CIVIL GRAND JURY | 2017-2018

CITY AND COUNTY OF SAN FRANCISCO

Open Source Voting in San Francisco

"If you want to go quickly, go alone. If you want to go far, go together."

CIVIL GRAND JURY | 2017-2018

CITY AND COUNTY OF SAN FRANCISCO

Jurors 2017-2018

Lori Campbell, Foreperson
Heather Dolan, Secretary
Richard Bogan
Paul Buxbaum
Charles Dworetz
William Hannan
Rasha Harvey
Hon. Alfred Knoll (ret.)
Douglas Lam
John Lee
Paul Pferdner
Charles Raznikov
John Sandoval
Derek Schaible
P Segal
Grady Ward

Executive Summary

Over the last nine years, San Francisco's Open Source Voting project has had affirmations of support from the Mayor, the Board of Supervisors, the Elections Commission, local experts, independent consultants, the Secretary of State, and members of the State Assembly and Senate. Consensus usually builds to action, but it has yet to do so with this project, which has remained in a state of hypothetical exploration for the better part of a decade. The Civil Grand Jury has found clear structural and organizational obstacles to completing the project and achieving its anticipated benefits.

We found three reasons that the project has failed to gain traction. First, despite having a plethora of stakeholders with the right intentions and knowledge, expertise is scattered within City agencies and organizations. Experts in elections administration, open source experience, and purchasing are not structurally aligned. Second, there are too many people who need to sign off on the project for any one of them to make impactful and informed decisions. Third, there is not a definitive project owner with both an institutional responsibility for the project and an eagerness to tackle it. Without structural changes that align city institutions and establish a clear leader, the dispersal of experts and decision-makers ensures the project's current pattern of disjointed progress. This leaves the project at an impasse, where repeated expensive and equivocal investigations are mistaken for progress.

The Civil Grand Jury has determined a set of procedural and structural adjustments that have the potential to increase the likelihood that the project will complete successfully. These recommendations center around clear ownership and creative partnerships. The project needs an unambiguous owner in order to move forward. Partnerships with other counties, software-focused non-profits, and the California Secretary of State can render this undertaking tractable.

The Civil Grand Jury has found that if the Open Source Voting project is completed, many of its promised benefits are likely to come to fruition in the long term. These include cost savings, flexibility for the city, and transparency. However, many claims of immediate advantages for the City are unsubstantiated. Open source software does not possess inherent benefits for San Francisco taxpayers, instead it will increase costs and add compliance and administration risks in the intermediate term. Finally, there is not a convincing time horizon to realize these benefits because the project remains too nebulous to concretely evaluate.

San Francisco has taken a decade to debate and assess the value of open source voting. If this project continues, in ten more years, San Francisco will either have created new critical democratic infrastructure or will have wasted taxpayer dollars by perpetually planning for an unrealized future. What separates these two scenarios is strategic multilateral partnerships, open source best practices and culture, and strong commitment under unambiguous ownership.

Methodology

To produce this report, the Civil Grand Jury compiled information from a broad array of sources. Secondary sources such as news media and academic literature formed our initial lines of inquiry, and primary sources like interviews with experts and government reports provided us with corroborating details. However, all facts within this report were either substantiated by three interviewed sources or have been extracted from documents produced by city, county, state or federal governments.

We interviewed San Francisco officials, non-profit advocates, academics, software developers, and procurement specialists. These interviews gave insight into the progress of the Open Source Voting project, and an understanding of how project stakeholders are aligned and informed.

We examined government records, including internal documents about the project, contracts with voting system providers, and external consultant reports. These provided structural details, and views into how the project has progressed (and stalled) over its history.

We attended and watched and meetings of the Election Commission, the Open Source Voting Technical Advisory Committee, and the Board of Supervisors, and utilized the minutes and resolutions from each.

We dug through a wide array of evidence and data available on the websites of the Elections Commission and the Department of Elections.

Members of the Jury worked as poll workers in the June 2018 primaries to get first hand experience with the process of election administration.

We thoroughly explored the findings and conclusions of other reports on this subject from the last fifteen years, produced by the City, the State, and by other parties. These included a 2006 statement issued by the Secretary of State, a 2017 feasibility study performed by the city, a 2014 summary published by the Election Commission, and additional reports devoted to Open Source Voting generated by non-profits and consultants. These reports informed our line of questioning and gave a sense of how the project has changed over its history.

Compiling and digesting all of this information gave us a deep understanding of the current state of the project, as well as visibility into the potential pitfalls that the OSV project is likely to face. This report has two goals: to increase the efficiency of the City’s progress toward Open Source Voting, and to make the important issues of this project accessible to non-technical citizens.

Background

California elections are administered at the county level, and San Francisco (as both a city and county) tasks the San Francisco Department of Elections (DoE) to conduct its city and county elections.[1] The DoE acts under the authority of the Elections Commission (EC), an independent body of appointed officials. Though the DoE is not directly under the Mayor's office, its budget is drafted and approved within the standard allocations process–funding is proposed in the Mayor's budget and approved by the Board of Supervisors (BoS).

The Elections Commission (EC) oversees the creation of election policy and hires the Director of the DoE. Some members of the Elections Commission have been vocal advocates for Open Source Voting Systems, both in and out of their roles as commissioners. In April of 2017, the EC established a subcommittee called the Open Source Voting Technical Advisory Committee (OSVTAC) to provide technical guidance on the best way forward for a San Francisco OSV project.

