VotingWorks

Comments to EAC on VVSG 2.0

June 22nd, 2020

Dear EAC Commissioners & Staff,

Please find, attached, our comments on the VVSG 2.0 Requirements.

As we all know, each certification requirement raises the bar to produce a compliant voting system. Most of the time, that's exactly right: there ***should*** be a high bar for the systems that safeguard our democracy. Sometimes, though, a requirement imposes a burden without a corresponding benefit. Especially where requirements with no clear benefit impact the usability, affordability, or security of the system, we argue that the requirement should be changed or removed. Affordability of voting equipment should be a major goal of the VVSG, as it is a major consideration for election officials. Each requirement's burden should be weighed against its benefit.

To that end, we've provided feedback in three categories:

1. **Promoting innovation and competition** – the VVSG should promote competition and innovation, producing a dynamic market of vendors working to produce the best possible system at the best possible price. This benefits jurisdictions and voters.

2. **Goal-driven requirements** – the VVSG requirements should be driven by goals, not by specific implementation choices that meet those goals in a particular way. This is important for flexibility, innovation, and future-proofing of the requirements.

3. **Modern requirements** – the VVSG requirements should be pruned of carry-over requirements from prior versions of the VVSG that are outdated. Software development methods have evolved significantly in the last 15 years, and some requirements are carried over from that long ago.

We hope this feedback is helpful, and we welcome any further discussion, of course.

Sincerely,

Ben Adida, PhD
Executive Director
VotingWorks

Table of Contents

Promoting Innovation and Competition
- Interoperability & Certification Modularity
- User-Centered Design
- All Vendors in the Same Boat
- Transparency

Goal-Driven Requirements
- Scanning
- Workflow Management
- Resiliency, Durability, Maintenance
- Accessibility
- User Experience & Design
- Security

Modern Requirements
- Internet connectivity & secure boot
- Modern L&A
- Continuous Operation
- COTS
- Coding Requirements

Promoting Innovation and Competition

In 2016, a Wharton Business School report on the business of voting systems[1] concluded that the market for voting equipment is deeply broken, because of difficult requirements combined with a small market size. In fact, our organization, VotingWorks, is the first new vendor in this space in the last 12 years.

Understanding how the certification process privileges legacy vendors over newcomers is critical to understanding the lack of competition in the industry. For example, legacy vendors are allowed to continue to certify systems under the VVSG 1.0 standards from 2005. This indefinite loophole for old systems means that only new vendors are currently required to certify to VVSG 1.1. No legacy vendor has certified to 1.1 in the 5 years that it has been available, and we fear the same loophole might be provided to allow legacy vendors to sidestep VVSG 2.0. This essentially creates two different certification requirements for old and new vendors.

We want to urge the EAC to take these critical issues into account when defining the VVSG 2.0. The EAC should find ways to promote new vendors, innovation, and competition. Election officials and voters will benefit from such a dynamic market of vendors.

Interoperability & Certification Modularity

We commend the EAC for **Section 4** -- Interoperability. This set of requirements will significantly improve competition in the marketplace by making it possible for jurisdictions to mix and match components from different vendors. This **must be paired with modular certification**. A new vendor should be able to build and sell an excellent BMD without also having to offer hand-marked paper ballot support, or vice-versa. Of course, jurisdictions would still need to procure all the systems needed to help every voter vote independently and secretly, but there's no reason this must come from one vendor alone.

The use of open specifications that are well documented (**Section 1.1.7H**) is great. We strongly support this.

We also support the use of CDF standards (**Sections 1.1.1Z, 1.1.10T, 4.1, 4.2**). We want to point out two things:

- The CDF standards, while strong, have not yet been fully implemented by any mainstream voting system. It is thus possible that changes to the standard will be necessary.

[1] https://publicpolicy.wharton.upenn.edu/business-of-voting/

- The CDF standards are very extensive, and, for security purposes, a voting system may want to only implement a subset of the data model, possibly using a simplified format.

For those two reasons, we encourage the EAC to only require the use of ***any*** openly documented and royalty-free format, and encourage but not require the use of CDF.

User-Centered Design

We strongly commend the EAC for the introduction of user-centered design as a core practice (**Section 2.2.A**). We encourage the EAC in two directions:

1. Allow a number of specific implementation requirements to be replaced with user testing that proves the goal, so that implementation may be more flexible. We will provide a number of examples in the remainder of this document.
2. Allow for limited field tests of systems prior to full certification, so that usability in real elections can be validated before system designs are locked in through certification.

