purpose test procedures including any assumptions or constraints;

d. Test data, including the data source, whether it is real or simulated, and how test data are controlled;

e. Expected test results; and

f. Criteria for evaluating test results.

Source: [2] II.2.7.1.

DISCUSSION

The vendor's test cases may help to speed up the open-ended testing portions of certification testing (Volume V Section 4.4 and Volume V Section 4.5).

Additional details for these requirements are provided by MIL-STD-498, Software Test Plan and Software Test Description. LOOK THIS UP.

1.5.2 National certification test specifications

Ⓡ 1.5-3 GeneratedID2226

The vendor shall provide usability test reports in Common Industry Format (CIF).

Source: New requirement.

Ⓡ 1.5-4 GeneratedID2229

The vendor shall provide specifications for verification and validation of overall software performance. These specifications shall cover:

a. Control and data input/output;

b. Acceptance criteria;

c. Processing accuracy;

d. Data quality assessment and maintenance;

e. Ballot interpretation logic;

f. Exception handling;

g. Security; and

h. Production of audit trails and statistical data.

Source: [2] II.2.7.2.

DISCUSSION

The vendor's test cases may help to speed up the open-ended testing portions of certification testing (Volume V Section 4.4 and Volume V Section 4.5).

Ⓡ 1.5-5 GeneratedID2241

The specifications shall identify procedures for assessing and demonstrating the suitability of the software for election use.

Source: [2] II.2.7.2.

1.6 Configuration management plan

This section to be rewritten by AG.

[2] I.8, to be provided by AG.

Make cross-references in this section more precise after AG section is done.

Vendors shall submit a Configuration Management Plan that addresses the configuration management requirements of Volume III Section 3.5.1.8. This plan shall describe all policies, processes, and procedures employed by the vendor to carry out these requirements. Information submitted by the vendor shall be used by the accredited test lab to assist in developing and executing the system certification test plan. This information is particularly important to support the design of test plans for system modifications. A well-organized, robust and detailed Configuration Management Plan will enable the accredited test lab to more readily determine the nature and scope of tests needed to fully test the modifications. The Configuration Management Plan shall contain the sections identified below.

1.6.1 Configuration management policy

The vendor shall provide a description of its organizational policies for configuration management, addressing the specific requirements of Volume III Section 3.5.1.8. These requirements pertain to:

a. Scope and nature of configuration management program activities; and

b. Breadth of application of vendor's policy and practices to the voting system.

1.6.2 Configuration identification

The vendor shall provide a description of the procedures and naming conventions used to address the specific requirements of Volume III Section 3.5.1.8. These requirements pertain to:

a. Classifying configuration items into categories and subcategories;

b. Uniquely numbering or otherwise identifying configuration items; and

c. Naming configuration items.

1.6.3 Baseline and promotion

The vendor shall provide a description of the procedures and naming conventions used to address the specific requirements of Volume III Section 3.5.1.8. These requirements pertain to:

a. Establishing a particular instance of a system component as the starting baseline;

b. Promoting subsequent instances of a component to baseline throughout the system development process for the first complete version of the system submitted for testing; and

c. Promoting subsequent instances of a component to baseline status as the component is maintained throughout its life cycle until system retirement (i.e., the system is no longer sold or maintained).

1.6.4 Configuration control procedures

The vendor shall provide a description of the procedures used by the vendor to approve and implement changes to a configuration item to prevent unauthorized additions, changes, or deletions to address the specific requirements of Volume III Section 3.5.1.8. These requirements pertain to:

a. Developing and maintaining internally developed items;

b. Developing and maintaining third party items;

c. Resolving internally identified defects; and

d. Resolving externally identified and reported defects.

1.6.5 Release process

The vendor shall provide a description of the contents of a system release, and the procedures and related conventions by which the vendor installs, transfers, or migrates the system to accredited voting system testing laboratories and customers to address the specific requirements of Volume III Section 3.5.1.8. These requirements pertain to:

a. A first release of the system to an accredited test lab;

b. A subsequent maintenance or upgrade release of a system, or particular components, to an accredited test lab;

c. The initial delivery and installation of the system to a customer; and

d. A subsequent maintenance or upgrade release of a system, or particular components, to a customer.

1.6.6 Configuration audits

The vendor shall provide a description of the procedures and related conventions for the two audits required by Volume III Section 3.5.1.8. These requirements pertain to:

a. Physical configuration audit that verifies the voting system components submitted for certification testing to the vendor's technical documentation; and

b. Functional configuration audit that verifies the system performs all the functions described in the system documentation.

1.6.7 Configuration management resources

The vendor shall provide a description of the procedures and related conventions for maintaining information about configuration management tools required by Volume III Section 3.5.1.8. These requirements pertain to information regarding:

a. Specific tools used, current version, and operating environment;

b. Physical location of the tools, including designation of computer directories and files; and

c. Procedures and training materials for using the tools.

1.7 Quality assurance program

This section to be rewritten by AG.

[2] I.7, to be provided by AG.

[2] I.7.7 is a duplicate spec for the contents of the whole TDP.

Make cross-references in this section more precise after AG section is done.

Vendors shall submit a Quality Assurance Program that addresses the quality assurance requirements of Volume III Section 3.5.1.7. This plan shall describe all policies, processes, and procedures employed by the vendor to ensure the overall quality of the system for its initial development and release and for subsequent modifications and releases. This information is particularly important to support the design of test plans by the accredited test lab. A well-organized, robust and detailed Quality Assurance Program will enable the accredited test lab to more readily determine the nature and scope of tests needed to test the system appropriately. The Quality Assurance Program shall, at a minimum, address the topics indicated below.

1.7.1 Quality assurance policy

The vendor shall provide a description of its organizational policies for quality assurance, including:

a. Scope and nature of Quality Assurance activities; and

b. Breadth of application of vendor's policy and practices to the voting system.

1.7.2 Parts and materials special tests and examinations

The vendor shall provide a description of its practices for parts and materials tests and examinations that meet the requirements of Volume III Section 3.5.1.7.

1.7.3 Quality conformance inspections

The vendor shall provide a description of its practices for quality conformance inspections that meet the requirements of Volume III Section 3.5.1.7. For each test performed, the record of tests provided shall include:

a. Test location;

b. Test date;

c. Individual who conducted the test; and

d. Test outcomes.

1.7.4 Documentation

The vendor shall provide a description of its practices for documentation of the system and system development process that meet the requirements of Volume III Section 3.5.1.7.

1.8 System change notes

1.8-1 GeneratedID2302

Vendors submitting modifications for a system that has been tested previously and received national certification shall submit system change notes.

Source: [2] II.2.13.

DISCUSSION

These will be used by the accredited test lab to assist in developing and executing the test plan for the modified system.

1.8-2 GeneratedID2306

The system change notes shall include the following information:

a. Summary description of the nature and scope of the changes, and reasons for each change;

b. A listing of the specific changes made, citing the specific system configuration items changed and providing detailed references to the documentation sections changed;

c. The specific sections of the documentation that are changed (or completely revised documents, if more suitable to address a large number of changes); and

d. Documentation of the test plan and procedures executed by the vendor for testing the individual changes and the system as a whole, and records of test results.

Source: [2] II.2.13.

1.9 Configuration for testing

Configuration of software, both operating systems and applications, is critical to proper system functioning. Correct test design and sufficient test execution must account for the intended and proper configuration of all system components.

R 1.9-1 GeneratedID2314

Vendors shall submit to the test lab, in the TDP, a record of all user selections made during software installation.

Source: [2] I.4.1.1.

R 1.9-2 GeneratedID2317

The vendor shall also submit a record of all configuration changes made to the software following its installation.

Source: [2] I.4.1.1.

Chapter 2 Voting equipment user documentation (vendor)

This section contains requirements on the content of the documentation that vendors supply to jurisdictions that use their systems. The user documentation is also included in the TDP given to test labs.

It is not the intent of these requirements to prescribe an outline for user documentation. Vendors are encouraged to innovate in the quality and clarity of their user documentation. The intent of these requirements is to ensure that certain information that is of interest to end users and test labs alike will be included somewhere in the user documentation. To speed the test lab review, vendors should provide test labs with a short index that points out which sections of the user documentation are responsive to which sections of these requirements.

2.1 System overview

Ⓡ 2.1-1 GeneratedID2322

In the system overview, the vendor shall provide information that enables the user to identify the functional and physical components of the system, how the components are structured, and the interfaces between them.

Source: [2] II.2.2.

2.1.1 System description

Ⓡ 2.1-2 GeneratedID2326

The system description shall include written descriptions, drawings and diagrams that present:

a. A description of the functional components (or subsystems) as defined by the vendor (e.g., environment, election management and control, vote recording, vote conversion, reporting, and their logical relationships);

b. A description of the operational environment of the system that provides an overview of the hardware, software, and communications structure;

c. A concept of operations that explains each system function, and how the function is achieved in the design;

d. Descriptions of the functional and physical interfaces between subsystems and components;

e. Identification of all WUUP hardware and software products and communications services used in the development and/or operation of the voting system, identifying the name, vendor, and version used for each such component, including operating systems, database software, communications routers, and modem drivers;

f. Dial-up networking software;

g. Interfaces among internal components and interfaces with external systems. For components that interface with other components for which multiple products may be used, the vendor shall identify file specifications, data objects, or other means used for information exchange, and the public standard used for such file specifications, data objects, or other means; and

h. Benchmark directory listings for all software (including firmware elements) and associated documentation included in the vendor's release in the order in which each piece of software would normally be installed upon system setup and installation.

Source: [2] II.2.2.1.

2.1.2 System performance

FIXME: Harmonize with limits specified elsewhere, clarify system vs. devices.

Ⓡ 2.1-3 GeneratedID2339

The vendor shall provide system performance information including:

a. The performance characteristics of each operating mode and function in terms of expected and maximum speed, throughput capacity, maximum volume (maximum number of voting positions and maximum number of ballot formats supported), and processing frequency;

b. Quality attributes such as reliability, maintainability, availability, usability, and portability;

c. Provisions for safety, security, privacy, and continuity of operation; and

d. Design constraints, applicable standards, and compatibility requirements.

Source: [2] II.2.2.2.

Ⓡ 2.1-3.1 GeneratedID2346

The capacity for a central tabulator shall be documented by the vendor. This documentation shall include the capacity for individual components that impact the overall capacity.

Applies to: Central tabulator

Source: [2] I.3.2.5.1.1.

DISCUSSION

The capacity to convert the punches or marks on individual ballots into signals is uniquely important to central count systems.

2.2 System functionality description

Ⓡ 2.2-1 GeneratedID2351

The vendor shall provide a listing of the system's functional processing capabilities, encompassing capabilities required by the Guidelines and any additional capabilities provided by the system, with a simple description of each capability.

a. The vendor shall explain, in a manner that is understandable to users, the capabilities of the system that were declared in the implementation statement;

b. Additional capabilities (extensions) shall be clearly indicated;

c. Required capabilities that may be bypassed or deactivated during installation or operation by the user shall be clearly indicated;

d. Additional capabilities that function only when activated during installation or operation by the user shall be clearly indicated; and

e. Additional capabilities that normally are active but may be bypassed or deactivated during installation or operation by the user shall be clearly indicated.

Source: [2] II.2.3.

Impact: Removed redundancy with implementation statement.

2.3 System security specification

This section to be rewritten by STS.

Threat analysis goes here.

Make cross-references in this section more precise after STS section is done.

Vendors shall submit a system security specification that addresses the security requirements of Volume III Section 3.2. This specification shall describe the level of security provided by the system in terms of the specific security risks addressed by the system, the means by which each risk is addressed, the process used to test and verify the effective operation of security capabilities and, for systems that use public telecommunications networks as defined in Volume III Section 3.2, the means used to keep the security capabilities of the system current to respond to the evolving threats against these systems.

Information provided by the vendor in this section of the TDP may be duplicative of information required by other sections. Vendors may cross reference to information provided in other sections provided that the means used provides a clear mapping to the requirements of this section.

Information submitted by the vendor shall be used to assist in developing and executing the system certification test plan. The Security Specification shall contain the sections identified below.

2.3.1 Access control policy

See new access control draft from STS.

The vendor shall specify the features and capabilities of the access control policy recommended to purchasing jurisdictions to provide effective voting system security. The access control policy shall address the general features and capabilities and individual access privileges indicated in Volume III Section 3.2.

2.3.2 Access control measures

See new access control draft from STS.

The vendor shall provide a detailed description of all system access control measures and mandatory procedures designed to permit access to system states in accordance with the access policy, and to prevent all other types of access to meet the specific requirements of Volume III Section 3.2.

