Guidance for voting system standards

Designing and testing election systems to be usable and accessible

Designing and testing a user-centered and usable voting system

An overview of what you need to know to meet the requirements in VVSG 2.0 and design a voting system that is usable for voters and poll workers, and accessible for voters with disabilities.

Updated December 2018

Contents

Part 1:
User-centered design and voting systems

User-centered design (UCD) is a way of building systems focused on meeting users' own goals. UCD includes activities and methods for discovering what users need, and what meets those needs.[1] In UCD, designers use these methods before development starts and on each new iteration of a system. This iterative, user-need-driven approach is an established best practice in public and private product development.[2]

The VVSG 2.0 requires voting system vendors report on their user-centered design methods to make their systems usable, accessible and less risky for election administrators.

This guidance offers suggestions on conducting and documenting UCD to meet this requirement. It describes key user-centered design elements and outputs. The companion documents suggest participant user-centered design methods to use during systems development.

This guidance builds on current professional practice in user-centered design and usability and international standards, applying these practices to voting systems. The ISO has defined UCD's methods in each of these categories in standards, including:

- Ergonomics of human-system interaction (ISO 9241)
- Common Industry Format (CIF) for usability test reports) (ISO/IEC 25062)
- Context of use description (ISO/IEC 25063)
- User needs report (ISO/IEC 25064)
- User requirements specification (ISO/IEC 25065)
- Evaluation reports (ISO/IEC 25066)

[1] UCD has a long and deep history in technology development (See *Foundations for Designing User-Centered* by F.E Ritter, Springer 2014). This definition of UCD is taken from World Wide Web Consortium's definition (https://www.w3.org/WAI/redesign/ucd) and the human-centered design ISO standard (ISO 9241-210:2010).
The U.S. Department of Health and Human Service's "Evidence-based guidelines for website usability" review much of the foundational literature in UCD (https://guidelines.usability.gov).

[2]See Usability.gov's summary of UCD's benefits (https://www.usability.gov/what-and-why/benefits-of-ucd.html). Also see one analysis showing how "industry standard" UCD has become:
Vredenburg, K., Mao, J. Y., Smith, P. W., & Carey, T. (2002, April). A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 471-478).

What is user-centered design?

UCD includes methods to conduct at each phase of a product's development. These methods fall into three categories.

Conducting research to understand what users need.

In UCD, the best way to learn what users need is asking or observing them.

UCD includes methods for conducting qualitative and quantitative research about what people need. This research is meant to help designers better understand user goals, preferences, environments and constraints. This research needn't be as robust as academic or peer-reviewed research. It should be just robust enough to decide what to start building.

Evaluating system iterations against user needs.

In UCD, the best way to learn whether a system meets user needs is having users try to use it.

UCD focuses on continuously asking users to try using systems or prototypes. Their experience is the best indication about whether a feature will work well. Repeated evaluation identifies what works and what doesn't.

Changing your system to better align with what users need.

In UCD, the best way to build a user-centered system is to continuously improve it.

UCD only accomplishes its goals if designers change their system based on what they learn in research and evaluation. Designers should change the system based on what users say is easy and hard. They should remove features that don't meet user needs and hone features that do.

UCD and the VVSG

VVSG 2.0 requires vendors to incorporate UCD methods into systems development.

The VVSG 2.0's second principle is "high quality implementation." User-centered design is one requirement for high quality implementations:

> Requirement 2.2: The voting system is implemented using best practice user-centered design methods, for a wide range of representative voters, including those with and without disabilities, and election workers.

To meet this requirement, vendors must submit a report describing: a listing of user-centered design methods used, the types of voters and election workers included in those methods, how those methods were integrated into the overall implementation process, how the results of those methods contributed to developing the final features and design of the voting system.

Meeting VVSG system usability testing requirements (8.3A, 8.4A) is only part of user-centered design.

VVSG 2.0 (8.3A and 8.4A) requires vendors to report on a system usability test with voters and poll workers. Meeting these requirements is one part of incorporating UCD into development. Usability testing ready-to-deploy systems is an important way to demonstrate their strengths.

The final test can provide information about the final product, such as typical time to vote on the standard NIST ballot. The final test also provides a baseline to compare against other systems or to show improvements with updated systems.

