

## **From Error to Error: Why Voters Could not Cast a Ballot and Verify Their Vote With Helios, Prêt à Voter, and Scantegrity II**

**Claudia Z. Acemyan<sup>1</sup>, Philip Kortum<sup>1</sup>, Michael D. Byrne<sup>1,2</sup>, Dan S. Wallach<sup>2</sup>**

<sup>1</sup>Department of Psychology, Rice University

<sup>2</sup>Department of Computer Science, Rice University

6100 Main Street, MS-25

Houston, TX 77005 USA

{claudiaz, pkortum, byrne}@rice.edu and dwallach@cs.rice.edu

### **ABSTRACT**

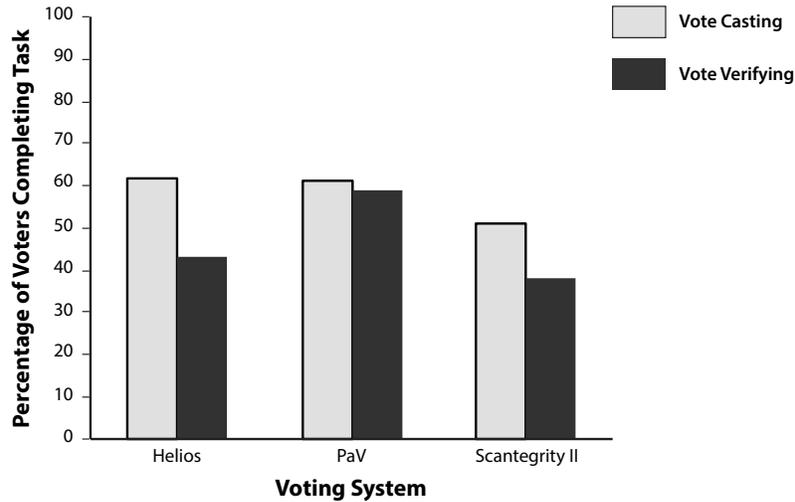
The aim of this paper is to identify user errors, and the related potential design deficiencies, that contributed to participants failing to vote cast and vote verify across three end-to-end voting systems: Helios, Prêt à Voter, and Scantegrity II. To understand why voters could not cast a vote 42% of the time and verify that their ballots were cast and counted with the tested e2e systems 53% of the time, we reviewed data collected during a system usability study. An analysis of the findings revealed subjects were most often not able to vote with Helios because they did not log in after encrypting their ballot but before casting it. For both Prêt à Voter and Scantegrity II, failing to vote was most frequently attributed to not scanning the completed ballot. Across all three systems, the most common reason participants did not verify their vote was due to not casting a ballot in the first place. While there were numerous usability failures identified in the study, these errors can likely be designed out of the systems. This formative information can be used to avoid making the same types of mistakes in the next generation of voting systems—ultimately resulting in more usable e2e methods.

### **INTRODUCTION**

Imagine a real, large-scale election in which nearly half of the voters cannot cast a vote. To make matters worse, many of these voters thought they cast a ballot that would be counted in the election tally so they did not take further action, such as asking for help. As for the voters who realized they could not figure out how to vote, some of them gave up, went home, and wanted to avoid voting in the future. An abysmal scenario like this could potentially result in altered election outcomes and disenfranchised voters, which is completely unacceptable.

While this scenario is made-up, it is not completely farfetched. In a previous paper (Acemyan, Kortum, Byrne, & Wallach, 2014), close to half of the participants (42%) could not cast a vote using the tested voter verifiable systems in a mock election (see Figure 1). As for the vote-verification process, even more participants could not verify their vote—an optional yet fundamental system feature that helps make sure that the end-to-end systems are accurate and transparent, while keeping individuals' votes private. If these systems—as tested—were deployed in a real election, it would not be surprising if similar outcomes resulted.

This previous usability paper by Acemyan et al. was summative. The purpose of the reported test was to assess the usability of three, representative e2e voting systems—Helios, Prêt à Voter (PaV), and the Rice implementation of Scantegrity II. Per ISO 9241-11, three measures were reported to assess system usability: efficiency (time to vote cast and time to vote verify), effectiveness (per-contest error rate and percentages of both cast votes and verified votes), and user satisfaction (System Usability Scale, SUS, rating for vote casting and vote verifying). Since these standard measures were used along with an established protocol to assess voting system usability, the paper was able to not only summarize the state-of-the-art of e2e systems at a given point in time, but also compare them to all of the previously tested voting methods. The advantage of conducting a summative assessment like this is that researchers are able to understand how the usability of various systems generally compares to one another (Rubin & Chisnell, 2008).



**Figure 1.** Percentage of voters completing vote casting and vote verification as a function of e2e voting system. These findings were reported in a previous paper (Acemyan et al., 2014).

Specific usability problems and their root causes were not reported in the previously published summative paper. Typically these types of findings are reported in a formative usability report. In a formative report, usability problems are identified so that they can be fixed in the next iteration of the systems (Sauro, 2010). Hence, the aim of this paper is to identify usability problems related to vote casting and vote verifying failures, since voting and verifying are two primary user goals. Concrete design suggestions that have the potential to address the most impactful usability impediments can aid in the development of future iterations of e2e systems, making them easier to use. While this method will not guarantee the identification and resolution of all system usability problems, it can likely improve the vote-casting and vote-verification completion rates in e2e systems.

Helios, PaV, and Scantegrity II have evolved since the study was conducted. The systems' development teams have likely even recognized some of the same usability failures reported in this paper and addressed them through alterations to system designs. Nonetheless, this formative usability assessment is still of value. It is still important for the voting research community to understand exactly why the systems failed as configured in this study so that developers of any voting system, or any party who selects and sets up voting methods for an election, can avoid making the same mistakes in the future.

## METHODS

The participants, materials, procedures, and general research design were the same as those described in our previous paper (Acemyan et al., 2014). To summarize, 37 participants who were U.S. eligible voters participated in a mock election. This sample size was sufficient to identify usability problems related to vote casting and vote verifying. Research has shown that relatively small sample sizes can be used to identify the majority of usability problems (e.g., Lewis, 1994; Schmettow, 2012). Participation in the study entailed vote casting and vote verifying with each of three voter verifiable methods: Helios, Prêt à Voter, and Scantegrity II (all orders were used; participants were randomly assigned to an order). After using each system, participants completed surveys while keeping the particular system in mind. At the end of the study, participants completed a final general survey, were thanked, and debriefed.

If a participant struggled using a system or asked the experimenter for help, they were told, "Please do whatever you think you should do to cast your vote [or when verifying: check on or verify your vote]." Participants were not assisted during the usability test because then the usability of the system would have no longer been evaluated. Instead, the efficacy of the assistance would have become the object of evaluation. To withhold help during a usability test was to maintain equality across tested systems. If different types of assistance were offered for each system depending on its unique set of usability problems, then the internal validity of the study would have been

compromised. It would no longer have been clear whether the observed effects were due to the system design or the type and/or amount of assistance offered. For these reasons, in this study, we controlled for this variable by not offering any aid to participants. This decision aligns with strong usability protocols, which rarely allow for user assistance during the completion of tasks (Rubin & Chisnell, 2008).

A “control system” was not required in this standard usability assessment. As is standard in human factors usability research, the data regarding each tested system was sufficient as a metric to judge the usability of the evaluated system.

The Helios voting system and election was set up and run through Helios’ website at [vote.heliosvoting.org](http://vote.heliosvoting.org) during the winter of 2013-2014. The tested version of PaV was developed by our research team. At the time this research study was conducted, the PaV system had not been developed to handle long ballots with numerous races like those found in the United States. Hence, a Rice version of PaV was developed based on published papers about PaV (e.g., Lundin & Ryan, 2008; Ryan et al., 2009; Ryan & Peacock, 2010; Ryan & Schneider, 2006), the PaV website (Prêt à Voter, n.d.), and in consultation with Peter Ryan. The tested variant of Scantegrity II was also put together by our team. The Scantegrity II systems used in the Tacoma Park elections were not available for usability testing. For this reason, a Rice version of Scantegrity II was developed based on printed materials used in the 2009 Takoma Park, Maryland election (Carback et al., 2010; [www.takomaparkmd.gov](http://www.takomaparkmd.gov)), published articles about the system (e.g., Chaum et al., 2008; Chaum et al., 2009; Carback et al., 2010; Sherman et al., 2010), and email correspondence with both Aleks Essex and Richard Carback, researchers who have direct experience with the implementation (R. Carback, personal communication, December 13, 2012; A. Essex, personal communication, December 14, 2012). It should be noted that this study included only the Scantegrity II instructions available to voters during the actual use of the voting system because systems cannot rely on users to have engaged with optional instructions that are unavailable when voting with the method. When aspects of the system that might have the potential to impact usability were not specified—such as the exact scanner used and its setup—human factors principles were followed.

The critical incident technique was used to analyze the data. This method was selected because it is widely associated with engineering studies of human error; is useful as a rapid trouble shooting technique in initial, global system analyses; and has significantly contributed to the reduction of human error (Kirwan & Ainsworth, 1992).

Critical incident analysis involves, first, identifying events that have a major impact on system objectives, second, developing a taxonomy in which to categorize the events, and third, analyzing these events in order to identify the root causes of the issues so that system changes can be suggested in order to mitigate the problems and reduce errors (sometimes fixing these types of problems will require major alterations to the system design, not just tweaks to the existing system). The focus for change is on the equipment and system, not the users, as poor system designs contribute to errors and usability problems (Shneiderman, 1987).