The vendor also shall define and provide a detailed description of the methods used to preclude unauthorized access to the access control capabilities of the system itself.

2.3.3 Equipment and data security

The vendor shall provide a detailed description of system capabilities and mandatory procedures for purchasing jurisdictions to prevent disruption of the voting process and corruption of voting data to meet the specific requirements of Volume III Section 3.2. This information shall address measures for polling place security and central count location security.

STS should fill this in, and/or insert references to relevant requirements in STS sections. [2] I.6, et al.

2.3-1 SpecLocks

The user documentation shall contain specifications for the locks and/or seals on ballot boxes and ballot transfer boxes.

Source: [2] I.3.2.4.2.6.b.

Applies to: Paper-based device

2.3-2 GeneratedID2379

For systems containing paper-based tabulators, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for handling ballot boxes.

Source: Reworded from [2] I.6.3.2

Applies to: Paper-based device ^ *Tabulator*

Test reference: Volume V Section 2.1

2.3-3 GeneratedID2384

For systems containing paper-based tabulators, the Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for preparing ballots for counting.

Source: Reworded from [2] I.6.3.2

Applies to: Paper-based device ^ *Tabulator*

Test reference: Volume V Section 2.1

2.3-4 GeneratedID2389

The Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for counting operations.

Source: Reworded and generalized from [2] I.6.3.2

Test reference: Volume V Section 2.1

Ⓡ 2.3-5 GeneratedID2393

The Voting Equipment User Documentation shall detail the measures to be taken related to the physical and procedural controls for reporting data.

Source: Reworded and generalized from [2] I.6.3.2

Test reference: Volume V Section 2.1

Ⓡ 2.3-6 DontLeaveRecordsInSun

The Voting Equipment User Documentation shall detail the care and handling precautions necessary for removable media and records to satisfy Requirement III.3.6-1.

Source: Added precision.

Test reference: Volume V Section 2.1

2.3.4 Software installation

The vendor shall provide a detailed description of the system capabilities and mandatory procedures for purchasing jurisdictions to ensure secure software (including firmware) installation to meet the specific requirements of Volume III Section 3.2. This information shall address software installation for all system components.

2.3.5 Telecommunications and data transmission security

The vendor shall provide a detailed description of the system capabilities and mandatory procedures for purchasing jurisdictions to ensure secure data transmission to meet the specific requirements of Volume III Section 3.2:

a. For all systems, this information shall address access control, and prevention of data interception; and

b. For systems that use public communications networks as defined in Volume III Section 3.2, this information shall also include:

1. Capabilities used to provide protection against threats to third party products and services;
2. Policies and processes used by the vendor to ensure that such protection is updated to remain effective over time;
3. Policies and procedures used by the vendor to ensure that current versions of such capabilities are distributed to user jurisdictions and are installed effectively by the

jurisdiction;

4. A detailed description of the system capabilities and procedures to be employed by the jurisdiction to diagnose the occurrence of a denial of service attack, to use an alternate method of voting, to determine when it is appropriate to resume voting over the network, and to consolidate votes cast using the alternate method;
5. A detailed description of all activities to be performed in setting up the system for operation that are mandatory to ensure effective system security, including testing of security before an election; and
6. A detailed description of all activities that should be prohibited during system setup and during the timeframe for voting operations, including both the hours when polls are open and when polls are closed.

2.3.6 Other elements of an effective security program

The vendor shall provide a detailed description of the following additional procedures required for use by the purchasing jurisdiction:

a. Administrative and management controls for the voting system and election management, including access controls;

b. Internal security procedures, including operating procedures for maintaining the security of the software for each system function and operating mode;

c. Adherence to, and enforcement of, operational procedures (e.g., effective password management);

d. Physical facilities and arrangements; and

e. Organizational responsibilities and personnel screening.

This documentation shall be prepared such that these requirements can be integrated by the jurisdiction into local administrative and operating procedures.

2.4 System operations manual

Ⓡ 2.4-1 GeneratedID2419

The system operations manual shall provide all information necessary for system use by all personnel who support pre-election and election preparation, polling place activities and central counting activities, as applicable, with regard to all system functions and operations identified in Volume IV Section 2.2.

Source: [2] II.2.8.

DISCUSSION

The nature of the instructions for operating personnel will depend upon the overall system design and required skill level of system operations support personnel.

Ⓡ 2.4-2 Operations manual, support training

The system operations manual shall contain all information that is required for the preparation of detailed system operating procedures and for the training of administrators, central election officials, election judges and poll workers.

Source: [2] II.2.8.

2.4.1 Introduction

Ⓡ 2.4-3 GeneratedID2426

The vendor shall provide a summary of system operating functions and modes in sufficient detail to permit understanding of the system's capabilities and constraints.

Source: [2] II.2.8.1.

Ⓡ 2.4-4 GeneratedID2429

The roles of operating personnel shall be identified and related to the operating modes of the system.

Source: [2] II.2.8.1.

Ⓡ 2.4-5 GeneratedID2432

Decision criteria and conditional operator functions (such as error and failure recovery actions) shall be described.

Source: [2] II.2.8.1.

Ⓡ 2.4-6 GeneratedID2435

The vendor shall also list all reference and supporting documents pertaining to the use of the system during election operations.

Source: [2] II.2.8.1.

2.4.2 Operational environment

2.4-7 GeneratedID2439

The vendor shall describe the system environment and the interface between the user or operator and the system.

Source: [2] II.2.8.2.

2.4-8 GeneratedID2442

The vendor shall identify all facilities, furnishings, fixtures, and utilities that will be required for equipment operations, including equipment that operates at the:

a. Polling place;

b. Central count facility; and

c. Other locations.

Source: [2] II.2.8.2.

2.4-9 GeneratedID2448

The user documentation supplied by the vendor shall include a statement of all requirements and restrictions regarding environmental protection, electrical service, recommended auxiliary power, telecommunications service, and any other facility or resource required for the proper installation and operation of the system.

Source: [2] I.3.2.2.

2.4.3 System installation and test specification

2.4-10 GeneratedID2452

The vendor shall provide specifications for validation of system installation and readiness.

Source: [2] II.2.8.3.

2.4-10.1 GeneratedID2455

These specifications shall address all components of the system and all locations of installation (e.g., polling place, central count facility), and shall address all elements of system functionality and operations identified in Volume IV Section 2.2 above, including general capabilities and functions specific to particular voting activities.

Source: [2] II.2.8.3.

Impact: Removed references to acceptance testing (out of scope).

2.4.4 Operational features

2.4-11 GeneratedID2460

The vendor shall provide documentation of system operating features that includes:

a. A detailed description of all input, output, control, and display features accessible to the operator or voter;

b. Examples of simulated interactions to facilitate understanding of the system and its capabilities;

c. Sample data formats and output reports; and

d. Illustration and description of all status indicators and information messages.

Source: [2] II.2.8.4.

2.4-12 Document scratch vote algorithms

For systems that support straight party voting, the vendor shall document the available algorithms for counting scratch votes.

Source: New requirement.

Applies to: Straight party voting

DISCUSSION

See Requirement III.4.8-4.11.

2.4.5 Operating procedures

2.4-13 GeneratedID2472

The vendor shall provide documentation of system operating procedures that:

a. Provides a detailed description of procedures required to initiate, control, and verify proper system operation;

b. Provides procedures that clearly enable the operator to assess the correct flow of system functions (as evidenced by system-generated status and information messages);

c. Provides procedures that clearly enable the administrator to intervene in system operations to recover from an abnormal system state;

d. Defines and illustrates the procedures and system prompts for situations where operator intervention is required to load, initialize, and start the system;

e. Defines and illustrates procedures to enable and control the external interface to the system operating environment if supporting hardware and software are involved. Such information also shall be provided for the interaction of the system with other data processing systems or data interchange protocols;

f. Provides administrative procedures and off-line operator duties (if any) if they relate to the initiation or termination of system operations, to the assessment of system status, or to the development of an audit trail;

g. Supports successful ballot and program installation and control by central election officials, provides a detailed work plan or other form of documentation providing a schedule and steps for the software and ballot installation, which includes a table outlining the key dates, events and deliverables; and

h. Supports diagnostic testing, specifies diagnostic tests that may be employed to identify problems in the system, verify the correction of maintenance problems, and isolate and diagnose faults from various system states.

Source: [2] II.2.8.5.

2.4-14 GeneratedID2483

The user documentation supplied by the vendor shall provide a schedule and steps for the software and ballot installation, including a table outlining the key dates, events and deliverables.

Source: [2] I.2.3.3.a.

2.4.6 Operations support

Ⓡ 2.4-15 GeneratedID2487

The vendor shall provide documentation of system operating procedures that:

a. Defines the procedures required to support system acquisition, installation, and readiness testing; and

b. Describes procedures for providing technical support, system maintenance and correction of defects, and for incorporating hardware upgrades and new software releases.

Source: [2] II.2.8.6.

2.4.7 Transportation and storage

Ⓡ 2.4-16 GeneratedID2493

The user documentation shall include any special instructions for preparing voting devices for shipment.

Source: New requirement.

Ⓡ 2.4-17 GeneratedID2496

The user documentation shall include any special storage instructions for voting devices.

Source: [2] I.3.2.2.1.

2.4.8 Appendices

The vendor may provide descriptive material and data supplementing the various sections of the body of the system operations manual. The content and arrangement of appendices are at the discretion of the vendor. Topics recommended for discussion include:

a. **Glossary:** A listing and brief definition of all terms that may be unfamiliar to persons not trained in either voting systems or computer operations;

b. **References:** A list of references to all vendor documents and to other sources related to operation of the system;

c. **Detailed Examples:** Detailed scenarios that outline correct system responses to faulty operator input. Alternative procedures may be specified depending on the system state; and

d. **Manufacturer's Recommended Security Procedures:** Security procedures that are to be executed by the system operator.

2.5 System maintenance manual

Ⓡ 2.5-1 GeneratedID2505

The system maintenance manual shall provide information in sufficient detail to support election workers, information systems personnel, or maintenance personnel in the adjustment or removal and replacement of components or modules in the field.

Source: [2] II.2.9.

DISCUSSION

Technical documentation needed solely to support the repair of defective components or modules ordinarily done by the manufacturer or software developer is not required.

Ⓡ 2.5-2 GeneratedID2509

Recommended service actions to correct malfunctions or problems shall be discussed, along with personnel and expertise required to repair and maintain the system, and equipment, materials, and facilities needed for proper maintenance.

Source: [2] II.2.9.

2.5.1 Introduction

Ⓡ 2.5-3 GeneratedID2513

The vendor shall describe the structure and function of the equipment (and related software) for election preparation, programming, vote recording, tabulation, and reporting in sufficient detail to provide an overview of the system for maintenance, and for identification of faulty hardware or software.

Source: [2] II.2.9.1.

Ⓡ 2.5-3.1 GeneratedID2516

The description shall include a concept of operations that fully describes such items as:

a. The electrical and mechanical functions of the equipment;

b. How the processes of ballot handling and reading are performed (paper-based systems);

c. For electronic vote-capture devices, how vote selection and casting of the ballot are performed;

d. How transmission of data over a network is performed (if applicable);

e. How data are handled in the processor and memory units;

f. How data output is initiated and controlled;

g. How power is converted or conditioned; and

h. How test and diagnostic information is acquired and used.

Source: [2] II.2.9.1.

2.5.2 Maintenance procedures

Ⓡ 2.5-4 GeneratedID2528

The vendor shall describe preventive and corrective maintenance procedures for hardware and software.

Source: [2] II.2.9.2.

2.5.2.1 Preventive maintenance procedures

Ⓡ 2.5-5 GeneratedID2532

The vendor shall identify and describe:

a. All required and recommended preventive maintenance tasks, including software tasks such as software backup, database performance analysis, and database tuning;

b. Number and skill levels of personnel required for each task;

c. Parts, supplies, special maintenance equipment, software tools, or other resources needed for maintenance; and

d. Any maintenance tasks that must be coordinated with the vendor or a third party (such as coordination that may be needed for off-the-shelf items used in the system).

Source: [2] II.2.9.2.1.

2.5.2.2 Corrective maintenance procedures

2.5-6 GeneratedID2540

The vendor shall provide fault detection, fault isolation, correction procedures, and logic diagrams for all operational abnormalities identified by design analysis and operating experience.

Source: [2] II.2.9.2.2.

2.5-7 GeneratedID2543

The vendor shall identify specific procedures to be used in diagnosing and correcting problems in the system hardware or user-controlled software. Descriptions shall include:

a. Steps to replace failed or deficient equipment;

b. Steps to correct deficiencies or faulty operations in software;

c. Modifications that are necessary to coordinate any modified or upgraded software with other software modules;

d. The number and skill levels of personnel needed to accomplish each procedure;

e. Special maintenance equipment, parts, supplies, or other resources needed to accomplish each procedure; and

f. Any coordination required with the vendor, or other party, for off the shelf items.