But there's more to UCD than usability testing ready-to-deploy systems. Vendors should include UCD methods throughout their development process, not just at the end. Good results on usability testing on the final product is a sign of a good user-centered design process conducted throughout design and development.

Other VVSG requirements still apply.

Adopting UCD methods does not exempt vendors from other VVSG requirements. Development choices based on UCD still have to meet VVSG requirements.

But user-centered design methods can show systems meet VSG requirements. Think of UCD as an approach to gathering evidence your system meets the VVSG's requirements.

UCD helps ensure systems meet broader principles of the VVSG.

UCD also helps meet the VVSG's broader principles. With the help of UCD, ballots should be:

- **Equivalent and consistent:** All voters have access to mark and cast their ballot without discrimination.
- **Marked as intended**: Ballots are presented in a clear, understandable way, and is operable by all voters.
- **Cast as marked:** Ballots are cast as marked, both secretly and privately.
- **Usability tested:** Meet performance standards for usability and accessibility.
- **Meet web accessibility standards:** Browser-based systems meet web accessibility standards, in addition to voting standards.

What does UCD help accomplish?

User-centered design methods help make systems usable, accessible, and follow the principles of universal design. ISO standards define each of these outcomes. In turn, these systems are lower risk for election administrators to adopt and manage.

Usability

Usability is a feature or attribute of a design. NIST relies on the International Standards Organization (ISO) definition of usability in ISO 9241:

- The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.
- Effectiveness: the accuracy and completeness with which specified users can achieve specified goals in particular environments
- Efficiency: the resources expended in relation to the accuracy and completeness of goals
- Satisfaction: the comfort and acceptability of the work system to its users and other people affected by its use
- The system won't meet the standard if it is frustrating, and time-consuming to use, and unpleasant as an experience.

Accessibility and Principles of Universal Design

The ISO/IEC standards also address accessibility. Accessibility is:

The extent to which products, systems, service, environments and facilitates can be used by people from a population with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

Voting systems are legally required to be accessible for people with disability. Designers can aim for accessibility through the **Principles of Universal Design**[3], or designing a product or service so that it can be accessed, understood and used to the greatest extent possible by all people regardless of their age, size, ability or disability. Systems with universal design follow seven principles.

- Equitable Use
- Flexibility in Use
- Simple and Intuitive Use
- Perceptible Information
- Tolerance for Error
- Low Physical Effort
- Size and Space for Approach and Use

These principles are incorporated into the principles and guidelines for voting systems in the VVSG to help ensure that voters with the widest range of capabilities can use the primary voting system because it has been designed with their range of needs in mind. It does not mean one universal voting system that everyone must use.

Mitigating risk in election administration

Adopting UCD ultimately mitigates risk by exposing systems to a few people at a time throughout development, instead of entire jurisdictions as systems are adopted. Voters and poll workers are less likely to make mistakes on usable systems. Universally designed systems allow almost all people to vote independently and limit poll worker involvement in the voting booth. Attention to user needs early in development solves upstream and downstream problems. Purchasers feel confident that they will not encounter problems of usability or accessibility for users.

Who should UCD include?

To realize its benefits, UCD should include all the people who use a system, including:

- Voters, including those who
- use different assistive technologies to use a voting system
- use the system in a language other than English
- are experienced and inexperienced voters

[3] The Principles of Universal Design are seven general principles to guide the design process or evaluate existing designs. You can find the principles and more information at the Center for Universal Design at North Carolina State University

- Election workers

When does UCD happen?

UCD can and should happen anywhere and everywhere throughout systems development.

- **Before development starts**, designers can conduct open-ended, "generative" research about what voters, poll workers and election administrators need. This research might include interviews, observations of existing systems, or customer satisfaction surveys.
- **As development continues**, designers can ask users to try using prototypes (rough, basic versions) of potential product ideas. These prototypes can be as simple as paper sketches attached to cardboard boxes.
- **As the system takes shape**, designers can usability test individual features (like selecting a candidate in a context). When it's possible to use the system for an "end to end task" (like voting and casting a ballot), designers can test individual features.
- **When the system is ready for deployment**, designers should conduct a final usability test, covering all types of voters and poll workers.