This critical incident method of task analysis has been previously used in areas like pilot error and aircraft operations to change safety procedures and the design of equipment (e.g., Green, 1990), software documentation iterative design in order to ensure that the final documents met users’ needs (Galdo et al., 1986), and medical errors in the contexts of reporting culture and safety performance to understand how hospitals differ (Itoh et al., 2010). Likewise, it can also be used to identify and categorize usability problems that led to voting system failures.

In the context of voting with e2e methods in this study, critical incidents are defined as a voter not being able to vote cast (goal 1) and/or vote verify (goal 2), two of the main system objectives. A user error is defined as the actions, whether intended or not, that prevented the voter from casting a ballot or from checking that their ballot had been cast. The system design deficiencies are the system features that likely contributed to the error occurrence. This paper does *not* attempt to address every possible usability struggle, slip, or mistake that might have occurred, or could occur, while vote casting and vote verifying; this list has the potential to be endless (Reason, 1990). For this reason, this paper focuses on these user errors that were deemed to be most serious.

To collect the data, the experimenter directly observed each participant use the three e2e systems. If a participant could not cast a ballot or check on their vote, the experimenter took notes about how the participant deviated from the procedures that were required to complete each task (see Appendices A, B, and C for the typical steps that users go through to vote cast and vote verify with each system). When the data were analyzed, the errors were grouped by type. Then based on the resulting taxonomy and the notes on observed participant behaviors, the system design deficiencies that might have contributed to the behavioral errors were identified. System design deficiencies were identified by referring to resources that present best practices of interface design (e.g., Johnson, 2010; Sanders & McCormick, 1993; Shneiderman, 1987; and Tidwell, 2011).

The methods outlined in this section help to ensure external validity, or the generalizability of the study's findings to voting systems used in a real election. All systems have been tested in a well-controlled, laboratory environment to ensure that they were utilized under identical conditions. This type of environment of course differs from crowded polling stations filled with voters and election officials on election day. Nonetheless, if usability problems are found in a lab-based usability test, the problems should be fixed so that real voters also do not encounter them.

## RESULTS

### Helios

#### Vote Casting

As can be seen in Table 1, 14 (38%) participants were not able to cast a vote using Helios. The most frequent reason that voters did not cast a ballot is that [A] after encrypting their ballot, they did not log in with their email account to both verify their eligibility to vote in an election and have their most recent cast ballot counted in the final election tally. Twelve participants (32%) made this error. There are many design deficiencies that possibly contributed to this particular error, each of which are discussed in the following paragraphs.

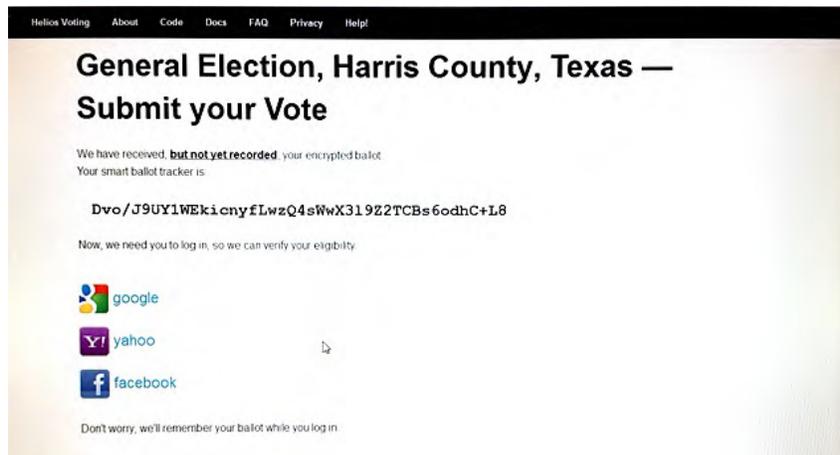
[A1] *Voting system design is not aligned with users' prior voting mental models.* Figure 2 shows a screenshot of the user interface that the thirteen voters viewed when they announced to the experimenter that they had finished casting their ballot. At this point they had already specified their choices and reviewed them before being brought to the log in authentication screen. According to their prior voting mental model (likely based on typical voting procedures) they would be done voting—since authentication would usually occur before starting the voting process. Yet, with Helios, authentication had not yet taken place and needed to occur at this point in the vote-casting process. This mismatch between Helios' voting process and participants' mental models for typical voting procedures misguided subjects to conclude that they were finished, when in fact there was still more to do in order to vote cast (for more information about user mental models in interaction design, see Stammers and Norcio, 1993). For this reason it is essential for voters' mental models to align with the voting method that they are using, even if the e2e method is novel.

[A2] *Users are expected to read and process all text displayed on the interface.* Another potential design deficiency that could account for participants announcing prematurely that they were finished voting at the screen shown in Figure 2 is that this interface does not account for users who do not want to read large amounts of text. In this study, it appeared that participants did not read and/or cognitively process every piece of information shown on the webpage. Instead they likely saw the words, "We have received..." and concluded that they were done voting. At this point they thought their goal was fulfilled and did not put in further resources to make sure that it was actually achieved. The design of the interface did not support the user who scans pages, picking out individual words, instead of reading every word. Since a previous research study found that only 16% of their participants read web pages word by word (Nielsen, 1997), the text on this Helios page should be scannable, simple, and clearly worded. In addition, the most critical text should be the most prominent, and only the most essential information related directly to the particular step should be displayed.

**Table 1.** Errors and Likely Contributing Design Deficiencies that Led to Helios Vote-Casting and Vote-Verification Failures

Number of Participants (Percentage of Participants)	Error	Contributing Design Deficiencies
<i>Vote Casting</i>		
12 (32%)	Encrypted the ballot but did not log in with e-mail [A]	<ul style="list-style-type: none"> <li>• Voting system design is not aligned with users' prior voting mental models [A1]</li> <li>• Users are expected to read and process all text displayed on the interface [A2]</li> <li>• Log in information is not entered directly on page [A3]</li> <li>• Too many steps are required to cast a vote [A4]</li> <li>• Incorrect organization of steps [A5]</li> </ul>
1 (3%)	Did not press the cast button after a successful log-in [B]	<ul style="list-style-type: none"> <li>• Users are expected to read and process all text displayed on the interface [B1]</li> <li>• Too many steps are required to cast a vote [B2]</li> </ul>
1 (3%)	Refused to cast a ballot due to concerns about errors [C]	<ul style="list-style-type: none"> <li>• Poor system usability degraded voter confidence [C1]</li> </ul>
<i>Vote Verification</i>		
14 (38%)	Did not cast a ballot [D]	<ul style="list-style-type: none"> <li>• See above vote-casting deficiencies</li> </ul>
6 (16%)	Performed the wrong action [E]	<ul style="list-style-type: none"> <li>• No instructions are provided for vote-verification process [E1]</li> </ul>

**Notes:** This table includes observed user errors that led to a vote casting and/or vote verification failures. It does not include every critical incident observed throughout the study.



**Figure 2.** Helios screen that voters see after encrypting their ballot but before logging in so that they can later cast their ballot

**[A3]** *Log in information is not entered directly on page.* Another possible problem with the tested Helios system design is that when it is time to authenticate, the voters do not enter their email address and password directly on the page shown in Figure 2. Instead, voters are supposed to click the e-mail client icon, which takes them through another series of pages, and then returns them back to a page that allows them to finally cast their ballot. Due to this design, voters got lost in the voting process. Some participants did not have the appropriate knowledge required to successfully navigate the site (Spool, 1999, p.15). These subjects could not figure out how to log in because it was not apparent to them that they should click on one of the Google, Yahoo, or Facebook icons to proceed. Of those participants who managed to figure out this step, two participants navigated away from the Helios voting system and could never find their completed ballot again—ultimately giving up before casting their ballot. In this scenario, the site structure did not align with their behavior and consequent expectations (Spool, 1999). In addition they had to shift their attention away from their primary goal of vote casting to a new task of figuring out the voting interface. This shift can cause users to lose track of what they were doing, or where they were doing it (Johnson, 2010, p. 97). If, instead, participants could log in with their email credentials directly on the page shown in Figure 2 so that the process was more linear and streamlined, then there would be less of a chance that they would fail at logging in and/or become permanently lost in the process.

**[A4, A5]** *Too many steps are required to cast a vote and incorrect organization of steps.* Between the time the voter completes his or her ballot and then casts it, the person views at least five different screens (more if not already logged in). The developers of Helios were likely aiming to have the interface guide the user step by step in the prescribed order, which is generally considered to be a good interaction design. However, in the tested version of Helios, the number of steps makes the voting process feel drawn out. To make matters worse, the steps, accompanying information, and actions required on behalf of the system user are not chunked correctly. Particularly, the system does not strike a balance between the division of steps (and sub-steps) and the number of them, making the process so tedious that users look for the first instance that they might be done (i.e., seeing that the ballot was received), and then leave prematurely before the cast-ballot button ever gets pressed. To prevent fleeing voters (Everett, 2007), it would be best to keep both the distance short and organization of steps simple, yet meaningful, between the ballot completion page and the vote-casting page in order to make the stepwise process the most efficient, and effective.

