UOCAVA PILOT PROGRAM TESTING REQUIREMENTS

Uniformed and Overseas Citizens Absentee Voting Act Pilot Program Testing Requirements

MARCH 24, 2010

Table of Contents

Section 1: Overview 5
1.1 Background 5
1.2 UOCAVA Remote Electronic Voting System Scope 8
1.3 Conformance Clause 10
1.4 Effective Date 14
Section 2: Functional Requirements 15
2.1 Accuracy 15
2.2 Operating capacities 18
2.3 Pre-Voting Capabilities 19
2.4 Voting Capabilities 20
2.5 Post Voting Capabilities 22
2.6 Audit and Accountability 24
2.7 Performance Monitoring 28
Section 3: Usability 29
3.1 General Principles 29
3.2 Alternative Languages 29
3.3 Clarity of Instructions 29
3.4 Voting Input Fields 29
3.5 Interaction Issues 30
3.6 Ballot Legibility 31
3.7 Perceptual Issues 32
Section 4: Software 33
4.1 Selection of Programming Languages 33
4.2 Selection of General Coding Conventions 33
4.3 Software Modularity and Programming 34
4.4 Structured Programming 34
4.5 Comments 35
4.6 Executable Code and Data Integrity 36
4.7 Error Checking 37
4.8 Recovery 39
Section 5: Security 42
5.1 Access Control 42
5.2 Identification and Authentication 45
5.3 Cryptography 48
5.4 Voting System Integrity Management 51
5.5 Communications Security 52
5.6 Logging 54
5.7 Incident Response 60
5.8 Physical and Environmental Security 60
5.9 Penetration Resistance 63
Section 6: Quality Assurance 67
6.1 General Requirements 67
6.2 Components from Third Parties 67
6.3 Responsibility for Tests 67
6.4 Parts and Materials, Special Tests, and Examinations 68
6.5 Quality Conformance Inspections 68
Section 7: Configuration Management 69
7.1 Scope 69
7.2 Configuration Identification 69
7.3 Baseline and Promotion Procedures 70
7.4 Configuration Control Procedures 70
7.5 Configuration Audits 71

Section 8: Technical Data Package 73
8.1 Scope 73
8.2 Implementation Statement 75
8.3 System Hardware Specification 75
8.4 Application Logic Design and Specification 77
8.5 System Security Specifications 89
8.6 System Test Specification 96
8.7 Configuration for Testing 98
Section 9: System Users Manual 100
9.1 Scope 100
9.2 System Overview 100
9.3 System Functionality Description 102
9.4 System Security Specification 103
9.5 Software 105
9.6 Setup Inspection 107
9.7 System Operations Manual 111
9.8 System Maintenance Manual 116
9.9 Personnel Deployment and Training Requirements 119
Appendix A: Definitions of Words with Special Meanings 121
Appendix B: List of References 125
Appendix A: Accuracy Test Case 131

Intentionally left blank

Section 1: Overview

1.1 Background

1.1.1 UOCAVA Pilot Projects

The Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) of 1986 protects the right to vote in federal elections for this defined category of citizens. UOCAVA sets out federal and state responsibilities to assist these voters in exercising their voting rights. The Secretary of Defense is the presidential designee responsible for the federal functions of the Act. The Federal Voting Assistance Program (FVAP) administers this law on behalf of the Secretary of Defense and works cooperatively with other federal agencies and state and local election officials to carry out its provisions.

UOCAVA legislation was enacted before the advent of today's global electronic communications technology. Consequently it relied on U.S. domestic and military mail systems as well as foreign postal systems for the worldwide distribution of election materials. By the mid-1990s it became apparent that the mail transit time and unreliable delivery posed significant barriers for many UOCAVA citizens, preventing them from successfully exercising their right to vote. At the same time the Internet was being widely adopted by businesses, governments and the general public. Therefore it was a natural development for FVAP and states to consider the potential of the Internet as an alternative to the "by-mail" UOCAVA process.

FVAP sponsored Voting Over the Internet (VOI), a small pilot project for the November 2000 general election, to examine the feasibility of using Internet technology. Four states participated in this experiment, which enabled voters to use their own personal computers to securely register to vote, request and receive absentee ballots, and return their voted ballots. Following the successful completion of the VOI project, in the Fiscal Year 2002 National Defense Authorization Act (§1604 of P.L. 107-107:115 Stat.1277), Congress instructed the Secretary of Defense to carry out a larger demonstration project for the November 2002 general election. This project was to be "carried out with participation of sufficient numbers of absent uniformed services voters so that the results are statistically significant".

Since there was not sufficient time to define and implement a large project for 2002, the project was planned for implementation for the November 2004 election. Seven states agreed to participate and worked with FVAP to develop system requirements and operating procedures. However, the Secure Electronic Registration and Voting Experiment (SERVE) was cancelled before it was deployed due to concerns raised by several computer scientists. These individuals contended that the use of personal computers over the Internet could not be made secure enough for voting and consequently called for the project to be terminated. The Department of Defense, citing a lack of public confidence in the SERVE system, decided the project could not continue under these circumstances.

In response to this development, the Fiscal Year 2005 National Defense Authorization Act (§567 of P.L. 108-375;118 Stat.119) repealed the requirement for the Secretary of Defense to conduct an electronic voting demonstration project "until the first regularly scheduled general election for federal office which occurs after the Election Assistance Commission (EAC) notifies the Secretary that the Commission has established electronic absentee voting guidelines and certifies that it will assist the Secretary in carrying out the project". Pursuant to this legislation, in September 2005, the EAC requested its voting system advisory group, the Technical Guidelines Development Committee (TGDC), to add this subject on their research agenda; however the request was declined.

Since the State of Florida conducts its own voting system certification process, Okaloosa County, Florida, decided to field a small pilot for the 2008 general election. Instead of allowing voters to use their own personal computers, Okaloosa County set up staffed absentee voting locations in England, Germany and Japan. Voters that visited these sites were allowed to cast their ballots electronically using laptop computers supplied by the Supervisor of Elections office. Election workers that staffed these sites verified voter identity and eligibility using an on-line connection to the voter registration system. A paper record of each vote was printed and used to verify the electronic results when the votes were tabulated.

1.1.2 Testing Pilot Systems

Most states require voting systems to undergo a testing and certification process before the system may be used in an election. This provides a level of assurance that the system provides the required functionality and operates reliably and securely. The four states participating in the VOI project agreed to test that system utilizing the Department of Defense Information Technology Security Certification and Accreditation (DITSCAP) process combined with the State of Florida Division of Elections Voting Systems Certification process. The testing regimen planned for the SERVE system was a combined DITSCAP, National Association of State Election Directors (NASED), and State of Florida certification and accreditation process. The system used for Okaloosa County's remote voting pilot was tested and certified by the State of Florida Division of Elections.

Due to the nature of these new systems, existing voting system standards were not sufficient for testing specific aspects. Therefore, additional security requirements were needed to test the use of digital signatures, cryptography and secure communications protocols. The hardware and software standards, developed for DRE and optical scan systems used in polling places, also needed to be revised to reflect the characteristics of the remote voting technologies. Each of the pilot projects established a working group, comprised of election officials, security experts and test engineers, to define the additional requirements needed to supplement the existing voting system standards. Reference materials for the working groups came from various national and international sources of information technology standards, such as the Federal Information Processing Standards (FIPS), Common Criteria, and the International Standards Organization. These efforts resulted in testing requirements documents that were specific to the technical features of each of the pilot systems, which supplied the criteria for testing and certifying these particular pilot systems.

Since 2008, several states have enacted legislation enabling them to conduct electronic voting projects for UOCAVA voters, beginning with the 2010 elections. To be prepared to support the states with these projects, in July

2009 the EAC convened a UOCAVA Working Group to consider how to adapt the EAC's Testing and Certification Program to accommodate UOCAVA pilot systems. It was concluded that two products were needed: a modified set of system testing requirements; and a revised testing and certification process. It was determined that the working group would assist the EAC in drafting the testing requirements and EAC staff would adapt the certification process to accommodate the UOCAVA pilot program.

The EAC UOCAVA Working Group has taken much the same approach as the pilot project working groups. The source materials drawn on for this effort included: the Voluntary Voting System Guidelines (VVSG) 1.0 ; the VVSG 1.1; the VVSG 2.0; the VOI, SERVE and Okaloosa Project requirements documents; FIPS; and NIST Special Publications. One significant difference in the EAC Working Group approach was the technology scope covered by the requirements. The VOI, SERVE and Okaloosa system requirements were tailored specifically for the particular system implementations developed for those projects. However, since many different types of remote voting systems could be submitted to the EAC certification program, the EAC Working Group defined generic system requirements to provide for system design flexibility.

1.1.3 Scope of EAC Pilot Project Testing Requirements

Pilot projects are small in scale and short in duration. Consequently, certification for pilot systems needs to be quicker and less expensive than the regular process currently used for conventional systems with an expected life of more than 10 years. Nevertheless, since actual votes will be cast using the voting systems utilized in the pilot project, the certification process must retain sufficient rigor to provide reasonable assurance that the pilot systems will operate correctly and securely.

There is a fundamental dichotomy in complexity in remote voting architectures: those where the voting platform is controlled (e.g., provided by the election jurisdiction); and those where it is not controlled (e.g., the voter uses his own personal computer). Since the EAC planned to have the pilot certification process ready for implementation during the first half of 2010, it was decided that the EAC would focus its efforts on controlled platform architectures servicing multiple jurisdictions. This is a highly secure remote voting solution and the Okaloosa Project provides an implementation example for reference. Defining requirements for this class of system architecture was determined to provide a reasonable test case that could be completed within the available timeframe. In addition, most of the core system processing functions are the same for both types of architectures, so a substantial number of requirements will carry over as this work is expanded to include other methods of remote electronic voting.

1.1.4 Next Steps

While the EAC was working to ensure that the pilot certification effort was underway, legislation dealing with a number of UOCAVA voting issues were under consideration by Congress. Ultimately, passed as part of the Fiscal Year 2010 National Defense Authorization Act (NDAA) (§581 of P.L. 111-84), the Military and Overseas Voters Empowerment Act contains a provision allowing the Secretary of Defense to establish one or more pilot programs to test the feasibility of new election technology for UOCAVA voters. This provision requires the EAC and the National Institute of Standards and Technology (NIST) to provide best practices or standards to support these pilot programs, "in accordance with electronic absentee voting guidelines established under" the earlier FY2005 NDAA. In December 2009, the EAC directed the TGDC to begin this work as a top research priority. The EAC expects this work to result in the comprehensive set of remote electronic voting system guidelines as mandated by the FY2005 NDAA. The TGDC has been tasked to consider the full range of remote voting architectures, including instances where the voter can use his own personal computer for voting. The pilot testing requirements, that the EAC is currently developing, will be provided to the TGDC as the basis and starting point for their research and deliberations.

1.2 UOCAVA Remote Electronic Voting System Scope

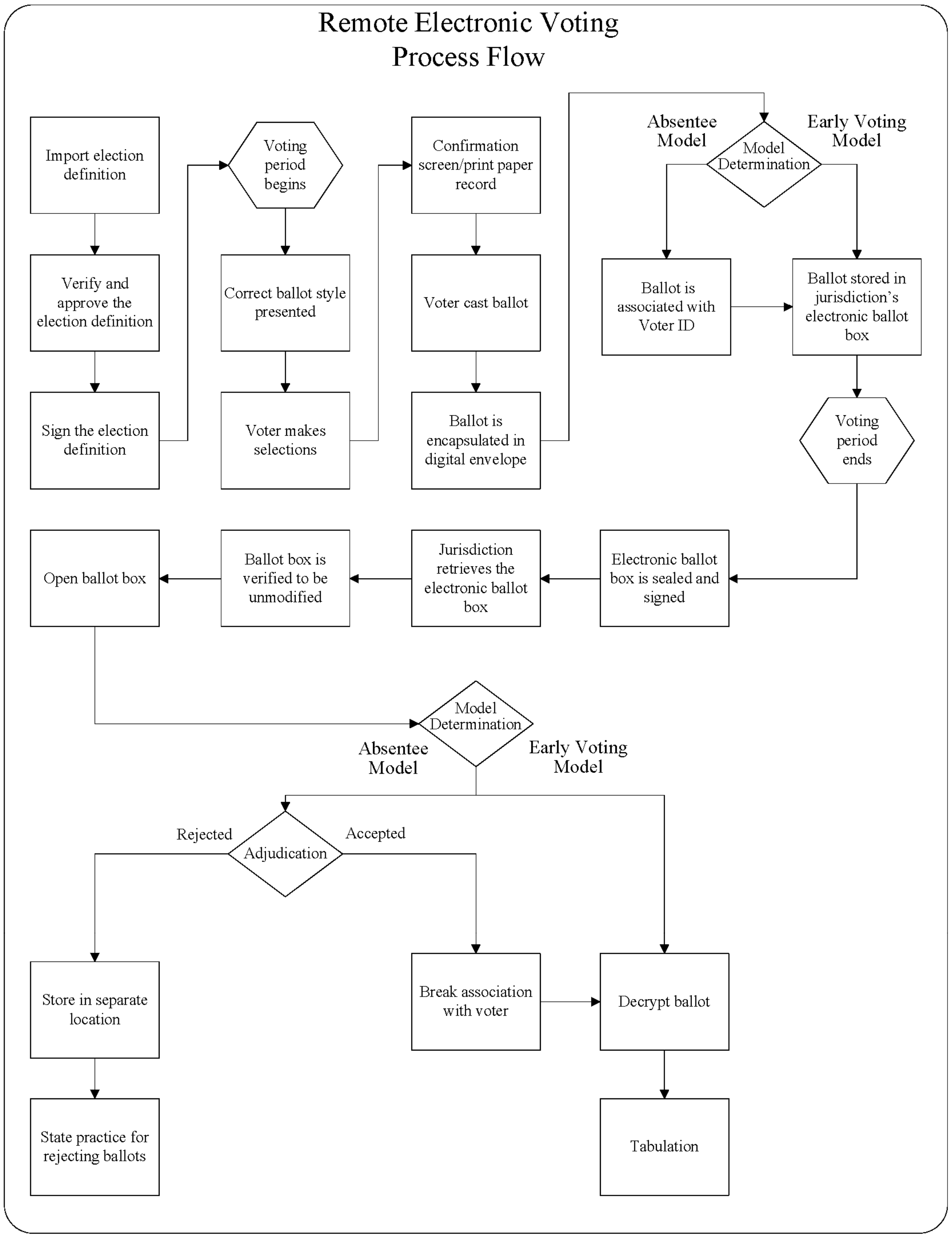
An initial step in a system certification process is to define the scope of what should be included in the certification. UOCAVA pilot project systems operate as adjuncts to the various polling site systems used by the jurisdictions that are participating in the pilot project. The systems will require linkages to the local Election Management System in order to obtain election definition data and to report election results. The systems also will require linkages to the Voter Registration Database to authenticate voters and determine their eligibility to vote, match them with the correct ballot style, and record voter history. Processes that are handled procedurally for polling place systems may be implemented in a software application in a remote electronic system. Another difference is that the UOCAVA voting period currently extends for 45 days. So these absentee systems have to be in operation for a fairly long time before polling places are open. Most, if not all, states prohibit tabulation of absentee ballots until the polls are closed, so voted ballots may have to be stored on the system for several weeks. Therefore, the functions and the architectures of remote voting systems demonstrate some notable differences from conventional polling site systems.

Figure 1-1 illustrates a generic process flow for remote electronic voting that does not presuppose any particular architectural solution. Even at this high level of abstraction, two alternative processing paths are needed to accommodate differences in individual state requirements. The first path, called the absentee model, has two distinguishing features. This is essentially an electronic rendering of the UOCAVA by-mail process. In this path, the voter's identity must remain linked to the cast ballot until the close of the voting period. At that time an adjudication is made by the local jurisdiction on whether to accept or not accept the ballot. If the ballot is accepted, any identifiable link to the voter is removed. The now anonymous ballot is placed in the ballot box to be tabulated. If the ballot is rejected, the link is not removed and the disposition of the 'unopened' ballot is made in accordance with individual state procedures.

The second path, called the early voting model, does not maintain any association between the voter and the cast ballot. When the voter presses the 'Vote' button and receives notification that the ballot has been recorded, the ballot goes directly into the ballot box. There is no ballot adjudication step and therefore no need to maintain a connection between the voter and the ballot.

There are many of ways in which systems can be designed to perform these absentee functions. However, as noted in 1.1, only one type of system architecture is covered in this document. The voting platform envisioned is a remote voting location staffed and managed by election workers, which services a number of different election jurisdictions. The election workers verify the voter's identity and eligibility to vote and update voter history in much the same manner as poll workers perform these functions at a polling place. The voter uses a laptop computer or similar device provided by the project to view the ballot, make his selections and cast his ballot. For security purposes, no vote data is permanently retained by the voting device. The cast ballot is transmitted to an electronic ballot box which is stored at another location. The voting device is equipped with a printer to produce a paper record of the voter's choices that the voter can review for verification purposes. The paper record must be deposited in a secure receptacle and transported to the appropriate jurisdiction for system audit purposes. Other elements of the system architecture are not specified. All systems submitted for pilot certification must support both the absentee and the early voting models.

Figure 1-1 UOCAVA Process


Remote Electronic Voting Process Flow
Import election definition
Verify and approve the election definition
Sign the election definition
Voting period begins
Correct ballot style presented
Voter makes selections
Confirmation screen/print paper record
Voter cast ballot
Ballot is encapsulated in digital envelope
Absentee Model
Model Determination
Early Voting Model
Ballot is associated with Voter ID
Ballot stored in jurisdiction's electronic ballot box
Voting period ends
Electronic ballot box is sealed and signed
Jurisdiction retrieves the electronic ballot box
Ballot box is verified to be unmodified
Open ballot box
Model Determination
Absentee Model
Early Voting Model
Rejected
Adjudication
Accepted
Store in separate location
State practice for rejecting ballots
Break association with voter
Decrypt ballot
Tabulation

1.3 Conformance Clause

1.3.1 Scope and Applicability

This document defines requirements for conformance of remote electronic voting systems intended for use in UOCAVA pilot programs that manufacturers of such systems SHALL meet pursuant to EAC pilot program certification. These pilot programs consist of staffed kiosks connected to multiple state data centers with paper records to support system performance validation. Pilot system functionality excludes voter registration and election management system except for defined data interchange interfaces. This document also provides the framework, procedures, and requirements that voting system testing labs (VSTLs) and manufacturers responsible for the certification testing of such pilot program systems SHALL follow. The requirements and procedures in this document may also be used by states to certify remote electronic voting systems for their own pilot programs.