UCD self-assessment

Vendors can ask themselves these questions to evaluate their UCD process.

- How were users included in the process of designing and developing the product?
- How many hours of exposure did the product team have with directly observing users of the systems during the development of this release of the product?
- What kinds of inspection, evaluation, and observation did the product team engage in with users?
- What were the major insights the product team gained from each of the activities they included users in?

Self-assessment table

Objective	**Assessment**				
	Not met **1**	**Minimal** **2**	**Acceptable** **3**	**Good** **4**	**Strong** **5**
Users were included in the design and development process		Company employees in the beginning of the lifecycle	Community and advisory groups throughout the lifecycle	Range of individual users at every phase	Diverse, large sample that was geographically distributed
Team* exposure / direct observation of users			2 hours per person every 6 weeks		
Methods used to learn about and understand users' needs	Surveys and / or focus groups – asking about feelings and opinions	Presentations and demos to community groups; other forms of public comment	Basic usability testing of features and functionality	Field research, observing voters and poll workers in polling places	Co-design, small and large usability tests with diverse users

* "Team" means everyone in the vendor company who influences design decisions, from business analysts to legal and compliance.

How is UCD documented?

The best sign of user-centered design is a usable, accessible system. But documents data, and other artifacts can also demonstrate UCD. Different documents are appropriate for different phases of a project.

During development, internal documents help track progress towards the final product.

- Early user research (with methods like surveys or interviews) might produce a list of user needs, preferences and constraints.
- Usability testing during the product development life cycle might produce a list of issues to address in future iterations.

For system certification, standard templates report on work to meet UCD requirements. Purchasers can also use these reports to compare systems.

- The UCD process is reported using the [TBD] template to demonstrate High Quality Implementation in VVSG 2.0 Principle 2, *Requirement 2.2-A-User centered design process*
- Final system usability testing is reported using the *CIF for Voting Systems* template, based the Common Industry Format for Usability (ISO/IEC 25062). This report template meets the requirements for usability for voters in Principle 8, requirement *8.3-A-Usability tests with voters*.
- Final system usability testing with poll workers is reported using the *Poll Worker Usability Testing to* meet the requirements for usability for poll workers in Principle 8, requirement 8*.4-A-Usabilty tests with poll workers*

Part 2: Implementing user-centered design (UCD) for voting systems

Introduction: How to do UCD for voting systems

Building usable, robust systems requires a range of user-centered methods, including:

- **Inspection** to identify usability defects and potential usability problems
- **User surveys** and **interviews** that elicit problems, opinions, and impressions from users and potential users
- **Observation** of users in a controlled or field setting
- **Usability testing** of prototypes, features or systems

These methods can be used at many parts of the development process. For example, a vendor might:

- Observe users in the field during election preparations
- Interview and survey voters and election administrators about specific questions that arise from the observation
- Repeatedly usability test low-fidelity prototypes of new systems
- Inspect near-finished software for usability problems.
- Conduct large-sample usability tests to establish a system's general usability

Each method and stage requires different documentation. Early research might generate a list of user needs and requirements, while later usability might generate reports. Final system usability testing reports should use the Common Industry Format (CIF) for usability test results.

Key UCD methods for voting systems

Although UCD encompasses a range of methods (see ISO/IEC 15288), five are particularly relevant to voting systems design.

Inspection

Goal: Identify large usability problems in an existing system or prototype. This method is less useful for identifying minor or subtle problems or gathering user needs.

Related standards: ISO/IEC 15288

Process (Based on the U.S. General Services Administration's 18F Method Cards[4])

1. Recruit a group of three to five people familiar with usability best practices. These people aren't necessarily designers, but they might be people know a lot about designing usable voting systems.
2. Create a checklist of "heuristics" or general usability best practices the system should meet. Heuristics can be system specific or broad. For example:
    a. "The website should keep users informed about what is going on, through appropriate feedback within reasonable time."
    b. "The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms."
    c. "Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue."
3. Ask each evaluator to individually use the system and go through the checklist.
4. After individual inspections, gather evaluators to compare checklists. Note issues that multiple people observed. Go back to the system and confirm the problems.