All Vendors in the Same Boat

Though this next issue is not covered in the current requirements, it is critical: ***all vendors should be in the same boat***. The VVSG cannot provide loopholes for legacy vendors. We recommend that the EAC disallow selling any voting system not compliant with the VVSG 2.0 past a particular date in the near future. If that isn't possible, then at least, if any one vendor is allowed to sell equipment under a prior version of the VVSG, all vendors should be allowed to do so, too.

Importantly, we strongly urge the EAC to ***make this decision now***, before the details of the VVSG 2.0 are finalized. Only if legacy vendors know they will be forced to abide by the VVSG 2.0 will they have the incentive to thoroughly evaluate the requirements and point out where these requirements may be too onerous.

Transparency

We strongly commend the EAC for the transparency requirements in **Section 3**. We suggest going further: full documentation of voting systems should be public. Let the public and world know how these systems work.

Goal-Driven Requirements

One of the most important and positive innovations in the VVSG 2.0 process is the focus on high-level principles and goals. Some particularly positive examples of this, that clearly state a goal without prescribing how to accomplish it, include:

- **Section 7.3.D** on consistent relationship
- **Section 7.3.E** on feedback to user action.
- **Section 7.3.F** on the ability to correct a ballot before it is cast and counted.
- **Sections 8.3 and 8.4** focusing on user tests: we would like to see other parts of the requirements rely on user tests that prove outcomes, rather than prescribed implementations.
- **Sections 9 and 10** are particularly well focused on goals rather than implementation mechanism.

In the remainder of this section, we will point out areas that could benefit from the same goal-driven approach exemplified in the sections above.

Scanning

- **Sections 1.1.7-D and 1.1.7-F** require physical ballot separation (e.g. outstacking or scanning interruption) when batch feeding. This precludes the use of a number of high-quality COTS scanners, when the desired outcome can be achieved in a different way - in particular, digital adjudication of write-ins is faster and easier.
- **Section 1.1.7-G** requires a scanner to indicate, on a jam, whether the jammed ballot has been read or not. This is particularly difficult to accomplish, and is only one of many methods to prevent double-counting, the actual goal. We suggest this be phrased as a desired outcome:

  “the scanning process should be able to recover from a paper jam relatively easily without causing a double count of the ballot that caused the jam.”

- **Section 1.1.7L** - not allowing scanners to record marks outside the contest option position means precluding more advanced mechanisms for detecting a voter’s intent, e.g. a circling of the candidate name. This requirement seems too tight, and could be loosened to allow for evolving technology

Workflow Management

- **Section 1.1.8E** prevents reopening of the polls once they are closed. In our experience in the field, any action that is unrecoverable leads to disproportionate negative impact on voters when poll workers inevitably make a mistake. The goal here is clearly to prevent cheating if a rogue poll worker later reopens the polls and casts extra ballots. But this can be achieved in other ways: extensive logging of all poll-opening-and-closing actions, maintaining separate tallies for each span of time where polls are open so this can later be corrected, or disallowing reopening of the polls only past a certain time on the clock. The requirement should be more focused on the goal -- preventing ballot box stuffing -- without mandating a particular implementation.
- **Section 1.1.10I** prevents tally reports before polls close. This is difficult to enforce in software, especially if legal orders extend the close of polls to unexpected time. This requirement could be fulfilled more easily by well-defined process.

Resiliency, Durability, Maintenance

- **Section 1.2.J** requires every component to survive a storage failure. This is quite expensive and not the only way to build a reliable system. Instead, components can simply be affordable enough, and the architecture resilient enough, that on failure, the whole component -- e.g. the ballot card encoder -- can be replaced. The requirement here should be focused on removing single point of failure and mandating fast-enough recovery from failure, not on mandating that each component have redundant storage.
- **Section 2.1.1E** requires durability for 10 years. What if a voting system can be built that costs ¼ the price but only lasts 5 years? This is better for election officials and voters, but is disallowed by the standard. We suggest striking this requirement altogether, as we expect states will simply compare the durability and price of systems as part of their own RFP process.
- **Section 2.1.2B** requires a particular path to maintainability in the field -- what if a vendor wants to simply have defective equipment shipped back to them and a new unit sent out? We suggest this approach should not be precluded.