This document defines the minimum requirements for remote electronic voting systems in the context of pilot programs conducted by states and local jurisdictions and the process of testing these systems. The requirements are intended for use by:

- Designers and manufacturers of voting systems;
- VSTLs performing the analysis and testing of systems in support of the EAC certification process;
- Election officials, including ballot designers and officials responsible for the installation, operation, and maintenance of voting machines for UOCAVA pilot programs; and
- VSTLs and consultants performing the state certification of voting systems for pilot programs.

Minimum requirements specified in this document include:

- Functional capabilities;
- Performance characteristics, including security;
- Documentation; and
- Test evaluation criteria.

1.3.2 Conformance Framework

This section provides the framework in which conformance is defined. It identifies the entities to which these requirements apply, the relationships among the various entities, the structure of the requirements, and the terminology used to indicate conformance.

1.3.2.1 Applicable entities

The requirements, prohibitions and options specified in these requirements apply to remote electronic voting systems, voting system manufacturers, and VSTLs. These requirements apply to all systems submitted for pilot certification under the EAC program.

1.3.2.2 Requirements of entities

It is the voting system manufacturer that must implement these requirements and provide the necessary documentation for the system. In order to claim conformance to the requirement, the voting system manufacturer SHALL satisfy the specified requirements. The voting system manufacturer SHALL successfully complete the prescribed test campaign with an EAC VSTL in order to obtain EAC certification.

The VSTL SHALL satisfy the requirements for conducting pilot program certification testing. Additionally, as indicated in the document, certain requirements SHALL be tested by the manufacturer rather than the VSTL. The VSTL may use an operational environment emulating that used by election officials as part of their testing to ensure that the voting system can be configured and operated in a secure and reliable manner according to the manufacturer's documentation and as specified by the requirements. The VSTL SHALL coordinate and deliver the requisite documentation, including a Test Plan and a Test Report, to the EAC for review and approval.

The EAC SHALL review the test results and associated documentation from both the VSTL and the manufacturer and make a determination that all requirements have been appropriately tested and the test results are acceptable. The EAC may conduct audits of manufacturer testing to ensure its adequacy. The EAC will issue a pilot program certification number that indicates conformance of the specified system to these requirements.

1.3.3 Extensions

Extensions are additional functions, features, and/or capabilities included in a voting system that are not required by this document. To accommodate the needs of states that may impose additional requirements and to accommodate changes in technology, this document allows extensions. The use of extensions SHALL NOT contradict nor cause the nonconformance of functionality required by this document.

1.3.4 Implementation Statement

The implementation statement SHALL describe the remote electronic voting system and SHALL document the requirements that have been implemented by the voting system. It SHALL also identify optional features and capabilities supported by the voting system, as well as any extensions (i.e., additional functionality beyond what is required in this document). The implementation statement SHALL include a checklist identifying all the requirements for which a claim of conformance is made.

The implementation statement SHALL be submitted with the manufacturer's application to the EAC for pilot program certification testing. It SHALL provide a concise summary and narrative description of the voting system's capabilities. It SHALL include identifying information about the voting system, including the hardware and software components, version number and date.

1.3.5 Equivalent Configurations

1.3.5.1 Background

Under the current EAC certification program (prior to this document), the scope of certification is very specific and extends only to the exact voting system configuration tested. The certificate specifically identifies each of the various configurations of the voting system's components that were tested and certified, including the OS version and service pack, as well as the CPU. Any modification to the system not authorized by the EAC will void the certificate. The certificate is applicable to the system configuration that has been tested during certification and is not applicable when any modification to hardware, software or COTS products has occurred.

There is a tradeoff between requiring the exact configuration that was tested and certified to be deployed and allowing "equivalent configurations" that have been tested by the voting system manufacturer and attested to perform identically on these configurations. Requiring only exact configurations that have been certified to be deployed guarantees that the customer is using the actual system that has been tested by the VSTL, but does not allow the flexibility needed to accommodate routine and expected changes to COTS systems. The requirements in this document are designed to allow for such flexibility.

1.3.5.2 Procedures for changes to baseline configuration

Testing for UOCAVA Pilot Certification is conducted by the VSTL and voting system manufacturer on the baseline configuration consisting of:

1. Specific hardware;
2. Major Version of operating system and third-party COTS applications.
    - Major Versions are changed when an updated version is downloaded; major versions are not considered changed when a patch is applied to fix an individual item.
    - In Microsoft Operating Systems, Major Versions would include Service Packs– New Service Packs would be considered a different Major Version.
    - Downloading patches (i.e., security) would not be considered a change to the Major Version. However, manufacturers SHALL create a log of all patches downloaded and supply them to the EAC upon request.

Any change to hardware or software (Major Versions) SHALL be regression tested by the voting system manufacturer to ensure that all requirements affected by the change have been adhered to. Regression testing SHALL be documented and legally affirmed to by the manufacturer, and accepted by the EAC. Regression testing SHALL be done by the manufacturer when the EAC certified version differs from the one being deployed in any of the following ways:

a. Any hardware is changed. However, de minimis changes, as defined in the EAC Certification Manual, SHALL NOT undergo regression testing;

b. Any change to Major Version of the OS is made; and

c. Any major change to a third-party COTS application is made.

All regression testing by manufacturers SHALL include accuracy and reliability testing. Other tests SHALL be repeated for requirements closely related to the functionality that was modified with the hardware or software (Major Version) changes.

Any change to the voting system application not covered by 3 a, b or c SHALL undergo testing by the VSTL.

Test Reports describing the manufacturer regression testing SHALL be submitted to the EAC. The EAC may conduct random audits to ensure that the manufacturer regression testing performed was sufficient.

1.3.6 Requirements Language and Structure

1.3.6.1 Language

Understanding how language is used is a pre-requisite to understanding this document. Language in this document is divided into two categories: normative, i.e., the requirements language itself, and informative. Normative language is prescriptive and must be followed to obtain conformance to this document and ultimately EAC certification. Informative parts of this document include discussion, examples, extended explanations, and other matter that are necessary for proper understanding of the requirements and how to ensure conformance. Informative text is not prescriptive and serves to clarify requirements.

Normative language is specifically for requirements. The following keywords are used within requirements text to indicate the conformance aspects of the requirement:

- SHALL indicates a mandatory requirement to do something;
- SHALL NOT indicates a mandatory requirement not to do something.

1.3.6.2 Structure of requirements

Each remote electronic voting system requirement in this document is identified according to a hierarchical scheme in which higher-level requirements (e.g., "Voter SHALL make application to request to vote absentee by remote electronic method") are supported by lower-level requirements (e.g., "The application SHALL include name, date of birth, legal residence address, etc."). Thus, requirements are nested. When the nesting hierarchy has reached four levels (i.e., 1.1.1.1), further nested requirements are designated with lowercase letters, then roman numerals. Therefore, all requirements are traceable by a distinct reference.

Some requirements are directly testable and some are not. Lower-level requirements (i.e., leaf-node requirements that have no requirements directly beneath them) are directly testable. Higher-level requirements (i.e., requirements with directly testable requirements beneath them) are not directly testable. Higher-level requirements 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 requirement. Satisfying all the lower-level requirements will result in satisfying the corresponding higher-level requirement. Thus, VSTLs need to only directly test lower-level requirements, not higher-level requirements. However, if non-conformance with a higher-level requirement is determined through any other means (e.g., OEVT testing, inspection, etc.) then the voting system is deemed not to conform to that higher-level requirement.

1.4 Effective Date

The UOCAVA Pilot Program Testing requirements SHALL become effective for pilot certification testing upon adoption by the EAC. At that time, all pilot systems submitted for EAC certification SHALL be tested for conformance with these requirements.

These requirements are voluntary in that each of the states can decide whether to require the voting systems used in pilot programs for their state to have an EAC certification. States may decide to adopt these requirements in whole or in part at any time, irrespective of the effective date. In addition, states may specify additional requirements that pilot voting systems used in their jurisdictions must meet. The EAC certification program does not, in any way, pre-empt the ability of the states to have their own voting system certification process.

Section 2: Functional Requirements

2.1 Accuracy

Voting system accuracy addresses the accuracy of data for each of the individual ballot selections that could be selected by a voter, including the positions that are not selected. Accuracy is defined as the ability of the voting system to capture, record, store, consolidate and report the specific selections and absence of selections, made by the voter on each ballot without error.

For each processing function in the following list, the voting system SHALL achieve a target error rate of no more than one in 10,000,000 ballot positions, with a maximum acceptable error rate in the test process of one in 500,000 ballot positions. Types of functions include:

- Recording voter selections of candidates and contest into voting data storage;
- Recording voter selections into ballot image storage independently from voting data storage; and
- Consolidation of vote selection data from multiple voting sites to generate jurisdiction-wide vote totals.

2.1.1 Components and Hardware

2.1.1.1 Component accuracy

Memory hardware, such as semiconductor devices and magnetic storage media, SHALL be accurate.

Test Method:* *Functional

Test Entity:* *VSTL

2.1.1.2 Equipment design

The design of equipment in all voting systems SHALL provide for protection against mechanical, thermal, and electromagnetic stresses that impact voting system accuracy.

Test Method:* *Functional

Test Entity:* *VSTL

2.1.1.3 Voting system accuracy

To ensure vote accuracy, all voting systems SHALL:

a. Record the election contests, candidates, and issues exactly as defined by election officials;

b. Record the appropriate options for casting and recording votes;

c. Record each vote precisely as indicated by the voter and be able to produce an accurate report of all votes cast;

d. Include control logic and data processing methods incorporating parity and check-sums (or equivalent error detection and correction methods) to demonstrate that the voting system has been designed for accuracy; and

e. 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.

Test Method:* *Functional

Test Entity:* *VSTL

2.1.2 Environmental Range

All voting systems SHALL meet the accuracy requirements over manufacturer specified operating conditions and after storage under non-operating conditions.

Test Method:* *Functional

Test Entity:* *VSTL

2.1.3 Content of Data Verified for Accuracy

2.1.3.1 Election management system accuracy

Voting systems SHALL accurately record all election management data entered by the user, including election officials or their designees.

Test Method:* *Functional

Test Entity:* *VSTL

2.1.3.2 Recording accuracy

For recording accuracy, all voting systems SHALL:

a. Record every entry made by the user;

b. Accurately interpret voter selection(s) and record them correctly to memory;

c. Verify the correctness of detection of the user selections and the addition of the selections correctly to memory;

d. Verify the correctness of detection of data entered directly by the user and the addition of the selections correctly to memory; and

e. Preserve the integrity of election management data stored in memory against corruption by stray electromagnetic emissions, and internally generated spurious electrical signals.

Test Method:* *Functional

Test Entity: ***VSTL***

2.1.4 Telecommunications Accuracy

The telecommunications components of all voting systems SHALL meet the requirements specified in section 2.1.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.1.5 Accuracy Test Content

Voting system accuracy SHALL be verified by a specific test conducted for this objective. The overall test approach is described in Appendix C.

2.1.5.1 Simulators

If a simulator is used, it SHALL be verified independent of the voting system in order to produce ballots as specified for the accuracy testing.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.1.5.2 Ballots

Ballots used for accuracy testing SHALL include all the supported types (i.e., rotation, languages, etc.) of contest and election types (primary, general)

Test Method: ***Functional***

Test Entity: ***VSTL***

2.1.6 Reporting Accuracy

Processing accuracy is defined as the ability of the voting system to process stored voting data. Processing includes all operations to consolidate voting data after the voting period has ended.

UOCAVA voting systems SHALL produce reports that are consistent with no discrepancy among reports of voting data.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.2 Operating capacities

2.2.1 Maximum Capacities

The manufacturer SHALL specify at least the following maximum operating capacities for the voting system (i.e. server, vote capture device, communications links):

- Throughput,
- Memory,
- Transaction processing speed, and
- Election constraints:
  - Number of jurisdictions
  - Number of ballot styles per jurisdictions
  - Number of contests per ballot style
  - Number of candidates per contest

Test Method: Functional

Test Entity: VSTL

2.2.1.1 Capacity testing

The voting system SHALL achieve the maximum operating capacities stated by the manufacturer in section 2.2.1

Test Method: Functional

Test Entity: VSTL

2.2.2 Operating Capacity notification

The voting system SHALL provide notice when any operating capacity is approaching its limit.

Test Method: Functional

Test Entity: VSTL

2.2.3 Simultaneous Transmissions

The voting system SHALL protect against the loss of votes due to simultaneous transmissions.

Test Method: Functional

Test Entity: VSTL

2.3 Pre-Voting Capabilities

2.3.1 Import and Verify Election Definition

2.3.1.1 Import the election definition

The voting system SHALL:

a. Keep all data logically separated by, and accessible only to, the appropriate state and local jurisdictions;

b. Provide the capability to import or manually enter ballot content, ballot instructions and election rules, including all required alternative language translations from each jurisdiction;

c. Provide the capability for the each jurisdiction to verify that election definition was imported accurately and completely;

d. Support image files (e.g., jpg or gif) and/or a handwritten signature image on the ballot so that state seals, official signatures and other graphical ballot elements may be properly displayed; and

e. Support multiple ballot styles per each local jurisdiction.

Test Method:* *Inspection/Functional

Test Entity:* *VSTL

2.3.1.2 Protect the election definition

The voting system SHALL provide a method to protect the election definition from unauthorized modification.

Test Method:* *Functional

Test Entity:* *VSTL

2.3.2 Readiness Testing

2.3.2.1 Voting system test mode

The voting system SHALL provide a test mode to verify that the voting system is correctly installed, properly configured, and all functions are operating to support pre-election readiness testing for each jurisdiction.

Test Method:* *Functional

Test Entity:* *VSTL

2.3.2.2 Test data segregation

The voting system SHALL provide the capability to zero-out or otherwise segregate test data from actual voting data.

Test Method:* *Functional

Test Entity: ***VSTL***

2.4 Voting Capabilities

2.4.1 Opening the Voting Period

2.4.1.1 Accessing the ballot

The voting system SHALL:

a. Present the correct ballot style to each voter;

b. Allow the voting session to be canceled; and

c. Prevent a voter from casting more than one ballot in the same election.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.2 Casting a Ballot

2.4.2.1 Record voter selections

The voting system SHALL:

a. Record the selection and non-selection of individual vote choices for each contest and ballot measure;

b. Record the voter's selection of candidates whose names do not appear on the ballot, if permitted under state law, and record as many write-ins as the number of candidates the voter is allowed to select;

c. Prohibit the voter from accessing or viewing any information on the display screen that has not been authorized and preprogrammed into the voting system (i.e., no potential for display of external information or linking to other information sources);

d. Allow the voter to select his preferences on the ballot in any legal number and combination;

e. Provide unambiguous feedback regarding the voter’s selection, such as displaying a checkmark beside the selected option or conspicuously changing its appearance;

f. Indicate to the voter when no selection, or an insufficient number of selections, has been made for a contest (e.g., undervotes);

g. Provide the voter the opportunity to correct the ballot for an undervote before the ballot is cast;

h. Prevent the voter from making more than the allowable number of selections for any contest (e.g., overvotes); and

i. In the event of a failure of the main power supply external to the voting system, provide the capability for any voter who is voting at the time to complete casting a ballot, allow for the successful shutdown of the voting system without loss or degradation of the voting and audit data, and allow voters to resume voting once the voting system has reverted to back-up power.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.2.2 Verify voter selections

The voting system SHALL:

a. Generate a paper record identifier. This SHALL be a random identifier that uniquely links the paper record with the cast vote record;

b. Produce a paper record each time the confirmation screen is displayed;

c. After reviewing the confirmation screen and paper record, a voter SHALL be able to either cast the ballot or return to the vote selection process to make changes; and

d. Prompt the voter to confirm his choices before casting the ballot, signifying to the voter that casting the ballot is irrevocable and directing the voter to confirm his intention to cast the ballot.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.2.3 Cast ballot

The voting system SHALL:

a. Store all cast ballots in a random order;

b. Notify the voter after the vote has been stored successfully that the ballot has been cast;

c. Notify the voter that the ballot has not been cast successfully if it is not stored successfully, including storage of the ballot, and provide clear instruction as to steps the voter should take to cast his ballot should this event occur; and

d. Prohibit access to voted ballots until such time as state law allows for processing of absentee ballots.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.2.4 Ballot linking to voter identification

2.4.2.4.1 Absentee model

The cast ballot SHALL be linked to the voter’s identity without violating the privacy of the voter.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.2.4.2 Early voting model

The cast ballot SHALL NOT be linked to the voter’s identity.

Test Method: ***Inspection***

Test Entity: ***VSTL***

2.4.3 Vote Secrecy

2.4.3.1 Link to voter

The voting system SHALL be capable of producing a cast vote record that does not contain any information that would link the record to the voter.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.4.3.2 Voting session records

The voting system SHALL NOT store any information related to the actions performed by the voter during the voting session.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.5 Post Voting Capabilities

2.5.1 Ballot Box Retrieval and Tabulation

2.5.1.1 Seal and sign the electronic ballot box

The voting system SHALL seal and sign the electronic ballot box, by means of a digital signature, to protect the integrity of its contents.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.5.1.2 Electronic ballot box retrieval

The voting system SHALL allow each jurisdiction to retrieve its electronic ballot box.

Test Method: Functional

Test Entity: VSTL

2.5.1.3 Electronic ballot box integrity check

The voting system SHALL perform an integrity check on the electronic ballot box verifying that is has not been tampered with or modified before opening.

Test Method: Functional

Test Entity: VSTL

2.5.1.4 Open ballot box

The voting system SHALL allow only an authorized entity to open the ballot box.

Test Method: Functional

Test Entity: VSTL

2.5.1.5 Absentee model

2.5.1.5.1 Adjudication

The voting system SHALL allow the designation of electronic ballots as “accepted” or “not accepted” by an authorized entity.

Test Method: Functional

Test Entity: VSTL

2.5.1.5.2 Digital envelope removal

After a ballot is accepted, the voting system SHALL remove the digital envelope breaking all correlation between the voter and the ballot.

Test Method: Functional

Test Entity: VSTL

2.5.1.6 Ballot decryption

The decryption process SHALL remove all layers of encryption, producing a record that is in clear text.

Test Method: Functional

Test Entity: VSTL

2.5.2 Tabulation

2.5.2.1 Tabulation report format

The voting system SHALL have the capability to generate a tabulation report of voting results in an open and non-proprietary format.

Test Method: Functional

Test Entity: VSTL

2.6 Audit and Accountability

2.6.1 Scope

This section presents requirements for the voting system to provide the capability for certain types of audits listed below. The audits work together to ensure independent agreement between what is presented to the voters by the paper record and the electronic tabulation results. The audits addressed in this section are:

a. Hand audit - Validate electronic tabulation results ballot style through comparison with results of manual count of paper records; and

b. Random sampling comparison of ballot images and the corresponding paper records.

2.6.2 Electronic Records

In order to support independent auditing, a voting system SHALL be able to produce electronic records that contain the necessary information in a secure and usable manner. Typically, this includes records such as:

- Vote counts;
- Counts of ballots recorded;
- Paper record identifier;
- Event logs and other records of important events; and
- Election archive information.