Selected references

- "Heuristic Evaluations and Expert Reviews" from Usability.gov. (https://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html)
- "Heuristic Evaluation" in *Usability Inspection Methods* by Jakob Nielsen (John Wiley & Sons 1994).
- "Heuristic evaluation of user interfaces" by Jakob Nielsen and Rolf Molich in *Proceedings of the ACM CHI'90 Conference*, page 249-256.

[4] Based on "Heuristic Evaluation" from the 18F Design Method Cards (https://methods.18f.gov/discover/heuristic-evaluation/)

Interviews

Goal: To better understand the variety of user needs, challenges and contexts (e.g. voters or election administrators). Less useful for establishing quantitative estimates of how many users have what needs and use what features, or validating particular designs.

Related standards: ISO/IEC 15288

Process (Based on the U.S. General Services Administration's 18F Method Cards[5])

1. Write down some topics you'd like to ask about, and then some specific questions for each topic. Good topics might include the individual's history with elections, how they prepare for elections, what they do on election day and what challenges they encounter.
2. Recruit interviewees best able to discuss your topics of interest. If you are asking about voter's experiences, you might look for people new to voting. If you want to learn about election officials without experience with your system, you might look for new poll workers.
3. Sit down one-on-one with each interviewee. (Or two-on-one: a participant, interviewer and note-taker.) Introduce yourself. Explain why you're conducting the interview.
4. Let the conversation flow freely, but keep coming back to your topics. Be comfortable with silences that let your interviewee elaborate. Ask lots of "why is that" and "how do you do that" questions. Take thorough notes.
5. After you've interviewed several people, compare notes. Look for common goals, challenges and environments.

Selected references

- "Individual interviews" from Usability.gov (https://www.usability.gov/how-to-and-tools/methods/individual-interviews.html)
- "Tips for capturing the best data from user interviews" by Ryan Sibley from the 18F blog. (https://18f.gsa.gov/2016/02/09/tips-for-capturing-the-best-data-from-user-interviews/)
- *Interviewing Users* by Steve Portigal (Rosenfeld 2013)

Surveys

Goal: To measure how many users report having certain opinions or behaviors. Less useful for understanding why participants have those opinions or how they might respond to a particular design.

[5] Based on "Stakeholder and user interviews" from the 18F Design Method Cards (https://methods.18f.gov/discover/stakeholder-and-user-interviews/)

Related standards: ISO/IEC 15288

Process

1. Identify the goals of your survey. In particular, select the type of people you want to learn about (for example, election administrators unfamiliar with your voting system). Then decide what you want to learn about them (for example, what type of voting system they prefer).
2. Prepare survey questions related to your goals. Write a couple questions for each goal. Try to make them as fast to answer as possible. Use mostly fill-in-the-blank or multiple choice questions. Ask only a couple open-ended questions.
3. Pilot test or ask a colleague or friend to take your survey. Ask them what they thought each question meant. Edit any unclear questions.
4. Distribute your survey to an appropriate sample of people. See the references for more suggestions on survey sampling.
5. Use appropriate statistics to summarize your results. If you chose to present inferential statistics (like test results or p-values), explain what they mean.

Selected references

- "Four Tips for Survey Design" from Digital.gov (https://digital.gov/2014/11/10/4-tips-on-great-survey-design/)
- "Surveys" in Observing the User Experience: A Practitioner's Guide to User Research by Elizabeth Goodman, et al. (Elsevier 2012)
- "Using self-report questionnaires in research: A common on the use of a controversial method" by P.E. Specter in *Journal of Organizational Behavior* (*15*, 385-392).

Observation

Goal: To see how users actually complete a task (as opposed to how they describe it). Less useful for eliciting users' feelings and reflections on a task.

Related standards: ISO/IEC 15288

Process (Based on the U.S. General Services Administration's 18F Method Cards[6])

1. Decide what task you want to observe. Then figure out who conducts that task. For example, you might decide to observe voting machine set up. Poll workers normally do that, so you might want to observe them.