Source: [2] II.2.9.2.2.

2.5.3 Maintenance equipment

2.5-8 GeneratedID2553

The vendor shall identify and describe any special purpose test or maintenance equipment recommended for fault isolation and diagnostic purposes.

Source: [2] II.2.9.3.

2.5.4 Parts and materials

2.5-9 GeneratedID2557

Vendors shall provide detailed documentation of parts and materials needed to operate and maintain the system.

Source: [2] II.2.9.4.

2.5.4.1 Common standards

2.5-10 GeneratedID2561

The vendor shall provide a complete list of approved parts and materials needed for maintenance. This list shall contain sufficient descriptive information to identify all parts by:

a. Type;

b. Size;

c. Value or range;

d. Manufacturer's designation;

e. Individual quantities needed; and

f. Sources from which they may be obtained.

Source: [2] I.3.4.1.b, II.2.9.4.1.

2.5.4.2 Paper-based systems

R 2.5-11 GeneratedID2571

The user documentation shall identify specific marking devices that, if used to make the prescribed form of mark, produce readable marked ballots so that the system meets the performance requirements for accuracy.

Source: Simplified from [2] I.3.2.4.2.3.

Applies to: Paper-based device

DISCUSSION

Includes pens and pencils.

Impact: Deleted requirement to specify performance characteristics of marking devices because the certification only covers the ones used in testing.

R 2.5-11.1 GeneratedID2577

For marking devices manufactured by multiple external sources, the vendor shall specify a listing of sources and model numbers that satisfy these requirements.

Source: [2] I.3.2.4.2.3.c and II.2.9.4.2.

R 2.5-12 TDPPaperSpecs

The user documentation shall specify the required paper stock, weight, size, shape, opacity, color, watermarks, field layout, orientation, size and style of printing, size and location of punch or mark fields used for vote response fields and to identify unique ballot formats, placement of alignment marks, ink for printing, and folding and bleed-through limitations for preparation of ballots that are compatible with the system.

Source: [2] I.2.3.1.3.1.c, I.3.2.4.2.1.c, II.2.9.4.2.

Applies to: Paper-based device

R 2.5-13 GeneratedID2583

User documentation for marksense systems shall include specifications for ballot materials to ensure that vote selections are read from only a single ballot at a time, without detection of marks from multiple ballots concurrently (e.g., reading of bleed-through from other ballots).

Source: [2] I.2.3.1.3.2.

Applies to: Optical scanner

2.5.5 Maintenance facilities and support

Ⓡ 2.5-14 GeneratedID2588

The vendor shall identify all facilities, furnishings, fixtures, and utilities that will be required for equipment maintenance.

Source: [2] II.2.9.5.

Ⓡ 2.5-15 GeneratedID2591

Vendors shall specify the typical system configuration that is to be used to assess availability, and any assumptions made with regard to any parameters that impact the Mean Time To Repair (MTTR). These factors shall include at a minimum:

a. Recommended number and locations of spare devices or components to be kept on hand for repair purposes during periods of system operation;

b. Recommended number and locations of qualified maintenance personnel who need to be available to support repair calls during system operation; and

c. Organizational affiliation (i.e., jurisdiction, vendor) of qualified maintenance personnel.

Source: [2] I.3.4.5, II.2.9.5.

2.5.6 Appendices

The vendor may provide descriptive material and data supplementing the various sections of the body of the system maintenance manual. The content and arrangement of appendices are at the discretion of the vendor. Topics recommended for amplification or treatment in appendix include:

a. **Glossary:** A listing and brief definition of all terms that may be unfamiliar to persons not trained in either voting systems or computer maintenance;

b. **References:** A list of references to all vendor documents and other sources related to maintenance of the system;

c. **Detailed Examples:** Detailed scenarios that outline correct system responses to every conceivable faulty operator input; alternative procedures may be specified depending on the system state; and

d. **Maintenance and Security Procedures:** Technical illustrations and schematic representations of electronic circuits unique to the system.

2.6 Personnel deployment and training requirements

Ⓡ 2.6-1 GeneratedID2603

The vendor shall describe the personnel resources and training required for a jurisdiction to operate and maintain the system.

Source: [2] II.2.10.

2.6.1 Personnel

Ⓡ 2.6-2 GeneratedID2607

The vendor shall specify the number of personnel and skill levels required to perform each of the following functions:

a. Pre-election or election preparation functions (e.g., entering an election, contest and candidate information; designing a ballot; generating pre-election reports);

b. System operations for voting system functions performed at the polling place;

c. System operations for voting system functions performed at the central count facility;

d. Preventive maintenance tasks;

e. Diagnosis of faulty hardware or software;

f. Corrective maintenance tasks; and

g. Testing to verify the correction of problems.

Source: [2] II.2.10.1.

Ⓡ 2.6-3 GeneratedID2617

The vendor shall distinguish which functions may be carried out by user personnel and which must be performed by vendor personnel.

Source: [2] II.2.10.1.

2.6.2 Training

R 2.6-4 GeneratedID2621

The vendor shall specify requirements for the orientation and training of administrators, central election officials, election judges, and poll workers.

Source: [2] II.2.10.2.

Impact: Deleted "vendor personnel" from list, harmonized with newly defined roles.

Chapter 3 Certification Test Plan (test lab)

[5] II.A. Revise drastically or delete, as appropriate.

Chapter 4 Test report for EAC certification (test lab)

This section assigned to Horlick?

Issue #1317: saved to /home/dflater/HAVA/pubs/S1317_EAC_Vol2_1.9_Transparency.pdf

[5] II.B.

4.1 Requirements

R 4.1-1 GeneratedID2630

Whether or not a system is qualified, the test lab shall report all of the data collected for estimation of MTBF and error rate.

R 4.1-2 GeneratedID2632

If a system is not qualified, the test lab shall report on all failed tests and the reasons for failure, including all applicable evidence (e.g., vote data report, citation of logic error in source code).

Ⓡ 4.1-3 GeneratedID2634

The test lab shall report on all applicable, non-mandatory ("should") requirements that the system failed to satisfy, along with any related justifications.

Ⓡ 4.1-4 GeneratedID2636

The test lab shall report the open-ended functional tests performed and the test verdicts. No system shall be qualified if any open-ended functional tests are assigned the verdict Fail using the test lab's defined pass criteria.

Ⓡ 4.1-5 prepostfinding

For each callable unit (function, method, operation, subroutine, procedure, etc.), in source code or analogous logic design, the test lab shall report a finding on whether the preconditions and postconditions correctly describe the behavior of the unit in all cases. This finding shall be one of Correct, Incorrect, or Unable to Determine.

Ⓡ 4.1-6 limitsfinding

The test lab shall report a finding whether all the assumptions about capacities and limits that appear in the preconditions, postconditions, and proofs are consistent with the capacities and limits that the devices are claimed to be capable of processing correctly. This finding shall be one of Consistent, Inconsistent, or Unable to Determine.

Ⓡ 4.1-7 assertionsfinding

For the software or logic design as a whole, and for each invariant and assertion indicated in Volume III Section 5.3 for the classes claimed in the implementation statement, the test lab shall report a finding whether the assertion is satisfied in all cases within the aforementioned capacities and limits. This finding shall be one of Satisfied, Unsatisfied, or Unable to Determine.

Say something about device vs. system level, integration logic. See Volume V Section 3.5.

STS, HFP: add your stuff.

Chapter 5 Public Information Package (test lab)

5.1 Requirements

Ⓡ 5.1-1 GeneratedID2644

If a system is qualified, the test lab shall publish a statement to that effect that includes the following information:

a. The implementation statement, as made by the vendor;

b. A list of the tests for which the test verdict was Waived;

c. The estimated error rate and MTBF of the system as calculated from the statistics collected during testing;

d. For paper-based tabulators, the speed or rate at which tabulation was performed in typical case and capacity tests;

e. A confirmation that the verdict on every applicable test was Pass, all preconditions and postconditions were found Correct, all the assumptions about capacities and limits were found Consistent, and all assertions were found Satisfied;

f. A summary of all failures, errors, uncorrected nonconformities and anomalies that were noted during conformity assessment, no matter how minor;

g. A timeline of the certification process for the qualified system.

STS, HFP: add your stuff.

Volume V Testing standard

Chapter 1 Overview of certification testing

Grab verbiage from [5] II.1.

[2] defines qualification testing as "the examination and testing of a computerized voting system by an Independent Test Authority using qualification test standards to determine if the system complies with the qualification performance and test standards and with its own specifications. This process occurs prior to state certification."

The purpose of voting system (qualification) testing is to provide the states and other affected stakeholders with some level of assurance that a voting system is fit for use. States have the option to subject a voting system to additional scrutiny before purchasing and deploying it; however, most states require qualification by an ITA as an entry condition.

Even if procedural controls and audit trails ensured that any miscount would be detected, it could still be catastrophic for a state to have to rerun a compromised election and to remedy the faulty equipment. It is in the states' interests for the certification process to eliminate voting systems that are not trustworthy before they are purchased and deployed.

Chapter 2   Introduction to test methods

This section contains high-level descriptions of general test methods. A citation of one of these methods in the Test Reference of a requirement in the Product Standard provides guidance on how conformity to the requirement could be assessed without giving a specific procedure. When a specific procedure for assessing conformity is provided in these Guidelines, the Test Reference instead cites an item in Documentation and Design Reviews (Volume V Chapter 3) or Test Protocols (Volume V Chapter 4), as applicable.

2.1   Inspection

Inspection is the examination of a product design, product, process or installation and determination of its conformity with specific requirements or, on the basis of professional judgement, with general requirements. [11]

Inspection is indicated when there is no operational test for assessing conformity to a given requirement. Inspection can be as simple as a visual confirmation that a particular design element or function is present or review of documentation to ensure inclusion of specific content, or it can be as complex as formal evaluation by an accredited specialist.

Logic verification is an example of inspection. Although formal proofs can be checked automatically, the determination that the premises correctly describe the behavior of the system requires professional judgement.

2.2   Functional testing

Functional testing is the determination through operational testing of whether the behavior of a system or device in specific scenarios conforms to requirements. Functional tests are derived by analyzing the requirements and the behaviors that should result from implementing those requirements. For example, one could determine through functional testing that a tabulator reports the correct totals for a specific simulated election day scenario.

Functional testing is indicated when the requirements on the behavior of a system or device are sufficiently precise and constraining that conformance can be objectively determined.

Strategies for conducting functional testing are broadly characterized as either "black box" or "white box" (see Volume I Section 1.4). However, a given test is neither black-box nor white-box. That distinction pertains to the strategy by which applicable tests are developed and/or selected, not to the tests themselves. For example, if a given input is tested because it is a special case in the functional specification of the system, then it is black box testing; but if that same input is tested because it exercises an otherwise unused block of code found during the review of source code, then it is white box testing.

Functional testing can be performed using a test suite (Volume V Section 4.2) or it can be open-ended (Volume V Section 4.4).

2.3 Performance testing (benchmarking)

Performance testing, a.k.a. benchmarking, is the measurement of a property of a system or device in specific scenarios. For example, one could determine through performance testing the amount of time that a tabulator takes to report its totals in a specific simulated election day scenario.

What distinguishes performance testing from functional testing is the form of the experimental result. A functional test yields a yes or no verdict, while a performance test yields a quantity. This quantity may subsequently be reduced to a yes or no verdict by comparison with a benchmark, but in the case of functional testing there is no such quantity to begin with. (E.g., there is no concept of "*x* % correct" for reported totals. Either they are correct or they are not.)

Performance testing is indicated when the requirements supply a benchmark for a measurable property.

Usability testing is an example of performance testing. The property being measured in usability testing involves the behavior of human test subjects.

2.4 Security penetration testing

Security penetration testing is an attempt by a lab technician or an accredited specialist to bypass or break the access or integrity controls of a system or device.

Like functional testing, security penetration testing can falsify a general assertion (namely, that the system or device is secure) but it cannot verify it (show it to be true in all cases).

Security penetration testing can be performed using a test suite (Volume V Section 4.3) or it can be open-ended (Volume V Section 4.5).

Chapter 3 Documentation and design reviews (inspections)

General for all subsections: need to specify pass criteria and what happens if they flunk. (Issue #4890 from Systest)

3.1 Initial review of documentation

R 3.1-1 GeneratedID2678

Prior to initiating the software review, the test lab shall verify that the documentation submitted by the vendor in the TDP is sufficient to enable:

a. Review of the source code; and

b. Design and conducting of tests at every level of the software structure to verify that the software meets the vendor's design specifications and the requirements of the performance standards.