The following requirements apply to records produced by the voting system for any exchange of information between devices, support of auditing procedures, or reporting of final results:

a. Requirements for which electronic records must be produced by tabulators; and

b. Requirements for printed reports to support auditing steps.

2.6.2.1 All records capable of being exported

The voting system SHALL provide the capability to export its electronic records.

Test Method: Functional

Test Entity: VSTL

2.6.2.2 Ballot images

The voting system SHALL have the capability to generate ballot images in a human readable format.

Test Method: Functional

Test Entity: VSTL

2.6.2.3 Ballot image content

The voting system SHALL be capable of producing a ballot image that includes:

a. Election title and date of election;

b. Jurisdiction identifier;

c. Ballot style;

d. Paper record identifier; and

e. For each contest and ballot question:

    i. The choice recorded, including write-ins; and

    ii. Information about each write-ins.

Test Method: Functional

Test Entity: VSTL

2.6.2.4 All records capable of being printed

The voting system SHALL provide the ability to produce printed forms of its electronic records. The printed forms SHALL retain all required information as specified for each record type other than digital signatures.

Test Method: Functional

Test Entity: VSTL

2.6.2.5 Summary count record

The voting system SHALL produce a summary count record including the following:

a. Time and date of summary record; and

b. The following, both in total and broken down by ballot style and voting location:

   i. Number of received ballots

   ii. Number of counted ballots

   iii. Number of rejected electronic CVRs

   iv. Number of write-in votes

   v. Number of undervotes.

Test Method: Functional

Test Entity: VSTL

2.6.3 Paper Records

The vote capture device is required to produce a paper record. This record SHALL be available to the voter to review and verify, and SHALL be retained for later auditing or recounts, as specified by state law. Paper records provide an independent record of the voter's choices that can be used to verify the correctness of the electronic record created by the voting device.

2.6.3.1 Paper record creation

Each vote capture device SHALL print a human readable paper record.

Test Method: Functional

Test Entity: VSTL

2.6.3.2 Paper record contents

Each paper record SHALL contain at least:

a. Election title and date of election;

b. Voting location;

c. Jurisdiction identifier;

d. Ballot style;

e. Paper record identifier; and

f. For each contest and ballot question:

   i. The recorded choice, including write-ins; and

   ii. Information about each write-in.

Test Method: Inspection

Test Entity: VSTL

2.6.3.3 Privacy

The voting system SHALL be capable of producing a paper record that does not contain any information that could link the record to the voter.

Test Method: Inspection

Test Entity: VSTL

2.6.3.4 Multiple pages

When a single paper record spans multiple pages, each page SHALL include the voting location, ballot style, date of election, and page number and total number of the pages (e.g., page 1 of 4).

Test Method: Functional

Test Entity: VSTL

2.6.3.5 Machine-readable part contains same information as human-readable part

If a non-human-readable encoding is used on the paper record, it SHALL contain the entirety of the human-readable information on the record.

Test Method: Inspection

Test Entity: VSTL

2.6.3.6 Format for paper record non-human-readable data

Any non-human-readable information on the paper record SHALL be presented in a non-proprietary format.

Test Method: Inspection

Test Entity: VSTL

2.6.3.7 Linking the electronic CVR to the paper record

The paper record SHALL:

a. Contain the paper record identifier; and

b. Identify whether the paper record represents the ballot that was cast.

Test Method: Inspection

Test Entity: VSTL

2.7 Performance Monitoring

2.7.1 Voting system and Network Status

2.7.1.1 Network monitoring

The voting system SHALL provide for system and network monitoring during the voting period.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.7.1.2 Tool access

The system and network monitoring functionality SHALL only be accessible to authorized personnel from restricted consoles.

Test Method: ***Functional***

Test Entity: ***VSTL***

2.7.1.3 Tool privacy

System and network monitoring functionality SHALL NOT have the capability to compromise voter privacy or election integrity.

Test Method: ***Functional***

Test Entity: ***VSTL***

Section 3: Usability

3.1 General Principles

The goal of a voting system is to have the simplest design needed to meet its intended functions. This design needs to provide guidance to the voter to assist them through the balloting process. In addition, the voting system should minimize the amount of voter inputs required to complete the balloting process.

3.2 Alternative Languages

The voting system SHALL be capable of presenting the ballot, ballot selections, review screens and instructions in any language required by state or federal law.

Test Method: Inspection

Test Entity: Manufacturer

3.3 Clarity of Instructions

The system SHALL:

a. Provide clear instructions and assistance to allow voters to successfully execute and cast their ballots independently;

b. Provide instructions for all valid operations; and

c. Clearly state the nature of the problem, when warnings and alerts are issued by the vote capture device and the set of responses available to the voter. The warning SHALL clearly state whether the voter has performed or attempted an invalid operation or whether the voting equipment itself has malfunctioned in some way.

Test Method: Inspection

Test Entity: Manufacturer

3.4 Voting Input Fields

The design of the voting input field should make it clear where and how to vote and the voting system should provide feedback that the vote was accepted by the voting system. The guidance in this section addresses these design features.

3.4.1.1 User input; voting system

The voting system shall:

a. Provide a consistent relationship between names of the candidates and where to cast a vote.

b. Clearly indicate the maximum number of candidates for whom one can vote for within a single contest; and

c. Provide sufficient computational performance in the vote capture device to provide responses to each voter entry in no more than three seconds

Test Method: ***Functional***

Test Entity: ***Manufacturer***

3.4.1.2 User input; vote capture device

The vote capture device SHALL:

a. On touch screens, provide sensitive touch areas that have a minimum height of 0.5 inches and minimum width of 0.7 inches. The vertical distance between the centers of adjacent areas shall be at least 0.6 inches, and the horizontal distance at least 0.8 inches; and

b. Provide input mechanisms designed to minimize accidental activation.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

3.5 Interaction Issues

The voting process SHALL be designed to minimize interaction difficulties for the voter.

3.5.1 Navigation

3.5.1.1 Page scrolling

The vote capture device SHALL NOT require page scrolling by the voter.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

3.5.1.2 Displaying contest

The vote capture device SHALL display all necessary information to cast a vote for a single contest in one place without the need to turn pages or page to other screens

Test Method: ***Functional***

Test Entity: ***Manufacturer***

3.5.1.3 Movement

The means by which voters navigate through the voting system SHALL be simple and not require complex or complicated actions (e.g., clicking on a "Next Page" button rather than scrolling).

Test Method: Functional

Test Entity: Manufacturer

3.5.1.4 Navigation features

Navigation features SHALL be provided that are distinct and clearly separated from voting response fields

Test Method: Functional

Test Entity: Manufacturer

3.5.1.5 Pace

Voters SHALL be able to control the pace and sequence of their use of the ballot. Voters SHALL be able to freely move back and forward through the ballot.

Test Method: Functional

Test Entity: Manufacturer

3.5.1.6 Additional time

If the vote capture device requires a response by a voter within a specific period of time, it SHALL issue an alert at least 20 seconds before this time period has expired and provide a means by which the voter may receive additional time.

Test Method: Functional

Test Entity: Manufacturer

3.6 Ballot Legibility

In order to facilitate usability, voting system designers should pay close attention to design elements that affect the voter's ability to clearly read and easily understand the information provided. The following guidance addresses these design features:

a. The font size and style used SHALL ensure that written material can be easily and unambiguously read.

b. Text (except labels) SHALL be presented using upper and lower case characters.

c. All text intended for the voter SHALL be presented in a sans serif font.

d. All electronic voting machines SHALL provide a minimum font size of 3.0 mm (measured as the height of a capital letter) for all text.

e. A clearly legible font SHALL be utilized. Fonts SHALL have true ascenders and descenders, uniform stroke width, and uniform aspect ratio.

f. For a given font, it SHALL be possible to clearly distinguish between the following characters: X and Y, T and Y, I and L, I and 1, O and Q, O and 0, S and 5, and U and V.

g. Instructions SHALL be concise and be designed to communicate information clearly and unambiguously so that they can be easily understood and interpreted without error.

h. Instruction steps SHALL be written in active voice as positive commands (focusing on what to do, not what not to do).

i. Punctuation SHALL conform to standard usage of the language used.

j. The use of color by the voting system SHALL agree with common conventions:

    i. Green, blue or white is used for general information or as a normal status indicator;
    ii. Amber or yellow is used to indicate warnings or a marginal status; and
    iii. Red is used to indicate error conditions or a problem requiring immediate attention.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

3.7 Perceptual Issues

The voting system SHALL be designed to minimize perceptual difficulties for the voter.

a. No vote capture device display screen SHALL flicker with a frequency between 2 Hz and 55 Hz.

b. Any aspect of the vote capture device that is adjustable by the voter or remote voting location workers, including font size, color contrast, and audio volume, SHALL automatically reset to a standard default value upon completion of that voter’s session.

c. The minimum figure-to-ground ambient contrast ratio for all text and information graphics (including icons) intended for the voter SHALL be 3:1.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

Section 4: Software

4.1 Selection of Programming Languages

4.1.1 Acceptable Programming Language Constructs

Application logic SHALL be produced in a high-level programming language that has all of the following control constructs:

a. Sequence;

b. Loop with exit condition (e.g., for, while, and/or do-loops);

c. If/Then/Else conditional;

d. Case conditional; and

e. Block-structured exception handling (e.g., try/throw/catch).

Test Method: Inspection

Test Entity: Manufacturer

4.2 Selection of General Coding Conventions

4.2.1 Acceptable Coding Conventions

Application logic SHALL adhere to (or be based on) a published, credible set of coding rules, conventions or standards (herein simply called "coding conventions") that enhance the workmanship, security, integrity, testability, and maintainability of applications.

Test Method: Inspection

Test Entity: Manufacturer

4.2.1.1 Published

Coding conventions SHALL be considered published if they appear in publicly available media.

Test Method: Inspection

Test Entity: Manufacturer

4.2.1.2 Credible

Coding conventions SHALL be considered credible if at least two different organizations independently decided to adopt them and made active use of them at some time within the three years before conformity assessment was first sought.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.3 Software Modularity and Programming

4.3.1 Modularity

Application logic SHALL be designed in a modular fashion.

4.3.1.1 Module testability

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

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.3.1.2 Module size and identification

Modules SHALL be small and easily identifiable.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.4 Structured Programming

4.4.1 Exception Handling

4.4.1.1 Exception handling

Application logic SHALL handle exceptions using block-structured exception handling constructs.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.4.1.2 Legacy library units must be wrapped

If application logic makes use of any COTS or third-party logic callable units that do not throw exceptions when exceptional conditions occur, those callable units SHALL be wrapped in callable units that check for the relevant error conditions and translate them into exceptions, and the remainder of application logic SHALL use only the wrapped version.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.4.2 Unstructured Control Flow is Prohibited

Application logic SHALL contain no unstructured control constructs.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.4.2.1 Branching

Arbitrary branches (a.k.a. GoTos) SHALL NOT be allowed.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.4.2.2 Intentional exceptions

Exceptions SHALL only be used for abnormal conditions. Exceptions SHALL NOT be used to redirect the flow of control in normal ("non-exceptional") conditions.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.4.2.3 Unstructured exception handling

Unstructured exception handling (e.g., On Error GoTo, setjmp/longjmp) SHALL NOT be allowed.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.4.2.4 Separation of code and data

Application logic SHALL NOT compile or interpret configuration data or other input data as a programming language.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.5 Comments

4.5.1 Header Comments

Application logic modules SHALL 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; and

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

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.6 Executable Code and Data Integrity

4.6.1 Code Coherency

Application logic SHALL conform to the following sub-requirements:

a. Self-modifying code SHALL NOT be allowed;

b. Application logic SHALL be free of race conditions, deadlocks, livelocks, and resource starvation;

c. If compiled code is used, it SHALL only be compiled using a COTS compiler; and

d. If interpreted code is used, it SHALL only be run under a specific, identified version of a COTS runtime interpreter.

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.6.2 Prevent Tampering With Code

Programmed devices SHALL defend against replacement or modification of executable or interpreted code.

Test Method:* *Inspection

Test Entity:* *VSTL

4.6.3 Prevent Tampering With Data

The voting system SHALL prevent access to or manipulation of configuration data, vote data, or audit records.

Test Method:* *Inspection

Test Entity:* *VSTL

4.7 Error Checking

4.7.1 Detect Garbage Input

4.7.1.1 Validity check

Programmed devices SHALL check information inputs for completeness and validity.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.1.2 Defend against garbage input

Programmed devices SHALL ensure that incomplete or invalid inputs do not lead to irreversible error.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2 Mandatory Internal Error Checking

4.7.2.1 Error checking

Application logic that is vulnerable to the following types of errors SHALL check for these errors at run time and respond defensively (as specified by Requirement 4.7.2.8) when they occur:

- Out-of-bounds accesses of arrays or strings (includes buffers used to move data);
- Stack overflow errors;
- CPU-level exceptions such as address and bus errors, dividing by zero, and the like;
- Variables that are not appropriately handled when out of expected boundaries;
- Numeric overflows; and
- Known programming language specific vulnerabilities.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.2 Range checking of indices

If the application logic uses arrays, vectors, character sequences, strings 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.3 Stack overflows

If stack overflow does not automatically result in an exception, the application logic SHALL explicitly check for and prevent stack overflow.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.4 CPU traps

The application logic SHALL implement such handlers as are needed to detect and respond to CPU-level exceptions including address and bus errors and dividing by zero.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.5 Garbage input parameters

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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.6 Numeric overflows

If the programming language does not provide automatic run-time detection of numeric overflow, all arithmetic operations that could potentially overflow the relevant data type SHALL be checked for overflow.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

4.7.2.7 Nullify freed pointers

If pointers are used, any pointer variables that remain within scope after the memory they point to is deallocated SHALL be set to null or marked as invalid (pursuant to the idiom of the programming language used) after the memory they point to is deallocated.

Test Method: ***Inspection***

Test Entity: ***VSTL***

4.7.2.8 React to errors detected

The detection of any of the errors enumerated in Requirement 4.7.2.1 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.

Test Method: Inspection

Test Entity: VSTL

4.7.2.9 Do not disable error checks

Error checks detailed in Requirement 4.7.2.1 SHALL remain active in production code.

Test Method: Inspection

Test Entity: VSTL

4.7.2.10 Roles authorized to respond to errors

Exceptions resulting from failed error checks or CPU-level exceptions SHALL require intervention by an election official or administrator before voting can continue.

Test Method: Inspection

Test Entity: Manufacturer

4.7.2.11 Election integrity monitoring

The voting system 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 official or administrator if such violations they occur.

Test Method: Inspection

Test Entity: Manufacturer

4.8 Recovery

4.8.1 Voting system Device Failure

4.8.1.1 Resuming normal operations

All voting systems SHALL be capable of resuming normal operations following the correction of a failure in any device.

Test Method: Functional

Test Entity: Manufacturer

4.8.1.2 Failures not compromise voting or audit data

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

Test Method:* *Inspection

Test Entity:* *Manufacturer

4.8.1.3 Device survive component failure

All vote capture device SHALL be capable of resuming normal operation following the correction of a failure in any component (e.g., memory, CPU, printer) provided that catastrophic electrical or mechanical damage has not occurred.

Test Method:* *Functional

Test Entity:* *Manufacturer

4.8.2 Controlled Recovery

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

Test Method:* *Functional

Test Entity:* *Manufacturer

4.8.2.1 Nested error conditions

Nested error conditions that are corrected without reset, restart, reboot, or shutdown of the vote capture device SHALL be corrected in a controlled sequence so that voting system status may be restored to the initial state existing before the first error occurred.

Test Method:* *Functional

Test Entity:* *Manufacturer

4.8.2.2 Reset CPU error states

CPU-level exceptions that are corrected without reset, restart, reboot, or shutdown of the vote capture device SHALL be handled in a manner that restores the CPU to a normal state and allows the voting system to log the event and recover as with a software-level exception.

Test Method:* *Functional

Test Entity:* *Manufacturer

4.8.3 Restore Device to Checkpoints

When recovering from non-catastrophic failure or from any error or malfunction that is within the operator's ability to correct, the voting 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.

Test Method:* *Functional

Test Entity:* *Manufacturer

Section 5: Security

5.1 Access Control

This section states requirements for the identification of authorized system users, processes and devices and the authentication or verification of those identities as a prerequisite to granting access to system processes and data. It also includes requirements to limit and control access to critical system components to protect system and data integrity, availability, confidentiality, and accountability.

This section applies to all entities attempting to physically enter voting system facilities or to request services or data from the voting system.

5.1.1 Separation of Duties

5.1.1.1 Definition of roles

The voting system SHALL allow the definition of personnel roles with segregated duties and responsibilities on critical processes to prevent a single person from compromising the integrity of the system.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.1.2 Access to election data

The voting system SHALL ensure that only authorized roles, groups, or individuals have access to election data.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.1.3 Separation of duties

The voting system SHALL require at least two persons from a predefined group for validating the election configuration information, accessing the cast vote records, and starting the tabulation process.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.2 Voting system Access

The voting system SHALL provide access control mechanisms designed to permit authorized access and to prevent unauthorized access to the system.

5.1.2.1 Identity verification

The voting system SHALL identify and authenticate each person, to whom access is granted, and the specific functions and data to which each person holds authorized access.

Test Method: Functional

Test Entity: VSTL

5.1.2.2 Access control configuration

The voting system SHALL allow the administrator group or role to configure permissions and functionality for each identity, group or role to include account and group/role creation, modification, and deletion.

Test Method: Functional

Test Entity: VSTL

5.1.2.3 Default access control configuration

The voting system's default access control permissions SHALL implement the least privileged role or group needed.

Test Method: Functional

Test Entity: VSTL

5.1.2.4 Escalation prevention

The voting system SHALL prevent a lower-privilege process from modifying a higher-privilege process.

Test Method: Functional

Test Entity: VSTL

5.1.2.5 Operating system privileged account restriction

The voting system SHALL NOT require its execution as an operating system privileged account and SHALL NOT require the use of an operating system privileged account for its operation.

Test Method: Functional

Test Entity: VSTL

5.1.2.6 Logging of account

The voting system SHALL log the identification of all personnel accessing or attempting to access the voting system to the system event log.

Test Method: Functional

Test Entity: VSTL

5.1.2.7 Monitoring voting system access

The voting system SHALL provide tools for monitoring access to the system. These tools SHALL provide specific users real time display of persons accessing the system as well as reports from logs.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.2.8 Login failures

Vote capture device located at the remote voting location and the central server SHALL have the capability to restrict access to the voting system after a preset number of login failures.

