National Institute of Standards and Technology
U.S. Department of Commerce

VVSG 2.0
Core Requirements

Benjamin Long
John Wack

September 19-20, 2019

How did we get here?

Baseline requirements:

- VVSG 1.0, VVSG 1.1
- 2007 VVSG Recommendations

New Material:

- Interoperability and common data formats
- Reworkings of accuracy/reliability

VVSG Interoperability and Testing Public Working Groups

- Feedback and input from public working group calls

Where to find the Core Requirements?

- The majority of the core requirements fall under Principles 1 and 2
- Documentation is covered in Principle 3
- Requirements in Human Factors and Security are linked from the core requirements in those areas

	Principle
1	High Quality Design
2	High Quality Implementation
3	Transparency
4	Interoperability

Principle 1: High Quality Design

The voting system is designed to accurately, completely, and robustly carry out election processes.

- **Meaning**
  - This is about
    - a) **specification** and
    - b) **evaluation** of implementations based on specifications.
  - **Focuses on 3 things** in 3 guidelines 1.1, 1.2, and 1.3
    - **1.1** preconditions for specification: voting processes, functions, and variations
    - **1.2** accuracy, realistic volume, defined limits
    - **1.3** testability
  - Has 146 total requirements
- **Guidelines**
  - **1.1** - The voting system is designed using commonly-accepted election process specifications.
  - **1.2** - The voting system is designed to function correctly under real-world operating conditions.
  - **1.3** - Voting system design supports evaluation methods enabling testers to clearly distinguish systems that correctly implement specified properties from those that do not.

Guideline 1.1: Specification of Process, Functions, and Logic

- **Guideline 1.1** - The voting system is designed using commonly-accepted election process specifications.
- **Summary:**
  - This is about specification of process, functions, and logic.
- **Notes:**
  - One can't expect to build or test functions that aren't well-specified.
  - Guidelines in the first principle are about making sure we have a sufficient specification of processes and functions.
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 - Election processes (Requirements by Voting Activity)
  - VVSG 1.0, 1.1, 2007 - Voting variations
  - EAC 17 functions
- **Requirements**
  - 1.1.1 Election definition
  - 1.1.2 Equipment setup
  - 1.1.3 Opening the polls
  - 1.1.4 Ballot activation
  - 1.1.5 Casting
  - 1.1.6 Recording voter choices
  - 1.1.7 Ballot handling for paper scanners
  - 1.1.8 Closing the polls
  - 1.1.9 Tabulation
  - 1.1.10 Reporting results

Guideline 1.2: Accuracy, Logical Limitations, and Volume Testing

- **Guideline 1.2** - The voting system is designed to function correctly under real-world operating conditions.
- **Notes:**
  - Any process is not realistically correct if it is not accurate.
    - Accuracy testing is essential for ensuring that well-specified processes are correctly implemented in whatever technologies are selected.
    - This is where accuracy from previously standards resides (which tests to ensure that correct functionality is preserved across operations).
  - Specifying technology configurations so that they can support realistic elections sizes, complexities, and loads - i.e., realistically modeling the logic and limits of real elections and basing them on good logical models - is another aspect of the first principle.
    - This is where traditional volume and load testing (mock elections) is addressed from previous standards.
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 accuracy, misfeed rate - requirements/testing
  - VVSG 1.0 and 1.1 reliability testing
  - VVSG 1.0, 1.1, 2007 logical limits (TDP, e.g., max number ballots that can be counted, possible vote total (counter capacity), or number of ballot styles, etc.)
- **Requirements:**
  - 1.2-A Assessment of accuracy
  - …
  - 1.2-B Assessment of reliability
  - …
  - 1.2-F Misfeed rate benchmark
  - 1.2-G Respond gracefully to stress of system limits
  - 1.2-H Handle realistic volume