[6] Stakeholder and user interviews. (2017) *18F Design Method Cards.* Retrieved from https://methods.18f.gov/discover/stakeholder-and-user-interviews/

2. Arrange to go to the place where people complete the task. For example, you might want to go to poll worker training to see how poll workers learn. Make sure you have permission from the person being observed and their supervisor, if applicable.
3. While observing, ask the participant to act normally. Pretend you're a student learning how to do the job. Ask questions to help you understand what the person is doing and why.
4. At the end of the session, explain what you have learned and check for errors.
5. Immediately after, write up your notes. When you've observed several people doing the same task, try writing a step by step description of how people complete the task and what challenges they encounter.

Selected references

- "Task analysis and observation" from Usability.gov (https://www.usability.gov/how-to-and-tools/methods/task-analysis.html)
- "Contextual inquiry" in Observing the User Experience: A Practitioner's Guide to User Research by Elizabeth Goodman, et al. (Elsevier 2012)

Usability testing

Goal: Learning what works well for users (and doesn't) about a current prototype, feature or system.

Related standards: ISO/IEC 15288

Process[7].

1. Decide why you're conducting the usability test. For example, you might want to learn how easily voters/poll workers can use your system, what mistakes they make, or whether the system works like they expect.
2. Schedule one-on-one sessions with a few real users. For example, find five voters or five poll workers. Pick people who haven't been involved in developing the system.
3. When people arrive for the test:
4. Go over what will happen.
5. Ask them to complete a couple tasks, like voting or setting up the machine. Watch, listen for questions (don't answer them) and comments (write them down).
6. When they are done voting, ask them to walk you through what they did and why.

[7] Based on "Usability testing" from the 18F Design Method Cards (https://methods.18f.gov/validate/usability-testing/)

7. After you've complete for few sessions, look for patterns in what challenge people. Compile a list of issues.

Further reading

- See "Usability testing and voting systems" later in this guidance
- "Usability testing" from Usability.gov. https://www.usability.gov/how-to-and-tools/methods/usability-testing.html
- Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems by Steve Krug (New Riders, 2009)

When to use each method

UCD methods should shape products at all their phases. UCD can join linear, agile, rapid and other development approaches, as described by ISO/IEC/IEEE 12207:2017. Regardless of your development approach, choose a method based on the goals of a development stage.

Goal	Related methods		
Develop voting system requirements and understand user needs	**Observation** of current or potential people using existing voting systems	**Interviews** with election administrators or voters	**Surveys** of existing customers about how they use their current systems
Validate initial feature ideas and prototypes	**Usability testing** of rough prototypes with a small group	**Accompanying interviews** exploring new ideas after testing	
Identify usability problems in features under-development	**Usability testing** of in-development software, focusing on particular features	**Inspection** of near-completed features by experts for common usability issues	
Establish evidence of finished system usability	**Usability testing** with a diverse set of potential users	**Surveys** of people using the new system to identify usability issues discovered in the wild	

How to document UCD

Documenting UCD serves dual purposes:

- Recording your understanding of users and their needs to inform future product development work
- Generating evidence that you used a UCD process throughout systems development

A product should have its own UCD portfolio of related documentation. The portfolio should have artifacts, diagrams, and reports that describe plans, data, and findings (following ISO/IEC standards). Include all of these in user needs reports to show that your team uses an iterative, UCD process.

Requirements documents and user needs inventory

Develop lists and narratives of what you know about user needs from your observations, interviews and surveying. These documents should focus on the *context of use*: who the users are, what their tasks are, what their surroundings are like as they perform the tasks (ISO/IEC 15288).[8]

Prototypes and sketches

Demonstrate early ideas through prototypes of progressive fidelity. They should be tested iteratively with a wide range of users (ISO/IEC 15288) and annotated with what you learned from each.[9]

Issues lists

As product development progresses, document the results of inspection and small scale usability testing with a list of issues it discovered. You can prioritize the issues, describe their effects on the user experience and note which matter to whom. Note which issues were solved and which weren't.[10]

---

[8] See the 18F method cards for journey mapping, mental modeling and personas at methods.18f.gov.

[9] See Usability.gov's prototyping page for more information (https://www.usability.gov/how-to-and-tools/methods/prototyping.html). Also see *Presumptive design: Design provocations for innovation* by Frishberg and Lambdin (New Riders, 2017).