- The lockout threshold SHALL be configurable by appropriate administrators/operators
- The voting system SHALL log the event
- The voting system SHALL immediately send a notification to appropriate administrators/operators of the event.
- The voting system SHALL provide a mechanism for the appropriate administrators/operators to reactivate the account after appropriate confirmation.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.2.9 Account lockout logging

The voting system SHALL log a notification when any account identification is locked.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.2.10 Session time-out

Authenticated sessions on critical processes SHALL have an inactivity time-out control that will require personnel re-authentication when reached. This time-out SHALL be implemented for administration and monitor consoles on all voting system devices.

Test Method:* *Functional

Test Entity:* *VSTL

5.1.2.11 Screen lock

Authenticated sessions on critical processes SHALL have a screen-lock functionality that can be manually invoked.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.2 Identification and Authentication

Authentication mechanisms and their associated strength may vary from one voting system capability and architecture to another but all must meet the minimum requirement to maintain integrity and trust. It is important to consider a range of roles individuals may assume when operating different components in the voting system and each may require different authentication mechanisms.

The requirements described in this section vary from role to role. For instance, a remote voting location worker will have different identification and authentication characteristics than a voter. Also, for selected critical functions there may be cases where split knowledge or dual authorization is necessary to ensure security. This is especially relevant for critical cryptographic key management functions.

5.2.1 Authentication

5.2.1.1 Strength of authentication

Authentication mechanisms supported by the voting system SHALL support authentication strength of at least 1/1,000,000.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.2.1.2 Minimum authentication methods

The voting system SHALL authenticate users per the minimum authentication methods outlined below.

Test Method: ***Functional***

Test Entity: ***VSTL***

Table 5-1 Roles

GROUP OR ROLE	MINIMUM AUTHENTICATION STRENGTH
Election Judge	Two factor
Remote Voting Location Worker	One factor
Voter	Not required

Election Official	Two factor
Administrator	Two-factor
Application or Process	Digital signature 112 bits of security[1]

5.2.1.3 Multiple authentication mechanisms

The voting system SHALL provide multiple authentication methods to support multi-factor authentication.

Test Method: Functional

Test Entity: VSTL

5.2.1.4 Secure storage of authentication data

When private or secret authentication data is stored by the voting system, it SHALL be protected to ensure that the confidentiality and integrity of the data are not violated.

Test Method: Functional

Test Entity: VSTL

5.2.1.5 Password reset

The voting system SHALL provide a mechanism to reset a password if it is forgotten in accordance with the system access/security policy.

Test Method: Functional

Test Entity: VSTL

5.2.1.6 Password strength configuration

The voting system SHALL allow the administrator group or role to specify password strength for all accounts including minimum password length, use of capitalized letters, use of numeric characters, and use of non-alphanumeric characters per NIST 800-63 Electronic Authentication Guideline standards.

Test Method: Functional

Test Entity: VSTL

[1] NIST Special Publication 800-57 Part 1 Table 2

5.2.1.7 Password history configuration

The voting system SHALL enforce password histories and allow the administrator to configure the history length when passwords are stored by the system.

Test Method: Functional

Test Entity: VSTL

5.2.1.8 Account information password restriction

The voting system SHALL ensure that the user name is not used in the password.

Test Method: Functional

Test Entity: VSTL

5.2.1.9 Automated password expiration

The voting system SHALL provide a means to automatically expire passwords.

Test Method: Functional

Test Entity: VSTL

5.2.1.10 Device authentication

The voting system servers and vote capture devices SHALL identify and authenticate one another using NIST - approved cryptographic authentication methods at the 112 bits of security.

Test Method: Functional

Test Entity: VSTL

5.2.1.11 Network authentication

Remote voting location site Virtual Private Network (VPN) connections (i.e., vote capture devices and authentication device connections) to voting servers SHALL be authenticated using strong mutual cryptographic authentication at the 112 bits of security.

Test Method: Functional

Test Entity: VSTL

5.2.1.12 Message authentication

Message authentication SHALL be used for applications to protect the integrity of the message content using a schema with 112 bits of security

Test Method: Functional

Test Entity: VSTL

5.2.1.13 Message authentication mechanisms

IPsec, SSL, or TLS and MAC mechanisms SHALL all be configured to be compliant with FIPS 140-2 using approved algorithm suites and protocols.

Test Method:* *Functional

Test Entity:* *VSTL

5.3 Cryptography

Cryptography serves several purposes in voting systems. They include:

- Confidentiality: where necessary the confidentiality of voting records can be provided by encryption;
- Authentication: data and programs can be authenticated by a digital signature or message authentication codes (MAC), or by comparison of the cryptographic hashes of programs or data with the reliably known hash values of the program or data. If the program or data are altered, then that alteration is detected when the signature or MAC is verified, or the hash on the data or program is compared to the known hash value. Typically the programs loaded on voting systems and the ballot definitions used by voting systems are verified by the systems, while systems apply digital signatures to authenticate the critical audit data that they output. For remote connections cryptographic user authentication mechanism SHALL be based on strong authentication methods; and
- Random number generation: random numbers are used for several purposes including the creation of cryptographic keys for cryptographic algorithms and methods to provide the services listed above, and as identifiers for voting records that can be used to identify or correlate the records without providing any information that could identify the voter.

5.3.1 General Cryptography Requirements

5.3.1.1 Cryptographic functionality

All cryptographic functionality SHALL be implemented using NIST-approved cryptographic algorithms/schemas, or use published and credible cryptographic algorithms/schemas/protocols.

Test Method:* *Inspection

Test Entity:* *VSTL

5.3.1.2 Required security strength

Cryptographic algorithms and schemes SHALL be implemented with a security strength equivalent to at least 112 bits of security to protect sensitive voting information and election records.

***Test Method:* Inspection**

***Test Entity:* VSTL**

5.3.1.3 Use NIST-approved cryptography for communications

Cryptography used to protect information in-transit over public telecommunication networks SHALL use NIST-approved algorithms and cipher suites.

***Test Method:* Function**

***Test Entity:* VSTL**

5.3.2 Key Management

The following requirements apply to voting systems that generate cryptographic keys internally.

5.3.2.1 Key generation methods

Cryptographic keys generated by the voting system SHALL use a NIST-approved key generation method, or a published and credible key generation method.

***Test Method:* Inspection**

***Test Entity:* VSTL**

5.3.2.2 Security of key generation methods

Compromising the security of the key generation method (e.g., guessing the seed value to initialize the deterministic random number generator (RNG)) SHALL require as least as many operations as determining the value of the generated key.

***Test Method:* Inspection**

***Test Entity:* VSTL**

5.3.2.3 Seed values

If a seed key is entered during the key generation process, entry of the key SHALL meet the key entry requirements see 5.3.3.1. If intermediate key generation values are output from the cryptographic module, the values SHALL be output either in encrypted form or under split knowledge procedures.

***Test Method:* Inspection**

***Test Entity:* VSTL**

5.3.2.4 Use NIST-approved key generation methods for communications

Cryptographic keys used to protect information in-transit over public telecommunication networks SHALL use NIST-approved key generation methods. If the approved key generation method requires input from a random number generator, then an approved (140-2) random number generator SHALL be used.

Test Method:* *Inspection

Test Entity:* *VSTL

5.3.2.5 Random number generator health tests

Random number generators used to generate cryptographic keys SHALL implement one or more health tests that provide assurance that the random number generator continues to operate as intended (e.g., the entropy source is not stuck).

Test Method:* *Inspection

Test Entity:* *VSTL

5.3.3 Key Establishment

Key establishment may be performed by automated methods (e.g., use of a public key algorithm), manual methods (use of a manually transported key loading device), or a combination of automated and manual methods.

5.3.3.1 Key entry and output

Secret and private keys established using automated methods SHALL be entered into and output from a voting system in encrypted form. Secret and private keys established using manual methods may be entered into or output from a system in plaintext form.

Test Method:* *Inspection

Test Entity:* *VSTL

5.3.4 Key Handling

5.3.4.1 Key storage

Cryptographic keys stored within the voting system SHALL NOT be stored in plaintext. Keys stored outside the voting system SHALL be protected from disclosure or modification.

Test Method:* *Inspection

Test Entity:* *VSTL

5.3.4.2 Key zeroization

The voting system SHALL provide methods to zeroize all plaintext secret and private cryptographic keys within the system.

Test Method:* *Functional

Test Entity:* *VSTL

5.3.4.3 Support for rekeying

The voting system SHALL support the capability to reset cryptographic keys to new values.

Test Method:* *Functional

Test Entity:* *VSTL

5.4 Voting System Integrity Management

This section addresses the secure deployment and operation of the voting system, including the protection of removable media and protection against malicious software.

5.4.1 Protecting the Integrity of the Voting System

5.4.1.1 Cast vote integrity; transmission

The integrity and authenticity of each individual cast vote SHALL be protected from any tampering or modification during transmission.

Test Method:* *Functional

Test Entity:* *VSTL

5.4.1.2 Cast vote integrity; storage

The integrity and authenticity of each individual cast vote SHALL be preserved by means of a digital signature during storage.

Test Method:* *Functional

Test Entity:* *Manufacturer

5.4.1.3 Cast vote storage

Cast vote data SHALL NOT be permanently stored on the voting device.

Test Method:* *Functional

Test Entity:* *Manufacturer

5.4.1.4 Electronic ballot box integrity

The integrity and authenticity of the electronic ballot box SHALL be protected by means of a digital signature.

Test Method: Functional

Test Entity: Manufacturer

5.4.1.5 Malware detection

The voting system SHALL use malware detection software to protect against known malware that targets the operating system, services, and applications.

Test Method: Inspection

Test Entity: Manufacturer

5.4.1.6 Updating malware detection

The voting system SHALL provide a mechanism for updating malware detection signatures.

Test Method: Inspection

Test Entity: Manufacturer

5.5 Communications Security

This section provides requirements for communications security. These requirements address ensuring the integrity of transmitted information and protecting the voting system from external communications-based threats.

5.5.1 Data Transmission Integrity

5.5.1.1 Data integrity protection

Voting systems that transmit data over communications links SHALL provide integrity protection for data in transit through the generation of integrity data (digital signatures and/or message authentication codes) for outbound traffic and verification of the integrity data for inbound traffic.

Test Method: Functional

Test Entity: VSTL

5.5.1.2 TLS/SSL

Voting systems SHALL use at a minimum TLS 1.0, SSL 3.1 or equivalent protocols.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.5.1.3 Virtual private networks

Voting systems deploying VPNs SHALL configure them to only allow FIPS-compliant cryptographic algorithms and cipher suites.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.5.1.4 Unique system identifier

Each communicating device SHALL have a unique system identifier.

Test Method: ***Inspection***

Test Entity: ***VSTL***

5.5.1.5 Mutual authentication required

Each device SHALL mutually strongly authenticate using the system identifier before additional network data packets are processed.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

5.5.1.6 Secrecy of ballot data

Data transmission SHALL preserve the secrecy of voters' ballot selections and SHALL prevent the violation of ballot secrecy and integrity.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.5.2 External Threats

Voting systems SHALL implement protections against external threats to which the system may be susceptible.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.5.2.1 Disabling network interfaces

Voting system components SHALL have the ability to enable or disable physical network interfaces.

Test Method: ***Functional***

***Test Entity:* VSTL**

5.5.2.2 Minimizing interfaces

The number of active ports and associated network services and protocols SHALL be restricted to the minimum required for the voting system to function.

***Test Method:* Inspection/Vulnerability**

***Test Entity:* VSTL**

5.5.2.3 Prevention of attacks and security non-compliance

The voting system SHALL block all network connections that are not over a mutually authenticated channel.

***Test Method:* Functional/Vulnerability**

***Test Entity:* VSTL**

5.6 Logging

5.6.1 Log Management

5.6.1.1 Default settings

The voting system SHALL implement default settings for secure log management activities, including log generation, transmission, storage, analysis, and disposal.

***Test Method:* Inspection**

***Test Entity:* VSTL**

5.6.1.2 Log access

Logs SHALL only be accessible to authorized roles.

***Test Method:* Functional**

***Test Entity:* Manufacturer**

5.6.1.3 Log access

The voting system SHALL restrict log access to append-only for privileged logging processes and read-only for authorized roles.

***Test Method:* Functional**

***Test Entity:* VSTL**

5.6.1.4 Logging events

The voting system SHALL log logging failures, log clearing, and log rotation.

Test Method: Functional

Test Entity: VSTL

5.6.1.5 Log format

The voting system SHALL store log data in a publicly documented format, such as XML, or include a utility to export log data into a publicly documented format.

Test Method: Inspection

Test Entity: VSTL

5.6.1.6 Log separation

The voting system SHALL ensure that each jurisdiction's event logs and each component's logs are separable from each other.

Test Method: Functional

Test Entity: Manufacturer

5.6.1.7 Log review

The voting system SHALL include an application or program to view, analyze, and search event logs.

Test Method: Functional

Test Entity: Manufacturer

5.6.1.8 Log preservation

All logs SHALL be preserved in a useable manner prior to voting system decommissioning.

Test Method: Inspection

Test Entity: Manufacturer

5.6.1.9 Voter privacy

Logs SHALL NOT contain any data that could violate the privacy of the voter's identity.

Test Method: Functional

Test Entity: Manufacturer

5.6.1.10 Timekeeping format

Timekeeping mechanisms SHALL generate time and date values, including hours, minutes, and seconds.

Test Method:* *Functional

Test Entity:* *VSTL

5.6.1.11 Timekeeping precision

The precision of the timekeeping mechanism SHALL be able to distinguish and properly order all log events.

Test Method:* *Inspection

Test Entity:* *VSTL

5.6.1.12 System clock security

Only the system administrator SHALL be permitted to set the system clock.

Test Method:* *Functional

Test Entity:* *VSTL

5.6.2 Communications Logging

5.6.2.1 General

All communications actions SHALL be logged.

Test Method:* *Inspection

Test Entity:* *VSTL

5.6.2.2 Log content

The communications log SHALL contain at least the following entries:

- Times when the communications are activated and deactivated;
- Services accessed;
- Identification of device to which data was transmitted to or received from;
- Identification of authorized entity; and
- Successful and unsuccessful attempts to access communications or services.

Test Method:* *Functional

Test Entity:* *VSTL

5.6.3 System Event Logging

This section describes requirements for the voting system to perform event logging for system maintenance troubleshooting, recording the history of system activity, and detecting unauthorized or malicious activity. The operating system, and/or applications software may perform the actual event logging. There may be multiple logs in use for any system component.

5.6.3.1 Event log format

The voting system SHALL log the following data for each event:

a. System ID;

b. Unique event ID and/or type;

c. Timestamp;

d. Success or failure of event, if applicable;

e. User ID triggering the event, if applicable; and

f. Jurisdiction, if applicable.

Test Method: Inspection

Test Entity: VSTL

5.6.3.2 Critical events

All critical events SHALL be recorded in the system event log.

Test Method: Functional

Test Entity: Manufacturer

5.6.3.3 System events

At a minimum the voting system SHALL log the events described in the table below.

Test Method: Inspection

Test Entity: Manufacturer

Table 5-2 System events

SYSTEM EVENT	DESCRIPTION
GENERAL SYSTEM FUNCTIONS	
Error and exception messages	Includes but not limited to: • The source and disposition of system interrupts resulting in entry into exception handling routines. • Messages generated by exception handlers.

System Event	Description
	• The identification code and number of occurrences for each hardware and software error or failure. • Notification of physical violations of security. • Other exception events such as power failures, failure of critical hardware components, data transmission errors or other types of operating anomalies. • All faults and the recovery actions taken. • Error and exception messages such as ordinary timer system interrupts and normal I/O system interrupts do not need to be logged.
Critical system status messages	Critical system status messages other than information messages displayed by the device during the course of normal operations. Includes but not limited to: • Diagnostic and status messages upon startup. • The "zero totals" check conducted before opening the voting location.
Non-critical status messages	Non-critical status messages that are generated by the data quality monitor or by software and hardware condition monitors.
Events that require election official intervention	Events that require election official intervention, so that each election official access can be monitored and access sequence can be constructed.
Shutdown and restarts	Both normal and abnormal shutdowns and restarts.
Changes to system configuration settings	Configuration settings include but are not limited to registry keys, kernel settings, logging settings, and other system configuration settings.
Integrity checks for executables, configuration files, data, and logs	Integrity checks that may indicate possible tampering with files and data.
The addition and deletion of files	Files added or deleted from the system.
System readiness results	Includes but not limited to: • System pass or fail of hardware and software test for system readiness. • Identification of the software release, identification of the election to be processed, voting location identification, and the results of the software and hardware diagnostic tests. • Pass or fail of ballot style compatibility and integrity test. • Pass or fail of system test data removal.
Removable media events	Removable media that is inserted into or removed from the system.
Backup and restore	Successful and failed attempts to perform backups and restores.
Authentication related events	Includes but not limited to:

SYSTEM EVENT	DESCRIPTION
	• Login/logoff events (both successful and failed attempts). • Account lockout events. • Password changes.
Access control related events	Includes but not limited to: • Use of privileges. • Attempts to exceed privileges. • All access attempts to application and underlying system resources. • Changes to the access control configuration of the system.
User account and role (or groups) management activity	Includes but not limited to: • Addition and deletion of user accounts and roles. • User account and role suspension and reactivation. • Changes to account or role security attributes such as password length, access levels, login restrictions, permissions, etc. • Administrator account and role password resets.
Installation, upgrading, patching, or modification of software or firmware	Logging for installation, upgrading, patching, or modification of software or firmware include logging what was installed, upgraded, or modified as well as a cryptographic hash or other secure identifier of the old and new versions of the data.
Changes to configuration settings	Includes but not limited to: • Changes to critical function settings. At a minimum critical function settings include location of ballot definition file, contents of the ballot definition file, vote reporting, location of logs, and system configuration settings. • Changes to settings including but not limited to enabling and disabling services. • Starting and stopping processes.
Abnormal process exits	All abnormal process exits.
Successful and failed database connection attempts (if a database is utilized).	All database connection attempts.
Changes to cryptographic keys	At a minimum critical cryptographic settings include key addition, key removal, and re-keying.
Voting events	Includes: • Opening and closing the voting period. • Casting a vote. • Success or failure of log and election results exportation.

5.7 Incident Response

5.7.1 Incident Response Support

5.7.1.1 Critical events

Manufacturers SHALL document what types of system operations or security events (e.g., failure of critical component, detection of malicious code, unauthorized access to restricted data) are classified as critical.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.7.1.2 Critical event alarm

An alarm that notifies appropriate personnel SHALL be generated on the remote voting device or server, dependant upon which device has the error, if a critical event is detected.