Source: [2] II.5.3.

3.2 Physical configuration audit

R 3.2-1 GeneratedID2684

The test lab shall verify that the classes claimed in the implementation statement accurately characterize the devices submitted for testing.

Source: New requirement.

DISCUSSION

Any *Electronic device* that includes software or firmware installed or commissioned by the voting system vendor is a *Programmed device*. Vendors claiming that an electronic device is not programmed must demonstrate to the satisfaction of the testing and certifying authorities that the device contains no software or firmware that should be subject to the requirements indicated for programmed devices.

[2] I.8.7.1, II.6.6.

3.3 Functional configuration audit

[2] I.8.7.2, II.6.7.

Ⓡ 3.3-1 GeneratedID2691

The test lab shall confirm the propriety and correctness of the configuration choices described in Volume IV Section 1.9.

Source: [2] I.4.1.1.

Verify that languages claimed are supported.

3.4 Examination of vendor practices for configuration management and quality assurance

[2] II.7, to be provided by AG.

3.5 Logic verification

Because of its high complexity, the scope of logic verification is necessarily limited to the core tabulating functions of the voting system. Specifically, the scope extends from the election definition and cast vote records in abstract form to the totals that are output in vote data reports. Software modules that are solely devoted to interacting with the user or formatting reports are not subject to logic verification. However, they are required to conform with Requirement III.3.1-1, which is tested in Volume V Section 3.7.

Paste in standard text on logic verification.

Define how to verify system while respecting device boundaries and per-device limits.

It is acceptable, even expected, that logic verification will show that some or most exception handlers in the source code cannot logically be invoked. These exception handlers are not redundant—they provide defense-in-depth against faults that escape detection during logic verification.

Ⓡ 3.5-1 GeneratedID2701

No system shall be qualified unless all preconditions and postconditions (see Requirement IV.1.4-23.b and Requirement IV.4.1-5) are found Correct.

Ⓡ 3.5-2 GeneratedID2703

No system shall be qualified unless all the assumptions about capacities and limits (see Requirement III.2.5-1.e and Requirement IV.4.1-6) are found Consistent.

R 3.5-3 GeneratedID2705

No system shall be qualified unless all invariants and assertions (see Requirement IV.1.4-24 and Requirement IV.4.1-7) are found Satisfied.

3.6 Source code review

Scope these requirements to non-WUUP software and firmware per resolution of COTS issue (see PREFACE).

WUUP software is not subject to code examination. However, source code generated by a WUUP package and embedded in software modules for compilation or interpretation may be inspected by the test lab to determine testing requirements or to verify that the code is unmodified and that the default configuration options have not been changed.

R 3.6-1 GeneratedID2709

The test lab shall compare the source code to the vendor's software design documentation to ascertain how completely the software conforms to the vendor's specifications.

Source: [2] II.5.4.

R 3.6-2 GeneratedID2712

The test lab shall assess the extent to which the code adheres to the published, credible coding conventions chosen by the vendor.

Source: [2] II.5.4, II.5.4.2.

DISCUSSION

See Requirement III.3.5-2.

R 3.6-3 GeneratedID2716

The test lab shall assess the extent to which the code adheres to the requirements of Volume III Section 3.2.1 and Volume III Section 3.5.1.

Source: [2] II.5.4.

3.6-3.1 Assess conformity with code length limit

The test lab shall assess the extent to which the code adheres to Requirement III.3.5-4.

Source: [2] II.5.4.2.i.

DISCUSSION

See Requirement IV.1.4-25. The reviewer should consider the functional organization of each module or callable unit and the use of formatting, such as blocking into readable units, that supports the intent of Requirement III.3.5-4.

3.6-4 CheckTheChecker

The test lab shall verify the efficacy of built-in measurement, self-test, and diagnostic capabilities described in Volume III Section 4.4.1.

Source: [2] I.2.3.4.1.a2 (the second a).

3.7 Verification of design requirements in product standard

3.7-1 GeneratedID2724

For each of the design requirements enumerated below, the test lab shall review the source code (if applicable) and design of the voting system to verify that the requirement is satisfied. For each one, the test lab shall publish a finding whether the requirement is met. This finding shall be one of Satisfied, Unsatisfied, or Unable to Determine. No system shall be qualified unless all design requirements are found Satisfied.

a. All requirements in Volume III Section 3.1

b. Requirement III.3.6-1[14]

c. Requirement III.3.7-1[15]

d. Requirement III.4.6-9

e. Requirement III.4.6-15

f. Requirement III.4.9-3

g. Requirement III.4.10-1

h. Requirement III.4.10-2.1

i. Requirement III.4.10-2.3

Complete this list

3.8 *Et seq.* Stuff retained from [2]

Chapter 4 Test protocols

****
THIS SECTION IS IN DISREPAIR
****

Go through test protocols, separate devices from systems.

Separating device from system means that there is no longer a single limit on number of ballots that can be counted, etc.

Stuff to be provided by STS ([2] II.6.4) and HFP ([2] II.3.4, II.6.5).

The tests specified in this chapter are system tests. They apply regardless of whether part or all of the system is WUUP. See COTS issue (PREFACE).

4.1 Hardware

[2] II.4, to be provided by AG.

4.2 Counting and reporting

4.2.1 General test template

Merge in templates from [2] II.3.3.1 and II.3.3.2, if not redundant. Some steps should be done once for the entire test campaign, not before each test. Fix.

Most test cases will follow this general template. Different test cases will elaborate on the general template in different ways, depending on what is being tested.

a. Establish initial state (clean out data from previous tests, verify resident software/firmware)

b. Program election and prepare ballots

c. Generate pre-election audit reports

d. Configure polling equipment

e. Generate system readiness audit reports

f. Open poll

g. Run test ballots

h. Close poll

i. Generate in-process audit reports

j. Generate data reports for the specified reporting contexts

k. Inspect ballot counters

l. Inspect reports

4.2.2 General pass criteria

[2] I.9.6.2.6 contains some relevant items. It assumes that the test lab will be able to diagnose the source of the failure (if failure is due to software defect, system is returned to vendor for correction). Hmmm.

Ⓡ 4.2-1 GeneratedID2762

The test lab need only consider tests that apply to the class specified in the implementation statement, including those tests that are designated for all systems. The test verdict for all other tests shall be Not Applicable.

Ⓡ 4.2-2 GeneratedID2764

If the documented assumptions for a given test (indicated by the presence of an **Assumptions:** field in the test case description) are not met, the test verdict shall be Waived and the test shall not be executed.

Ⓡ 4.2-3 GeneratedID2766

If the test lab is unable to execute a given test because the system does not support functionality that is required per the implementation statement or is required for all systems, the test verdict shall be Fail.

Ⓡ 4.2-4 GeneratedID2768

If the test lab executes a test, the test verdict shall be assigned based on the following inputs, which are described in more detail below. The test verdict shall be Pass if and only if none of these inputs indicates a verdict of Fail.

a. Mean Time Between Failure (MTBF)

b. Error rate

c. Additional pass criteria

d. General performance requirements

Ⓡ 4.2-5 GeneratedID2774

No system shall be qualified if any test verdicts are Fail.

4.2.2.1 Mean Time Between Failure

[2] II.C needs to be revised and included somewhere.

Ⓡ 4.2-6 GeneratedID2778

During execution of all tests except Dangling ref: ForcedErrorRecovery, the test lab shall keep track of real time and the number of failures (see definition in Volume II). These statistics shall be collected and accumulated across all tests.

Ⓡ 4.2-7 GeneratedID2780

If a failure should occur during the execution of any test except Dangling ref: ForcedErrorRecovery, the test lab shall note the failure for use in the calculation of MTBF. The test lab shall then follow the vendor's documented procedures for recovering from failures.

Ⓡ 4.2-7.1 GeneratedID2782

If recovery is not possible or not successful, the test verdict shall be Fail. Otherwise, after recovery, the test lab shall attempt to re-execute the test that was affected by the failure from the beginning.

4.2-7.2 GeneratedID2784

If the failure recurs, the test verdict shall be Fail. If the failure does not recur, the following system-level MTBF decision criteria shall be applied:

a. If statistical analysis of the cumulative behavior across all tests executed so far indicates with at least 90 % confidence that the MTBF is worse than 744 hours (one month), the test verdict shall be Fail.

b. Otherwise, the failure shall be noted, the test verdict shall be assigned based on the other inputs (disregarding the failure), and testing shall continue.

744 hours is just a suggestion—this is for AG to determine (Volume III Section 3.4.1).

For more information, see Volume III Section 3.4.1.

4.2.2.2 Error rate

4.2-8 GeneratedID2790

During all test executions, the test lab shall keep track of the number of ballot positions counted and the number of errors (see definition in Volume II). These statistics shall be collected and accumulated across all tests.

DISCUSSION

The definition of error effectively treats data that denote ballot format or precinct as ballot positions. If reports attribute the counts and totals to the wrong contests or candidates, every ballot position is considered to have been reported erroneously.[16]

4.2-9 GeneratedID2792

If a test runs to completion, the test lab shall inspect the data reports and verify that counts and totals are reported in compliance with the requirements in Volume III Section 4.9. If all reported counts and totals are identical to the specified values, the test verdict shall be Pass. Otherwise, the following system-level accuracy decision criteria shall be applied:

a. If statistical analysis of the cumulative behavior across all tests executed so far indicates with at least 95 % confidence that the error rate is worse than 1 in 10,000,000 ballot positions, the test verdict shall be Fail.

b. Otherwise, the error shall be noted, the test verdict shall be assigned based on the other inputs (disregarding the error), and testing shall continue.

For more information, see Volume III Section 3.4.2.

4.2.2.3 Additional pass criteria

When certain performance requirements of the VVSG are of particular relevance to a particular test, these are noted after **Additional pass criteria:** in the test case description.

Ⓡ 4.2-10 GeneratedID2799

The test lab shall verify that **Additional pass criteria** requirements are satisfied during the execution of test cases. If they are not, the test verdict shall be Fail.

4.2.2.4 General performance requirements

Ⓡ 4.2-11 GeneratedID2802

A demonstrable violation of any requirement of the VVSG during the execution of any test case shall result in a test verdict of Fail, irrespective of whether this requirement was explicitly noted in the **Additional pass criteria** for that test case.

DISCUSSION

For example, if any of the audit reports should be incomplete or incorrect with respect to any of the many applicable requirements in Volume III Section 3.2.2, the test verdict would be Fail.

For example, if a DRE system should take longer than 5 minutes for each device to generate a consolidated report, Requirement III.4.8-16.1 would be violated and the test verdict would be Fail.

Add test case for all candidates get same number of votes (will the system break if it can't rank the candidates).

4.2.3 Null case test

The purpose of the null case test is to verify that closing the polls after processing zero ballots is correctly handled. This case can arise in practice, for example, in precincts where a single DRE is provided alongside other equipment, if no voters use the DRE.

4.2.3.1 All systems

Test 1 Null Case

References: Requirement III.4.9-12

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Null Case Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

No ballots shall be cast.

4.2.4 Simple case tests

The purpose of a simple case test is to establish that one or more functional features that are required to be supported, are supported. Simple case tests are not stress tests, although by their minimality, they may unintentionally test boundary conditions. For stress tests, refer to the Capacity Tests section.

Following subsections are organized by classes. Simple case tests apply only if the class specified in the implementation statement is a subclass of the class indicated in the subsection name.

4.2.4.1 All systems

Test 2 1-of-M Trivial Case

References: Requirement III.4.1-5.1, Requirement III.4.1-5.2, Requirement III.4.6-7.1, Requirement III.4.8-4.1

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the 1-of-M Trivial Case Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

Two ballots shall vote for Unopposed Candidate.

Test 3 1-of-M Simple Case

References: Requirement III.4.1-5.1, Requirement III.4.1-5.2, Requirement III.4.6-7.1, Requirement III.4.8-4.1

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the 1-of-M Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1
> Ballot Position 2
> Ballot Position 3

Reporting contexts: Single context.

Scenario:

Four ballots shall vote for Ballot Position 1

Three ballots shall vote for Ballot Position 2

Two ballots shall vote for Ballot Position 3

One ballot shall vote for none (undervote).

Test 4 Reporting Levels Test

References: Requirement III.4.1-4

Six vote-capture devices, three precinct tabulators and one central tabulator are required to execute this test.

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the Reporting Levels Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts:

Machines 1 and 2 shall be in Precinct 1.

Machines 3 and 4 shall be in Precinct 2.