**[B, B1, B2]** *Users are expected to read and process all text displayed on the interface and too many steps are required to cast a vote.* One participant (3%) failed to cast a ballot because they completed the log in process, but on the very last page did not press the final vote-casting button. One reason this error likely occurred was because on the final vote-casting screen there was too much text presented, some of which was confusing and not directly related to the task of pressing the button that casts the voter's ballot (refer to [A1] for more information on why this is a problem). The second reason that the error might have occurred has to do with there being too many discrete, cumbersome tasks required to vote—from ballot completion and review to multi-step encryption, logging in, and finally vote submission. Again, if some of the early steps were to be made invisible to the user (i.e., the system still includes the security mechanisms but does not require the voters to be aware of them and serve as an active participant in the process), then the number of cumbersome tasks and the associated errors might be reduced (for further information about this design deficiency, see [A3]).

**[C, C1]** *Poor system usability degraded voter confidence.* One participant (3%) refused to cast a ballot because they were concerned about errors. Specifically the user stated that they were worried both that the system could be making mistakes and that they were not using the system correctly. For these reasons, the participant wanted to talk to an election official to get help. Regarding their concerns about system errors, research should be conducted in the future to determine how to strategically increase a voter's confidence and trust in the system's reliability and accuracy. As for their concerns about whether or not they were performing the correct actions required to cast a vote, addressing system usability issues would improve the participant's confidence that they were using the system "correctly" and that the intended selections would be cast and counted accurately.

Vote Verification

In the vote-verification process, 14 (38%) participants were not able to check on their vote because they [D] did not cast a ballot (refer to Table 1). In other words, after subjects thought they voted, they could not do anything else with the system because there was not a cast ballot to check on. To correct this problem, the system needs to be redesigned so that every voter can both easily cast a ballot and recognize if they have or have not done so.

Of the 23 participants who were able to cast a ballot, six (26%) were not able to verify due to the following possible design deficiencies:

[E, E1] *No instructions are provided for the vote-verification process.* Another reason that voters were not able to check on their ballots with Helios is that they performed the wrong actions. Six participants encountered this type of error because vote-checking instructions were not provided to voters.

In this study, before the participants started to use any of the tested systems, the experimenter directed participants to first vote, then verify their vote. For this reason, participants knew vote verification was possible, somehow—an advantage real voters might not have. Since there were no vote-verification instructions and cues, there were many actions participants performed to achieve what they thought was a verification, when in fact it was not: participants viewed the review screen (see Figure 3) and then “googled” their smart ballot tracker (and found nothing), and/or they printed the smart ballot tracker from the review screen. Some participants had no idea what to do, and so they did not take any further action. To reduce the frequency of these mistakes, the system should provide voters with simple, concise vote-verification instructions both on the final vote-cast confirmation screen and through the vote-confirmation email. It would also be beneficial to provide simple, inline instructions throughout the verification process.

Of the 16 (43% of total participants) who were able to complete some form of vote verification, only half of these subjects completed verification procedures as outlined in the published papers describing Helios (e.g., Adida, 2008; Adida et al., 2009). In other words, *only 8 (22%) of all the voters who participated in this study completed a full verification* in which they checked their vote (associated with the smart ballot tracker that they recorded) on the online bulletin board. Voters who only partially verified did not do this—and instead relied on methods like viewing a vote confirmation email—likely because instructions on the vote-verification process were not provided to the users. Even though this deficiency did not lead to a total failure, the voters did not use the system as the designers intended in order to make sure that the system is working properly and that there is not any evidence of tampering.

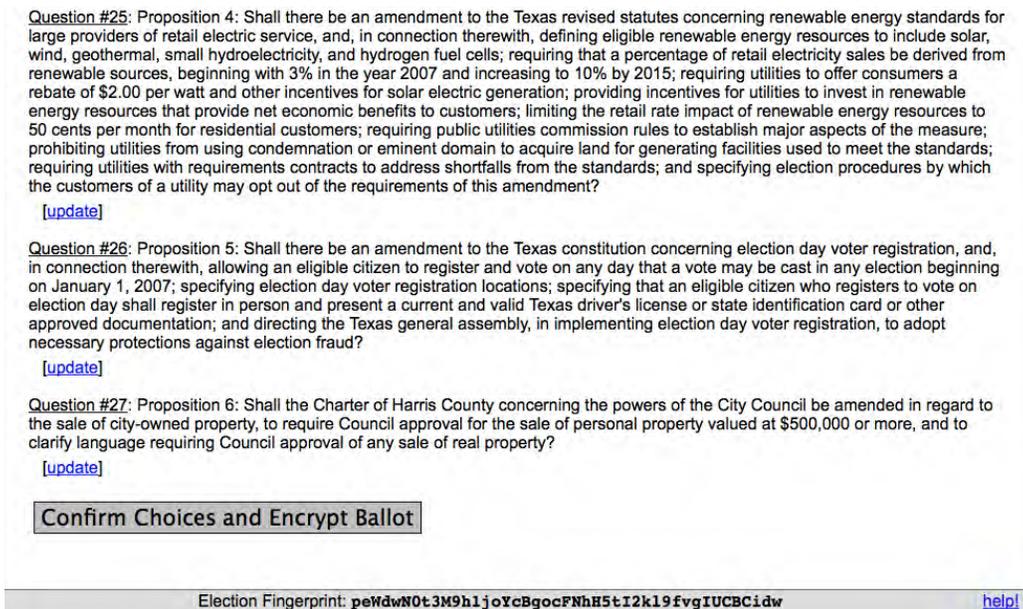


Figure 3. Helios vote selection review screen

**Prêt à Voter**

Vote Casting

As can be seen in Table 2, when voting with the PaV voting system, 8 (22%) of participants [F] did not scan in their ballot before placing it in the ballot box. It should be pointed out that the simple, single-use scanner (see Figure 4) was easily accessible. It automatically fed and scanned the ballot cards when they were placed into the feeder slot. In addition, the ballot box was placed off to the side of the vote-casting station. There are several design deficiencies that might have contributed to this scanning error resulting in a failure to vote cast.

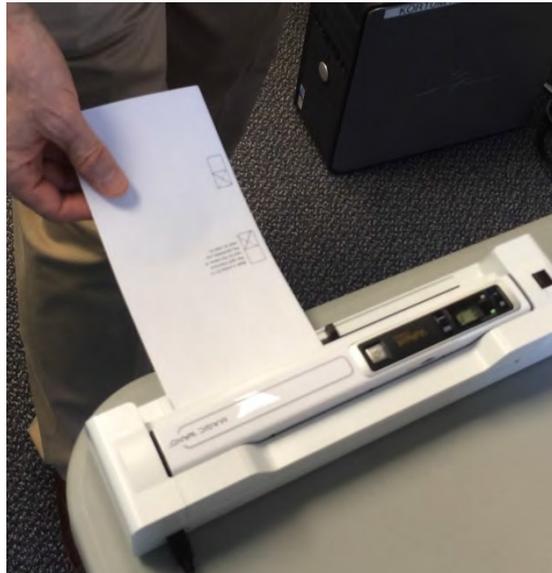
[F1, K1] *No physical mechanisms are on the ballot box preventing an unscanned ballot from being placed in it.* The ballot box used in this study was not designed to keep unscanned ballot cards (i.e., uncast ballots) from being placed in it. A design solution might be to place the scanner behind the ballot box opening. A voter would simply drop in their ballot into the ballot box for it to be both scanned and stored (for an example see Belton, Kortum, & Acemyan, in press).

**Table 2.** Errors and Likely Contributing Design Deficiencies that Led to PaV Vote-Casting and Vote-Verification Failures

Number of Participants (Percentage of Participants)	Error	Contributing Design Deficiencies
<i>Vote Casting</i>		
8 (22%)	Did not scan-in ballot; placed it directly in the locked ballot box [F]	<ul style="list-style-type: none"> <li>No physical mechanisms are on ballot box preventing an unscanned ballot from being placed in it [F1]</li> <li>System does not support prior voting mental model [F2]</li> <li>No inline instructions [F3]</li> </ul>
5 (14%)	Could not operate scanner [G]	<ul style="list-style-type: none"> <li>Scanner is not explicitly identified [G1]</li> <li>Operating instructions are not available [G2]</li> <li>Design of scanner [G3]</li> </ul>
1 (3%)	Refused to detach the candidates list and then proceed with the voting process due to being confused [H]	<ul style="list-style-type: none"> <li>Numerous novel steps are required to cast a ballot [H1]</li> <li>No inline instructions [H2]</li> </ul>
<i>Vote Verification</i>		
14 (38%)	Did not cast a ballot [I]	<ul style="list-style-type: none"> <li>See above deficiencies</li> </ul>
1 (3%)	Could not navigate to the verification website [J]	<ul style="list-style-type: none"> <li>Requires use of the internet [J1]</li> <li>Website address is too complex [J2]</li> </ul>

**Note:** This table includes observed user errors that led to a vote casting and/or vote verification failures. It does not include every critical incident observed throughout the study.