Test Method: ***Functional***

Test Entity: ***Manufacturer***

5.8 Physical and Environmental Security

5.8.1 Physical Access

5.8.1.1 Unauthorized physical access requirement

Any unauthorized physical access SHALL leave physical evidence that an unauthorized event has taken place.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.2 Physical Ports and Access Points

5.8.2.1 Non-essential ports

The voting system SHALL disable physical ports and access points that are not essential to voting operations, testing, and auditing.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3 Physical Port Protection

5.8.3.1 Physical port shutdown requirement

If a physical connection between the vote capture device and a component is broken, the affected vote capture device port SHALL be automatically disabled.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.2 Physical component alarm requirement

The voting system SHALL produce a visual alarm if a connected component is physically disconnected.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.3 Physical component event log requirement

An event log entry that identifies the name of the affected device SHALL be generated if a vote capture device component is disconnected.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.4 Physical port enablement requirement

Disabled ports SHALL only be re-enabled by authorized administrators.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.5 Physical port restriction requirement

Vote capture devices SHALL be designed with the capability to restrict physical access to voting machine ports that accommodate removable media with the exception of ports used to activate a voting session.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.6 Physical port tamper evidence requirement

Vote capture devices SHALL be designed to give a physical indication of tampering or unauthorized access to ports and all other access points, if used as described in the manufacturer's documentation.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.3.7 Physical port disabling capability requirement

Vote capture devices SHALL be designed such that physical ports can be manually disabled by an authorized administrator.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.4 Door Cover and Panel Security

5.8.4.1 Access points security requirement

Access points such as covers and panels SHALL be secured by locks or tamper evident or tamper resistance countermeasures and SHALL be implemented so that voting system owners can monitor access to vote capture devices components through these points.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.5 Secure Paper Record Receptacle

5.8.5.1 Secure paper record container requirement

If the voting system provides paper record containers then they SHALL be designed such that any unauthorized physical access results in physical evidence that an unauthorized event has taken place.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.6 Secure Physical Lock and Key

5.8.6.1 Secure physical lock access requirement

Voting equipment SHALL be designed with countermeasures that provide physical indication that unauthorized attempts have been made to access locks installed for security purposes.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.6.2 Secure locking system key requirement

Manufacturers SHALL provide locking systems for securing voting devices that can make use of keys that are unique to each owner.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.8.7 Media Protection

These requirements apply to all media, both paper and digital, that contain personal privacy related data or other protected or sensitive types of information.

5.8.7.1 Remote voting site protection

The voting system SHALL meet the following requirements:

a. All paper records (including rejected ones) printed at the remote voting locations SHALL be deposited in a secure container;

b. Vote capture device hardware, software and sensitive information (e.g., electoral roll) SHALL be physically protected to prevent unauthorized modification or disclosure; and

c. Vote capture device hardware components, peripherals and removable media SHALL be identified and registered by means of a unique serial number or other identifier.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

5.9 Penetration Resistance

5.9.1 Resistance to Penetration Attempts

5.9.1.1 Resistant to attempts

The voting system SHALL be resistant to attempts to penetrate the system by any remote unauthorized entity.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.1.2 System information disclosure

The voting system SHALL be configured to minimize ports, responses and information disclosure about the system while still providing appropriate functionality.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.1.3 System access

The voting system SHALL provide no access, information or services to unauthorized entities.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.1.4 Interfaces

All interfaces SHALL be penetration resistant including TCP/IP, wireless, and modems from any point in the system.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.1.5 Documentation

The configuration and setup to attain penetration resistance SHALL be clearly and completely documented.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2 Penetration Resistance Test and Evaluation

5.9.2.1 Scope

The scope of penetration testing SHALL include all the voting system components. The scope of penetration testing includes but is not limited to the following:

- Server system;
- Vote capture devices;
- All items setup and configured per Technical Data Package (TDP) recommendations;
- Local wired and wireless networks; and
- Internet connections.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2.2 Test environment

Penetration testing SHALL be conducted on a voting system set up in a controlled lab environment. Setup and configuration SHALL be conducted in accordance with the TDP, and SHALL replicate the real world environment in which the voting system will be used.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2.3 White box testing

The penetration testing team SHALL conduct white box test using manufacturer supplied documentation and voting system architecture information. Documentation includes the TDP and user documentation. The testing team SHALL have access to any relevant information regarding the voting system configuration. This includes, but is not limited to, network layout and Internet Protocol addresses for system devices and components. The testing team SHALL be provided any source code included in the TDP.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2.4 Focus and priorities

Penetration testing seeks out vulnerabilities in the voting system that might be used to change the outcome of an election, interfere with voter ability to cast ballots, ballot counting, or compromise the ballot secrecy. The penetration testing team SHALL prioritize testing efforts based on the following:

a. Threat scenarios for the voting system under investigation;

b. Remote attacks SHALL be prioritized over in-person attacks;

c. Attacks with a large impact SHALL be prioritized over attacks with a more narrow impact; and

d. Attacks that can change the outcome of an election SHALL be prioritized over attacks that compromise ballot secrecy or cause non-selective denial of service.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2.5 Penetration testing team establishment

The test lab SHALL establish a penetration testing team with at least two security experts. One of these experts SHALL have at least 4 years of experience in penetration testing, and the others SHALL have at least 2 years of experience.

Test Method: ***Functional***

Test Entity: ***VSTL***

5.9.2.6 Penetration testing level of effort-test plan

In determining the level of effort to apply to penetration testing, the test lab SHALL take into consideration the size and complexity of the voting system, any available results from the “closed end” functional, security, and usability testing activities and laboratory analysis and testing activities.

Test Method: Functional

Test Entity: VSTL

5.9.2.7 Penetration testing level of effort

The penetration testing team SHALL devote a minimum period of 4 staff weeks to examining and testing the voting system and to generating the reports of the testing results.

Test Method: Functional

Test Entity: VSTL

Section 6: Quality Assurance

6.1 General Requirements

At a minimum, this program SHALL:

a. Include procedures for specifying, procuring, inspecting, accepting, and controlling parts and raw materials of the requisite quality;

b. Require the documentation of the software development process;

c. Require the documentation of the hardware specification and selection process;

d. Identify and enforce all requirements for:

   i. In-process inspection and testing that the manufacturer deems necessary to ensure proper fabrication and assembly of hardware;

   ii. Installation and operation of software and firmware;

e. Include plans and procedures for post-production environmental screening and acceptance testing; and

f. Include a procedure for maintaining all data and records required to document and verify the quality inspections and tests.

Test Method: Inspection

Test Entity: VSTL

6.2 Components from Third Parties

A manufacturer who does not manufacture all the components of its voting system, but instead procures components as standard commercial items for assembly and integration into a voting system, SHALL verify that the supplier manufacturers follow documented quality assurance procedures that are at least as stringent as those used internally by the voting system manufacturer.

Test Method: Inspection

Test Entity: Manufacturer

6.3 Responsibility for Tests

Manufacturer SHALL be responsible for performing all quality assurance tests, acquiring and documenting test data, and providing test reports for examination by the VSTL as part of the national certification process. These reports SHALL also be provided to the purchaser upon request.

Test Method: Inspection

Test Entity: Manufacturer

6.4 Parts and Materials, Special Tests, and Examinations

In order to ensure that voting system parts and materials function properly, manufacturers SHALL:

a. Select parts and materials to be used in voting systems and components according to their suitability for the intended application. Suitability may be determined by similarity of this application to existing standard practice or by means of special tests;

b. Design special tests, if needed, to evaluate the part or material under conditions accurately simulating the actual voting system operating environment; and

c. Maintain the resulting test data as part of the quality assurance program documentation.

Test Method: Inspection

Test Entity: Manufacturer

6.5 Quality Conformance Inspections

The manufacturer performs conformance inspections to ensure the overall quality of the voting system and components delivered to the VSTL for national certification testing and to the jurisdiction for implementation. To meet the conformance inspection requirements the manufacturer SHALL:

a. Inspect and test each voting system or component to verify that it meets all inspection and test requirements for the voting system; and

b. Deliver a record of tests or a certificate of satisfactory completion with each voting system or component

Test Method: Inspection

Test Entity: Manufacturer

Section 7: Configuration Management

7.1 Scope

7.1.1 Configuration Management Requirements

The configuration management documentation provided for manufacturer registration SHALL be sufficient for pilot projects.

Test Method:* *Inspection

Test Entity:* *EAC

7.1.2 Audit of Configuration Management Documentation

The manufacturer SHALL provide the following documentation to the EAC for review. This documentation will be audited during the registration review which will be conducted during the pilot testing period. The items which the EAC will audit are the following:

a. Application of configuration management requirements;

b. Configuration management policy;

c. Configuration identification;

d. Baseline, promotion, and demotion procedures;

e. Configuration control procedures;

f. Release process;

g. Configuration audits; and

h. Configuration management resources.

Test Method:* *Inspection

Test Entity:* *EAC

7.2 Configuration Identification

Configuration identification is the process of identifying, naming, and acquiring configuration items. Configuration identification encompasses all voting system components.

7.2.1 Classification and Naming Configuration Items

Manufacturers SHALL describe the procedures and conventions used to classify configuration items into categories and subcategories, uniquely number or otherwise identify configuration items and name configuration items.

Test Method:* *Inspection

Test Entity:* *Manufacturer

7.2.2 Versioning Conventions

When a voting system component is part of a higher level system element such as a subsystem, the manufacturer SHALL describe the conventions used to:

a. Identify the specific versions of individual configuration items and sets of items that are incorporated in higher level system elements such as subsystems;

b. Uniquely number or otherwise identify versions; and

c. Name versions.

Test Method:* *Inspection

Test Entity:* *Manufacturer

7.3 Baseline and Promotion Procedures

Manufacturers SHALL establish formal procedures and conventions for establishing and providing a complete description of the procedures and related conventions used to:

a. Establish a particular instance of a component as the starting baseline;

b. Promote subsequent instances of a component to baseline status as development progresses through to completion of the initial completed version released to the VSTL for testing; and

c. Promote 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 by the manufacturer).

Test Method:* *Inspection

Test Entity:* *Manufacturer

7.4 Configuration Control Procedures

Configuration control is the process of approving and implementing changes to a configuration item to prevent unauthorized additions, changes or deletions. The manufacturer SHALL establish such procedures and related conventions, providing a complete description of those procedures used to:

a. Develop and maintain internally developed items;
b. Acquire and maintain third-party items;
c. Resolve internally identified defects for items regardless of their origin; and
d. Resolve externally identified and reported defects (i.e., by customers and VSTLs).

Test Method:* *Inspection

Test Entity:* *Manufacturer

7.5 Configuration Audits

7.5.1 Physical Configuration Audit

For the PCA, a manufacturer SHALL provide:

a. Identification of all items that are to be a part of the pilot voting system release;
b. Specification of compiler (or choice of compilers) to be used to generate executable programs;
c. Identification of all hardware that interfaces with the software;
d. Configuration baseline data for all hardware that is unique to the voting system;
e. Copies of all software documentation intended for distribution to users, including program listings, specifications, operations manual, voter manual, and maintenance manual;
f. Identification of any changes between the physical configuration of the voting system submitted for the PCA and that submitted for the FCA, with a certification that any differences do not degrade the functional characteristics; and
g. Complete descriptions of its procedures and related conventions used to support this audit by
    i. Establishing a configuration baseline of the software and hardware to be tested; and
    ii. Confirming whether the voting system documentation matches the corresponding system components.

Test Method:* *Inspection

Test Entity:* *VSTL

7.5.2 Functional Configuration Audit

The Functional Configuration Audit is conducted by the VSTL to verify that the voting system performs all the functions described in the system documentation. Manufacturers SHALL:

a. Completely describe its procedures and related conventions used to support this audit for all voting system components; and

b. Provide the following information to support this audit:

   i. Copies of all procedures used for module or unit testing, integration testing, and system testing;

   ii. Copies of all test cases generated for each module and integration test, and sample ballot formats or other test cases used for system tests; and

   iii. Records of all tests performed by the procedures listed above, including error corrections and retests .

Test Method: ***Functional / Inspection***

Test Entity: ***VSTL***

Section 8: Technical Data Package

8.1 Scope

This section contains a description of manufacturer documentation relating to the voting system that must be submitted with the system as a precondition of conformity assessment. These items are necessary to define the product and its method of operation; to provide technical and test data supporting the manufacturer'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 media, materials, source code, object code, and sample output report formats, must be submitted along with this documentation.

This documentation is used by the VSTL in constructing the test plan. Testing of systems submitted by manufacturers 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 manufacturer's system development process must be submitted for conformity assessment. Documentation describing the system development process permits assessment of the manufacturer'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. The VSTL must design and conduct the appropriate tests to cover all elements of the system and to ensure conformance with all system requirements.

8.1.1 Content and Format

The content of the Technical Data Package (TDP) is intended to provide clear, complete descriptions of the following information about the voting system:

- Overall system design, including subsystems, modules and the interfaces among them;
- Specific functional capabilities provided by the system;
- Performance and design specifications;
- Design constraints, applicable standards, and compatibility requirements;
- Personnel, equipment, and facility requirements for system operation, maintenance, and logistical support;
- Manufacturer practices for assuring system quality during the system's development and subsequent maintenance; and
- Manufacturer practices for managing the configuration of the system during development and for modifications to the system throughout its life cycle.

8.1.1.1 Required content for initial conformity assessment

8.1.1.1.1 Identify full system configuration

Manufacturers SHALL submit to the VSTL documentation necessary for the identification of the full system configuration submitted for evaluation and for the development of an appropriate test plan by the VSTL.

Test Method: Inspection

Test Entity: Manufacturer

8.1.1.1.2 Required content for pilot certification

Manufacturers SHALL provide a list of all documents submitted controlling the design, construction, operation, and maintenance of the voting system. At minimum, the TDP SHALL contain the following documentation:

a. implementation statement;

b. The voting equipment user documentation (Section 9 “Voting Equipment User Documentation”);

c. System hardware specification;

d. Application logic design and specification;

e. System security specifications;

f. System test specification;

g. Configuration for testing; and

h. Training Documentation.

Test Method: Inspection

Test Entity: Manufacturer

8.1.1.2 Format

The requirements for formatting the TDP are general in nature; specific format details are of the manufacturer's choosing.

8.1.1.2.1 Table of contents and abstracts

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.

Test Method: Inspection

Test Entity: Manufacturer

8.1.1.2.2 Cross-index

A cross-index SHALL be provided indicating the portions of the documents that are responsive to documentation requirements enumerated in section 8.1.1.1.2.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.1.2 Protection of Proprietary Information

8.1.2.1 Identify proprietary data

Manufacturers SHALL identify all documents, or portions of documents, containing proprietary information that is not releasable to the public.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.2 Implementation Statement

8.2.1 TDP Implementation Statement

The TDP SHALL include an implementation statement.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.3 System Hardware Specification

8.3.1 TDP System Hardware Specification

Manufacturers SHALL expand on the system overview included in the user documentation by providing detailed specifications of the hardware components of the voting system, including specifications of hardware used to support the telecommunications capabilities of the voting system, if applicable.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.3.2 System Hardware Characteristics

8.3.2.1 TDP system hardware characteristics

Manufacturers SHALL provide a detailed discussion of the characteristics of the system, indicating how the hardware meets individual requirements defined in this document, including:

a. Performance characteristics: 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: Suitability for intended use, requirements for security criteria, and vulnerability to adverse environmental factors;

c. Reliability: 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; and

d. Environmental conditions: 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.

Test Method: Inspection

Test Entity: Manufacturer

8.3.3 Design and Construction

8.3.3.1 Identify system configuration

Manufacturers SHALL provide sufficient data, or references to data, to identify unequivocally the details of the system configuration submitted for testing.

Test Method: Inspection

Test Entity: Manufacturer

8.3.3.2 Photographs for hardware validation

Manufacturers SHALL provide photographs of the exterior and interior of devices included in the system to identify the hardware of the system configuration submitted for testing.

Test Method: Inspection

Test Entity: Manufacturer

8.3.3.3 List of materials

Manufacturers 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.

Test Method: Inspection

Test Entity: Manufacturer

8.3.3.4 Design and construction miscellany

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. Electromagnetic environment generated by the system; and

c. Operator and voter safety considerations and any constraints on system operations or the use environment.

Test Method: Inspection

Test Entity: Manufacturer

8.3.4 Hardwired Logic

8.3.4.1 Hardwired and mechanical implementations of logic

For each non-COTS hardware component (e.g., an Application-Specific Integrated Circuit or a manufacturer-specific integration of smaller components), manufacturers SHALL provide complete design and logic specifications, such as Computer Aided Design and Hardware Description Language files.

Test Method: Inspection

Test Entity: Manufacturer

8.3.4.2 Logic specifications for PLDs, FPGAs and PICs

For each Programmable Logic Device (PLD), Field-Programmable Gate Array (FPGA), or Peripheral Interface Controller (PIC) that is programmed with non-COTS logic, manufacturers SHALL provide complete logic specifications, such as Hardware Description Language files or source code.

Test Method: Inspection

Test Entity: Manufacturer

8.4 Application Logic Design and Specification

8.4.1 TDP Application Logic Design and Specification

Manufacturers SHALL expand on the system overview included in the user documentation by providing detailed specifications of the application logic components of the system, including those used to support the telecommunications capabilities of the system, if applicable.

Test Method: Inspection

Test Entity:* *Manufacturer

8.4.2 Purpose and Scope

8.4.2.1 Describe application logic functions

Manufacturers SHALL describe the function or functions that are performed by the application logic comprising the system, including that used to support the telecommunications capabilities of the system, if applicable.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.3 Applicable Documents

8.4.3.1 List documents controlling application logic development

Manufacturers SHALL list all documents controlling the development of application logic and its specifications.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.4 Application Logic Overview

8.4.4.1 Application logic overview

Manufacturers SHALL provide an overview of the application logic.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.4.2 Application logic architecture

The overview SHALL include a description of the architecture, the design objectives, and the logic structure and algorithms used to accomplish those objectives.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.4.3 Application logic design

The overview SHALL include the general design, operational considerations, and constraints influencing the design.

Test Method:* *Inspection

Test Entity: Manufacturer

8.4.4.4 Application logic overview miscellany

The overview SHALL include the following additional information for each separate software package:

a. Package identification;

b. General description;

c. Requirements satisfied by the package;

d. Identification of interfaces with other packages that provide data to, or receive data from, the package; and

e. Concept of execution for the package.

Test Method: Inspection

Test Entity: Manufacturer

8.4.5 Application Logic Standards and Conventions

8.4.5.1 Application logic standards and conventions

Manufacturers SHALL provide information on application logic standards and conventions developed internally by the manufacturer as well as published industry standards that have been applied by the manufacturer.

Test Method: Inspection

Test Entity: Manufacturer

8.4.5.2 Application logic standards and conventions, checklist

Manufacturers SHALL provide information that addresses the following standards and conventions related to application logic:

a. Development methodology;

b. Design standards, including internal manufacturer procedures;

c. Specification standards, including internal manufacturer procedures;

d. Coding conventions, including internal manufacturer procedures;

e. Testing and verification standards, including internal manufacturer procedures, that can assist in determining the correctness of the logic; and

f. Quality assurance standards or other documents that can be used to examine and test the application logic. These documents include standards for logic diagrams, program documentation, test planning, and test data acquisition and reporting.