Accessibility

- **Sections 5.2D, 5.2E, and 7.2-A** require synchronized audio and coordinated visual/audio cues in the context of different explicit modes of operation of the ballot-marking device. While this may be one good implementation path, it comes

with significant added operational complexity: the machine now needs to be placed in a given mode, rather than being workable by any voter at all times. A better requirement could mandate user-centered design with a specific requirement to test with well defined voter cohorts that cover various disabilities. Testing would then validate that the approach chosen by the vendor works.

- **Section 6.1C** requires that the voter be have to disable video output. Like the previous point, this mandates operational complexity. Even if this approach is one good implementation path, the requirement could focus on user-centric design where a manufacturer can try different approaches to accessibility, as long as these are tested successfully with appropriate voter cohorts.
- **Section 6.2A** requires independent casting by voters with disabilities. This is the requirement that has caused significant difficulty for manufacturers in VVSG 1.1, as it is particularly hard to achieve with paper ballots and, where the goal can be achieved, it contradicts the security requirement of **Section 9.1.5-G** that disallows any physical capacity in a BMD to alter a ballot after it has been cast. A better requirement here would focus on the ***privacy*** of casting, not the requirement for assistance.
- **Sections 7.1.K, 7.1.L, and 7.2.G** mandate certain rates of speech, certain speech frequencies, and specific control by the voter -- we recommend instead that this requirement be met through user testing. If a manufacturer can design a system that is usable by enough voters, it should be acceptable, regardless of how many frequencies and rates of speech it supports. **Section 7.1.M** is a good example of a goal-driven version of this, requiring that the sound be understandable by voters, as verified by testing.
- **Section 7.1.N** requires Braille which we know is not used that broadly. We recommend focusing on user testing for blind voters.
- **Section 7.1.P** requires different controls of different colors. This may not be the only way to achieve usability by voters with limited dexterity. We recommend focusing on user testing instead.

User Experience & Design

- **Sections 7.1.E and 7.1F** require specific use of color. These seem like solid recommendations, but, again, other approaches might be just as good, and it would be useful to leave them open and focus, again, on user testing.
- **Section 7.1.G** mandates text sizes -- once again we recommend user testing instead.
- **Section 7.1.H** mandates no scrolling up to 200%, which seems to imply that the screen should be mostly blank at 100% magnification. This is another requirement that could be much more robustly enforced as a goal of usability, validated by user testing, rather than specific implementation.
- **Section 7.1J** requires sans-serif -- can this be replaced with a user test?

- **Section 7.2.E** specifies touchscreen gestures. While these make sense today, they may not in 5 years as new patterns emerge. This can and should be evaluated with user testing.

Security

Some security requirements are excellent in that they focus on goals:

- **Section 15.2** on presentation of errors clearly outlines goals without constraining implementation.
- **Section 15.4** outlines least privilege design while leaving to the manufacturer to explain how they have achieved this design goal. This is also great.

Other security requirements are too specific and would be better as outcomes:

- **Section 11.2.2** on RBAC is quite constraining: it defines specific roles and permissions. A manufacturer might well find a simpler path if they are allowed flexibility here, as long as the roles and permissions are documented and reviewed appropriately.
- **Section 11.3** requires multi-factor authentication for software updates, which doesn't always make sense depending on the architecture. In particular, a manufacturer may choose to perform updates only as complete reinstalls of the entire software stack, which can then be verified as correctly installed. Multi-factor authentication is not always sensible in that context. The goal here should be to ensure that only authorized software is running on the machine, and the vendor should be required to document how they achieve this goal.
- **Section 11.3** also requires 2fa for poll workers to open/close polls. In our field experience, this is operational complexity that will likely lead to more failures (e.g. "I can't open the polls because I forgot my PIN") rather than security. A better approach might be to log all poll open/close events for later auditing in case of concerns.
- **Section 12** requires a number of alarms, though it's not clear these are needed if the manufacturer can clearly explain how they meet the goal -- to ensure that machines cannot be corrupted on the day of the election.
- **Sections 12.2 and 14.2.M** mandate that ports be physically blocked from access. If the manufacturer can show that they're inactive, this is unnecessary and costly, especially for COTS equipment. The goal should be focused on preventing unauthorized software modification or data modifications, not on the specifics of how that's done.
- **Section 13.2** requires digital signatures on all data transferred. This is a significant operational overhead for election officials, who now need to manage a public-key infrastructure. This is well understood by security professionals to be particularly onerous and error-prone. The goal is to build layers of defenses against modification of election results. This could be done with a combination of hash comparisons,

RLAs, etc. Digital signatures will likely lead to more operational failures than protection.