Guideline 1.3: Testability, Conformance Clause, and Implementation Clause

- **Guideline 1.3 -** Voting system design supports evaluation methods enabling testers to clearly distinguish systems that correctly implement specified properties from those that do not.
- **Summary:**
  - This is essentially about:
    - testability
    - the conformance-clause for any system (as defined by the standard)
    - the implementation-clause for any system (as claimed by a manufacturer)
    - Information supporting evaluation (TDP, documentation, test information)
- **Notes:**
  - The standard should have a way to
    - map requirements to applicable aspects of given system
    - determine clearly and unambiguously:
      - What the precise configuration of functions is intended for that system,
      - How they are realized,
      - How they can be observed, and
      - How they can be compared to the specified functions and properties
  - **This guideline is realized:**
    - in the way the standard is organized - so that it may be effectively mapped to particular systems/configurations (i.e., **conformance clause, implementation statement, TDP,** etc.); and
    - in the ability of **specific tests** to carry out their function (i.e., clarity and accuracy of test methods and assertions).
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 conformance clause for each system of functions (as defined by the standard)
  - VVSG 1.0, 1.1, 2007 implementation statement/clause for each system of functions (as claimed by a manufacturer)
  - VVSG 1.0, 1.1, 2007 TDP and documentation
  - VVSG 1.0, 1.1, 2007 tests and testing information
- **Requirements**:
  - Covered in TDP, documentation, and testing requirements being transferred to EAC manuals

Principle 2: High Quality Implementation

The voting system is implemented using high quality best practices.

- **Meaning**
  - This is about
    - implementation of specifications using best practices in hardware, software, telecom, data, quality assurance, and configuration management: the primary technology-oriented components of previous standards
  - Has 7 guidelines
  - Has 85 total requirements
- **Guidelines**
  - **2.1** - The voting system software is implemented using trustworthy materials and best practices in software development.
  - **2.2** - The voting system is implemented using best practice user-centered design methods that consider a wide range of representative voters, including those with and without disabilities, and election workers.
  - **2.3** - Voting system logic is clear, meaningful, and well-structured.
  - **2.4** - Voting system structure is modular, scalable, and robust.
  - **2.5** – The voting system supports system processes and data with integrity.
  - **2.6** - The voting system handles errors robustly and gracefully recovers from failure.
  - **2.7** - The voting system performs reliably in anticipated physical environments.

Guideline 2.1: Implement Systems Using High Quality Materials and Software Development Best Practices

- **Guideline 2.1** - The voting system software is implemented using trustworthy materials and best practices in software development.
- **Summary:**
  - This is essentially about engineering and implementation best practices.
- **Notes:**
  - Indicates
    - Use of trustworthy materials (in general)
    - Use of specific best practices for software (in specific)
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 Requirements about uses of high-quality materials and parts
  - VVSG 1.0, 1.1, 2007 SW engineering, workmanship, and assurance
  - VVSG 1.0, 1.1, 2007 QA/CM best practices
- **Requirements:**
  - 2.1-A Acceptable programming languages
  - …
  - 2.1-C Acceptable coding conventions
  - 2.1.1-A General build quality
  - 2.1.1- B, C High quality products, parts
  - …
  - 2.1.1-E Durability
  - 2.1.2 - Maintainability

Guideline 2.2: Implement Systems Using Human Factors Best Practices

- Covered by Dr. Laskowski in HF briefing

Guideline 2.3: Implement System Logic Using (HW, FW, SW) Logic Development Best Practices

- **Guideline 2.3 -** Voting system logic is clear, meaningful, and well-structured.
- **Summary:**
  - This is about the design and implementation of logic in a voting system, no matter where it occurs – in software, firmware, hardware, processes, functions, etc.
- **Notes:**
  - This is about the following primary ideas: a) **Clear logic** (Good syntax), b) **Meaningful logic** (Good semantics), c) **Well-structured logic** (Good structure)
  - This guideline (2.3) focuses on the **logic** of the system whereas the next guideline (2.4) focuses on the **architecture** of the system.
  - **Logic** - include many specific kinds of logic such as: logic in fully developed by a manufacturer, in COTS, in libraries (border logic), obtained from 3rd-parties by modified by manufacturers, in firmware, in hardware, etc.
  - **Meaningfulness** - related to important considerations regarding:
    - the ability of a tester / reviewer / test (method, tool, protocol) / maintainer to
      - easily and clearly understand the intended logic/process
      - be able to trace its flows of inputs/outputs/control, etc.
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 Workmanship (for HW, SW, etc.)
  - VVSG 2007 types of logic
  - VVSG 1.0, 1.1, 2007 SW modularity, modules described in TDP and other documentation
- **Requirements:**
  - 2.3-A Block-structured exception handling
  - 2.3-B Wrapping legacy code
  - 2.3-C Separation of code and data
  - 2.3.1 Control flow structures