[F2, K2] *System does not support prior voting mental models.* The tested version of PaV does not support participants’ general how-to-vote procedure for voting with a paper ballot (Acemyan, Kortum, Byrne, & Wallach, in press; Norman, 1983). Specifically, in elections that use a paper voting method, voters typically would complete a paper ballot and then place it into a locked ballot box to cast it. Therefore, in this study when participants used PaV to vote—which also requires the use of a paper ballot—they likely referenced their general model for voting with this type of ballot and then dropped it into the box instead of first scanning it. Skipping this critical step resulted in a failure to vote. To support these prior voting models formed through experience and general knowledge (Kaplan & Kaplan, 1983), the PaV voting system should mirror the typical how-to-vote-with-a-paper-ballot procedure as closely as possible.



**Figure 4.** Photograph of the scanner used in this research project for both PaV and Scantegrity II systems

**[F3, H2, K3, O3]** *No inline instructions.* PaV does not provide the voters with instructions when they are needed at key points in the voting process. Instead voters are given lengthy instructions on a single page before they start the voting process (see Figure 5). This has the potential to pose a large cognitive load on the user (Johnson, 2010) who is voting with a novel system requiring them to do unusual things in the context of voting, like tearing their ballot in half. A system improvement would be to give the voter clear, concise instructions at each step in the vote-casting process—allowing voters to focus on their task while working within their short-term memory capacity (Shneiderman, 1987). Doing so might also prevent voters from accidentally shredding their instruction sheet and then having no idea how to proceed, which happened to one participant.

Five more participants did not cast their ballots because they **[G]** could not identify and/or operate the scanner. Three potential design deficiencies have been identified.

**[G1, L1]** *Scanner was not explicitly identified.* Some voters had never used, or seen, a scanner before. This lack of previous exposure made it difficult to identify the single-function scanner (see Figure 4) and/or differentiate it from a single-function printer. Labeling the equipment with both the word “scanner” and a visual icon might allow some people to at least identify the piece of equipment. Of those that could identify the scanner, some participants expressed that they were afraid of it because they have never scanned a document before. Even though a scanner is a relatively common piece of equipment, it should not be assumed that every voter knows how to use even a simple one. Thus there should be support in identifying equipment and operating it with confidence.

**[G2, L2]** *Operating instructions for scanner are not available.* Inline instructions for operating the scanner were not provided to participants. Especially for participants who never used a scanner before, it would have been helpful to tell them how to operate the scanner, even at a global level: e.g., “Insert ballot cards in feeder slot for automatic scanning. No other action is required.”

**[G3, L3]** *Design of scanner is not optimal.* In order for voters to not have to learn how to identify and operate a scanner to vote cast, the scanner could be redesigned. If the scanner is placed inside the ballot box so that voters only have to drop in the ballot (a step that they already familiar with) and the scanner—*hidden from view*—completes the process, then the voters would not have to worry about “scanning.” (For a specific design suggestion, see Belton et al., in press.)

**General Election Ballot  
Harris County, Texas  
November 8, 2016**

**INSTRUCTIONS TO VOTERS**

**1. Mark a cross (x) in the right hand box next to the name of the candidate you wish to vote for.** For an example, see the completed sample ballot below. Use only the marking device provided or a number 2 pencil. Please note that this ballot has multiple cards. If you make a mistake, don't hesitate to ask for a new ballot. If you erase or make other marks, your vote may not count.

Cathy	<input type="checkbox"/>	Mark a cross (X) in the right hand box next to the name of the candidate you wish to vote for.	Vote Verification Code
Eliot	<input type="checkbox"/>		
Geena	<input checked="" type="checkbox"/>		
Daniel	<input type="checkbox"/>		
Ben	<input type="checkbox"/>		
Ivy	<input type="checkbox"/>		
Hannah	<input type="checkbox"/>		
Frederick	<input type="checkbox"/>		
Ali	<input type="checkbox"/>		

**2. After marking all of your selections, detach the candidates lists (left side of cards).**

**3. Shred the candidates lists.**

**4. Feed your voting slips into the scanner.**

**5. Take your receipts.** Receipts can be used to confirm that you voted by visiting [votingstudy.rice.edu](http://votingstudy.rice.edu).

**Figure 5.** PaV instruction card, which was first page of ballot

**[H, H1]** Numerous novel steps are required to cast a ballot. One participant (3%) refused to detach their candidates list and then proceed with the voting process because they were confused. They did not understand why they would detach their selections from the people that they voted for and then shred them so there was not any record. Requiring the voter to perform so many atypical, unexpected steps seemed to erode their confidence. It was also clear that they did not understand why they were doing what they were doing to make their votes private (with the PaV system, a randomly ordered list of candidates is separated from the voter's selections and then shredded so that another person could not look at the ballot cards to figure out for whom they voted). To overcome this design deficiency in the future, the system should align as close as possible with typical how-to-vote procedures (Acemyan et al., in press; Nielsen & Molich, 1990), and if a voter must perform a novel procedure, it should be easy to accomplish. It would also help the user if they understood why they were performing the action.

Vote Verification

Fourteen (38%) participants did not verify that their vote was cast with PaV because they **[I]** did not cast a ballot. Addressing system design deficiencies would likely reduce this rate of failure. Of the participants who were able to cast a ballot (23 in total), one (4%) was not able to verify their vote. The one participant who was not able to check on their vote failed because they **[J]** could not navigate to the vote-verification website. This error can be accounted for by two likely design deficiencies:

[J1, J2] *Voter must use the internet and website address is too complex.* PaV’s vote-verification system required the participants to use the internet. However, one participant could not use it. In particular, they did not know in which field they should enter the website address; they kept entering it into the search engine’s search field, a common error among novice internet users (Sisson, 2014). Repeatedly, this participant also did not correctly type the address, “votingresearch.rice.edu”—highlighting that the verification sites should have the simplest and shortest address possible.

**Scantegrity II**

Vote Casting

As can be seen in Table 3, 16 (49%) participants were not able to cast their ballot with the Rice implementation of Scantegrity II.

**Table 3.** Errors and Likely Contributing Design Deficiencies that Led to Scantegrity II Vote-Casting and Vote-Verification Failures

Number of Participants (Percentage of Participants)	Error	Contributing Design Deficiencies
<b>Vote Casting</b>		
16 (43%)	Did not scan-in the ballot; placed it directly in the locked ballot box [K]	<ul style="list-style-type: none"> <li>No physical mechanisms are on the ballot box that prevented an unscanned ballot from being placed in it [K1]</li> <li>System does not support prior voting mental models [K2]</li> <li>No inline instructions [K3]</li> </ul>
1 (3%)	Could not operate scanner [L]	<ul style="list-style-type: none"> <li>Scanner is not explicitly identified [L1]</li> <li>Operating instructions are not available [L2]</li> <li>Design of scanner [L3]</li> </ul>
1 (3%)	No data available [M]	
<b>Vote Verification</b>		
18 (49%)	Did not cast a ballot [N]	<ul style="list-style-type: none"> <li>See above deficiencies [N1]</li> </ul>
3 (8%)	Did not record ballot ID [O]	<ul style="list-style-type: none"> <li>No inline instructions [O1]</li> <li>Ballot ID is not located in a prominent location on the ballot [O2]</li> <li>Ballot ID is too long and complex, which deterred voter from making the effort to record it [O3]</li> </ul>
1 (3%)	Could not navigate to the website [P]	<ul style="list-style-type: none"> <li>Voter did not know how to use the internet [P1]</li> <li>Website address too complex [P2]</li> </ul>

**Notes:** This table includes all observed critical incidents that led to vote casting and vote verification failures. The table does not include every critical incident observed throughout the entire study.

Four of the 37 participants (11%) did not use the special marking device as instructed, and instead used pen or pencil to complete their ballots. One of the Scantegrity II developers pointed out that completing a ballot with pen or pencil “does not necessarily invalidate the ballot” (A. Sherman, personal communication, August 20, 2014). However, three of these four participants never scanned in their ballot, meaning they never cast it. Therefore, the vote-casting rate would fall by 1 participant (3%) if failing to use the marking device invalidates the ballot. In this report of the data, a ballot was considered valid regardless of the marking device used—even though we explicitly asked participants to take all steps necessary to check on their votes after the election, which would include using the decoder pen to mark each selection.

[K, L] The same ballot box and scanner used in the PaV mock election were used with this system. For this reason, the other possible design deficiencies for vote casting are identical to those provided in the PaV section above.

#### Vote Verification

Eighteen (49%) participants were not able to verify their vote because they [N] did not cast a ballot. Of those that were able to cast their ballots, four (21%) were not able to verify successfully.