[10] See "Rolling issues lists" in Handbook of usability testing: how to plan, design and conduct effective tests by Dana Chisnell and Jeffrey Rubin (John Wiley & Sons, 2008). Also see "Reporting results" in Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems by Steve Krug (New Riders, 2009)

Simplified report

Summarize mid-project usability tests with simple reports. This report should be detailed enough to give outsiders a sense of what you did. It starts by describing the test itself and then progresses to what you learned from it.[11]

Suggested headings for a simplified usability report

Executive summary

About the test (method: Inspection, survey, observation, evaluation)

- Who participated
- What did we test and why
- Materials tested (short description)
- Tasks performed by participants (list of users' tasks)
- Test facilitator tools (script, demographic questionnaire, SUS, etc.)

Results

- What did participants find confusing or difficult?
- General issues (bulleted list)
- Issues specific to features, functionality, or platform (bulleted list and / or screenshots / illustrations with descriptions of issues)
- Suggested changes to improve user performance (screenshots, mockups, or other illustrations of recommended changes)
- Changes implemented (description and screenshots or other illustrations)

Common industry format for usability test reporting

Usability testing reports submitted as part of certification must follow the Common Industry Format for Usability described in ISO/IEC 25062. For more information on this report structure,

[11] You can download a template for this simplified report and see an example at electiontools.org. This outline is roughly the same as the Common Industry Format for Usability in ISO/IEC 25062. Also see "Reporting usability test results" from Usability.gov (https://www.usability.gov/how-to-and-tools/methods/reporting-usability-test-results.html)

see “CIF for Voting Systems: A modified Common Industry Standard template for reporting on usability testing of voting systems.”

Getting started: run a simple usability test

Start applying UCD wherever your product is in its lifecycle. The only way to reach deep understanding of users' needs is by observing users' behavior and how they interact with voting systems. Most teams start by conducting a simple, first usability test on a product or feature. This can be informal. At its essence, usability testing is simple.

All you need for a usability test is:

- a person who is like your target user
- a version of the product you want to test
- a quiet and comfortable space to be with the participant
- someone to moderate the interview
- someone to take notes.

For example, it doesn't need a usability lab. Instead, you can invite voters or poll workers to your offices and set up the least equipment needed for what you want to test. You need not set up a situation that is exactly like a polling place.

Everyone on the product team should observe the individual sessions. After each session, the team should discuss what they heard and saw.

After you have done one or two informal usability studies, it is likely that you will have more questions about users. At that point, you will branch out to other methods.

If you are starting from scratch on a product, the process might have a different starting point. You might want to start with market research and surveys to understand the problems your customers are feeling pain on. After that, graduate to observing someone performing the task in the current way. In these sessions, you can ask follow up questions to understand users' goals and tasks.

Part 3: Usability testing and voting systems

What is usability testing?

A key practice in user-centered design (UCD)—and a key requirement in the Voluntary Voting System Guidelines (VVSG) 2.0—is ensuring that the product you've made is usable by the end users, in this case voters and election workers.

One way to do this is through directly observing users like voters as they perform typical tasks to reach their own goals. This is called **usability testing**.

Usability testing–testing the voting system with voters and poll workers–is not the same as the conformance testing to usability and accessibility requirements in VVSG 2.0. It is also different from beta testing, quality assurance or user acceptance testing:

- Unlike beta testing, usability testing involves a structured session with a user and a facilitator and observers, not just gathering feedback after a limited release.
- Quality assurance focuses on testing whether the product meets its specifications. Usability testing focuses on whether the system meets its users' needs. Usability testing can identify some bugs that QA would. However, its real power is testing whether system created something useful to users.
- User acceptance testing focuses on whether a system meets the designer's *assumptions* about what users need. Typical user acceptance testing involves project managers stepping through new features with a user's needs in mind. UAT is a helpful tool, but doesn't substitute for having actual users test a new feature or system.

Usability testing and the VVSG

VVSG 2.0 (8.3A and 8.4A) requires vendors to report on a system usability test with voters and poll workers.

Ideally, the usability tests to meet these requirements are not a system's only usability tests. VVSG Requirement 2.2 requires systems be implemented with best practice UCD methods. usability tests during development are also key to building usable systems.