Guideline 2.4: Implement System Structure Using System Engineering Best Practices

- **Guideline 2.4 -** Voting system structure is modular, scalable, and robust.
- **Summary:**
  - This is about the design and implementation of structure/architectures (systems of elements) (and not about logic, which is covered in 2.3).
- **Notes:**
  - **Modular -** apply the same kinds of ideas from 2.3, but at the systems level
    - Organize the design of the overall system to manage the complexity it represents
    - Decompose system's deployable units into entities that connect, compose, and work together to achieve specified/implemented functions and processes
    - Supports system properties: portability, easily extensible, etc.
  - **Scalable -** can easily change the size of the system:
      - without it breaking and while still correctly and accurately perform its specified/implemented functions
  - **Robust** - can vary the demands on the system so that the system continues to function in a reliable fashion, over time
    - A robust system organizes the implementation of its processes and functions so that
      - the system does not become unstable or inoperable at the slightest variation in demands, loads, or operating conditions
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 modularity
  - VVSG 1.0, 1.1, 2007 reliability
- **Requirements:**
  - 2.4-A Modularity
  - 2.4-B Module testability
  - 2.4-C Module size and identification
  - 2.4-D Lookup tables in separate files

Guideline 2.5: Implement System with High Integrity Best Practices

- **Guideline 2.5 –** The voting system supports system processes and data with integrity.
- **Summary:**
  - This is about the **logical and structural integrity at all layers** of the implementation of the system.
- **Notes:**
  - To ensure the integrity of an implemented process or of data flowing through the system, one must take specific steps at each layer of the technology stack to make sure this has been addressed.
  - **Data integrity** -
    - ensure **data integrity** via error correction/detection at the level of **error correction codes/algorithms**
    - preserve the **integrity of transmitted data** – e.g., when casting a vote and the network/modem connection is lost
  - **Process integrity** to ensure that well-specified processes and logic (from P1) have been correctly implemented and are correctly and accurately operating
    - In testing as well as in operations, this can be achieved by:
      - Employing robust, defensive programming and design practices
      - wherein processes have defined conditions (pre- and post-conditions, as well as invariant conditions) that must be met - whether these are implemented in software, procedure, etc.
  - This guideline is different from other principles/guidelines in security – e.g., “system integrity" - e.g. **installation and baseline-management** *vs.* **implementation and operation**
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 integrity of data, transmission, error correction
- **Requirements:**
  - 2.5-A Avoid self-modifying code
  - 2.5-B Avoid unsafe concurrency
  - 2.5.1 Code integrity, 2.5.2 input/output errors, 2.5.3 output protection, 2.5.4 error handling, common cases: overflow, etc.

Guideline 2.6: Implement Systems with Robust Error Handling and Graceful Failure Recovery

- **Guideline 2.6 –** The voting system handles errors robustly and gracefully recovers from failure.
- **Summary:**
  - This is about the system's error detection, correction, and recovery.
- **Notes:**
  - **Graceful recovery from failure**
    - **if an error occurs**:
      - in logic, process, and data
      - during telecommunications, transmission, reception, decoding, encoding, encryption, decryption, etc...
      - in interaction with the system
      - in a node or a subsystem level
    - → **the system, as a whole, should** employ well-known **reliability strategies** in each case.
      - Reliability strategies often imply strategies of redundancy.
      - Reliability in error-correction codes (an element of previous standards) often operates on the principle of redundant coding.
      - Reliability at the system level implies that, where appropriate, one might need to load-share, creating redundant or failover systems so that the overall process being carried out by someone can continue successfully, even if a given subsystem fails, its failover configuration can robustly recover from that error.
  - Avoid **single points of failure**
  - These general ideas show up in every discipline – engineering, U/A, security, etc.
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 error handling
  - VVSG 1.0, 1.1, 2007 failure processing
  - VVSG 1.0, 1.1, 2007 failure recovery
  - VVSG 1.0, 1.1, 2007 error detection/correction
- **Requirements:**
  - 2.6-A Surviving device failure
  - 2.6-B No compromising voting or audit data
  - 2.6-C Surviving component failure
  - 2.6-D Controlled recovery
  - ...