Three participants were not able to verify that their ballot was cast because they [O] did not record the ballot ID, which is required to log in to the verification website (see Figure 6 for an example of this error). The following are the associated design deficiencies:

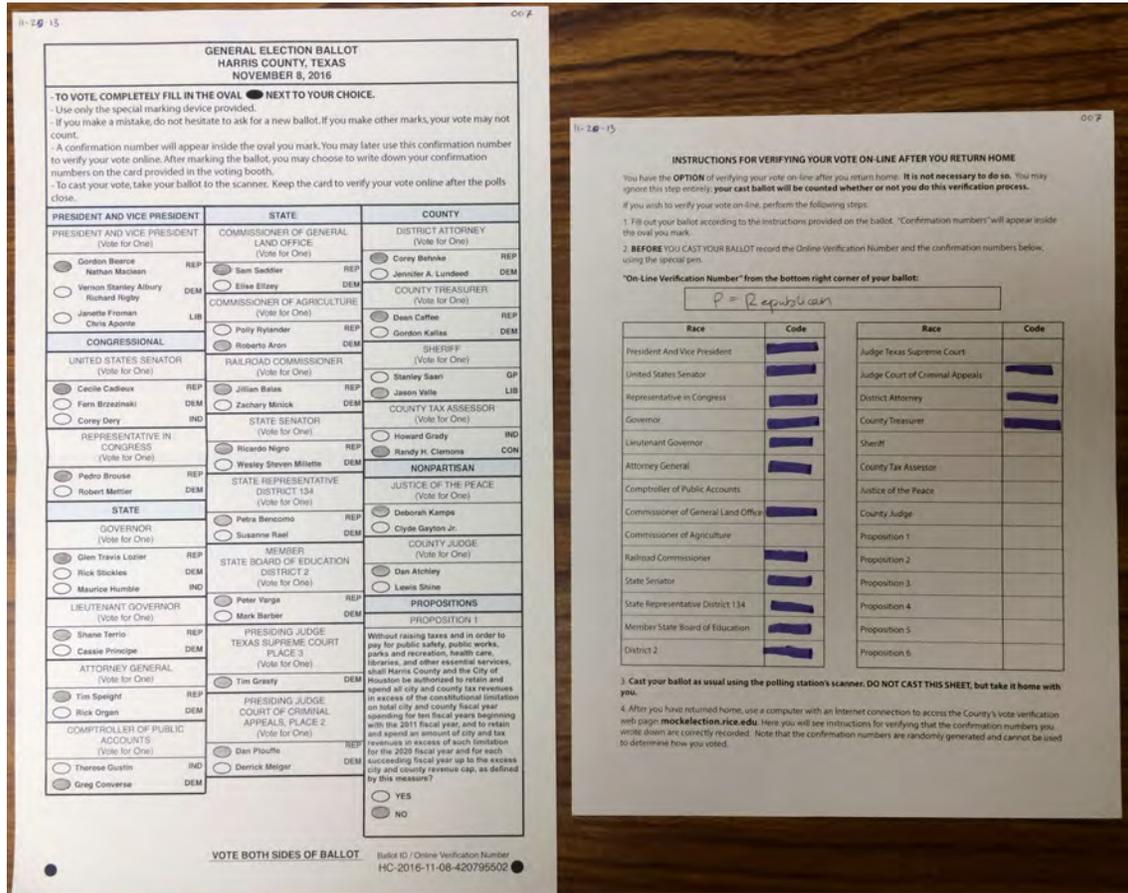
[O1] *No inline instructions.* Like PaV, the tested version of Scantegrity II did not give voters simple, instructions at the time that they were needed during the vote-casting and vote-verification processes. Instead, long blocks of instructions were found at the tops of the ballot and vote confirmation sheets. For the full explanation of this design deficiency, refer to the PaV section.

[O2] *Ballot ID is not located in a prominent location on the ballot.* Per the City of Takoma Park, Maryland, Municipal Election, November 8, 2011 sample ballot, the ballot ID was placed on the front, lower right corner of this study’s ballot. A similar font and font size was also used. It is possible that placing the ballot ID in a location that was not prominent, and in the voters’ visual periphery (Johnson, 2010, p.65), might have contributed to voters not being able to easily detect it—especially since there was a large quantity of information displayed (Tullis, Tranquada, & Siegel, 2011) on the ballot (i.e., election title heading, voting instructions, races with candidates, revealed codes, and the unique ballot identifier).

[O3] *Ballot ID is too long and complex.* The ballot identification on every participant’s ballot was “HC-2016-11-08-420795502.” HC stood for Harris County, 2016-11-08 was the hypothetical date for the mock election, and 420795502 was the unique number associated with the particular ballot (this 8-digit number was based on a sample of e2e ballot identification codes, many of which were even longer). This study’s ballot ID was not the same as the ID shown on the illustrative Scantegrity II ballots (i.e., #0001) featured in published and website materials (e.g., [https://www.usenix.org/legacy/events/evt08/tech/full\\_papers/chaum/chaum\\_html/index.html](https://www.usenix.org/legacy/events/evt08/tech/full_papers/chaum/chaum_html/index.html)). The published Scantegrity II illustrative ballot ID was too simple and not representative of a ballot ID used in an actual election.

Because the ID used in this study was so long and complex, it might have deterred participants from putting in the effort and time to record it. That is the perceived cost was too high for the return gain of being able to verify (Johnson, 2010), a goal secondary to vote casting. Or perhaps participants did not recognize the benefit of being able to verify, so they did not want to transcribe the ID. Based on the observed failures, if a ballot uses a code that voters must transcribe, those codes should be short, simple, and chunked (Miller, 1956) so that each group of numbers/letters is composed of 3-4 units—this will help reduce both voters’ cognitive load and the opportunity for them to make mistakes.

One participant, [P] could not use the internet to get to the website, the same error and [P1, P2] contributing design deficiencies encountered by a PaV voter.



**Figure 6.** One participant’s Scantegrity II ballot is shown left, and their vote confirmation sheet is shown right. For two reasons, this participant was not able to check on their votes. First, they did not use the decoder pen when completing their ballot, so the codes associated with each selection were not revealed or recorded. Second, they failed to record their ballot ID (located in the bottom right corner of their ballot) on their confirmation sheet (the top box is the area of the form reserved for the ballot ID). On account of these two errors, they did not record any information to track their ballot online.

Of the 15 (40%) of study participants who were able to complete some form of vote verification, only three (21%) of these completed verification procedures as outlined in the published papers describing Scantegrity II (e.g., Carback et al., 2010; Chaum et al., 2008). All told, *only 8% of all the voters who participated in the mock election completed a full verification* in which they correctly recorded their ballot ID, correctly wrote down all 27 codes, and looked at the vote-verification webpage associated with their ballot. Since there is no data to confirm that participants actually compared all 27 transcribed codes to those displayed on the site, the full verification rate has the potential to be even lower than 8%. Voters likely only partially verified because too much of a mental workload and physical burden was placed on them. After completing a ballot, participants had to transcribe by hand a long, complex ballot ID and small codes revealed inside each selected bubble. Then the subjects had to navigate to the vote-verification website, and only if they wrote down all of the information correctly could they type in the case sensitive ballot ID and compare all the codes one by one.

Another major problem with the Scantegrity II verification system was that there were too many opportunities for voters to make errors while transcribing and then comparing the verification information. Only 62% of all participants wrote down the ballot ID correctly, and just 8% recorded all 27 codes accurately. The ballot ID and each code that had to be written down on the confirmation sheet is an opportunity for the voter to make a mistake. Then there are even more opportunities for errors when the voters type the ballot ID into the verification system and

compare their recorded information with the information the system recorded. What if they notice one of the codes that they recorded does not match the corresponding code the system displays on the vote-verification screen? At this point the voter can contest the election. This brings up even more issues—such as the lack of concrete, specified procedures from the voters perspective to handle this type of event (e.g., who do they contact, how do they contact them, and what information do they need to provide?). This is applicable also to Helios and PaV. In addition, another concern about the Scantegrity II system is the possibility of potential deliberate attacks on the system. For example, imagine a voter claims the systems' online codes do not match their written ones, when in fact the malicious voter intentionally wrote-down the wrong codes so that they could report that the system was not operating as expected. This type of event could end up costing a precinct resources.

For all of these reasons discussed in this section, Scantegrity II had one of the weakest vote-verification systems tested in this study. Even though the identified system design deficiencies did not lead to a total failure, it is still highly problematic because the voters did not use the system as the designers intended in order to check that their votes were recorded accurately by the system.

## DISCUSSION

This study, which focused on all of the observed errors resulting in a total failure to vote cast and vote verify, found these errors were caused by only a few contributing design deficiencies. This is positive because a small number of fixes to the system designs could likely solve the majority of the issues that caused voters to fail at vote casting. As for the verification methods, if the voters need to complete full vote verifications for security reasons, then the Helios and Scantegrity II methods should be heavily reconsidered to reduce the amount of physical and cognitive work placed on voters.

### Some Cautionary Notes

*Errors and contributing design deficiencies listed for each system are not punch lists.*

While every usability issue observed while conducting the study was not identified in this paper, the data do provide direction for improving the systems to increase the likelihood that voters will fulfill the objectives of vote casting and vote verifying. What this paper does not provide is a punch list of elements to fix. First, even if every possible system design deficiency identified in this paper is addressed, the systems still might not be highly usable. The identified usability issues might have masked problems that have yet to be identified. Hence, once the outlined problems are fixed, new problems might be found that can only then be recognized. Second, there is the possibility that the design deficiencies presented in this paper may not be the root cause of the failures observed. Even though the deficiencies are based on basic human factors and human-computer interaction principles, until additional data is collected, there is no way to know for certain that they account for the observed voter behaviors. Third, a voting research team might think that they have the perfect solution for a problem, when in fact the solution does not fully resolve the problem, and may even causes new issues to arise. For instance, Scantegrity II, as deployed in the 2011 Takoma Park City Municipal Election, had the scanner attached to the ballot box (scantegrity.org). This design (which differs from published papers about the Scantegrity II system, e.g., Chaum et al., 2008; Carback et al., 2009; Chaum et al., 2009; Sherman et al., 2009) prevented the problem of failing to scan the ballot before placing it into the ballot box. What is unknown is if the voters had / would have trouble using the new scanner setup, or voters who have never scanned documents before were / would be still afraid to use it. In that case the design solution would still be insufficient and work would need to be done to improve system usability. For these reasons, the errors and contributing design deficiencies outlined here should be considered when designing any voting system so that the same mistakes are not repeated. With respect to the three systems tested in this paper, further testing must take place after making design modifications to assess if the system is more usable and all issues have been addressed.

