Gitub Intro:

This initial repository is to plan the best way to build open-source BallotBox.Vote apps for iOS and Android. The goal is to make simple, easy-to-use apps for creating, printing and scanning paper ballots for un-official ranked-choice elections (RCV). By un-official we mean for non-government elections such as class president or local civic or business association leaders (or any group decisions, such as what food to order for the office lunch). The apps will always be free to use, and the entity organizing this effort is applying for non-profit educational status. Please see the draft PDF app screenshots also posted.

Why are we creating this app and non-profit?

We are making this effort because we believe that despite RCV's recent momentum, it is still underperforming its vast potential as a democracy-strengthening reform.

We asked ourselves why is it so many very well-informed people we meet have still not even heard of RCV? Our theory is that it's an especially abstract idea to begin with, and that remains abstract because it is too hard to use in everyday life. Gerrymandering, money in politics, ballot access, etc. and other democracy-warping issues are also not exactly visceral, but they get attention and therefore build awareness when specific controversies arise. But ranked choice? We've got to get creative to bring about real grassroots awareness.

There are a few ways a busy teacher or a local chamber of commerce president, for example, could hold a ranked-choice election right now – whether with their own ballots and spreadsheet tabulation, or through a few, mostly for-profit sites that offer electronic means. But if those methods were going to bring RCV to your average prom king election, it would have happened already. But just as importantly, 99% of the teachers and community leaders right now are probably not even thinking about using ranked choice at all.

What's our plan?

So BallotBox.Vote plans to A) create super-simple apps for paper RCV ballots, and B) actively recruit volunteers to contact schools and civic groups in their communities to use the apps to hold their next elections. It's a two-part strategy. BallotBox.Vote is not going to lobby for RCV. Our entire mission is educational and begins and ends with just A and B above.

Our goal with the apps is to make voting with RCV not only easier, more auditable and transparent than any other un-official RCV system, but to make using RCV easier than any other system or voting method whatsoever. We want to be easier than dropping slips of paper into shoeboxes and counting the slips, or even having to ask people in a large room to raise their hands while someone points and counts.

GitHub Stuff

The founder of this effort has no software experience. Hopefully some professional developers will volunteer to take the technical leads on this project before too long. In addition to this document, please look at the separate PDF file of draft app screenshots. Those mock-ups communicate clearly what we want to build. I should add that the goal is for as much as possible of the ballot creation, scanning and tabulating to happen on the phone itself. Ideally elections with ballots being scanned by only one phone could be conducted with that phone on airplane mode.

Our assumption is that the initial stage of this effort should be to decide what computer languages and libraries to use and how to organize the actual programming in general and on GitHub specifically.

The easiest thing may be for participants to email advice on those topics to ballotboxinfo@gmail.com, or upload separate text files to this repository. One long read-me might not be the best method.

For now, let's say the iOS and Android apps have three main pieces:

1. The ballot-making section where the user can input candidate names and the number of ballots needed, and have a single PDF containing all the ballots created and then saved and/or e-mailed out. Each ballot page will have a unique serial number.
2. The ballot-scanning and tabulation section.
3. Wrapping #1 and #2 up into the phone apps and their user interfaces.

Also, below is link to a tutorial about using OpenCV to read classroom test bubble sheets. It uses Python, and that might not be the optimal language to then create smart phone apps. We've been told that OpenCV is a good tool for the camera scanning. Hopefully the Python image scanning tutorial can be adapted to a better app language - and to RCV from simple bubble test scoring - by someone with the right experience without too much trouble.

https://www.pyimagesearch.com/2016/10/03/bubble-sheet-multiple-choice-scanner-and-test-grader-using-omr-python-and-opencv/

Also, there is "Universal Ranked-Choice Tabulator" code that has been created by the RCV community of election officials. This project hopes to use that code to do the actual tabulation of the rounds.

It's important to note that the app is not intended to be super-duper secure with retina scans into blockchain, or ever be used for elections to government office. It will never communicate with official, government voter databases or any outside ID systems, or have proprietary systems like official voting machines do. While we are eager to be advised on security issues by experts, as of now, we do not intend to use any security methods beyond off-the-shelf tools built-into Internet browsers.

The goal is common-sense security. If a teacher downloads the app directly from a major app store, and holds an election for prom king an hour later, and the paper ballots are all scanned by the teacher standing right there as the students hand them in, we think that's good enough. Our goal is only that all of the participants in that election feel confident in the outcome, and not that the system might be theoretically hackable. It's up to the person running each election to manage their own ballot printing, distribution, collection, scanning and storage.

Also, by relying on paper ballots, any questions about the scanning and tabulation can be answered by hand-counting the ballots, or by scanning the same ballots with another phone that just downloaded the app again from a major app store. Our goal is, in part, security through transparency and redundancy.

Perhaps the largest near-term impact of this project concerns use at local political conventions. At least in our home state, state and local political parties choosing nominees at in-person conventions (as opposed to primaries) are free to count votes in any way they choose.

We've read about state and local conventions with 2,000 participants. Ideally, BallotBox.Vote will be able to handle an election on that scale with paper ballots being scanned by several devices with some centralized authorization (as outlined in the PDF of draft screenshots) and data sharing system.

Should BallotBox.Vote take hold at local or state political conventions, it could start to help our democracy before RCV gets approved by any other towns, cities or states for official elections. Simplifying and normalizing RCV for political parties obviously is a great way to educate elected leaders on the existence of and the benefits of RCV.

Once the paper ballots system works on phones, it might make sense to add voting by e-mail and/or by desktop computer in a year or two, so long as the system is still very, very simple to use. But for now it's paper ballots with smartphones.

There are of course many different variants of RCV. BallotBox.Vote plans only to offer the "standard" RCV, which we will somewhat arbitrarily define as the rules in use in Maine, as that is the largest RCV jurisdiction. Also, there are groups such as labor unions who can hold elections electronically by following special protocols. BallotBox.Vote does not see these more formal (but still not-for-government office) elections as a part of its mission. We do not plan to offer any telephone support or even much support by e-mail.

BallotBox.Vote wants only to be the decision system for basic, community elections. We want to remove all possible barriers that might stand in the way - such as cost and complexity - to average citizens trying RCV for the first time. The apps have to be very intuitive.

Lastly, one of the sites we look to as an example is SignUpGenius.com. It only does one thing, but everybody (we know) uses it for that one thing. People may only use it once a year for that bake sale sign-up, but the site does what it is supposed to do, and there is no reason to try another site. Similarly, we want to be the default system for local civic elections, and use that position to bring RCV into mainstream use.

Please email or post your advice on getting these apps built. Thanks.