Test Method: Inspection

Test Entity: ***Manufacturer***

8.4.5.3 Justify coding conventions

Manufacturers SHALL furnish evidence that the selected coding conventions are "published" and "credible" as specified in section 4.3.1.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.6 Application Logic Operating Environment

8.4.6.1 Application logic operating environment

Manufacturers SHALL describe or make reference to all operating environment factors that influence the design of application logic.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.7 Hardware Environment and Constraints

8.4.7.1 Hardware environment and constraints

Manufacturers SHALL identify and describe the hardware characteristics that influence the design of the application logic, such as:

a. 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.8 Application Logic Environment

8.4.8.1 Identify operating system

Manufacturers SHALL identify the operating system and the specific version thereof, or else clarify how the application logic operates without an operating system.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.8.2 Identify compilers and assemblers

For systems containing compiled or assembled application logic, manufacturers SHALL identify the COTS compilers or assemblers used in the generation of executable code, and the specific versions thereof.

Test Method: Inspection

Test Entity: Manufacturer

8.4.8.3 Identify interpreters

For systems containing interpreted application logic, manufacturers SHALL specify the COTS runtime interpreter that SHALL be used to run this code, and the specific version thereof.

Test Method: Inspection

Test Entity: Manufacturer

8.4.9 Application Logic Functional Specification

8.4.9.1 Application logic functional specification

Manufacturers SHALL provide a description of the operating modes of the system and of application logic capabilities to perform specific functions.

Test Method: Inspection

Test Entity: Manufacturer

8.4.10 Functions and Operating Modes

8.4.10.1 Functions and operating modes

Manufacturers SHALL describe all application logic functions and operating modes of the system, such as ballot preparation, election programming, preparation for opening the voting period, recording votes and/or counting ballots, closing the voting period, and generating reports.

Test Method: Inspection

Test Entity: Manufacturer

8.4.10.2 Functions and operating modes detail

For each application logic function or operating mode, manufacturers SHALL provide:

a. A definition of the inputs to the function or mode (with characteristics, limits, 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, limits, tolerances, or acceptable ranges, as applicable).

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.11 Application Logic Integrity Features

8.4.11.1 Application logic integrity features

Manufacturers SHALL describe the application logic'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.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.12 Programming Specifications

8.4.12.1 Programming specifications

Manufacturers SHALL provide in this section an overview of the application logic's design, its structure, and implementation algorithms and detailed specifications for individual modules.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.13 Programming Specifications Overview

The programming specifications overview SHALL document the architecture of the application logic.

8.4.13.1 Programming specifications overview, diagrams

This overview SHALL include such items as UML diagrams, data flow diagrams, and/or other graphical techniques that facilitate understanding of the programming specifications.

Test Method: Inspection

Test Entity: Manufacturer

8.4.13.2 Internal functioning of individual modules

This section SHALL be prepared to facilitate understanding of the internal functioning of the individual modules.

Test Method: Inspection

Test Entity: Manufacturer

8.4.13.3 Programming specifications overview, content

Implementation of the functions SHALL be described in terms of the architecture, algorithms, and data structures.

Test Method: Inspection

Test Entity: Manufacturer

8.4.14 Programming Specifications Details

8.4.14.1 TDP programming specifications details

The programming specifications SHALL describe individual application logic modules and their component units, if applicable.

Test Method: Inspection

Test Entity: Manufacturer

8.4.14.2 Module and callable unit documentation

For each application logic module and callable unit, manufacturers 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 module or callable unit; and

c. A description of its inputs, outputs, and other data elements as applicable with respect to communication over system interfaces (see section 8.4.16 “Interfaces”).

Test Method: Inspection

Test Entity: Manufacturer

8.4.14.3 Justify mixed-language software

If an application logic 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.14.4 References for foreign programming languages

If a module contains embedded border logic commands for an external library or package (e.g., 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.14.5 Source code

For each callable unit (function, method, operation, subroutine, procedure, etc.) in application logic, border logic, and third-party logic, manufacturers SHALL supply the source code.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.14.6 Inductive assertions

For each callable unit (function, method, operation, subroutine, procedure, etc.) in core logic, manufacturers SHALL specify:

a. Preconditions and postconditions of the callable unit, including any assumptions about capacities and limits within which the system is expected to operate; and

b. A sound argument (possibly, but not necessarily, a formal proof) that the preconditions and postconditions of the callable unit accurately represent its behavior, assuming that the preconditions and postconditions of any invoked units are similarly accurate.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.14.7 High-level constraints

Manufacturers SHALL specify a sound argument (possibly, but not necessarily, a formal proof) that the core logic as a whole satisfies each of the constraints for all cases within the aforementioned capacities and limits, assuming that the preconditions and postconditions of callable units accurately characterize their behaviors.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.14.8 Safety of concurrency

Manufacturers SHALL specify a sound argument (possibly, but not necessarily, a formal proof) that application logic is free of race conditions, deadlocks, livelocks, and resource starvation.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.15 System Database

8.4.15.1 System database

Manufacturers SHALL identify and provide a diagram and narrative description of the system's databases and any external files used for data input or output.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.15.2 Database design levels

For each database or external file, manufacturers SHALL specify the number of levels of design and the names of those levels (e.g., conceptual, internal, logical, and physical).

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.15.3 Database design conventions

For each database or external file, the manufacturer SHALL specify any design conventions and standards (which may be incorporated by reference) needed to understand the design.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.15.4 Data models

For each database or external file, manufacturers SHALL identify and describe all logical entities and relationships and how these are implemented physically (e.g., tables, files).

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.15.5 Schemata

Manufacturers 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 (meters, seconds, etc.);

e. Range or enumeration of possible values (0–99, etc.);

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).

Test Method: Inspection

Test Entity: Manufacturer

8.4.15.6 External file maintenance and security

For external files, manufacturers SHALL document the procedures for file maintenance, management of access privileges, and security.

Test Method: Inspection

Test Entity: Manufacturer

8.4.16 Interfaces

8.4.16.1 Identify and describe interfaces

Using a combination of text and diagrams, manufacturers SHALL identify and provide a complete description of all major internal and external interfaces.

Test Method: Inspection

Test Entity: Manufacturer

8.4.17 Interface Identification

8.4.17.1 Interface identification details

For each interface identified in the system overview, manufacturers 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 upon them).

Test Method: Inspection

Test Entity: Manufacturer

8.4.18 Interface Description

8.4.18.1 Interface types

For each interface identified in the system overview, manufacturers SHALL describe the type of interface (e.g., real-time data transfer or data storage-and-retrieval) to be implemented.

Test Method: Inspection

Test Entity: Manufacturer

8.4.18.2 Interface signatures

For each interface identified in the system overview, manufacturers 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 (meters, seconds, etc.);

e. Range or enumeration of possible values (0–99, etc.);

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).

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.18.3 Interface protocols

For each interface identified in the system overview, manufacturers 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 (e.g., 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.18.4 Protocol details

For each interface identified in the system overview, manufacturers 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.4.18.5 Characteristics of interfaces

For each interface identified in the system overview, manufacturers SHALL describe any other pertinent characteristics, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.).

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.4.19 Appendices

Manufacturers SHALL provide descriptive material and data supplementing the various sections of the body of the logic specifications. The content and arrangement of appendices are at the discretion of the manufacturer. Topics recommended for amplification or treatment in appendix form include:

- Glossary: A listing and brief definition of all module names and variable names, with reference to their locations in the logic structure. Abbreviations, acronyms, and terms should be included, if they are either uncommon in data processing and software development or are used with an unorthodox meaning;
- References: A list of references to all related manufacturer documents, data, standards, and technical sources used in logic development and testing; and
- Program Analysis: The results of logic configuration analysis algorithm analysis and selection, timing studies, and hardware interface studies that are reflected in the final logic design and coding.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.5 System Security Specifications

This section defines the documentation requirements for systems. These recommendations apply to the full scope of system functionality, including functionality for defining the ballot and other pre-voting functions, as well as functions for casting and storing votes, vote reporting, system logging, and maintenance of the system. User documentation includes all public information that is provided to end users. The Technical Data Package (TDP) includes the user documentation along with proprietary information that is viewed only by the VSTL.

8.5.1 General

8.5.1.1 Overall security

Manufacturers SHALL document in the TDP all aspects of system design, development, and proper usage that are relevant to system security. This includes, but is not limited to the following:

- System security objectives;
- All hardware and software security mechanisms;
- All cryptographic algorithms, protocols and schemes that are used;

- Development procedures employed to ensure absence of malicious code;
- Initialization, usage, and maintenance procedures necessary to secure operation;
- All attacks the system is designed to resist or detect; and
- Any security vulnerabilities known to the manufacturer.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.5.1.2 High level security

Manufacturers SHALL provide at a minimum the high-level documents listed in Table 8-1 as part of the TDP.

Test Method:* *Inspection

Test Entity:* *Manufacturer

Table 8-1 High level system documentation

DOCUMENT	DESCRIPTION
Security Threats Controls	This document identifies the threats the system protects against and the implemented security controls on system and system components.
Security Architecture	This document provides an architecture level description of how the security requirements are met, and SHALL include the various authentication, access control, audit, confidentiality, integrity, and availability requirements.
Interface Specification	This document describes external interfaces (programmatic, human, and network) provided by each of the computer components of the system.
Design Specification	This document provides a high-level design of each system component.
Development Environment Specification	This document provides descriptions of the physical, personnel, procedural, and technical security of the development environment including configuration management, tools used, coding standards used, software engineering model used, and description of developer and independent testing.
Security Testing and Vulnerability Analysis Documentation	This document describes security tests performed to identify vulnerabilities and the results of the testing. This also includes testing performed as part of software development, such as unit, module, and subsystem testing.

8.5.2 Access Control

8.5.2.1 General user

Manufacturers SHALL provide user and TDP documentation of access control capabilities of the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.2.2 General access control technical specification

Manufacturers SHALL provide descriptions and specifications of all access control mechanisms of the system including management capabilities of authentication, authorization, and passwords in the TDP.

Test Method: Inspection

Test Entity: Manufacturer

8.5.2.3 Unauthorized access technical specification

Manufacturers SHALL provide descriptions and specifications of methods to prevent unauthorized access to the access control mechanisms of the system in the TDP.

Test Method: Inspection

Test Entity: Manufacturer

8.5.2.4 Access control dependent system mechanisms

Manufacturers SHALL provide descriptions and specifications of all system mechanisms that are dependent upon, support, and interface with access controls in the TDP.

Test Method: Inspection

Test Entity: Manufacturer

8.5.2.5 Voting operations and roles

Manufacturers SHALL provide a list of all of the operations possible on the voting system and list the default roles that have permission to perform each such operation as part of the TDP.

Test Method: Inspection

Test Entity: Manufacturer

8.5.2.6 Critical event escalation

Manufacturers SHALL document a prioritized critical event escalation list of appropriate personnel to be notified.

Test Method: Inspection

Test Entity: Manufacturer

8.5.3 System Event Logging

8.5.3.1 General

Manufacturers SHALL provide TDP documentation of event logging capabilities of the system.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.5.4 Software Installation

8.5.4.1 Software list technical data package

Manufacturers SHALL provide a list of all software related to the system in the technical data package (TDP).

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.5.4.2 Software information

Manufacturers SHALL provide, at a minimum in the TDP, the following information for each piece of software related to the system:

- Software product name;
- Software version number;
- Software manufacturer name;
- Software manufacturer contact information;
- Type of software (application logic, border logic, third party logic, COTS software, or installation software);
- List of software documentation;
- Component identifier(s) (such as filename(s)) of the software; and
- Type of software component (executable code, source code, or data).

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.5.4.3 Software location information

Manufacturers SHALL provide the location (such as full path name or memory address) and storage device (such as type and part number of storage device) where each piece of software is installed on programmed devices of the system.

Test Method:* *Inspection

Test Entity: Manufacturer

8.5.4.4 Software functionality for programmed devices

Manufacturers SHALL document the functionality provided to the system by the software installed on programmed devices.

Test Method: Inspection

Test Entity: Manufacturer

8.5.4.5 Software dependencies and interaction

Manufacturers SHALL map the dependencies and interactions between software installed on programmed devices.

Test Method: Inspection

Test Entity: Manufacturer

8.5.4.6 Build environment software and hardware

Manufacturers SHALL provide a list of all software and hardware required to assemble the build environment used to create system software executable code including application logic, border logic, and third party logic.

Test Method: Inspection

Test Entity: Manufacturer

8.5.4.7 Build environment assembly procedures

Manufacturers SHALL document the procedures to assemble the build environment(s) used to create system software executable code including application logic, border logic, and third party logic.

Test Method: Inspection

Test Entity: Manufacturer

8.5.4.8 System software build procedures

Manufacturers SHALL document the procedures used to build the system software executable code including application logic, border logic, and third party logic.

Test Method: Inspection

Test Entity: Manufacturer

8.5.5 Physical Security

8.5.5.1 Unauthorized physical access

Manufacturers SHALL provide a list of all system components to which access must be restricted and a description of the function of each said component.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.5.5.2 Physical port and access point

Manufacturers SHALL provide a listing of all ports and access points.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.5.5.3 Physical lock documentation of use

For each lock, manufacturers SHALL document whether the lock was installed to secure an access point.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.5.5.4 Power usage

Manufacturer SHALL provide a list of all physical security countermeasures that require power supplies.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.5.5.5 Physical security

Manufacturer SHALL document the design and implementation of all physical security controls for the system and its components.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.5.6 System Integrity Management

8.5.6.1 Binaries per system

Manufacturers SHALL provide a list of the binaries that are required to be executed on the system devices.

Test Method: ***Inspection***

Test Entity: Manufacturer

8.5.7 Setup Inspection

8.5.7.1 Software integrity verification

Manufacturers SHALL provide a technical specification of how the integrity of software installed on programmed devices of the system is verified.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.2 Software integrity verification technique software non-modification

Manufacturers SHALL provide documentation of software integrity verification techniques that prevent the modification of software installed on programmed devices of the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.3 Register and variable value inspection

Manufacturers SHALL provide a technical specification of how the inspection of all the system registers and variables is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.4 Backup power inspection

Manufacturers SHALL provide a technical specification of how the inspection of the remaining charge of the backup power sources is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.5 Cabling connectivity inspection

Manufacturers SHALL provide a technical specification of how the inspection of the connectivity of cabling attached is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.6 Communications operational status inspection

Manufacturers SHALL provide a technical specification of how the inspection of the operational status of the communications capability is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.7 Communications on/off inspection

Manufacturers SHALL provide a technical specification of how the inspection of the on/off status of the communications capability is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.8 Consumable inspection

Manufacturers SHALL provide a technical specification of how the inspection of the remaining amount of each consumable is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.9 Calibration of voting device components inspection

Manufacturers SHALL provide a technical specification of how the inspection of the calibration for each component is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.5.7.10 Calibration of voting device components adjustment

Manufacturers SHALL provide a technical specification of how the adjustment to the calibration of each component is implemented by the system.

Test Method: Inspection

Test Entity: Manufacturer

8.6 System Test Specification

Manufacturers SHALL provide test specifications for:

a. Development test specifications; and

b. System test specifications.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.6.1 Development Test Specifications

8.6.1.1 Development test specifications

Manufacturers SHALL describe the plans, procedures, and data used during 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.

Test Method:* *Inspection

Test Entity:* *Manufacturer

8.6.2 System Test Specifications

RFI 2007-03 contains several requirements for usability testing by the manufacturer and that each of these requirements also mandates that the manufacturer report the test results as part of the TDP. These requirements are not present in this section but need to be considered as part of the system test specifications.

8.6.2.1 Specifications for verification and validation of system performance

Manufacturers SHALL provide specifications for verification and validation of overall system performance. These specifications SHALL cover:

a. Control and data input/output;

b. Processing accuracy;

c. Data quality assessment and maintenance;

d. ballot interpretation logic;

e. Exception handling;

f. Security;

g. Production of audit trails and statistical data;

h. Expected test results; and

i. Criteria for evaluating test results.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.6.2.2 Demonstrate fitness for purpose

The specifications SHALL identify procedures for assessing and demonstrating the suitability of the system for election use.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.7 Configuration for Testing

8.7.1 Configuration Description

Configuration of hardware and 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. If the system can be set up in both conforming and nonconforming configurations, the configuration actions necessary to obtain conforming behavior must be specified.

8.7.1.1 Hardware set-up

Manufacturers SHALL provide instructions and photographs illustrating the proper set-up of the system hardware.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.7.1.2 Provide answers to installation prompts

Manufacturers SHALL provide a record of all user selections that must be made during software/firmware installation for the system to meet the requirements of the UOCAVA Pilot Testing Requirements.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

8.7.1.3 Configuration data

Manufacturers SHALL submit all configuration data needed to set up and operate the system.

Test Method: ***Inspection***

8.7 Configuration for Testing

Test Entity: Manufacturer

Section 9: System Users Manual

9.1 Scope

This section contains requirements on the content of the documentation that manufacturers supply to jurisdictions that use their systems. In this context, "user" refers to election officials, others in the jurisdiction who implement systems, and VSTLs. The user documentation is also included in the TDP provided to the VSTL.

It is not the intent of these requirements to prescribe an outline for user documentation. Manufacturers 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 VSTLs will be included within the user documentation. To expedite the VSTL review, manufacturers SHALL provide the VSTL with a short index that relates the corresponding sections of the user documentation to the specific requirements in this document.