*Helping voters, instructions, and training are not “fixes.”*

It might be argued that if these systems had been used in an actual election, some of the behavioral errors and consequent system failures could have been avoided if the voters had asked an election official for assistance. This may be true. Assistance, offered at the right time, can prevent usability errors or rectify those that have occurred (similarly, assisting participants in a usability study greatly impacts test results; Rubin and Chisnell, 2008). While there are always officials present at polling sites, they should not be the first line of defense, and voters should not have to rely on them to successfully use the systems. There are high costs associated with voters asking someone to help them: voters might become frustrated and give up voting before asking for help, they might avoid voting in the future, their votes could be revealed to another person, officials might have to be trained to know how to address

every possible usability issue, election officials might not be able to assist in a timely manner in busy polling stations, the availability of help is not uniform across jurisdictions as some locations do not have a plentiful supply of well-trained poll workers, racial bias in poll workers might result in race-based discrimination when helping voters complete and cast ballots (Page & Pitts, 2009)—meaning there is the potential for some races to be helped more effectively than other races, and there is the potential for voting times to increase, making lines at some polling stations even longer.

It is also not wise to rely heavily on instructions and/or training in order for people to be able to use a system with usability problems, even if it is new and different—as is the case with all three of the e2e systems studied in this research. Instead it would be best to redesign the systems in order to make them easier to use and minimally reliant on instructions.

When Scantegrity II was used in a real election, “instructions along waiting lines, instructional video[s]... instructions in marking booth[s], instructions in newspapers and TV prior to election, [and] verbal instructions as people entered” were relied upon in addition to the instructions printed on the ballot and confirmation sheet (A. Sherman, personal communication, August 20, 2014). The researchers were trying to help the voters understand what to do and convey that they were able to verify their votes afterwards if they completed extra steps while voting. Despite these efforts, some voters ignored all instructions and ended up “not realiz[ing] that they could verify their vote and that they had to make a receipt to do so” (A. Sherman). It is also possible that not every voter who participated in the election received the training or encountered all instructions sets. So rather than trying to change user behavior, instead change the system design (Ross, 2010) to make sure the system is usable in the first place. If a system is highly usable, then there is no need for training—saving everyone time and money—and figuring out how to use the system becomes more effective (Rohn, 2005). Further, training and instructions do not fix inefficiencies of use (i.e., every user still has to go through inefficient steps) or improve user satisfaction (i.e., being able to accomplish a goal does not necessarily mean it was a good experience; Ross, 2010), two other aspects of usability not expressly addressed in this paper’s formative analysis.

In summary, instructions and videos do not fix a system’s usability problems and should not be used as a first line of defense, especially in a walk-up-and-use system. The systems must be designed so that any user can walk up and effortlessly use them, without any assistance, which is especially important in voting. Simple, inline instructions written in plain language might even help to achieve this goal (for more information about writing instructions, refer to Dana Chisnell’s *Writing Instructions Voters Understand*, 2013).

#### *Test, iterate, and retest.*

Summative testing is a great tool to understand the current state of e2e system usability. Formative studies are useful for identifying particular designs that need improved upon or avoided altogether moving forward. It is expected that using both of these methods early in the development process would reveal many usability deficiencies. It is unrealistic to expect a perfectly secure and usable system in a single pass of development because addressing both system security and usability at the same time is challenging, yet possible (Goodman et al., 2012). This is why system developers need to continually refine their voting systems and test them with real users. This process will then arm them with empirical data—versus anecdotes and personal opinions—to support a system design, modification, or equipment setup as being an usability improvement (or detriment) for voters.

#### **Future Research**

Besides conducting summative and formative usability studies on the next generation of e2e voting systems, future research must address the usability of e2e voting systems for those with reading, cognitive, and physical disabilities. For example, after observing this study’s participants, we believe that Helios would be the most usable for voters with several classes of disabilities such as movement and vision. Helios is implemented in an electronic format, meaning existing software and hardware that is used by the disabled to interact with computers and DREs can be paired with the voting system. Non-sighted voters could use screen readers, and voters with movement impairments could use equipment like jelly switches and sip-and-puff devices to help them vote.

In contrast, PaV and Scantegrity II would likely pose many extremely difficult accessibility problems. For example, how would non-sighted voters complete their paper ballots? Even if a device was available to read the PaV ballot to them, these voters would have difficulty knowing which pieces of paper to shred (i.e., the candidates lists) and which pieces to eventually scan (i.e., the completed ballot cards). The receipt issued to the non-sighted voters would

likely be meaningless because visual information is displayed on a ballot card. It is unclear if even a reading device would be able to help them interpret it. Similar challenges are associated with Scantegrity II, as non-sighted voters would not be able to read the revealed codes inside each selected bubble and would probably struggle to create their own receipt without assistance. While Audiotegrity has been proposed as a solution for unsighted voters to complete their Scantegrity ballot and print-off a confirmation card that lists the ballot ID and confirmation numbers for each of their selections (Kaczmarek et al., 2013), testing would still need to be conducted to assess the usability of the system, especially since Audiotegrity does not solve problems like how an unsighted voter would check that their printed ballot is correctly marked and that the confirmation card lists the correct confirmation numbers.

Voters with other physical disabilities, like movement impairments, will also probably have trouble with some of the tasks required to vote. For instance, consider the process of voting with PaV. It requires a voter to tear the ballot in half, shred the candidates list, and feed the remaining ballot cards into a scanner. In this study, a participant was observed who could not use the right half of their body. They resorted to using their teeth to complete these tasks.

Another participant in this study had problems standing and moved slowly. While vote casting with Scantegrity II, they filled in the bubble for their selection very slowly, shuffled back and forth between the ballot and vote confirmation sheets, switched between writing devices (a decoder pen was used on the ballot and a pen was used to record the codes on the vote confirmation sheet), and carefully wrote down the code that was revealed before going onto the next race. Since the task was so labor and time intensive for the participant, they kept sitting down to rest. This same participant seemed to more easily use the Helios and PaV systems since they only required them to interact with a single interface while completing their ballot. For reasons like these, research must be conducted to identify usability issues that the disabled will face and determine if and how the issues could be solved.

## CONCLUSION

This paper is not intended to be a directed criticism of any of the tested e2e systems that we selected. Rather, the intent of the paper was to demonstrate that deviations from the standard paper/ballot box voting method can lead to failures in the ability of the user to successfully complete their desired task. Going forward, we need to be certain to understand any changes that are made from the voters' expected voting model and ensure that these deviations have minor repercussions for user success. Accordingly, conducting controlled, high-quality usability evaluations of these new systems is imperative.

In summary, the complex security mechanisms found in e2e voting systems likely do not comprise an insurmountable barrier to the development of usable designs. Even though it is difficult to create error-resistant systems for diverse populations, it is possible. By addressing the design deficiencies and implementing solutions using industry-standard methods like iterative design, secure e2e systems can become usable enough to be deployed in large-scale, actual elections.

## ACKNOWLEDGEMENTS

This research was supported in part by the National Institute of Standards and Technology under grant #60NANB12D249. The views and conclusions expressed are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of NIST, the U.S. government, or any other organization.

## REFERENCES

- Acemyan, C.Z., Kortum, P., Byrne, M.D., & Wallach, D.S. (2014). Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. *USENIX Journal of Election Technology and Systems (JETS)*, 2(3), 26-56.
- Acemyan, C.Z., Kortum, P., Byrne, M.D., & Wallach, D.S. (in press). Users' mental models for three end-to-end voting systems: Helios, Prêt à Voter, and Scantegrity II. *Lecture Notes in Computer Science*.
- Adida, B. (2008). Helios: Web-based open-audit voting. *Proceedings of the 17<sup>th</sup> USENIX Security Symposium, USA*, 17, 335-348.