Why conduct usability tests on voting systems?

Usability testing throughout development is key to developing usable systems. It's particular important for voting systems because:

- Robust usability testing generates evidence a voting system will work as intended when it's deployed. Usability testing, reported in the Common Industry Format, can build state and local jurisdiction confidence.
- Mid-development usability testing is an easy way for designers--who are usually very comfortable with technology--to find what works with people who are less comfortable. Voters and poll workers have diverse technical skill and language ability and its particularly important voting systems work for everyone.
- Many voting systems require voters and poll workers to interact with multiple devices simultaneously. Multi-device systems open a new range of possible usability problems. End-to-end usability testing identifies those issues.
- Like most hardware systems, voting systems are complicated and time-consuming to manufacture. Usability testing early prototypes can save development time and money later.

Important elements of voting system usability tests

There are numerous introductions to usability testing and guidelines for usability testing with voting systems (See *Further reading*). This guidance focuses on important elements usability testing for voting systems.

Finding a range of people like your systems' eventual users.

Voting systems have a particularly wide range of potential users. To make systems usable for all of them, usability tests have to include all of them. When designing usability tests, at any stage of development, make sure your participants have a diversity of:

- **Experience voting or administering elections.** People new to voting or election administration behave differently than experienced voters or administrators. If your test participants have many degrees of experience, you'll find problems specific to new users as well as the nuanced needs of experts.
- **Experience with different styles and brands of voting systems.** People experienced with a particular voting machine style (DRE, BMD, etc.) or brand will compare your system with what they're using now. If your system is similar to what they're used to, they'll often find it easier to use. Make sure you include people who's previous experiences will help and not help them navigate your system.

- **Comfort and proficiency with technology.** People's comfort with technology in general has a large effect on their attitudes towards new voting systems. Your usability test should include people who will be excited to use new technology, and also people will be hesitant to try it.
- **Assistive technology users.** People who use assistive technology should be able to independently use your system. "Assistive technology" is a large umbrella, so consider testing with at least one member of each of the following groups:
- People who use screen readers
- People who use screen magnification
- People who have limited use of their hands
- People who use only tactile keypads or dual-switch controls (without screen readers)
- People who use wheelchairs
- **Language preferences.** Make sure your test includes people who want to use your system in languages other than English. Usability test your systems' alternative language interface with speakers of several other languages to catch problems switching languages, translations of system messages, or other language-specific issues.
- **Literacy.** People read their language(s) with varying degrees of fluency. Usability testing a system with low- and high-literacy people ensures the text and formatting are simple enough for everyone to follow.

Ask people to complete tasks that will reveal potential problems.

Usability testing centers on asking users to complete tasks with the system. The main voting system tasks are relatively simple (e.g. select candidates, cast ballots, set up the system, etc.) Just asking people to complete simple tasks with the system often doesn't conjure real world scenarios voting systems must handle. To elicit the subtle complexity of elections, try these strategies:

- **Give participants instructions about who to vote for.** Tell voters to vote for particular candidates, change their previous votes, overvote, undervote. Otherwise, many will take the simplest route through the system and not surface common problems.
- **Use the NIST test ballot for usability testing.** One way to surface problems is using a test ballot designed with a contests that can elicit common voting system problems, including types of contests, contests that are very long or very short, contests with similar names, and other complexities. NIST has created a sample ballot specification with realistic names you can use or adapt.

Observe what's hard for people, not what they say they want.

In usability testing sessions, people often say what they like and dislike without the system. In voting system usability testing, likes and dislikes can be driven by what's similar to their current

voting system. Focus on people's behaviors and understanding instead of their stated questions. Explore questions like:

- What tasks do they struggle to complete? Are there tasks they can't complete at all?
- Do they understand what they've selected or entered?
- Does their understanding of what's happening match what the system is actually doing?

Respond to patterns across users.