Guideline 2.7: Implement Systems Using Reliability and Testing (HW, Environment) Best Practices

- **Guideline 2.7 –** The voting system performs reliably in anticipated physical environments.
- **Summary:**
  - This is about the reliability, hardware, and physical climate, environmental, and stress testing.
- **Notes:**
  - This guideline
    - Addresses **traditional hardware testing**
    - Addresses **various kinds of physical/stress testing** - whether it relates to electrical stress (EMC, etc.), shock stress, environmental stress (temperature, humidity), etc.
    - Reflects the **physical environments** in which a voting system may be stored, transported, or operated
    - Requires that the system **continue to correctly and accurately perform its functions** under these conditions
- **Based on Information from Previous Standards:**
  - VVSG 1.0, 1.1, 2007 reliability
  - VVSG 1.0, 1.1, 2007 HW / physical environmental testing
- **Requirements:**
  - 2.7-A Ability to function in physical environment (non-operating)
  - 2.7-B Ability to support maintenance and repair in physical environment conditions
  - 2.7-C Ability to support transport and storage in physical environment conditions
  - 2.7-D Ability to support storage temperatures in physical environment
  - 2.7-E Ability to support storage humidity levels in physical environment
  - …
  - 2.7-G Ability to operate as intended at low and high temperatures
  - 2.7.1 Ability to withstand electrical disturbances

Principle 3: Transparency

The voting system and voting processes are designed to provide transparency.

- The voting system is designed and implemented such that it provides transparency to its operations and accuracy
- Requires clear documentation and capability to inspect its workings
- Has 3 guidelines, 82 total requirements
  - **3.1** - The documentation describing the voting system design, operation, accessibility features, security measures, and other aspects of the voting system can be read and understood
  - **3.2** - The processes and transactions, both physical and digital, associated with the voting system are readily available for inspection
  - **3.3** - The public can understand and verify the operations of the voting system throughout the entirety of the election

3.1: The documentation describing the voting system design, operation, accessibility features, security measures, and other aspects of the voting system can be read and understood

- Addresses transparency via requirements to have complete, clear documentation
- Complete, usable instructions for how to use the voting system and all features
- Explanations of how the voting system works so that it can be better understood and audited for correctness
- Requirements:
  - 3.1.1 – System overview
  - 3.1.2 – System performance
  - 3.1.3 – System security documentation
  - 3.1.4 – Software Installation
  - 3.1.5 – System operations
  - 3.1.6 – System Maintenance
  - 3.1.7 – Training material

3.2: Processes and transactions, both physical & digital, associated with the voting system are readily available for inspection

- Addresses documentation that explains how the voting system must be configured before opening the polls
- Complete, usable overviews of the setup process
- Assurance that the voting system is configured properly
- Requirements:
  - 3.2-A – Setup inspection process
  - 3.2-B – Minimum properties included in the setup inspection process
  - 3.2-C – Setup inspection record generation
  - 3.2-D – Installed software identification procedure
  - 3.2-E – Software integrity verification procedure
  - 3.2-F – Election information value
  - 3.2-G – Maximum and minimum values of election information storage
  - 3.2-H – Variable value inspection procedure
  - 3.2-I – Backup power operational range
  - 3.2-J – Backup power inspection procedure
  - 3.2-K – Cabling connectivity inspection procedure
  - …

3.3: The public can understand and verify the operations of the voting system throughout the entirety of the election

- Addresses documentation for any use of data encodings, logging, and other features that otherwise would prevent inspection and harm transparency.
- Open specification of logging file and contents
- Full documentation of encodings including data packing prior to encoding
- Requirements:
  - 3.3-A – System security, system event logging
  - 3.3-B – Specification of common data format usage
  - 3.3-C – Bar and other codes
  - 3.3-D – Encodings
  - 3.3-E – Audit

Principle 4: Interoperability

The voting system is designed to support interoperability in its interfaces to external systems, its interfaces to internal components, its data, and its peripherals.

- Deals with the voting system's devices interfacing to each other and to the external world using NIST common data formats and industry-standard protocols and interfaces.
- Goal is that devices can be more easily used and swapped with devices from different manufacturers and that the EAC can implement component certification.
- Has 4 guidelines, 11 total requirements
  - **4.1** - Voting system data that is imported, exported, or otherwise reported, is in an interoperable format
  - **4.2** - Standard, publicly-available formats for other types of data are used, where available
  - **4.3** - Widely-used hardware interfaces and communications protocols are used
  - **4.4** - Commercial-off-the-shelf (COTS) devices can be used if they meet applicable VVSG requirements

4.1: Capability to use common data formats for imports and exports

- Use of the CDFs for many imports and exports as an alternative to manufacturer’s format.
- Must also document how the manufacturer implements the format.
- Requirements:
  - 4.1-A – Data export and exchange format
  - 4.1-B – Election programming data input and output
  - 4.1-C – Tabulator report data
  - 4.1-D – Exchange of cast vote records (CVRs)
  - 4.1-E – Exchange of voting device election event logs
  - 4.1-F – Voting device event code documentation
  - 4.1-G – Specification of common format usage

4.2: Standard, publicly-available formats for other types of data are used, where available.