- Adida, B., De Marneffe, O., Pereira, O., & Quisquater, J. J. (2009). Electing a university president using open-audit voting: Analysis of real-world use of Helios. *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, USA, 18*.
- Belton, G., Kortum, P., & Acemyan, C.Z. How hard can it be to place a ballot into a ballot box? Usability of ballot boxes in tamper resistant voting systems. *Journal of Usability Studies*, in press.
- Carback, R., Chaum, D., Clark, J., Conway, J., Essex, A., Herrnson, P.S., . . . . Vora, P.L. (2010). Scantegrity II municipal election at Takoma Park: The first e2e binding governmental election with ballot privacy. *Proceedings of the 19th USENIX Security Symposium, USA, 19*.
- Chaum, D., Carback, R.T., Clark, J., Essex, A., Popoveniuc, S., Rivest, R.L., . . . A.T., Vora, P.L. (2009). Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE Transactions on Information Forensics and Security, 4*(4), 611-626.
- Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., & Vora, P. (2008). Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy, 6*(3), 40-46.
- Chisnell, Dana. (2013). *Writing instructions voters understand* (Field Guides to Ensuring Voter Intent, Vol. 02). Retrieved from the Presidential Commission on Election Administration website: <https://www.supportthevoter.gov/files/2013/08/Field-Guide-Vol-02-20130620.pdf>
- Everett, S. P. (2007). *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection* (Doctoral dissertation, Rice University). Retrieved from <http://chil.rice.edu/alumni/petersos/EverettDissertation.pdf>
- Del Galdo, E.M., Williges, R.C., Williges, B.H., & Wixon, D.R. (1986). An evaluation of critical incidents for software documentation design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, USA, 30*(1), 19-23.
- Goodman, E., Kuniavsky, M., & Moed, A. (2012). Balancing needs through iterative development. In *Observing the User Experience: A Practitioner's Guide to User Research* (pp. 21-44). Waltham, MA: Morgan Kaufmann.
- Green, R. (1990). Human error on the flight deck. In D.E. Broadbent, A. Baddeley, and J.T. Reason (Eds.), *Human Factors in Hazardous Situations* (pp. 503-512). Oxford: Clarendon Press.
- How to Vote with Scantegrity*. (n.d.) Retrieved from [scantegrity.org](http://scantegrity.org)
- Itoh, K., Omata, N., & Andersen, H.B. (2010). A human error taxonomy for analyzing healthcare incident reports: Assessing reporting culture and its effects on safety performance. *Journal of Risk Research, 12*(3-4), 485-511.
- Johnson, J. (2010). *Designing with the mind in mind: Simple guide to understanding user interface design rules*. Burlington, MA, Morgan Kaufmann.
- Kaczmarek, T., Wittrock, J., Carback, R., Florescu, A., Rubio, J., Runyan, N., Vora, P.G., & Zagorski, F. (2013). Dispute resolution in accessible voting systems: The design and use of Audioteegrity. *Lecture Notes in Computer Science, 7985*, 127-141.
- Lewis, J.R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 36*(2), 368-378.
- Lundin, D., & Ryan, P.Y. (2008). Human readable paper verification of Prêt à Voter. In S. Jajodia & J. Lopez (Eds.), *Computer Security – ESORICS 2008: Proceedings of the 13<sup>th</sup> European Symposium on Research in Computer Security, Malaga, Spain, October 6-8, 2008* (pp. 379-395). Berlin, Germany: Springer Berlin Heidelberg.

- Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63, 81-97.
- Nielsen, J. (1997). *How Users Read on the Web*. Retrieved from <http://www.nngroup.com/articles/how-users-read-on-the-web/>
- Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of ACM CHI'90 Conference, USA*, 249-256.
- Norman, D.A. (1983). Some observations on mental models. In D. Gentner, & A.L. Stevens (Eds.). *Mental Models* (pp. 7-14). New York, NY: Lawrence Erlbaum Associates.
- Page, A., & Pitts, M.J. (2009). Poll workers, election administration, and the problem of implicit bias. *Michigan Journal of Race & Law*, 15(1), 1-57.
- Prêt à Voter*. (n.d.). Retrieved from <http://www.pretavoter.com>
- Reason, J. (1990). *Human error*. Cambridge, England: Cambridge University Press.
- Rubin, J., & Chisnell, D. (2008). Conduct the test sessions. In *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* (pp. 201-220). Indianapolis, IN: Wiley.
- Rohn, J.A. (2005). Cost-justifying usability in vendor companies. In R.G. Bias, & D.J. Mayhew (Eds.), *Cost-Justifying Usability* (pp. 185-191). San Francisco, CA: Morgan Kaufmann.
- Ross, J. (2010). *It's Not a Training Issue*. Retrieved from <http://www.uxmatters.com/mt/archives/2010/12/its-not-a-training-issue.php>
- Ryan, P. Y., Bismark, D., Heather, J., Schneider, S., & Xia, Z. (2009). Prêt à Voter: A voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4), 662-673.
- Ryan, P.Y., & Peacock, T. (2010). A threat analysis of Prêt à Voter. In D. Chaum, M. Jakobsson, R.L. Rivest, P.Y. Ryan, J. Benaloh, & M. Kutylowski, (Eds.), *Lecture notes in computer science: Vol. 6000. Towards trustworthy elections: New directions in electronic voting* (pp. 200-215). New York, NY: Springer.
- Ryan, P.Y., & Schneider, S.A. (2006). Prêt à Voter with re-encryption mixes. In D. Gollmann, J. Meier, & A. Sabelfeld (Eds.), *Computer security – ESORICS 2006: Proceedings of the 11<sup>th</sup> European symposium on research in computer security, Hamburg, Germany, September 18-20, 2006* (pp. 313-326). Berlin, Germany: Springer Berlin Heidelberg.
- Sanders, M.S., & McCormick, E.J. (1993). *Human factors in engineering and design*. New York, NY: McGraw-Hill.
- Sauro, S. (2010). *Are The Terms Formative And Summative Helpful Or Harmful?* Retrieved from <https://www.measuringu.com/blog/formative-summative.php>
- Schmettow, M. (2012). Sample size in usability studies. *Communications of the ACM*, 55(4), 64-70.
- Sherman, A.T., Carback, R., Chaum, D., Clark, J., Essex, A. .... Vora, P. (2010). Scantegrity mock election at Takoma Park. *Electronic Voting*, 45-61.
- Shneiderman, B. (1987). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.
- Sisson, D. (n.d.). *Assumptions about user search behavior*. Retrieved from [philosophe.com/search\\_topics/user\\_behavior/](http://philosophe.com/search_topics/user_behavior/)
- Spool, J.M. (1999). *Web site usability: A designer's guide*. San Diego, CA: Academic Press.

Staggers, N., & Norcio, A.F. (1993). Mental models: Concepts for human-computer interaction research. *International Journal of Man-Machine Studies*, 38, 587-605.

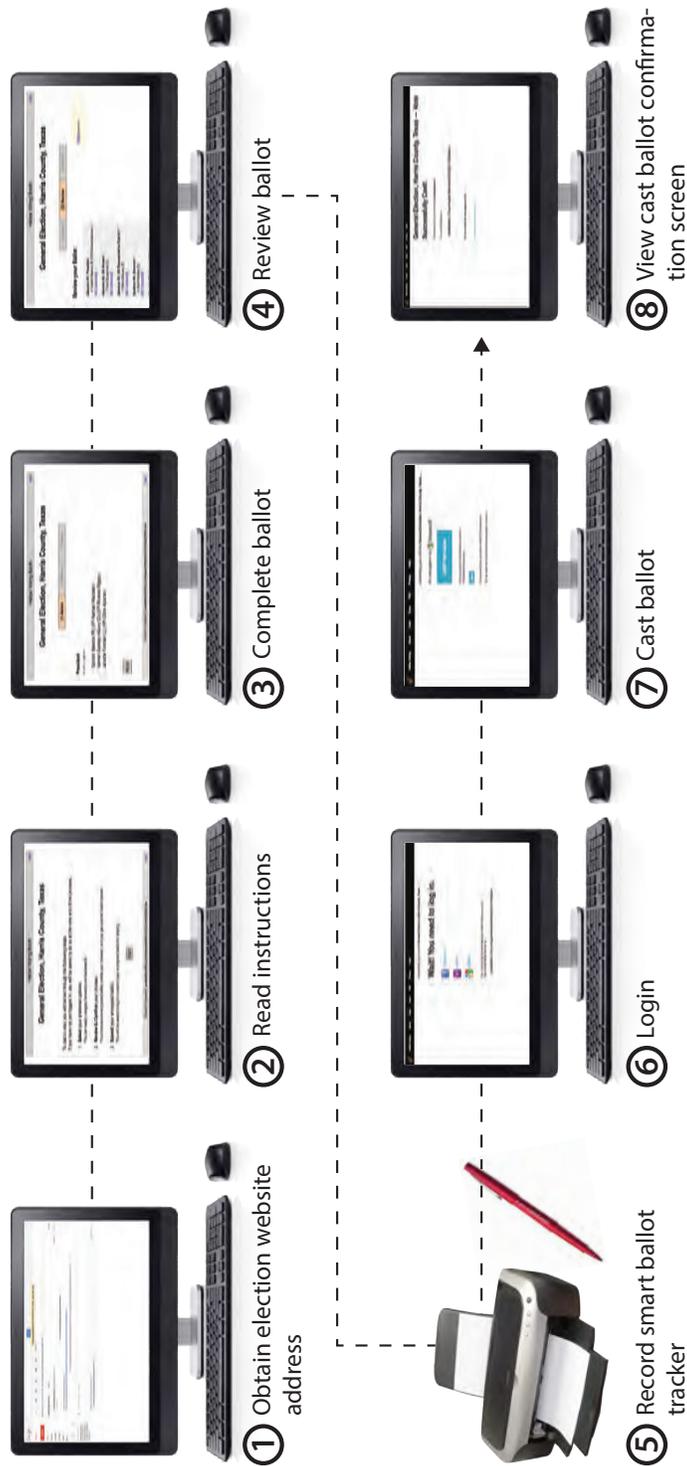
The City of Takoma Park, Maryland. (n.d.). Retrieved from <http://www.takomaparkmd.gov>

Tidwell, J. (2011). *Designing interfaces*. Sebastopol, CA: O'Reilly Media, Inc.

Tullis, T.S., Tranquada, F.J., & Siegel, M.J. (2011). Presentation of information. In K.L. Vu, & R.W. Proctor (Eds.), *Handbook of Human Factors in Web Design* (pp. 154-184). Boca Raton, FL: CRC Press.