Machines 5 and 6 shall be in Precinct 3.

Precincts 1 and 2 shall be in District 1.

Precinct 3 shall be in District 2.

All of the above shall be in the Jurisdiction.

Scenario:

On Machine 1, three ballots shall be cast for Ballot Position 1, two ballots shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.

On Machine 2, three ballots shall be cast for Ballot Position 1, one ballot shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.

On Machine 3, two ballots shall be cast for Ballot Position 1, one ballots shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.

On Machine 4, one ballot shall be cast for Ballot Position 1, one ballot shall be cast for Ballot Position 2, and one ballot shall be cast for Ballot Position 3.

On Machine 5, two ballots shall be cast for Ballot Position 2.

On Machine 6, one ballot shall be cast for Ballot Position 1.

4.2.4.2 DRE

Test 5 Ballot Images Simple Case

References: Requirement III.4.11-1.3, Requirement III.4.11-1.4

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the Ballot Images Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

Four ballots shall vote for Ballot Position 1

Three ballots shall vote for Ballot Position 2

Two ballots shall vote for Ballot Position 3

One ballot shall vote for none (undervote).

After close of polls, the test lab shall retrieve and review the ballot images.

Additional pass criteria:

The cast vote records (retrieved ballot images) shall be accurate. (Requirement III.4.11-1.3)

The cast vote records (retrieved ballot images) shall be human-readable. (Requirement III.4.11-1.4)

The cast vote records (retrieved ballot images) shall not reveal the order in which they were voted. (Dangling ref: PrivacyShuffleCVR)

4.2.4.3 Marksense and Punchcard

Test 6 Overvoting Simple Case

References: Requirement III.4.9-20, Requirement III.4.9-20.1

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the Overvoting Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

Three ballots shall vote for Ballot Position 1

Two ballots shall vote for Ballot Position 1 and Ballot Position 2

Two ballots shall vote for Ballot Position 2

Three ballots shall vote for Ballot Position 2 and Ballot Position 3

One ballot shall vote for Ballot Position 3

Four ballots shall vote for Ballot Position 1 and Ballot Position 3

One ballot shall vote for all three ballot positions

One ballot shall vote for none (undervote).

In addition to generating the usual reports, the test lab shall perform four ad-hoc queries to determine the number of overvotes combining ballot positions in each of the four applicable combinations (1+2, 1+3, 2+3, 1+2+3). (Requirement III.4.9-20.1)

4.2.4.4 Closed primaries

Test 7 Closed Primary Simple Case

References: Requirement III.4.1-5.3, Requirement III.4.2-1.6, Requirement III.4.6-7.3, Requirement III.4.8-4.7, Requirement III.4.9-16.2, Requirement III.4.9-17.1, Test 8, Test 8

Ballot format: Whig

2 1-of-M contests where M = 2.

The first contest shall be described as follows:

> This is the first contest in the Whig ballot format of the Closed Primary Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Whig 1
> Whig 2

The second contest shall be described as follows:

> This is the second contest, a nonpartisan office. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Whig 3
> Tory 3

Ballot format: Tory

2 1-of-M contests where M = 2.

The first contest shall be described as follows:

> This is the first contest in the Tory ballot format of the Closed Primary Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Tory 1
> Tory 2

The second contest (nonpartisan) shall be identical to the second contest in the Whig ballot format.

Reporting contexts: Single context.

Scenario:

Two Whig ballots shall vote for Whig 1 and Whig 3

One Whig ballot shall vote for Whig 2 and Tory 3

One Tory ballot shall vote for Tory 1 and Tory 3

Two Tory ballots shall vote for Tory 2 and skip the second contest (undervotes).

Additional pass criteria:

Separate counts for each party shall be reported in accordance with Requirement III.4.9-16.2 and Requirement III.4.9-17.1.

4.2.4.5 Open primaries

Test 8 Open Primary Simple Case

References: Requirement III.4.1-5.3, Requirement III.4.2-1.6, Requirement III.4.6-2.4, Requirement III.4.6-7.4, Requirement III.4.8-4.7, Requirement III.4.9-16.2, Requirement III.4.9-17.1

Ballot formats:

Ballot formats shall be identical to those in Test 7, except changing the name of the test case in the contest descriptions.

Reporting contexts: Single context.

Scenario: same as Test 7.

Additional pass criteria:

Separate counts for each party shall be reported in accordance with Requirement III.4.9-16.2 and Requirement III.4.9-17.1.

In a DRE system, the voter should be allowed to choose a party affiliation at the time of voting and vote the appropriate ballot format in privacy in accordance with Requirement III.4.6-2.4.

4.2.4.6 Write-ins

Test 9 Write-ins Simple Case

References: Requirement III.4.1-5.4, Requirement III.4.6-7.5, Requirement III.4.8-4.8

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Write-ins Simple Case Test. There are no candidates on the ballot. Write in at most one.

The only ballot position in the contest shall be a write-in opportunity.

Reporting contexts: Single context.

Scenario:

Four ballots shall write in First Write-In Candidate.

Three ballots shall vote for none (undervote).

Two ballots shall write in Second Write-In Candidate.

4.2.4.7 Ballot rotation

Test 10 Ballot Rotation Simple Case

References: Requirement III.4.2-1.7, Requirement III.4.6-7.6, Requirement III.4.6-7.7, Requirement III.4.8-4.9

Ballot format:

1 1-of-M contest where M = 3, with ballot rotation enabled.

The contest shall be described as follows:

> This is the only contest in the Ballot Rotation Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Candidate 1
> Candidate 2
> Candidate 3

Reporting contexts: Single context.

Scenario:

Four ballots shall vote for Candidate 1

Three ballots shall vote for Candidate 2

Two ballots shall vote for Candidate 3

Additional pass criteria:

In a DRE system, each candidate shall appear in each position on the ballot exactly three times in accordance with Requirement III.4.6-7.7.

4.2.4.8 Straight party voting

Test 11 Straight Party Voting Simple Case

References: Requirement III.4.1-5.5, Requirement III.4.6-7.8, Requirement III.4.8-4.10, Requirement III.4.8-4.11

Ballot format:

2 1-of-M contests.

The first contest shall be described as follows:

> STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The ballot positions shall be the following:

> Whig
> Tory

The second contest shall be described as follows:

> This is the only contest in the Straight Party Voting Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1 (Whig)
> Ballot Position 2 (Tory)
> Ballot Position 3 (Independent)

Reporting contexts: Single context.

Scenario:

Two ballots shall vote straight party Whig and skip the second contest (allowing the straight party vote to be effective)

One ballot shall skip the straight party vote and vote for Ballot Position 1 (Whig)

One ballot shall votes straight party Tory but then vote for Ballot Position 1 (Whig)

Two ballots shall vote straight party Tory and skip the second contest

One ballot shall skip the straight party vote and vote for Ballot Position 2 (Tory)

One ballot shall vote straight party Tory but then vote for Ballot Position 3 (Independent).

(Expected result: Ballot Position 1 (Whig), 4; Ballot Position 2 (Tory), 3; Ballot Position 3 (Independent), 1.)

4.2.4.9 Cross-party endorsement

Test 12 Cross-party Endorsement Simple Case

References: Requirement III.4.1-5.6, Requirement III.4.6-7.9, Requirement III.4.8-4.12

Ballot format:

2 1-of-M contests.

The first contest shall be described as follows:

> STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The ballot positions shall be the following:

> Whig
> Free-Soil
> National
> Federalist

The second contest shall be described as follows:

> This is the only contest in the Straight Party Voting Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1 (Whig/National/Federalist)
> Ballot Position 2 (Free-Soil)
> Ballot Position 3 (Independent)

Reporting contexts: Single context.

Scenario:

One ballot shall vote straight party Whig and skip the second contest

Two ballots shall vote straight party Free-Soil and skip the second contest

Three ballots shall vote straight party National and skip the second contest

Two ballots shall vote straight party Federalist and skip the second contest

One ballot shall skip the straight party vote and vote for Ballot Position 2 (Free-Soil)

One ballot shall skip the straight party vote and vote for Ballot Position 3 (Independent).

4.2.4.10 Split precincts

Test 13 Split Precinct Simple Case

References: Requirement III.4.1-5.7, Requirement III.4.2-1.8, Requirement III.4.6-7.10, Requirement III.4.8-4.13

Ballot format: District 1

1 1-of-M contest where M = 2.

The contest shall be described as follows:

> This is the only contest in the District 1 ballot format of the Split Precinct Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> District 1 Candidate 1
> District 1 Candidate 2

Ballot format: District 2

1 1-of-M contest where M = 2.

The contest shall be described as follows:

> This is the only contest in the District 2 ballot format of the Split Precinct Simple Case Test. There are two candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> District 2 Candidate 1
> District 2 Candidate 2

Reporting contexts: Precinct 1, District 1, District 2, Jurisdiction. (Precinct 1 is split between District 1 and District 2.)

Scenario:

Three District 1 ballots shall vote for District 1 Candidate 1

Two District 1 ballots shall vote for District 1 Candidate 2

Six District 2 ballots shall vote for District 2 Candidate 1

Additional pass criteria:

Only the District 1 ballots shall be reported in the District 1 reporting context.

Only the District 2 ballots shall be reported in the District 2 reporting context.

All ballots shall be reported in the Precinct 1 and Jurisdiction reporting contexts.

4.2.4.11 N of M voting

Test 14 N-of-M Simple Case

References: Requirement III.4.1-5.8, Requirement III.4.6-7.11, Requirement III.4.8-4.14

Ballot format:

1 N-of-M contest where N = 2 and M = 3.

The contest shall be described as follows:

> This is the only contest in the N-of-M Simple Case Test. There are three candidates on the ballot. Vote for at most two.

The ballot positions shall be the following:

> Ballot Position 1
> Ballot Position 2
> Ballot Position 3

Reporting contexts: Single context.

Scenario:

Four ballots shall vote for Ballot Position 1 and Ballot Position 2

Three ballots shall vote for Ballot Position 1 and Ballot Position 3

Two ballots shall vote for Ballot Position 2 and nobody else (single undervote)

One ballot shall vote for none (double undervote).

4.2.4.12 N of M voting + Write-ins

Test 15 N-of-M + Write-ins Simple Case

References: Requirement III.4.1-5.4, Requirement III.4.1-5.8, Requirement III.4.6-7.5, Requirement III.4.6-7.11, Requirement III.4.8-4.8, Requirement III.4.8-4.14

Ballot format:

1 N-of-M contest where N = 2 and M = 3.

The contest shall be described as follows:

> This is the only contest in the N-of-M + Write-ins Simple Case Test. There are no candidates on the ballot. Write in at most two.

The ballot positions shall be two write-in opportunities.

Reporting contexts: Single context.

Scenario:

Four ballots shall write in "Write-in Candidate 1" and "Write-in Candidate 2"

Two ballots shall write in "Write-in Candidate 1" and "Write-in Candidate 3"

Four ballots shall write in "Write-in Candidate 2" and nobody else (single undervote)

One ballot shall vote for none (double undervote).

4.2.4.13 Cumulative voting

Test 16 Cumulative Voting Simple Case

References: Requirement III.4.1-5.9, Requirement III.4.6-7.12, Requirement III.4.8-4.15

Ballot format:

1 cumulative voting contest where M = 3 and N(*r*) = 3.

The contest shall be described as follows:

> This is the only contest in the Cumulative Voting Simple Case Test. There are three candidates on the ballot. Cast at most three votes. You may cast multiple votes for the same candidate.

The ballot positions shall be the following:

> Ballot Position 1
> Ballot Position 2
> Ballot Position 3

Reporting contexts: Single context.

Scenario:

Four ballots shall vote once each for Ballot Position 1, Ballot Position 2, and Ballot Position 3

Three ballots shall vote twice for Ballot Position 2 and once for Ballot Position 3

Two ballots shall vote three times for Ballot Position 3

One ballot shall vote for none (undervote).

4.2.4.14 Ranked order voting

Test 17 Ranked Order Voting Simple Case

References: Requirement III.4.1-5.10, Requirement III.4.6-7.13, Requirement III.4.8-4.16, Requirement III.4.9-22

Ballot format:

1 1-of-M ranked order contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the Ranked Order Voting Simple Case Test. There are three candidates on the ballot. Please rank them in order of preference.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

Four ballots vote shall the ordering Ballot Position 1, Ballot Position 2, Ballot Position 3

Three ballots shall vote the ordering Ballot Position 2, Ballot Position 3, Ballot Position 1

Two ballots shall vote the ordering Ballot Position 3, Ballot Position 2, Ballot Position 1.