9.2 System Overview

9.2.1 User Documentation System Overview

In the system overview, manufacturers 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.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.2.2 System Overview Functional Diagram

The system overview SHALL include a high-level functional diagram of the system that includes all of its components. The diagram SHALL portray how the various components relate and interact.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.2.3 System Description

9.2.3.1 User documentation system description

The system description SHALL include written descriptions, drawings and diagrams that present:

a. A description of the functional components or subsystems, (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, firmware, software, and communications structure;

c. A description 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 COTS products (both hardware and software) included in the system and/or used as part of the system's operation, identifying the name, manufacturer, and version used for each such component;

f. Communications (dial-up, network) 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 manufacturers 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. Listings of all software and firmware and associated documentation included in the manufacturer's release in the order in which each piece of software or firmware would normally be installed upon system setup and installation.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.2.3.2 Identify software and firmware by origin

The system description SHALL include the identification of all software and firmware items, indicating items that were:

a. Written in-house;

b. Written by a subcontractor;

c. Procured as COTS; and

d. Procured and modified, including descriptions of the modifications to the software or firmware and to the default configuration options.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.2.3.3 Traceability of procured software

The system description SHALL include a declaration that procured software items were obtained directly from the manufacturer or from a licensed dealer or distributor.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.2.4 System Performance

9.2.4.1 User documentation system performance

Manufacturers SHALL provide system performance information including:

a. Device capacities and limits that were stated in the implementation statement;

b. 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 styles supported), and processing frequency;

c. Quality attributes such as reliability, maintainability, availability, usability, and portability;

d. Provisions for safety, security, voter privacy, ballot secrecy, and continuity of operations; and

e. Design constraints, applicable standards, and compatibility requirements.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.3 System Functionality Description

9.3.1 User Documentation, System Functionality Description

Manufacturers SHALL provide a listing of the system's functional processing capabilities, encompassing capabilities required by the UOCAVA Pilot Testing Requirements, and any additional capabilities provided by the system, with a description of each capability.

a. Manufacturers SHALL explain, in a manner that is understandable to users, the capabilities of the system 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.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.4 System Security Specification

9.4.1 Access Control

9.4.1.1 Access control implementation, configuration, and management

Manufacturers SHALL provide user documentation containing guidelines and usage instructions on implementing, configuring, and managing access control capabilities.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.4.1.2 Access control policy

Manufacturers SHALL provide, within the user documentation, the access control policy under which the system was designed to operate.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.4.1.3 Privileged account

Manufacturers SHALL disclose and document information on all privileged accounts included on the system.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.4.2 System Event Logging

9.4.2.1 System event logging

Manufacturers SHALL provide user documentation that describes system event logging capabilities and usage.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.4.2.2 Log format

Manufacturers SHALL provide fully documented log format information.

Test Method:* *Inspection

Test Entity: ***Manufacturer***

9.4.3 Ballot Decryption

9.4.3.1 Ballot decryption process

Manufacturers SHALL provide documentation on the proper procedures for the authorized entity to implement ballot decryption while maintaining the security and privacy of the data.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.4.3.2 Ballot decryption key reconstruction

Manufacturers SHALL provide documentation describing the proper procedure for the authorized entity to reconstruct the election private key to decrypt the ballots.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.4.3.3 Ballot decryption key destruction

Manufacturers SHALL document when any cryptographic keys created or used by the system may be destroyed. The documentation SHALL describe how to delete keys securely and irreversibly at the appropriate time.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.4.4 Physical Security

9.4.4.1 Physical security

Manufacturers SHALL provide user documentation explaining the implementation of all physical security controls for the system, including procedures necessary for effective use of countermeasures.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.4.5 Audit

9.4.5.1 Ballot count and vote total auditing

The system's user documentation SHALL fully specify a secure, transparent, workable and accurate process for producing all records necessary to verify the accuracy of the electronic tabulation result.

Test Method: Inspection

Test Entity: Manufacturer

9.4.5.2 Machine readability of paper record

Manufacturers SHALL provide documentation for a procedure to scan the paper record by optical character recognition.

Test Method: Inspection

Test Entity: Manufacturer

9.5 Software

9.5.1 Software installation

9.5.1.1 Software list

Manufacturers SHALL provide a list of all software to be installed on the programmed devices of the system and installation software used to install the software.

Test Method: Inspection

Test Entity: Manufacturer

9.5.1.2 Software information

Manufacturers SHALL provide at a minimum, the following information for each piece of software to be installed or used to install software on programmed devices of the system: software product name, software version number, software manufacturer name, software manufacturer contact information, type of software (application logic, border logic, third party logic, COTS software, or installation software), list of software documentation, component identifier(s) (such filename(s)) of the software, type of software component (executable code, source code, or data).

Test Method: Inspection

Test Entity: Manufacturer

9.5.1.3 Software location information

Manufacturers SHALL provide the location (such as full path name or memory address) and storage device (such as type and part number of storage device) where each piece of software is installed on programmed devices of the system.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.4 Election specific software identification

Manufacturers SHALL identify election specific software in the user documentation.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.5 Installation software and hardware

Manufacturers SHALL provide a list of software and hardware required to install software on programmed devices of the system in the user documentation.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.6 Software installation procedure

Manufacturers SHALL document the software installation procedures used to install software on programmed devices of the system.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.7 Compiler installation prohibited

The software installation procedures used to install software on programmed devices of the system SHALL specify that no compilers SHALL be installed on the programmed device.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.8 Procurement of system software

The software installation procedures SHALL specify that system software SHALL be obtained from the VSTL or approved distribution repositories.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.5.1.9 Open market procurement of COTS software

The software installation procedures SHALL specify that COTS software SHALL be obtained from the open market.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.5.1.10 Erasable storage media preparation

The software installation procedures SHALL specify how previously stored information on erasable storage media is removed before installing software on the media.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.5.1.11 Installation media unalterable storage media

The software installation procedures SHALL specify that unalterable storage media SHALL be used to install software on programmed devices of the system.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.5.1.12 Software hardening

Manufacturers SHALL provide documentation that describes the hardening procedures for the system.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.6 Setup Inspection

9.6.1 Setup inspection process

Manufacturers SHALL provide a setup inspection process that the system was designed to support.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.6.1.1 Minimum properties included in a setup inspection process

A setup inspection process SHALL, at a minimum, include the inspection of system software, storage locations that hold election information that changes during an election, other voting device properties, and execution of logic and accuracy testing related to readiness for use in an election.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.2 Setup inspection record generation

The setup inspection process SHALL describe the records that result from performing the setup inspection process.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.3 Installed software identification procedure

Manufacturers SHALL provide the procedures to identify all software installed on programmed devices.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.4 Software integrity verification procedure

Manufacturers SHALL describe the procedures to verify the integrity of software installed on programmed devices of system.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.5 Election information value

Manufacturers SHALL provide the values of system storage locations that hold election information that changes during the election, except for the values set to conduct a specific election.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.6 Maximum values of election information storage locations

Manufacturers SHALL provide the maximum values for the storage locations that the election information resides in.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.7 Register and variable value inspection procedure

Manufacturers SHALL provide the procedures to inspect the values of voting device storage locations that hold election information that changes for an election.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.8 Backup power operational range

Manufacturers SHALL provide the nominal operational range for the backup power sources of the voting device.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.9 Backup power inspection procedure

Manufacturers SHALL provide the procedures to inspect the remaining charge of the backup power sources of the voting device.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.10 Cabling connectivity inspection procedure

Manufacturers SHALL provide the procedures to inspect the connectivity of the cabling attached to the voting device.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.11 Communications operational status inspection procedure

Manufacturers SHALL provide the procedures to inspect the operational status of the communications capabilities of the voting device.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.12 Communications on/off status inspection procedure

Manufacturers SHALL provide the procedures to inspect the on/off status of the communications capabilities of the voting device.

Test Method: Inspection

Test Entity: Manufacturer

9.6.1.13 Consumables quantity of voting equipment

Manufacturers SHALL provide a list of consumables associated with the voting device, including estimated number of usages per quantity of consumable.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.6.1.14 Consumable inspection procedure

Manufacturers SHALL provide the procedures to inspect the remaining amount of each consumable of the voting device.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.6.1.15 Calibration of voting device components nominal range

Manufacturers SHALL provide a list of components associated with the voting device that require calibration and the nominal operating ranges for each component.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.6.1.16 Calibration of voting device components inspection procedure

Manufacturers SHALL provide the procedures to inspect the calibration of each component.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.6.1.17 Calibration of voting device components adjustment procedure

Manufacturers SHALL provide the procedures to adjust the calibration of each component.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.6.1.18 Checklist of properties to be inspected

Manufacturers SHALL provide a checklist of other properties of the system to be inspected.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.7 System Operations Manual

9.7.1 General

9.7.1.1 System operations manual

The system operations manual SHALL provide all information necessary for system set up and use by all personnel who administer and operate the system at the state and/or local election offices and at the remote voting locations, with regard to all system functions and operations identified in Section 9.3 System Functionality Description.

Test Method: Inspection

Test Entity: Manufacturer

9.7.1.2 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, state and/or local election officials, election judges, and remote voting site workers.

Test Method: Inspection

Test Entity: Manufacturer

9.7.2 Introduction

9.7.2.1 Functions

Manufacturers SHALL provide a summary of system operating functions to permit understanding of the system's capabilities and constraints.

Test Method: Inspection

Test Entity: Manufacturer

9.7.2.2 Roles

The roles of operating personnel SHALL be identified and related to the functions of the system.

Test Method: Inspection

Test Entity: Manufacturer

9.7.2.3 Conditional actions

Decision criteria and conditional operator functions (such as error and failure recovery actions) SHALL be described.

Test Method: Inspection

Test Entity: Manufacturer

9.7.2.4 References

Manufacturers SHALL list all reference and supporting documents pertaining to the use of the system during election operations.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.7.3 Operational Environment

9.7.3.1 Operational environment

Manufacturers SHALL describe the system environment and the interfaces between the system and State and/or local election officials, remote voting site workers, system administrators, and voters.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.7.3.2 Operational environment; equipment and facility

Manufacturers SHALL identify all facilities, furnishings, fixtures, and utilities that will be required for equipment operations, including equipment that operates at the:

a. Remote voting location;

b. State and/or local election offices; and

c. Other locations.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.7.3.3 Operational environment; installation

The operations manual 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.

Test Method:* *Inspection

Test Entity:* *Manufacturer

9.7.4 System Installation and Test Specification

9.7.4.1 Readiness testing

Manufacturers SHALL provide specifications for testing of system installation and readiness.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.7.4.1.1 Readiness test entire system

These specifications SHALL cover testing of all components of the system and all locations of installation (e.g., remote voting locations, state and/or local election offices), and SHALL address all elements of system functionality and operations identified in Section 9.3 System Functionality Description above, including general capabilities and functions specific to particular voting activities.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.7.5 Operational Features

9.7.5.1 Features

Manufacturers SHALL provide documentation of system operating features that includes:

a. Detailed descriptions 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.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.7.5.2 Document straight party override algorithms

For systems that support straight party voting, manufacturers SHALL document the available algorithms for counting straight party overrides.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.7.5.3 Document double vote reconciliation algorithms

For systems that support write-in voting, manufacturers SHALL document the available algorithms for reconciling write-in double votes.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.7.6 Operating Procedures

9.7.6.1 Operating procedures

Manufacturers 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. Enables the operator to assess the correct flow of system functions (as evidenced by system-generated status and information messages);

c. Enables 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 state and/or local election officials;

h. Provides a schedule and steps for the software and ballot installation, including a table outlining the key dates, events and deliverables; and

i. Specifies diagnostic tests that may be employed to identify problems in the system, verify the correction of problems, and isolate and diagnose faults from various system states.

***Test Method:* Inspection**

***Test Entity:* Manufacturer**

9.7.6.2 Printer error recovery guidelines

Manufacturers SHALL provide documentation for procedures to recover from printer errors and faults including procedures for how to cancel a vote suspended during an error.

Test Method: Inspection

Test Entity: Manufacturer

9.7.7 Transportation and Storage

9.7.7.1 Transportation

Manufacturers SHALL include any special instructions for preparing voting devices for shipment.

Test Method: Inspection

Test Entity: Manufacturer

9.7.7.2 Storage

Manufacturers SHALL include any special storage instructions for voting devices.

Test Method: Inspection

Test Entity: Manufacturer

9.7.7.3 Precautions for removable media

Manufacturers SHALL detail the care and handling precautions necessary for removable media and records.

Test Method: Inspection

Test Entity: Manufacturer

9.7.8 Appendices

Manufacturers SHALL 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 manufacturer. Topics required for discussion include:

- Glossary: A listing and brief definition of all terms that may be unfamiliar to persons not trained in either systems or computer operations;
- References: A list of references to all manufacturer documents and to other sources related to operation of the system;
- Detailed Examples: Detailed scenarios that outline correct system responses to faulty operator input; and

- Manufacturer's Recommended Security Procedures: Security procedures that are to be executed by the system operator.

Test Method: Inspection

Test Entity: Manufacturer

9.8 System Maintenance Manual

9.8.1.1 User documentation system maintenance manual

The system maintenance manual SHALL provide information to support election workers, information systems personnel, or maintenance personnel in the adjustment or removal and replacement of components or modules in the field.

Test Method: Inspection

Test Entity: Manufacturer

9.8.1.2 General contents

Manufacturers SHALL describe service actions recommended to correct malfunctions or problems; personnel and expertise required to repair and maintain the system, equipment, and materials; and facilities needed for proper maintenance.

Test Method: Inspection

Test Entity: Manufacturer

9.8.2 Introduction

9.8.2.1 Equipment overview, maintenance viewpoint

Manufacturers SHALL describe the structure and function of the hardware, firmware and 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.

Test Method: Inspection

Test Entity: Manufacturer

9.8.3 Maintenance Procedures

9.8.3.1 Maintenance manual maintenance procedures

Manufacturers SHALL describe preventive and corrective maintenance procedures for hardware, firmware and software.

Test Method: Inspection

Test Entity: Manufacturer

9.8.3.2 Maintenance manual preventive maintenance procedures

Manufacturers SHALL identify and describe:

a. All required and recommended preventive maintenance tasks, including software and data 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 referred to the manufacturer.

Test Method: Inspection

Test Entity: Manufacturer

9.8.3.3 Corrective maintenance procedures

9.8.3.3.1 Troubleshooting procedures

Manufacturers SHALL provide fault detection, fault isolation, correction procedures, and logic diagrams for all operational abnormalities identified by design analysis and operating experience.

Test Method: Inspection

Test Entity: Manufacturer

9.8.3.3.2 Troubleshooting procedures details

Manufacturers SHALL identify specific procedures to be used in diagnosing and correcting problems in the system hardware, firmware and software. Descriptions SHALL include:

a. Steps to replace failed or deficient equipment;

b. Steps to correct deficiencies or faulty operations in software or firmware;

c. Number and skill levels of personnel needed to accomplish each procedure;

d. Special maintenance equipment, parts, supplies, or other resources needed to accomplish each procedure; and

e. Any coordination required with the manufacturer.

Test Method: Inspection

Test Entity: Manufacturer

9.8.4 Maintenance Equipment

9.8.4.1 Special equipment

Manufacturers SHALL identify and describe any special purpose test or maintenance equipment recommended for fault isolation and diagnostic purposes.

Test Method: Inspection

Test Entity: Manufacturer

9.8.5 Parts and Materials

Manufacturers SHALL provide detailed documentation of parts and materials needed to operate and maintain the system.

Test Method: Inspection

Test Entity: Manufacturer

9.8.6 Maintenance Facilities and Support

9.8.6.1 Maintenance environment

Manufacturers SHALL identify all facilities, furnishings, fixtures, and utilities that will be required for equipment maintenance.

Test Method: Inspection

Test Entity: Manufacturer

9.8.6.2 Maintenance support and spares

Manufacturers SHALL specify:

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 (e.g., jurisdiction, manufacturer) of qualified maintenance personnel.

Test Method: Inspection

Test Entity: Manufacturer

9.8.7 Appendices

Manufacturers SHALL 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 manufacturer. Topics required for amplification or treatment in the appendix include:

- Glossary: A listing and brief definition of all terms that may be unfamiliar to persons not trained in either systems or computer maintenance;
- References: A list of references to all manufacturer documents and other sources related to maintenance of the system;
- Detailed Examples: Detailed scenarios that outline correct system responses to faulty operator input; and
- Maintenance and Security Procedures: Technical illustrations and schematic representations of electronic circuits unique to the system.

Test Method: Inspection

Test Entity: Manufacturer

9.9 Personnel Deployment and Training Requirements

Manufacturers SHALL describe the personnel resources and training required for a jurisdiction to operate and maintain the system for the duration of the pilot project.

Test Method: Inspection

Test Entity: Manufacturer

9.9.1 Personnel

9.9.1.1 Training manual personnel

Manufacturers SHALL specify the number of personnel and skill levels required to perform each of the following functions:

a. Pre-voting or election preparation functions;

b. System operations for system functions performed at the remote voting locations;

c. System operations for system functions performed at the State and/or local election office;

d. Preventive maintenance tasks;

e. Diagnosis of faulty hardware, firmware, or software;

f. Corrective maintenance tasks; and

g. Testing to verify the correction of problems.

Test Method: Inspection

Test Entity: ***Manufacturer***

9.9.1.2 User functions versus manufacturer functions

Manufacturers SHALL distinguish which functions may be carried out by user personnel and which must be performed by manufacturer personnel.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

9.9.2 Training

9.9.2.1 Training requirements

Manufacturers SHALL provide training materials to instruct system administrators, remote voting location workers, and state and/or local election officials on how to set up, configure and operate the system.

Test Method: ***Inspection***

Test Entity: ***Manufacturer***

Appendix A: Definitions of Words with Special Meanings

This section of the Pilot Program Requirements defines words (terms) that are used in the other parts of the Pilot Program Requirements, particularly in requirements text.

NOTE: Readers may already be familiar with definitions for many of the words in this section, but the definitions here often may differ in small or big ways from locality usage because they are used in special ways in the Pilot Program Requirements.

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 differ from common election and plain English usage, and may be unsuitable for applications that are beyond the scope of the Pilot Program Requirements. Readers are especially cautioned to avoid comparisons between this terminology and the terminology used in election law.

Any term that is defined neither in this terminology standard nor in any of the referenced documents has its regular (i.e., dictionary) meaning.