Usability testing elicits all sorts of idiosyncratic behavior. Voting system usability tests are no exception. You probably don't have time to solve every problem you see users encounter. Instead, focus on identifying:

- **Problems many participants have.** If many types of people struggle to complete a task, the root problem is probably wide-spread and worth addressing.
- **Problems that most affect the most vulnerable users.** Look for issues disproportionately experienced by users who can't ask for help without violating their own privacy, such as assistive technology users who have to invite poll workers into their voting booth. Also look for problems for people who can't easily ask for help, including people who aren't comfortable speaking English or who have a speech disability.
- **Problems that, even if rare, could have severe effects on data integrity.** For voters, these problems might be somewhat unclear instructions about casting or discarding ballots. Even if a small percentage of users experience the problem, the election effect could be severe. For poll workers, these problems might be inadvertently misconfiguring tabulation systems. Even if only one precinct misconfigured a tabulator, it could have severe election day results.

Reporting usability test results

As noted in *Part 2- Implementing user-centered design (UCD) for voting systems*, usability test results can take a variety of formats. What's most important is that the format matches the goal of the test. For example:

- Early usability tests might capture their results in **issues lists**. To enable quick iteration, early usability tests might put less emphasis on documenting process and focus more on documenting key learnings.
- Key end-to-end usability tests might be documented with a **simple report** detailed enough to give outsiders a sense of what you did. See the suggested headings in Part 2.
- Final usability testing reports used to meet VVSG 2.0 requirements for certification must follow the **Common Industry Format for Usability** described in ISO/IEC 25062. For more information and the VVSG reporting template, see *CIF for Voting Systems: A modified Common Industry Standard template for reporting on usability testing of voting systems*.

Possible types of usability testing for voting systems

There are several types of usability tests you might want to run on voting systems in development. The culmination of these tests is the VVSG-required usability test, but there are several other types of usability tests make systems more usable.

Type of test	What you test	What you learn	Common outputs
Early-stage prototype tests	Prototypes of possible systems or features (made from paper or digital prototyping tools)	What works or doesn't about a rough pre-development idea	Prototypes annotated with learnings; revised user needs documents
Feature-specific testing during development	A particular screen, function or feature of an in-development voting system	What elements of the current design work (and don't)	Issues lists or simple usability testing reports
End-to-end testing during development	The entire system a user would need to complete a task like "voting" or "setting up a voting system"	How the components of the system work well together for users (and don't)	Issues lists or simple usability testing reports
Pre-certification usability testing	A ready-to-deploy voting system	How your system will work on election day	Usability test report that follows the CIF for voting systems

UCD Reading List

Books

Usability Testing Essentials: Ready, Set...Test! by Carol M. Barnum (Morgan Kauffman, 2010)

Moderating Usability Tests - Principles & Practices for Interacting by Joe Dumas and Beth Loring (Morgan Kauffman, 2008)

A Web for Everyone: Designing Accessible User Experiences by Sarah Horton and Whitney Quesenbery (Rosenfeld Media, 2014)

Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems by Steve Krug (New Riders, 2009)

Forms that Work: Designing Web Forms for Usability by Caroline Jarrett (Morgan Kauffman, 2009)

Letting Go of the Words: Writing Web Content that Works, 2nd Edition by Janice (Ginny) Redish (Morgan Kauffman, 2012)

Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition by Jeffrey Rubin and Dana Chisnell (Wiley, 2008)

Websites

Usability.gov - https://www.usability.gov/
Tools, information, and resources about usability and UCD . Includes templates and forms for usability testing

18F Method Cards - https://methods.18f.gov/
A collection of tools to bring human-centered design into your project.

Plain language.gov - https://www.plainlanguage.gov/
Home of the federal plain language guidelines, including information about testing content and documentation. Includes checklists for plain language and writing tips.

WebAIM - https://webaim.org/
Resource site for web and digital accessibility from the Center for Persons with Disabilities and Utah State University. Includes checklists for WCAG 2.0 and Section 508.

W3C Internationalization - https://www.w3.org/standards/webdesign/i18n
Guidance on designing for multiple languages or writing systems

Related guidance documents

NIST offers guidance and templates specific to several types of voting system usability testing:

- "How to test for voter usability" - an outline of a protocol for usability testing to meet VVSG requirements
- "Ballot specifications and other usability testing materials"
- "CIF template for voting systems"
- "How to test for poll worker usability" and related appendices