Expected result:

Round 1:

- Ballot Position 1, 4
- Ballot Position 2, 3
- Ballot Position 3, 2

Round 2:

- Ballot Position 2, 5
- Ballot Position 1, 4

4.2.4.15 Provisional / challenged ballots

Test 18 Provisional Ballots Simple Case

References: Requirement III.4.6-7.14, Requirement III.4.8-4.3, Requirement III.4.8-4.4, Requirement III.4.9-16.3, Requirement III.4.9-17.2, Requirement III.4.9-26

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the Provisional Ballots Simple Case Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

Ballot Position 1
Ballot Position 2
Ballot Position 3

Reporting contexts: Single context.

Scenario:

Two regular ballots shall vote for Ballot Position 1

Five provisional ballots shall vote for Ballot Position 1, and two shall be accepted

Three regular ballots shall vote for Ballot Position 2

One regular ballot shall vote for Ballot Position 3

Four provisional ballots shall vote for Ballot Position 3, and one shall be accepted

One provisional ballot shall vote for none (undervote), and shall be accepted.

Additional pass criteria:

The number of provisional / challenged ballots read and counted shall be reported in compliance with Requirement III.4.9-16.3, and Requirement III.4.9-17.2.

4.2.5 Typical case tests

The purpose of typical case tests is to test the behavior of the voting system in scenarios that reflect typical use of the system in practice rather than artificial minimum and maximum conditions.

Election officials are encouraged to submit testing scenarios that more accurately reflect their use of voting systems in practice.

4.2.5.1 Special instructions for Marksense and Punchcard

For systems claiming the *Marksense* or *Punchcard* classes, all applicable typical case tests shall be executed at a tabulating rate no less than 30 ballots per minute,[17] or the maximum rate at which the tabulating equipment is documented to function reliably (L_P), whichever is less. To speed testing, a higher rate may be used if the vendor does not object.

4.2.5.2 All systems

Test 19 1-of-M Typical Case

References: Requirement III.4.1-5.1, Requirement III.4.1-5.2, Requirement III.4.6-7.1, Requirement III.4.8-4.1

Assumptions:

$L_R \geq 10$

$L_C \geq 10$

$L_B \geq 75000$

$L_T \geq 38375$

Ballot format:

10 1-of-M contests where M = 10.

The contests shall be described as follows (substituting numbers from 1 to 10 for *r*):

> This is Contest *r* in the 1-of-M Typical Case Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Precinct 1, Precinct 2, ... through Precinct 100, Jurisdiction.

Scenario:

A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 - *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

In all precincts,

345 ballots shall vote for ballot position 1 in every contest
345 ballots shall vote for ballot position 2 in every contest
1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest
1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest

In precincts 1 through 50, all remaining ballots shall vote for ballot position 1 in the first contest and undervote the rest.

In precincts 51 through 100, all remaining ballots shall vote for ballot position 2 in the first contest and undervote the rest.

Discussion:

This test is based loosely on the minimum acceptance test guidelines in Appendix J of [1]. It has been modified to remove the explicit requirement for a large number of voting machines for testing - such are unlikely to be available for a system that is not yet qualified. The requirement for N-of-M voting has been removed to permit the test to apply to all systems.

4.2.5.3 Marksense and Punchcard

Test 20 1-of-M Paper Typical Case

References: Requirement III.4.9-20

Assumptions:

$L_R \geq 10$

$L_C \geq 10$

$L_B \geq 75000$

$L_T \geq 34500$

Ballot format:

10 1-of-M contests where M = 10.

The contests shall be described as follows (substituting numbers from 1 to 10 for *r*):

> This is Contest *r* in the 1-of-M Paper Typical Case Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Precinct 1, Precinct 2, ... through Precinct 100, Jurisdiction.

Scenario:

A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 - *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

In all precincts,

> 345 ballots shall vote for ballot position 1 in every contest
> 345 ballots shall vote for ballot position 2 in every contest
> 1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
> 1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
> 1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest

1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest

In precincts 1 through 50, all remaining ballots shall overvote the first contest by voting for both ballot positions 1 and 3 and undervote the rest.

In precincts 51 through 100, all remaining ballots shall overvote the first contest by voting for both ballot positions 2 and 3 and undervote the rest.

4.2.5.4 N of M voting

Test 21 N-of-M Typical Case

References: Requirement III.4.1-5.8, Requirement III.4.6-7.11, Requirement III.4.8-4.14

Assumptions:

$L_R \geq 10$

$L_C \geq 10$

$L_B \geq 75000$

$L_T \geq 75000$

Ballot format:

There shall be 10 contests. The first contest shall be an N-of-M contest where M = N = 10.

The first contest shall be described as follows:

> This is Contest 1 in the N-of-M Typical Case Test. Vote for at most 10.

The other 9 contests shall be 1-of-M contests where M = 10. These contests shall be described as follows (substituting numbers from 2 to 10 for *r*):

> This is Contest *r* in the N-of-M Typical Case Test. Vote for at most one.

The ballot positions in all 10 contests shall be of the following form (substituting numbers from 1 to 10 for *c*):

Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Precinct 1, Precinct 2, ... through Precinct 100, Jurisdiction.

Scenario:

A total of 75000 ballots shall be cast. Precincts 1 through 50 shall each receive 750 - *n* ballots, for precinct number *n*. Precincts 51 through 100 shall each receive 700 + *n* ballots.

In precincts 1 through 50, all ballots shall vote for all 10 candidates in Contest 1.

In precincts 51 through 100, all ballots shall vote for only the first 8 ballot positions in Contest 1 (yielding two undervotes each).

In all precincts, for the remaining 9 contests,

345 ballots shall vote for ballot position 1 in every contest
345 ballots shall vote for ballot position 2 in every contest
1 ballot shall vote for ballot position 3 in contest 3 and undervote the rest
1 ballot shall vote for ballot position 4 in contest 4 and undervote the rest
1 ballot shall vote for ballot position 5 in contest 5 and undervote the rest
1 ballot shall vote for ballot position 6 in contest 6 and undervote the rest
1 ballot shall vote for ballot position 7 in contest 7 and undervote the rest
1 ballot shall vote for ballot position 8 in contest 8 and undervote the rest
1 ballot shall vote for ballot position 9 in contest 9 and undervote the rest
1 ballot shall vote for ballot position 10 in contest 10 and undervote the rest

In precincts 1 through 50, all remaining ballots shall vote for ballot position 1 in the first contest and undervote the rest.

In precincts 51 through 100, all remaining ballots shall vote for ballot position 2 in the first contest and undervote the rest.

4.2.5.5 Discussion

Write-ins, etc.: need a typical case test that combines all typical classes, if there is a most common combination.

4.2.6 Capacity tests (covers testing of maximum ballot counting rate)

Following subsections are organized by classes. Capacity tests apply only if the class specified in the implementation statement is a subclass of the class indicated in the subsection name.

4.2.6.1 Special instructions for Marksense and Punchcard

For systems claiming the *Marksense* or *Punchcard* classes, all applicable capacity tests shall be executed at the maximum speed or rate at which the tabulating equipment is documented to function reliably (L_P).

4.2.6.2 All systems

Test 22 1-of-M Contest/Ballot Capacity

References: Requirement III.4.1-5.1, Requirement III.4.1-5.2, Requirement III.4.6-7.1, Requirement III.4.8-4.1

Ballot format:

L_R 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the 1-of-M Contest/Ballot Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

A total of L_B ballots shall be cast.

$\min\left(L_T, \max\left(\left\lceil \frac{L_B}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* in every contest (*c* > 0).

Any ballots left over shall be blank (undervotes).

Test 23 Ballot format Capacity

References: Requirement III.4.2-1

Assumptions:

$L_T \geq \min(L_B, L_F) - L_C + 1$

Ballot formats:

L_F ballot formats shall be constructed. These forms shall share the same set of contests. (They shall be identical except for their form identifications.)

There shall be L_R 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the Ballot Format Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in any of the ballot formats.

Reporting contexts: Single context.

Scenario:

$\min(L_B, L_F)$ ballots shall be cast.

The *n*th ballot shall use ballot format *n* and shall vote for ballot position $\min(n, L_C)$ in every contest.

Test 24 Vote Register Capacity

References: Requirement III.4.9-12, Requirement III.4.9-19

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Vote Register Capacity Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for Unopposed Candidate.

Test 25 Undervote Register Capacity

References: Requirement III.4.9-21

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Undervote Register Capacity Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. All shall be blank.

Test 26 1-of-M Multi Capacity

References: Requirement III.4.9-21

Assumptions:

$L_R \geq L_F$

Ballot formats:

L_F ballot formats shall be constructed. The first contest on each form shall be unique to that form. It shall be described as follows (substituting numbers from 1 to L_F for *f*):

> This contest appears only in Ballot Format *f*. Vote for at most one.

The ballot positions in these contests shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Form *f* Position *c*

Each ballot format shall contain L_F - L_R other contests that are shared by all ballot formats. These contests shall be described as follows (substituting numbers from 1 to L_F - L_R for *r*):

> This is Shared Contest *r* in the 1-of-M Multi Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in any of the ballot formats.

Reporting contexts: Single context.

Scenario:

For each ballot format *f* , for *f* from 1 to L_F, $\left\lfloor \frac{L_B}{L_F} \right\rfloor$ ballots shall be cast. Let

$n = \left\lfloor \frac{L_B}{L_F} \right\rfloor \times L_F.$ (This may total fewer than L_B ballots.)

In the contest that is unique to each ballot format, $\min\left(L_T, \max\left(\left\lceil \frac{\left\lfloor \frac{L_B}{L_F} \right\rfloor}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* (*c* > 0). Any ballots left over shall undervote the unique contest.

In all other (shared) contests, $\min\left(L_T, \max\left(\left\lceil \frac{n}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* in every contest (*c* > 0). Any ballots left over shall undervote all of the shared contests.

4.2.6.3 Marksense and Punchcard

Test 27 Overvote Register Capacity

References: Requirement III.4.9-20

Ballot format:

1 1-of-M contest where M = 2.

The contest shall be described as follows:

> This is the only contest in the Overvote Register Capacity Test. There are two candidates on the ballot. Vote for one.

The ballot positions in the contest shall be the following:

> Ballot Position 1
> Ballot Position 2

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for both ballot positions (overvote).

4.2.6.4 Write-ins

Test 28 1-of-M Write-in Capacity 1

References: Requirement III.4.1-5.4, Requirement III.4.6-7.5, Requirement III.4.8-4.8

Ballot format:

L_R 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the 1-of-M Write-in Capacity 1 Test. Vote for at most one.

The ballot positions from 1 to L_C-1 in each contest shall be of the following form (substituting numbers from 1 to L_C-1 for *c*):

> Contest *r* Ballot Position *c*

The final ballot position in each contest shall be a write-in opportunity.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. All shall write in "Write-in Candidate *r*" in every contest, substituting contest numbers 1 to L_R for *r*.

Test 29 1-of-M Write-in Capacity 2

References: Requirement III.4.1-5.4, Requirement III.4.6-7.5, Requirement III.4.8-4.8

Ballot format:

L_R 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the 1-of-M Write-in Capacity 2 Test. There are no candidates on the ballot. Write in at most one.

The only ballot position in each contest shall be a write-in opportunity.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T, L_W)$ ballots shall be cast. All shall write in "Write-in Candidate *n*" in every contest, substituting ballot numbers 1 to $\min(L_B, L_T, L_W)$ for *n* (each ballot shall vote for a different write-in candidate).

4.2.6.5 Straight party voting

Test 30 1-of-M Straight Party Capacity

References: Requirement III.4.1-5.5, Requirement III.4.6-7.8, Requirement III.4.8-4.10, Requirement III.4.8-4.11

Ballot format:

The first contest shall be described as follows:

> STRAIGHT PARTY. If you desire to vote a straight party ticket for all offices, vote for at most one party here. Votes for individual candidates in subsequent contests will override the straight party vote in those contests only.

The only ballot position shall be the following:

> Whig

There shall be L_R-1 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R-1 for *r*):

> This is Contest *r* in the 1-of-M Straight Party Capacity Test. Vote for at most one.

The first ballot position in each contest shall be of the following form:

> Contest *r* Whig Candidate (Whig)

The remaining ballot positions in each contest shall be of the following form (substituting numbers from 2 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. Each one shall vote straight party Whig in the first contest and skip the remaining contests (allowing the straight party vote to be effective in every contest).

4.2.6.6 N of M voting

Test 31 N-of-M Capacity

References: Requirement III.4.1-5.8, Requirement III.4.6-7.11, Requirement III.4.8-4.14

Ballot format:

L_R N-of-M contests where N = M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the N-of-M Capacity Test. Vote for at most L_C.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. All shall vote for every candidate in every contest.