- **Sections 14.3.2A/B/C/D** requires software to be signed before it can be installed. Some systems, however, will choose to never install new software, only to install a complete stack and provide verification means to ensure it hasn't been modified. The goal here is to ensure that only acceptable software can run. This specific implementation path should not be mandated.
- **Sections 14.4.A / 14.4.B - 15.3.A, 15.3.1A, 15.3.1B / C / D / E / F / G** also focus on specific ways to prevent unauthorized software from running. They are specific to one architectural approach to software installation and updates. As per the previous point, the requirement could be simplified to "preventing unauthorized software from running" and letting the manufacturer describe how they do this.
- **Section 15.1.D** requires a very detailed set of logs that is quite onerous. Is this level required? What is the goal?
- **Sections 15.4.D + 15.4F** mandate a firewall or IDS, but what if all networking is turned off? The requirement should focus on inability to connect to the system from the outside, not on specific ways to do it.
- **Section 14.3.1C** - how can the software log anything if boot has been prevented? Especially with hardware-based trusted boot, this is particularly challenging.

Modern Requirements

We recommend modernizing certain requirements that have been carried over from previous certification versions. We want to commend the EAC for starting down this path of modernization already:

- **Sections 1.2.H and 1.2.I** are good modern approaches to reliability
- **Section 2.2.A** on user-centered design is the modern approach.

Internet connectivity & secure boot

We understand the fear that exists around internet connectivity for voting devices. However, the trend in modern systems is to do rapid deployment of security fixes over the air, as often as needed. When combined with signed updates and trusted boot to ensure that the system has not been otherwise modified, this is a very secure approach. We recommend that the VVSG 2.0 take into account that, 5 years from now, requiring in-person security updates, rather than over-the-air, may be hopelessly antiquated.

Modern L&A

L&A as a separate mode of operation is increasingly frowned upon by forward-thinking election officials, who prefer to truly simulate an election, since testing mode might not be exactly the same as real mode. It should be possible for a system to support this mode of testing. Removing “test mode” also makes for simpler, and thus better software, as much as more realistic testing. This applies to **sections 1.1.2L, 1.1.3B/C/D**.

Continuous Operation

Section 1.2.F mandates Continuous Operation, 163 hours “without error.” This is not the modern way to engineer resilient systems. Instead, in modern designs, a subsystem can fail, as long as that failure is detectable and quickly remedied without significantly impacting the voter. For example, a BMD might fail after 48 hours of operation, possibly auto-rebooting or requiring a manual reboot. If this can happen in under 2 minutes, voters are not affected. It is better to design systems that “Fail fast” and can recovery quickly, than to expand incredible resources on building systems that almost never fail, but also don’t know how to fail gracefully. For example, dealing with **Sections 2.5.2 and 2.5.3** might best be done with a reboot if garbage data is detected, in a fail fast approach. This should be possible.

COTS

We strongly encourage the EAC to lean on COTS hardware components as much as possible. The billions of dollars of investment that have gone into COTS hardware, including securing the supply chain, building hardware trusted boot, failing gracefully, etc. are properties to be leveraged, not recreated from scratch. To that end, we suggest two tweaks

- **Section 2.1.2C** requires nameplates and labels applied to systems, which precludes the possibility of EOs ordering COTS equipment and installing software on their own. Can this requirement be adjusted?
- **Section 8.1A** requires a minimum resolution of 1280x800 for informational screens, which is not a common COTS display. More common is 1366x768. Allowing a vertical resolution of 768 seems reasonable and would not rule out many 10-11” COTS displays.

Coding Requirements

- The coding requirements in **Sections 2.3 and 2.4** are fine, but the focus in modern software systems is on testing much more than on prescriptions of how the code is structured. Testing & test coverage provide a much better confidence level in quality, and should be the focus.
- **Section 2.5.B** - there is no way to prove that code is free of concurrency issues. This requirement should be tweaked to describe the measures taken to prevent / address concurrency issues.