- Use the CDFs if possible, otherwise use a format that is convenient for others.
- If no format/standard exists, manufacturer must make their format available without fee.
- Requirements:
  - 4.2-A – Standard formats
  - 4.2-B – Public documented manufacturer formats

4.3: Widely-used hardware interfaces and communications protocols are used.

- Use non-proprietary hardware interfaces and wireless protocols.
- Use of common standards such as USB and IEEE 802.x
- Whatever is used must be freely available
- Requirements:
  - 4.3-A – Standard device interfaces

4.4: COTS devices can be used if they meet applicable VVSG requirements.

- COTS devices are allowed as part of the voting system as long as all relevant VVSG requirements are still satisfied.
- Simplification of electrical requirements also helps
- Requirements:
    - 4.4-A - COTS devices meet applicable requirements

CDF Status - In a Nutshell

- Four CDF Specifications:
  - Election Results Reporting Versions 1, 2 – NIST SP 1500-100
  - Election Event Logging – NIST SP 1500-101
  - Cast Vote Records – NIST SP 1500-102
  - Voter Records Interchange – NIST SP 1500-103
- Election Modeling:
  - Definition of election processes, data and definitions, and data interchanges
  - Foundational for CDF and other election-related development
- Election Terminology Glossary Specification:
  - Also used in Next VVSG requirements and glossary
- Voting Methods Specification:
  - Formal definition of voting variations, including extensive work with RCV
  - Hope is that states coalesce around these methods

Who is Using the CDFs Currently?

- 1500-100 - Election Results Reporting:
  - AP ingests ERR feeds
  - Google in national and international elections
  - NC, OH, VA, WI
  - LA County
  - Several states require it in RFPs
  - Several vendors starting to support it
- 1500-103 - Voter Records Interchange:
  - OH, VA
  - A number of states require it in RFPs for VRDB upgrades or are using it in development
- 1500-102 - Cast Vote Records:
  - DARPA research project in secure voting systems
  - RCVRC Consortium (now certified for use with certain ES&S EVS versions)
  - Under development in MI, MA, MN, OR, UT
  - Other organizations

SP 1500-100 Election Results Reporting

- For pre, post-election data and election results
- An interoperable EMS, VRDB import/export format
- Used in NC, OH, LA County in 2016, Google use overseas and with VIP, WI in 2020, working with other states and manufacturers
- V2 has JSON format to reduce file sizes for large amounts of detailed results
- Includes support for Ranked Choice Voting results

SP 1500-101 Election Event Logging

- A basic format to hold election events, i.e., opening of polls, casting of a ballot, login of an admin, etc.
- Manufacturers will log as they do currently but include capability to export in this format
- Some log items will be useful in EAC EAVS, e.g., electronic pollbook logs items
- Essential that event code descriptions be made publicly available for election auditing

SP 1500-102 Cast Vote Records

- Deals with voted ballot information exported from scanners, EMS
- Increases interoperability between devices and simplifies collection and reporting of CVRs
- Includes support for newer voting variations, e.g., RCV
- Supports adjudication, auditing, and reporting

SP 1500-103 Voter Records Interchange

- Implements online voter registration exchanges with a (state) VR authority
- Includes data from NVRA and FPCA and state-specific forms
- Was extended to include VR updates or maintenance
- OH implemented previous version, is now being used in parts of VA
- Can be extended to deal with electronic poll book exchanges with VRDB

SP 1500-104 Election Terminology Glossary

- Started as a glossary for next VVSG
- Scope was expanded beyond VVSG terminology
- Also deals with synonyms for different terms used in same/similar ways across the states

Questions?

Benjamin Long - benjamin.long@nist.gov

John Wack - john.wack@nist.gov

CDF information – https://vote.nist.gov