4.2.6.7 N of M voting + Write-ins

Test 32 N-of-M Write-ins Capacity 1

References: Requirement III.4.1-5.4, Requirement III.4.1-5.8, Requirement III.4.6-7.5, Requirement III.4.6-7.11, Requirement III.4.8-4.8, Requirement III.4.8-4.14

Ballot format:

L_R N-of-M contests where $N = \left\lfloor \frac{L_C}{2} \right\rfloor$ and M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the N-of-M Write-ins Capacity 1 Test. Vote for at most N.

The first M-N ballot positions in each contest shall be of the following form (substituting numbers from 1 to M-N for *c*):

> Contest *r* Ballot Position *c*

The final N ballot positions shall be write-in opportunities.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. $\left\lfloor \frac{\min(L_B, L_T)}{2} \right\rfloor$ ballots shall vote for the first N candidates in each contest. The remaining ballots shall write in N candidates of the form "Contest *r* Write-in *n*," for *n* from 1 to N, in each contest.

Test 33 N-of-M Write-ins Capacity 2

References: Requirement III.4.1-5.4, Requirement III.4.1-5.8, Requirement III.4.6-7.5, Requirement III.4.6-7.11, Requirement III.4.8-4.8, Requirement III.4.8-4.14

Ballot format:

L_R N-of-M contests where N = M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the N-of-M Write-ins Capacity 2 Test. There are no candidates on the ballot. Write in at most L_C choices.

All L_C ballot positions in each contest shall be write-in opportunities.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_T)$ ballots shall be cast. In every contest, every ballot shall write in L_C choices of the form "Contest *r* Write-in *c*," substituting numbers 1 to L_C for *c*.

4.2.6.8 Cumulative voting

Test 34 Cumulative Voting Capacity

References: Requirement III.4.1-5.9, Requirement III.4.6-7.12, Requirement III.4.8-4.15

Ballot format:

L_R cumulative voting contests where M = N(*r*) = L_C.

The contests shall be described as follows (substituting L_C and numbers from 1 to L_R for *r*):

> This is Contest *r* in the Cumulative Voting Capacity Test. Cast at most L_C votes. You may cast multiple votes for the same candidate.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

A total of $\min\left(L_B, \left\lceil \frac{L_T}{L_C} \right\rceil\right)$ ballots shall be cast.

$\min\left(L_B, \left\lfloor \frac{L_T}{L_C} \right\rfloor\right)$ ballots shall cast L_C votes for the first ballot position in every contest.

Any ballot left over shall cast $L_T - \left\lfloor \frac{L_T}{L_C} \right\rfloor \times L_C$ votes for the first ballot position in every contest and undervote the rest.

4.2.6.9 Provisional / challenged ballots

Test 35 Provisional Ballot Capacity

References: Requirement III.4.6-7.14, Requirement III.4.8-4.3, Requirement III.4.8-4.4, Requirement III.4.9-16.3, Requirement III.4.9-17.2, Requirement III.4.9-26

Ballot format:

L_R 1-of-M contests where M = L_C.

The contests shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is Contest *r* in the Provisional Ballot Capacity Test. Vote for at most one.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to L_C for *c*):

> Contest *r* Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

A total of $\min(L_B, L_V)$ provisional ballots shall be cast and accepted for counting.

$\min\left(L_T, \max\left(\left\lceil \frac{\min(L_B, L_V)}{L_C} \right\rceil - c, 0\right)\right)$ ballots shall vote for Ballot Position *c* in every contest (*c* > 0).

Any ballots left over shall be blank (undervotes).

Additional pass criteria:

The number of provisional / challenged ballots read and counted shall be reported in compliance with Requirement III.4.9-16.3 and Requirement III.4.9-17.2.

4.2.6.10 Discussion

In all systems there is the possibility of a bottleneck in transmitting precinct results to central. Unfortunately there is no way to test this without having a large number of machines to work with—an unreasonable expectation for a system that is not even qualified yet.

Many more capacity tests could be written for different combinations of classes. There should be at least one torture test for the most common combination of classes, if there is one.

4.2.7 Error case tests

While most tests verify that the system does things that it is required to do, error case tests verify that the system does not do things that it is required not to do. As with all tests, passing an error case test does not conclusively show that the system is conforming, but failing an error case test conclusively shows that the system is non-conforming.

4.2.7.1 All systems

Test 36 Vote Register Overflow

References: Requirement III.4.6-17, Requirement III.4.6-17.1

Assumptions:

$L_B > L_T$

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Vote Register Overflow Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast L_T+1 ballots, all voting for Unopposed Candidate.

Additional pass criteria:

A DRE system shall not enable the L_T+1th ballot (Requirement III.4.6-17.1). The tabulator in a Marksense or Punchcard system shall not accept the L_T+1th ballot (Requirement III.4.6-17).

An audit log record shall exist for the counter reaching capacity event (Dangling ref: AuditLogCounterCapacityReq).

Discussion:

Overflow of the ballot counter is also implicitly tested by all of these register overflow tests.

Test 37 Undervote Register Overflow

References: Requirement III.4.6-17, Requirement III.4.6-17.1

Assumptions:

$L_B > L_T$

Ballot format:

1 1-of-M contest where M = 1.

The contest shall be described as follows:

> This is the only contest in the Undervote Register Overflow Test. There is only one candidate on the ballot.

The only ballot position in the contest shall be the following:

> Unopposed Candidate

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast L_T+1 ballots, all of them blank.

Additional pass criteria:

A DRE system shall not enable the L_T+1th ballot (Requirement III.4.6-17.1). The tabulator in a Marksense or Punchcard system shall not accept the L_T+1th ballot (Requirement III.4.6-17).

An audit log record shall exist for the counter reaching capacity event (Dangling ref: AuditLogCounterCapacityReq).

4.2.7.2 Marksense and Punchcard

Test 38 Overvote Register Overflow

References: Requirement III.4.6-17

Assumptions:

$L_B > L_T$

Ballot format:

1 1-of-M contest where M = 2.

The contest shall be described as follows:

> This is the only contest in the Overvote Register Overflow Test. There are two candidates on the ballot. Vote for one.

The ballot position in the contest shall be the following:

> Ballot Position 1
> Ballot Position 2

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast L_T+1 ballots. All shall vote for both ballot positions (overvote).

Additional pass criteria:

The tabulator shall not accept the L_T+1th ballot (Requirement III.4.6-17).

An audit log record shall exist for the counter reaching capacity event (Dangling ref: AuditLogCounterCapacityReq).

4.2.7.3 DRE

Test 39 DRE Overvoting

References: Requirement III.4.6-6.2

Ballot format:

1 1-of-M contest where M = 3.

The contest shall be described as follows:

> This is the only contest in the DRE Overvoting Test. There are three candidates on the ballot. Vote for at most one.

The ballot positions shall be the following:

> Ballot Position 1
> Ballot Position 2
> Ballot Position 3

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast one ballot that votes for both Ballot Position 2 and Ballot Position 3.

Additional pass criteria:

The DRE shall prevent the test lab from voting for more than one ballot position (Requirement III.4.6-6.2).

4.2.7.4 DRE + Write-ins

Test 40 DRE Write-ins Overvoting

References: Requirement III.4.6-6.2

Ballot format:

1 1-of-M contest where M = 2.

The contest shall be described as follows:

> This is the only contest in the DRE Write-ins Overvoting Test. There is one candidate on the ballot. Vote for at most one.

The first ballot position shall be the following:

> Ballot Position 1

The second ballot position in the contest shall be a write-in opportunity.

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast one ballot that both votes for Ballot Position 1 and writes in "Write-in candidate."

Additional pass criteria:

The DRE shall prevent the test lab from voting for more than one ballot position (Requirement III.4.6-6.2).

4.2.7.5 N of M voting

Test 41 N-of-M Vote Register Overflow

References: Requirement III.4.6-17, Requirement III.4.6-17.1

Assumptions:

$L_C \geq 10$

$L_B > \left\lfloor \frac{L_T}{10} \right\rfloor$

Ballot format:

1 N-of-M contest where N = M = 10.

The contest shall be described as follows (substituting numbers from 1 to L_R for *r*):

> This is the only contest in the N-of-M Vote Register Overflow Test. Vote for at most 10.

The ballot positions in each contest shall be of the following form (substituting numbers from 1 to 10 for *c*):

> Ballot Position *c*

There are no write-in ballot positions in this ballot format.

Reporting contexts: Single context.

Scenario:

The test lab shall attempt to cast $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$ ballots, all of which vote for all 10 ballot positions.

Additional pass criteria:

A DRE system shall not enable the $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$th ballot (Requirement III.4.6-17.1). The tabulator in a Marksense or Punchcard system shall not accept the $\left\lfloor \frac{L_T}{10} \right\rfloor + 1$th ballot (Requirement III.4.6-17).

An audit log record shall exist for the counter reaching capacity event (Dangling ref: AuditLogCounterCapacityReq).

Discussion:

A likely fault is for the system to only check that the count is less than L_T before enabling a ballot (should be less than L_T-9). This fault is masked in this test if L_T is a multiple of 10. We could test with other values, but it is possible to mask the fault in all tests by making L_T a product of those values.

4.2.7.6 Discussion

Mechanical / lever systems appear to be out of scope of the VVSG. But if not—what is supposed to happen in mechanical / lever systems when a register reaches its limit?

Other potential error case tests, not yet written:

Remote data delivery with intentional disruption of transmission (implementation-dependent).

Exception handling (requires creating an exception somehow—for testability, suggest adding a requirement for the capability to generate test exceptions. Could be of use in *in situ* L&A testing.)

4.3 Security

STS

4.4 Open-ended functional testing

4.4.1 Structural coverage

This text is retained from [2] II.A.4.3.3, "Software Module Test Case Design and Data," and II.6.3, "Testing Interfaces of System Components," with minor changes. As time permits, this section should be rewritten to enhance repeatability and reproducibility of the testing. At a minimum, the shalls should be formatted as requirements.

Text in [2] II.6.3 about regression testing.

The test lab shall review the vendor's program analysis, documentation, and, if available, module test case design. The test lab shall evaluate the test cases for each module, with respect to flow control parameters and data on both entry and exit. All discrepancies between the Software Specifications and the test case design shall be corrected by the vendor prior to initiation of the certification test.

If the vendor's module test case design does not provide conclusive coverage of all program paths, then the test lab shall perform an independent analysis to assess the frequency and consequence of error of the untested paths. The test lab shall define and execute additional module test cases as required to provide coverage of all modules containing untested paths with potential for untrapped errors. The test lab shall define pass criteria for implementation-dependent tests using the VVSG and the vendor-supplied system documentation to determine acceptable ranges of performance.

The test lab shall design and perform test procedures that test the interfaces of all system modules and subsystems with each other against the vendor's specifications. These tests shall be documented in the test lab's Certification Test Plan, and shall include the full range of system functionality provided by the vendor's specifications, including functionality that exceeds the specific requirements of the VVSG.

4.4.2 Functional coverage

This text is retained from [2] II.A.4.3.4, "Software Functional Test Case Design," and II.6.3, "Testing Interfaces of System Components," with minor changes. As time permits, this section should be rewritten to enhance repeatability and reproducibility of the testing. At a minimum, the shalls should be formatted as requirements.

[2] II.3, Functionality Testing, contains lots of related verbiage. Recycle as introductory text, or something?

The test lab shall review the vendor's functional test case designs. The test lab shall prepare a detailed matrix of system functions and the test cases that exercise them. The test lab shall also prepare a test procedure describing all test ballots, operator procedures, and the data content of output reports. Abnormal input data and operator actions shall be defined. Test cases shall also be designed to verify that the system is able to handle and recover from these abnormal conditions.

The vendor's test case design may be evaluated by any standard or special method appropriate.

In the event that the vendor's functional test data are insufficient, the test lab shall define and execute additional functional tests. The test lab shall define pass criteria for implementation-dependent tests using the VVSG and the vendor-supplied system documentation to determine acceptable ranges of performance.