Each term is followed by a normative definition.

absentee ballot:	A ballot cast from any location not defined as a polling place.
absentee model:	The ballot remains associated with the voter ID and is subject to an adjudication process to be accepted.
absentee voting:	The process of casting a ballot from any location not defined as a polling place.
administrator:	The role responsible for installing, configuring, and managing the technical operations of the system.
application logic:	Software, firmware, or hardwired logic from any source that is specific to the system, with the exception of border logic.
audit:	Systematic, independent, documented process for obtaining records, statements of fact or other relevant information and assessing them objectively to determine the extent to which specified requirements are fulfilled
authenticated session:	Process that requires all users to provide proof of identity.
ballot image:	Human-readable electronic representation of the ballot, including the voter's selections.
ballot question:	Contest in which the choices are Yes and No.
ballot secrecy:	Not being able to associate the selections of the ballot with the voter who cast it.
ballot style:	Particular set of contests to appear on the ballot for a particular election district, their order, the list of ballot positions for each contest, and the binding of candidate names to ballot positions
ballot:	The official presentation of all of the contests to be decided in a particular election. See also ballot image, cast vote record, and paper record.

baseline configuration:	The exact system configuration tested by the VSTL. It includes all the system components that were tested, including the specific hardware, operating system, application software, and third-party COTS applications.
border logic:	Software, firmware, or hardwired logic that is developed to connect application logic to COTS or third-party logic.
callable unit:	Function, method, operation, subroutine, procedure, or analogous structural unit that appears within a module (of a software program or analogous logical design).
candidate:	Person contending in a contest for office.
cast ballot:	Ballot in which the voter has taken final action in the selection of contest choices and which has been accepted.
cast vote record:	The record of all votes selected by a voter.
CIF:	Common Industry Format
common industry format:	Format described in ISO/IEC 25062:2006 "Common Industry Format (CIF) for Usability Test Reports".
component:	A discrete and identifiable element of hardware or software within a system.
concept of operations:	Description of roles and responsibilities for system administration, operation and use.
configuration data:	Non-executable input to software, firmware, or hardwired logic, not including vote data.
conformity assessment:	Demonstration that specified requirements relating to a product, process, system, person or body are fulfilled.
contest:	A single decision being put before the voters (e.g., the selection of candidates or the response to ballot questions).
core logic:	Subset of application logic that is responsible for vote recording and tabulation.
COTS:	Commercial Off the Shelf
credible:	Methodologies (e.g., coding conventions, cryptographic algorithms) are considered credible if at least two different organizations independently decided to adopt them and made active use of them at some time within the three years before conformity assessment was first sought.
CVR:	Cast vote record
device:	Functional unit that performs its assigned tasks as an integrated whole.
election definition:	Definition of the contests and questions that will appear on the ballot for a specific election.
election judge:	In this sense, an official on the canvassing board that adjudicates the acceptance of absentee ballots
election management system:	Set of processing functions and databases within a system that defines, develops and maintains election databases, performs election definitions and setup functions, format ballots, count votes, consolidates and report results, and maintains audit trails
election official:	The people associated with administering and conducting elections.
election title:	The heading on a ballot specifying the name of the election (e.g., General Election, Primary Election).

equivalent configuration:	A system configuration that has been attested to by the manufacturer to perform identically to the baseline configuration.
error rate:	Ratio of the number of errors detected in relation to the volume of data processed:
failure:	Events that result 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 device, operating system or application software, (d) a requirement for an unanticipated intervention by a person in the role of poll worker or technician before normal operation can continue, or (e) error messages and/or audit log entries indicating that a failure has occurred.
fault:	Flaw in design or implementation that may result in the qualities or behavior of the system deviating from the qualities or behavior that are specified in the Pilot Program Testing Requirements and/or in manufacturer-provided documentation.
hardwired logic:	Logic implemented through the design of an integrated circuit; the programming of a Programmable Logic Device (PLD), Field-Programmable Gate Array (FPGA), Peripheral Interface Controller (PIC), or similar; the integration of smaller hardware components; or mechanical design (e.g., as in lever machines).
implementation statement:	Statement by a manufacturer indicating the capabilities, features, and optional functions and extensions that have been implemented in a system.
inspection:	Examination of a product design, product, process or installation and determination of its conformity with specific requirements or, on the basis of professional judgment, with general requirements.
manufacturer:	Entity with ownership and control over a system submitted for testing.
module:	Structural unit of software or analogous logical design, typically containing several callable units that are tightly coupled.
paper record identifier:	Unique randomly generated code that links the paper record to the corresponding cast vote record.
paper record receptacle:	A secure unit for storing paper records at remote voting locations.
paper record:	Printed record of selections made by the voter.
programmed device:	Electronic device that includes application logic.
published:	Methodologies (e.g., coding conventions, cryptographic algorithms) are considered published if they appear in publicly available media.
remote voting location workers:	Election workers who staff the remote voting locations.
remote voting location:	Locations at which absentee voting takes place.
straight party override:	Ability to make an exception to straight party voting in selected races.
straight party voting:	Mechanism that allows voters to cast a single vote to select all candidates on the ballot from a single political party.
summative usability testing:	Evaluation of a product with representative users and tasks designed to measure the usability (defined as effectiveness, efficiency and satisfaction) of the complete product.

test:	Technical operation that consists of the determination of one or more characteristics of a given product, process or service according to a specified procedure.
third-party logic:	Software, firmware, or hardwired logic that is neither application logic nor COTS; e.g., general-purpose software developed by a third party that is either customized (e.g., ported to a new platform, as is Windows CE) or not widely used, or source code generated by a COTS package.
UOCAVA:	Uniformed and Overseas Citizens Absentee Voting Act
vote capture device:	Device that is used directly by a voter to vote a ballot.
voted ballot:	Ballot that contains all of a voter's selections and has been cast
voter privacy:	The inability of anyone to observe, or otherwise determine, what selections a voter has made.
voting process:	Entire array of procedures, people, resources, equipment and locations associated with the conduct of elections.
voting session:	Span of time beginning when a ballot is enabled or activated and ending when that ballot cast.
voting system:	Equipment (including hardware, firmware, and software), materials, and documentation used to define elections and ballot styles, configure voting equipment, identify and validate voting equipment configurations, perform readiness tests, activate ballots, capture votes, count votes, generate reports, transmit election data, archive election data, and audit elections.
VPN:	Virtual Private Network
VSTL:	Voting System Test Laboratory
white-box:	Uses an internal perspective of the system to design test cases based on internal structure. White box testing strategy deals with the internal logic and structure of the code.
write-in:	To make a selection of an individual not listed on the ballot.

Appendix B: List of References

The following is a list of documents or publications used in the creation of the UOCAVA Pilot Program Requirements

ANSI 02:	ANSI/TIA-968-A: 2002, Technical Requirements for Connection of Terminal Equipment to the Telephone Network.
BS 7799:	Data center certification standard
CERT 06:	CERT® Coordination Center, Secure Coding homepage, July 2006, Available from http://www.cert.org/secure-coding/.
DHS 06:	Department of Homeland Security, Build Security In, July 2006, Available from https://buildsecurityin.us-cert.gov/.
EAC06:	U.S. Election Assistance Commission, Testing and Certification Program Manual, Version 1.0, December 5, 2006. Available from http://www.eac.gov/program-areas/voting-systems/docs/testingandcertmanual.pdf/attachment_download/file.
FIPS 81:	(1980): DES Modes of Operation
FIPS 46-3:	(1999): Data Encryption Standard (DES)
FIPS 140-2:	Security Requirements for Cryptographic Modules
FIPS 180-2:	(2002): Secure Hash Standard (SHA1)
FIPS 186-2:	(2000): Digital Signature Standard (DSS)
FIPS 197:	(2001): Advanced Encryption Standard (AES)
FIPS 198:	(2002): The Keyed-Hash Message Authentication Code (HMAC)
FIPS 200:	Minimum security requirements for federal information and information systems.
FCC 07a:	Title 47, Part 68, Rules and Regulations of the Federal Communications Commission, Connection of Terminal Equipment to the Telephone Network: 2000.
GPO 90:	Performance and Test Standards for Punchcard, Marksense, and Direct Recording Electronic Voting Systems, January 1990 edition with April 1990 revisions, in Voting System Standards, U.S. Government Printing Office, 1990.14 Available from http://josephhall.org/fec_vss_1990_pdf/1990_VSS.pdf.
GPO 99:	Government Paper Specification Standards No. 11, February 1999.
HAVA 02:	The Help America Vote Act of 2002, Public Law 107-252. Available from http://www.fec.gov/hava/hava.htm.
HFP 07:	Human Factors and Privacy Subcommittee of the TGDC, "Usability Performance Benchmarks for the VVSG," August 2007. Available from http://vote.nist.gov/meeting-08172007/Usability-Benchmarks-081707.pdf.
IEEE 00:	IEEE 100:2000 The Authoritative Dictionary of IEEE Standard Terms, Seventh Edition.

IEEE 97:	IEEE/EIA 12207.1-1997, Industry implementation of International Standard ISO/IEC 12207:1995—(ISO/IEC 12207) standard for information technology—software life cycle processes—life cycle data.
IEEE 98:	IEEE Std 829-1998, IEEE standard for software test documentation.
IETF RFC 2246:	(1999): The TLS Protocol Version 1.0
IETF RFC 2510:	(1999): Internet X.509 PKI Certificate Management Protocols
IETF RFC 2817:	(2000): Upgrading to TLS within HTTP/1.1
IETF RFC 2818:	(2000): HTTP Over TLS
IETF RFC 3280:	(1999): Internet X.509 PKI Certificate and CRL Profile
IETF RFC 3369:	(2002): Cryptographic Message Syntax
IETF RFC 3370:	(2002): Cryptographic Message Syntax (CMS) Algorithms
IETF RFC 3546:	(2003): TLS Extensions
IETF RFC 3739:	(2004): Internet X.509 PKI Qualified Certificates Profile
IETF RFC 4279:	(2005): Pre-Shared Key Cipher suites for TLS
ISO 00:	ISO 9001:2000, Quality management systems – Requirements.
ISO 00a:	ISO/IEC TR 15942:2000, Information technology—Programming languages—Guide for the use of the Ada programming language in high integrity systems.
ISO 03:	ISO 10007:2003, Quality management systems – Guidelines for configuration management.
ISO 03a:	ISO/IEC 14882:2003, Programming languages—C.
ISO 04a:	ISO 17000:2004, Conformity assessment—Vocabulary and general principles.
ISO 05:	ISO 9000:2005, Quality management systems – Fundamentals and vocabulary.
ISO 06:	ISO/IEC 23270:2006, Information technology—Programming languages—C#.
ISO 06e:	ISO/IEC 25062:2006 Common Industry Format (CIF) for Usability Test Reports.
ISO 94:	ISO 9706:1994, Information and documentation—Paper for documents—Requirements for permanence.
ISO 95:	ISO/IEC 8652:1995, Information technology—Programming languages—Ada.
ISO 98a:	ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability.
ISO 99:	ISO/IEC 9899:1999, Programming languages—C.
ITU-T X.509:	(2000)/ISO/IEC 9594-8 (2001): Information Technology – Open Systems Interconnection – The Directory: Authentication Framework
Java 05:	The Java Language Specification, Third Edition, 2005. Available from http://java.sun.com/docs/books/jls/index.html.
LOTSE-V:	Legal, Operational and Technical Standards for E-Voting

MIL 83:	MIL-STD-810-D, Environmental Test Methods and Engineering Guidelines, 1983-7-19.
MIL 85:	MIL-STD-1521B (USAF) Technical Reviews and Audits for Systems, Equipments [sic], and Computer Software, rev. December 19, 1985.
MIL 96:	MIL-HDBK-781A, Handbook for Reliability Test Methods, Plans, and Environments for Engineering, Development, Qualification, and Production, April 1, 1996.
MIRA 04:	MISRA-C:2004: Guidelines for the use of the C language in critical systems, MIRA Limited, U.K., November 2004.
Morris 84:	F. L. Morris and C. B. Jones, "An Early Program Proof by Alan Turing," IEEE Annals of the History of Computing, v. 6, n. 2, April 1984, pp. 139-143.
Moulding 89:	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.
MS 05:	Request For Proposal #3443, Mississippi, April 28, 2005.
MS 05:	Paul Vick, The Microsoft® Visual Basic® Language Specification, Version 8.0, 2005. Available from Microsoft Download Center, http://go.microsoft.com/fwlink/?linkid=62990.
NGC 06:	Nevada Gaming Commission and State Gaming Control Board, Technical Standards for Gaming Devices and On-Line Slot Systems, March 2006. Available from http://gaming.nv.gov/stats_regs/reg14_tech_stnds.pdf.
NIST 02:	John P. Wack, Ken Cutler, Jamie Pole, National Institute of Standards and Technology Special Publication 800-41: Guidelines on Firewalls and Firewall Policy, January 2002. Available from http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf.
NIST 03:	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.
NIST 05:	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/.
NIST 05a:	Peter Mell, Karen Kent, Joseph Nusbaum, National Institute of Standards and Technology Special Publication 800-83: Guide to Malware Incident Prevention and Handling, November 2005. Available from http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf.
NIST 07:	Karen Scarfone, Peter Mell, National Institute of Standards and Technology Special Publication 800-94: Guide to Intrusion Detection and Prevention Systems, February 2007. Available from http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf.
NIST 75:	Saltman, Roy, National Institute of Standards Special Publication 500-30, Effective Use of Computing Technology in Vote-Tallying, 1975. Available from http://csrc.nist.gov/publications/nistpubs/NBS_SP_500-30.pdf.
ODBP CR:	ODBP Code Review
ODBP CRM:	ODBP Certification Matrix
ODBP DSF:	ODBP Description of System Features
ODBP P:	ODBP Plan

ODBP SPV:	ODBP System Performance Validation
ODBP SR:	ODBP System Requirements
ODBP SM:	ODBP Security Requirements Mapped to VVSG 2005
ODBP TR:	ODBP Test Report
OMG 07:	OMG Unified Modeling Language Superstructure Specification, version 2.1.1. Document formal/2007-02-05, Object Management Group, February 2007. Available from http://www.omg.org/cgi-bin/doc?formal/2007-02-05.
Oxford 93:	New Shorter Oxford English Dictionary, Clarendon Press, Oxford, 1993.
Pietrek 97:	Matt Pietrek, "A Crash Course on the Depths of Win32™ Structured Exception Handling," Microsoft Systems Journal, January 1997. Available from http://www.microsoft.com/msj/0197/exception/exception.aspx.
PKCS #1:	RSA Cryptography Standard
PKCS #5:	Password-based Encryption Standard
PKCS #7:	Cryptographic Message Syntax Standard
PKCS #8:	Private Key Information Syntax Standard
PKCS #10:	Certification Request Standard
PKCS #11:	Cryptographic Token Interface
PKCS #12:	Personal Information Exchange Syntax Standard
SCAM 01:	Joel Scambray, Stuart McClure, George Kurtz, Hacking Exposed: Network Security Secrets and Solutions, Second Edition, 2001.
SERVE DSF:	SERVE Description of System Features
SERVE EV:	SERVE Election Validation
SERVE R:	SERVE Requirements
SERVE SA:	SERVE Security Architecture
SERVE SACP:	SERVE System Accreditation and Certification Process
SERVE STC:	SERVE Security Test Conditions
SERVE TDP C:	SERVE TDP Checklist
SERVE TRA:	SERVE Threat Risk Assessment
SERVE VVP:	SERVE Vote Verification Process
SERVE WH:	SERVE White Hat
Sourceforge 00:	CEXCEPT (exception handling in C), software package, 2000. Available from http://cexcept.sourceforge.net/.
SP 800-53:	Rev 2 Recommended Security Controls for Federal Information Systems

SP 800-63:	Electronic Authentication Guideline, April 2006. Available from: http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.
SP 800-113:	(2007): DRAFT Guide to SSL VPNs
TRIVS RN:	Testing Requirements for Internet Voting Systems Robert Naegele
UL 05:	UL 60950-1:2005, Information Technology Equipment – Safety – Part 1: General Requirements.
UL 437:	UL 437:2003, Standard for Key Locks. (2003).
UOCAVA PT:	UOCAVA Penetration Testing
UT 04:	Solicitation #DG5502, Utah, 2004-07-09. January 27, 2006.
VOI CAR:	VOI Certification and Accreditation Report
VOI COD:	VOI Concepts of Operations
VOI DSF:	VOI Description of System Features
VOI LEO M:	VOI LEO Manual
VOI LEO SSRS:	VOI LEO Server Software Requirement Spec
VOI PPR:	VOI Pilot Peer Review
VOI PSR:	VOI Pilot System Requirements
VOI Report:	VOI Test Report 2001
VOI SA:	VOI System Arch
VOI SD:	VOI System Design
VOI SP:	VOI Security Policy
VOI SRS:	VOI Software Requirement Spec
VOI STEP:	VOI System Test and Evaluation Plan
VOI STP:	VOI Software Test Plan
VOI TP:	VOI Test Procedures
VOI TR:	VOI Test Report 1999
VSS 2002:	2002 Voting Systems Standards. Available from http://www.eac.gov/program-areas/voting-systems/docs/voting-systems-standards-volume-i-performance.pdf/attachment_download/file
VVSG 2005:	2005 Voluntary Voting System Guidelines, Version 1.0, March 6, 2006. Available from http://www.eac.gov/program-areas/voting-systems/docs/vvsgvolumei.pdf/attachment_download/file
VVSG 2.0:	VVSG Recommendations to the EAC, TGDC, August 31, 2007.
RFI 2007-03:	EAC Decision on Request for Interpretation 2007-03, 2005 VVSG Vol. 1 Section 3.1.1,

	September 5, 2007. Available from http://www.eac.gov/program-areas/voting-systems/docs/certification-docs-eac-decision-on-request-for-interpretation-2007-03.pdf-1/attachment_download/file.
Wald 47:	Abraham Wald, Sequential Analysis, John Wiley & Sons, 1947.

Appendix C: Accuracy Test Case

Some voting system performance attributes are tested by inducing an event or series of events, and the relative or absolute time intervals between repetitions of the event has no significance. Although equivalence between a number of events and a time period can be established when the operating scenarios of a system can be determined with precision, another type of test is required when such equivalence cannot be established. It uses eventbased failure frequencies to arrive at ACCEPT/REJECT criteria. This test may be performed simultaneously with time-based tests.

For example, the failure of a device is usually dependent on the processing volume that it is required to perform. The elapsed time over which a certain number of actuation cycles occur is, under most circumstances, not important. Another example of such an attribute is the frequency of errors in reading, recording, and processing vote data.

The error frequency, called “ballot position error rate,” applies to such functions as process of detecting the presence or absence of a voting punch or mark, or to the closure of a switch corresponding to the selection of a candidate.

Certification and acceptance test procedures that accommodate event-based failures are, therefore, based on a discrete, rather than a continuous probability distribution. A Probability Ratio Sequential Test using the binomial distribution is recommended. In the case of ballot position error rate, the calculation for a specific device (and the processing function that relies on that device) is based on:

- HO: Desired error rate = 1 in 10,000,000
- H1: Maximum acceptable error rate = 1 in 500,000
- a = 0.05
- b = 0.05

and the minimum error-free sample size to accept for qualification tests is 1,549,703 votes.

The nature of the problem may be illustrated by the following example, using the criteria contained in the *Guidelines* for system error rate. A target for the desired accuracy is established at a very low error rate. A threshold for the worst error rate that can be accepted is then fixed at a somewhat higher error rate. Next, the decision risk is chosen, that is, the risk that the test results may not be a true indicator of either the system's acceptability or unacceptability. The process is as follows:

- The desired accuracy of the voting system, whatever its true error rate (which may be far better), is established as no more than one error in every ten million characters (including the null character)
- If it can be shown that the system's true error rate does not exceed one in every five hundred thousand votes counted, it will be considered acceptable. This is more than accurate enough to declare the winner correctly in almost every election

- A decision risk of 5 percent is chosen, to be 95 percent sure that the test data will not indicate that the system is bad when it is good or good when it is bad

This results in the following decision criteria:

a. If the system makes one error before counting 26,997 consecutive ballot positions correctly, it will be rejected. The vendor is then required to improve the system

b. If the system reads at least 1,549,703 consecutive ballot positions correctly, it will be accepted

c. If the system correctly reads more than 26,997 ballot positions but less than 1,549,703 when the first error occurs, the testing will have to be continued until another 1,576,701 consecutive ballot positions are counted without error (a total of 3,126,404 with one error)