APPENDIX A

Typical Helios Vote-Casting Procedure

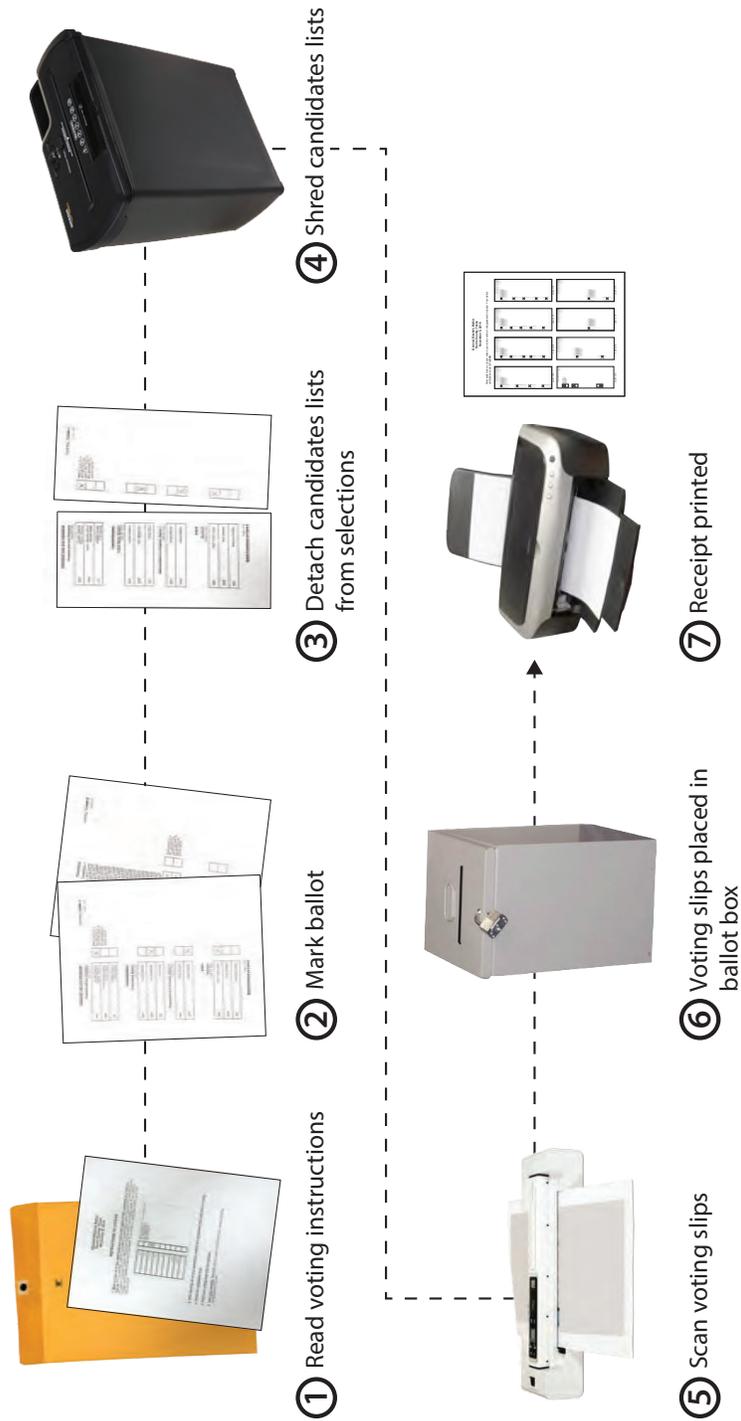


### Typical Helios Vote-Verification Procedure

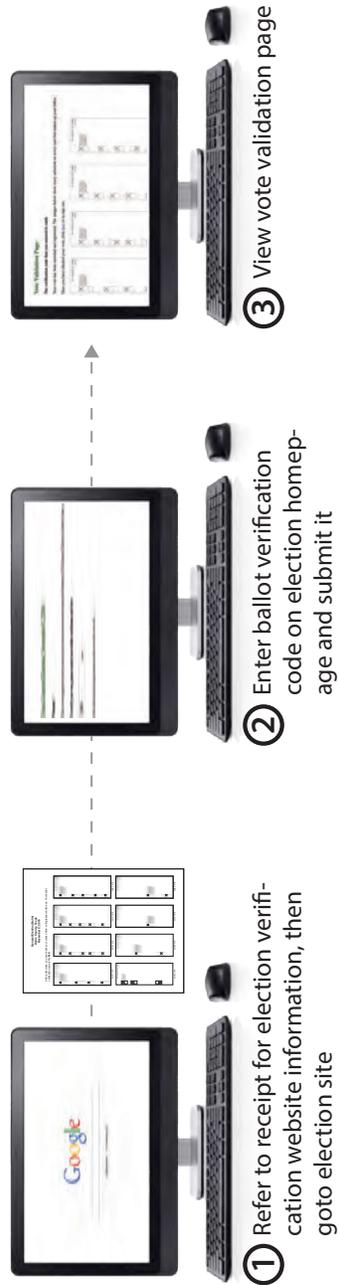


APPENDIX B

Typical Prêt à Voter Vote-Casting Procedure

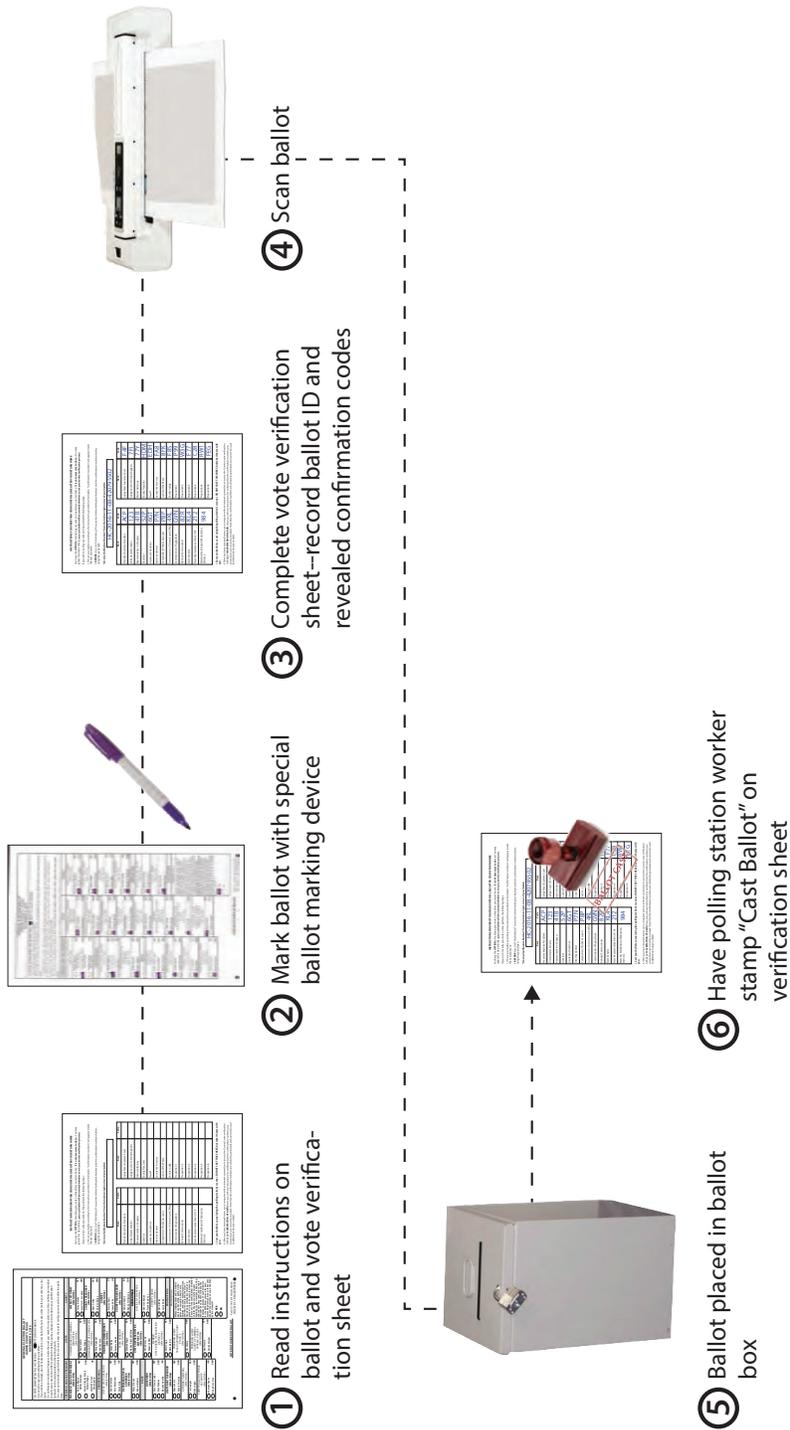


### Typical Prêt à Voter Vote-Verification Procedure



APPENDIX C

Typical Scantegrity II Vote-Casting Procedure



**Note:** Step six was based on *Scantegrity II: End-to-end Verifiability for Optical Scan Election Systems Using Invisible Ink Confirmation Codes* (Chaum et. al., 2008).

### Typical Scantegrity II Vote-Verification Procedure