Depending upon the design and intended use of the voting system, all or part of the functions listed below shall be tested.

a. Ballot preparation subsystem;

b. Test operations performed prior to, during, and after processing of ballots, including:

    1. Logic tests to verify interpretation of ballot formats, and recognition of precincts to be processed;
    2. Accuracy tests to verify ballot reading accuracy;
    3. Status tests to verify equipment statement and memory contents;
    4. Report generation to produce test output data; and
    5. Report generation to produce audit data records;

c. Procedures applicable to equipment used in the polling place for:

1. Opening the polls and enabling the acceptance of ballots;
2. Maintaining a count of processed ballots;
3. Monitoring equipment status;
4. Verifying equipment response to operator input commands;
5. Generating real-time audit messages;
6. Closing the polls and disabling the acceptance of ballots;
7. Generating election data reports;
8. Transfer of ballot counting equipment, or a detachable memory module, to a central counting location; and
9. Electronic transmission of election data to a central counting location; and

d. Procedures applicable to equipment used in a central counting place:

1. Initiating the processing of a ballot deck, programmable memory device, or other applicable media for one or more precincts;
2. Monitoring equipment status;
3. Verifying equipment response to operator input commands;
4. Verifying interaction with peripheral equipment, or other data processing systems;
5. Generating real-time audit messages;
6. Generating precinct-level election data reports;
7. Generating summary election data reports;
8. Transfer of a detachable memory module to other processing equipment;
9. Electronic transmission of data to other processing equipment; and
10. Producing output data for interrogation by external display devices.

For systems that use telecommunications capabilities, components that are located at the poll site or separate vote counting site shall be tested for effective interface, accurate vote transmission, failure detection, and failure recovery. For voting systems that use telecommunications lines or networks that are not under the control of the vendor (e.g., public telephone networks), the test lab shall test the interface of vendor-supplied components with these external components for effective interface, vote transmission, failure detection, and failure recovery.

4.5 Open-ended security testing

STS

Volume VI Supplemental guidance

This volume is not a part of the Voluntary Voting System Guidelines.

Chapter 1 Procedures required for correct system functioning

The requirements in this chapter provide context for what the functional requirements specify or, more often, for what they omit. These requirements do not pertain to the voting system and are not tested by an accredited test lab.

1.1 Requirements

R 1.1-1 bpgoodpaper

Election officials shall verify that paper ballots are produced in accordance with vendor specifications.

R 1.1-1.1 GeneratedID3490

Election officials shall ensure that paper ballots conform to vendor specifications for type of paper stock, weight, size, shape, size and location of punch or mark field used to record votes, folding, bleed through, and ink for printing.

Source: [2] I.2.3.1.3.1.c.

R 1.1-1.2 GeneratedID3493

For punchcards, election officials shall ensure that the vote response fields can be properly aligned with punching devices used to record votes.

Source: [2] I.2.3.1.1.2.b.

R 1.1-1.3 GeneratedID3496

For marksense ballots, election officials shall ensure that the timing marks align properly with the vote response fields.

Source: [2] I.2.3.1.1.2.c.

R 1.1-2 bp22mo

All printed copy records produced by the election database and ballot processing systems shall be labeled and archived for a period of at least 22 months after the election.

Source: Reworded from [2] I.2.2.11.

DISCUSSION

See also Requirement III.3.6-1 and Volume III Section 3.6.2.

1.1-3 bpeverybodyvotes

The voting process shall allow each eligible voter to cast a ballot.

Source: [2] I.2.4.2.b, generalized from DRE systems to the voting process.

DISCUSSION

See also Requirement III.4.6-13.1.

1.1-4 At most one cast ballot per voter

The voting process shall prevent a voter from casting more than one ballot in the same election.

Source: [2] I.2.4.2.d, generalized from DRE systems to the voting process.

DISCUSSION

See also Requirement III.4.6-1.1.

Impact: [2] restriction to DRE was probably to require DREs to do this automatically, which they don't.

1.1-5 bppreventwrongformat

The voting process shall prevent a voter from voting a ballot format to which he or she is not entitled.

Source: [2] I.2.4.2.c, generalized from DRE systems to the voting process.

DISCUSSION

See also Requirement III.4.2-1.2, Requirement III.4.2-1.3 and Requirement III.4.6-2.

Impact: [2] restriction to DRE was probably to require DREs to do this automatically, which they don't.

1.1-6 bpnochangevotes

The voting process shall prevent modification of the voter's vote after the ballot is cast.

Source: [2] I.2.4.3.3.n, generalized.

DISCUSSION

See also Requirement III.4.6-14, cast ballot.

1.1-7 bpnopeeking

The voting process shall prevent access to voted ballots until after the close of polls.

Source: [2] I.2.4.3.3.r, generalized.

DISCUSSION

See also Requirement III.4.7-1.

1.1-8 Paper-based tabulator, clearing misfeeds when ballot was read

If it is necessary to clear a misfed ballot that was read by a paper-based tabulator but became stuck on its way to the ballot box, election officials shall perform this task in the presence of a witness.

DISCUSSION

If an audit found that the contents of the ballot box and the records from the tabulator did not match, one would want to be able to rule out the possibility that something made its way into the ballot box while the tabulator was disconnected.

1.1-9 GeneratedID3521

All precincts shall account for all ballots pursuant to the current best practices for ballot accounting.

DISCUSSION

[Best Practice Reference Here]

Impact: Ballot accounting has repeatedly been raised as a serious concern both by the TGDC and by public comments.

1.1-10 Early voting, ballot accounting

In the presence of a witness, election officials shall record the value of the ballot counter from each tabulator at the end of each active period.

Source: Issue #1366, Issue #2143.

DISCUSSION

See Volume III Section 5.2. This procedure might be facilitated by designated functions of the voting equipment (i.e., printing of special early-voting end-of-day reports that include the timestamp, the value of the ballot counter, and little else).

1.1-11 Early voting, resumption practices

Election officials returning equipment to the ready state after it has been placed in the suspended state shall perform this operation in the presence of a witness, confirm that the equipment recorded no activity, and confirm that the ballot counter is unchanged from the value that was recorded when voting was suspended.

DISCUSSION

See Volume III Section 5.2. This procedure might be facilitated by designated functions of the voting equipment (i.e., printing of special early-voting resumption reports that include the timestamp, the value of the ballot counter, confirmation that nothing happened overnight, and little else).

1.1-12 unofficialbp

Any unofficial reports shall be clearly labelled as unofficial.

Source: [2] I.2.5.4.c, converted to procedural requirement.

Impact: See Volume I Section 1.6.

Volume VII Bibliography

[1] 1990 Voting Systems Standards, official version no longer available. Unofficial PDF produced by Joseph Lorenzo Hall, UC Berkeley, 2004-07-15, available at http://www.calvoter.org/issues/votingtech/vtstandards.html.

[2] 2002 Voting Systems Standards, available from http://www.eac.gov/election_resources/vss.html.

[3] IEEE Draft Standard for the Evaluation of Voting Equipment, draft P1583/D5.3.2b, 2005-01-04. Unpublished.

[4] Voluntary Voting System Guidelines Version I Initial Report, 2005-05-09, available from

http://www.epic.org/privacy/voting/eac_foia/.

[5] 2005 Voluntary Voting System Guidelines, 2006-02-01, available from http://www.eac.gov/vvsg_intro.htm.

[6] UML 2.0 Superstructure Specification, 2004-10-08, http://doc.omg.org/ptc/2004-10-02.

[7] Recommended Security Controls for Federal Information Systems, National Institute of Standards and Technology Special Publication 800-53, 2005-02, available from http://csrc.nist.gov/publications/nistpubs/.

[8] Fred R. Byers, Care and Handling of CDs and DVDs—A Guide for Librarians and Archivists, National Institute of Standards and Technology Special Publication 500-252, 2003-10, available from http://www.itl.nist.gov/div895/carefordisc/index.html.

[9] ISO 9706:1994, Information and documentation—Paper for documents—Requirements for permanence. Available from ISO, http://www.iso.org/.

[10] ISO 8601:2004, Data elements and interchange formats—Information interchange—Representation of dates and times. Available from ISO, http://www.iso.org/.

[11] ISO 17000:2004, Conformity assessment—Vocabulary and general principles. Available from ISO, http://www.iso.org/.

[12] Request For Proposal #08455, Kansas, 2005. Available from http://www.kssos.org/elections/05elec/Voting_Equipment_RFP.pdf, 2006-03-02.

[13] Request For Proposal #3443, Mississippi, 2005. Available from http://www.its.state.ms.us/rfps/3443.htm, 2006-01-25.

[14] Request for Proposals #108.6-03-001, North Dakota, 2003-10-31. Available from http://www.state.nd.us/hava/documents/docs/vsp-rfp-official.pdf, 2006-01-26.

[15] Solicitation #DG5502, Utah, 2004-07-09, available from http://purchasing.utah.gov/BidHeaders/8750.pdf, 2006-01-27.

[16] 2004 Presidential General Election Review Lessons Learned, http://www.truevotemd.org/Resources/Lessons_Learned.pdf.

[17] ES&S International, Comment on NIST Preliminary Report: Disposition of 2002 VSS Requirements, 2005-04-14, available at http://vote.nist.gov/ECPosStat.htm.

[18] Capability Maturity Model Integration, http://www.sei.cmu.edu/cmmi/.

[19] Philippe A. Martin, Petri Net Linear Form (PNLF), in "Using PIPE and Woflan," http://meganesia.int.gu.edu.au/~phmartin/workflow/PIPE/, 2005-07-22.

[20] New Shorter Oxford English Dictionary, Clarendon Press, Oxford, 1993.

[21] M. R. Moulding, "Designing for high integrity: the software fault tolerance approach," Section 3.4. In C. T. Sennett, ed., High-Integrity Software, Plenum Press, New York and London, 1989.

Notes

[1] [2] I.9.4: "Malfunctions that result from failure to comply fully with other requirements of this standard will not in every case warrant rejection;" [2] and [5] II.B.5: "Any uncorrected deficiency that does not involve the loss or corruption of voting data shall not necessarily be cause for rejection."

[2] At its September 2005 meeting, the TGDC supported the plan to revise and expand the prescriptive coding conventions addressing software integrity in a fairly conservative manner in the expectation that an expert review will follow.

[3] Commercial equipment and materials are identified in order to describe certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

[4] The understanding that a report is not mutable is important to the integrity of the voting process. For example, a vote data report represents a "snapshot" summary of vote data as it appeared at a specific time. Subsequent alterations to that vote data shall not result in any changes to the report that was generated, but a new report (with a later timestamp) may be generated to summarize the revised data. Analogously, a test lab forwards a report to the EAC at a specific time. If a system is sent back for retesting, the result is a new report. The original report is never "lost" due to subsequent changes.

[5] A prerequisite for device-level certification would be prescribing a system architecture so that the responsibilities of each device and the interfaces between those devices could be well-specified. Such prescription is undesirable. More importantly, even with a prescribed architecture, a device-level certification would provide no assurance that any particular system that included that component would function as specified. That assurance can only be obtained by evaluating the complete system in the configuration in which it is to be deployed.

[6] Portions of this section are derived from Section 5.6.2.2 of [3].

[7] This material is from an unapproved draft of a proposed IEEE Standard, P1583. As such, the

material is subject to change in the final standard. Because this material is from an unapproved draft, the IEEE recommends that it not be utilized for any conformance/compliance purposes. It is used at your own risk.

[8] Portions of this section are derived from Sections 5.6.2.2 and 6.6.4.2 of [3].

[9] In mathematical jargon, the word *domain* would be more appropriate than *range* for input variables; however, "range checking" is the common programming jargon.

[10] A compromised device could be programmed to give the correct answers during logic and accuracy testing but behave differently after polls are opened. This kind of fraud is detected and prevented through other means, beginning with the design review specified in Volume V Section 3.7 and Requirement III.3.1-1.

[11] A system conforming to the *Write-ins* class is required to be capable of counting and reporting totals for all candidates that are written in by voters. In some states, write-in votes are not counted unless they exactly match one of a list of registered, accepted write-in candidates. Voting systems may support reporting options that meet the requirements of such states without disruption to the counting logic.

[12] The use of preconditions and postconditions as we have recommended first appeared in C. A. R. Hoare, "An Axiomatic Basis for Computer Programming," Communications of the ACM, v. 12, n. 10, October 1969, pp. 576-580, 583, with ideas derived from Robert W Floyd, "Assigning Meanings to Programs," in J. T. Schwartz, ed., Mathematical Aspects of Computer Science: Proceedings of Symposia in Applied Mathematics, v. 19, American Mathematical Society, 1967, pp. 19-32.

[13] Informality is permitted here to bridge the gap between a programming language with informal semantics and the formality that we require. The size limit on callable units in source code (Requirement III.3.5-4.1) is intended to keep them small enough that preconditions and postconditions can be validated by inspection.

[14] The test lab should rely on authoritative information regarding the archivalness of various storage media. See also [8] and [9].

[15] Requirement III.3.7-1.3 and Requirement III.3.7-1.4 indicate acceptable designs.

[16] C.f. [2] I.3.2.6.2.2, "error-free" consolidation and reporting.

[17] Default tabulating rate is from [1] J-3.