A voting system is a sophisticated package of hardware, software, and logistics, each of which needs to work in careful coordination with the other components to operate as intended. Generally, governments license election systems from corporations that provide voting system services, through contracts that usually last between four and ten years. You can read details about what constitutes a voting system in Appendix A.

Open Source Software (OSS) is software that is generally free for anyone to download, use, modify, and redistribute. Some of the most widely distributed and used software in the world is open source, and much of the cutting edge software being developed today relies on open source projects. The security of an open source project is dependent on the number and size of the organizations and individuals that scrutinize it. You can read about what constitutes open source software in Appendix B, common properties of open source software development in Appendix C, and how to interpret the security of open source software in Appendix D.

An Open Source Voting System (OSVS), is a voting system where the software and logistical components of the system are licensed and available as Open Source Software, and the hardware is made up of commodity (or "off the shelf") components. A complete OSVS would provide a way for any jurisdiction to run an election at minimal cost, without licensing an election system from a vendor. San Francisco's Open Source Voting project is pushing for the city to build an Open Source Voting System to be owned and operated by San Francisco, but free for other counties and nations to use. An explanation of what would comprise this system is covered in Appendix E.

[1] This report will refer to San Francisco consistently as "the City", though this shorthand jointly refers to its capacities and responsibilities as both a city and county.

In California, voting systems must be certified by the California Secretary of State (SoS) before they are used. Certification occurs at a "snapshot" of the system, such that nothing can change after the certification is complete without requiring recertification. This process can be expensive, time-consuming, and has historically certified systems from only eight vendors (only four of which are in use today). California has never certified an Open Source Voting System. More information about the certification process is available in Appendix F.

Almost all American elections are conducted using election systems from three election software providers. These providers have consolidated over time due to the nature of their product, and display non-competitive tendencies that have drawn the attention of federal antitrust regulators. This, in combination with San Francisco's peculiarities, limits the City to only one system that can be used to conduct elections in San Francisco while complying with state and county laws. An overview of the forces shaping this market, and what it means for San Francisco is included in Appendix G.

A large number of materials discuss the pros and cons of developing an OSV system, and brief summaries of some past reports and recommendations are included in Appendix H.

There have been efforts to move forward an OSVS within San Francisco for the better part of the last decade, but San Francisco is not the only jurisdiction in the United States that has attempted to develop an OSVS. An analysis of some of the more significant efforts, and their implications for San Francisco are discussed in Appendix I.

These appendices serve as the basis for many of the facts presented in this report. The following discussion focuses on ways that San Francisco could improve upon its current efforts, drawing upon the background laid out in detail in the appendices.

Discussion

This discussion is organized to answer three motivating questions:

- What claims about the benefits and risks of an open source voting system are credible?
- Why has the Open Source Voting project failed to realize material progress?
- What does a path to successfully completing the project look like?

Each of these questions is answered in detail in a corresponding section below.

Evaluating Risks and Benefits of an Open Source Voting System in SF

More ink has been spilled on the value of an OSV system than on the code to build one. We have found strong evidence to support long-term cost savings to the City, and ambiguous analysis of the impact of openness on the security of the effort. We will attempt to shed light on the concrete types of value the system would bring to the city, and how that value can be maximized.

Cost

In the non-competitive market for Election Systems, owning software saves costs. In attempting to rent or lease voting systems from a for-profit provider, San Francisco has faced a series of challenges that stem from properties of the market. To start, there are only a small (and quickly consolidating) number of vendors that sell election technology.[2] This is compounded by a costly and time-intensive State certification process that disincentivizes new market entrants.[3] On the demand side, most bidding processes for elections systems expressly state historical system use in their vendor qualifications, further limiting the number of vendors that can participate. Additionally, the City cannot buy from the second largest election system provider because the firm is based in Austin, Texas, and California has decided to stop doing business with any state that enacts transphobic bathroom restrictions (which Texas did in 2016).[4] Finally, and critically, the City's elections demand Ranked Choice Voting capabilities which few vendors provide in their flagship products.[5]

These problems are not new. They have been noticed, but not addressed by San Francisco leaders for over a decade. The challenge of finding a certified vendor who supports ranked choice voting was outlined in a memo to the Board of Supervisors from the Director of Elections

[2] This is detailed in Appendix G, which provides an overview of the dominant vendors for election system technology.

[3] Details on the Secretary of State's Election System Certification Process are included in Appendix F.

[4] An article discussing the financial and logistical implications of the perspective of Texas is here.

[5] Details about the claimed support for Ranked Choice Voting from each of the major election systems providers is included in Appendix J.

in 2006.[6] It was a pivotal part of the 2007 contract for services from Sequoia (now Dominion) Systems.[7] It was highlighted in each Board of Supervisors approval of extensions of that contract in 2011, 2013, and 2017.[8] [9] [10] It is one of the reasons that the 2015 process to investigate potential new voting systems (including Open Source Systems) did not proceed beyond the RFI phase.[11] In the last decade, the problem has gone unchanged by market forces: only one corporation has a Ranked Choice Voting System certified by the California Secretary of State.[12] Therefore, it should not be surprising that the 2018 RFP[13] for a certified RCV voting system returned only one bidder: the one the City currently uses, and the only one that supports RCV in California.

Any market with a single supplier and a legally compelled purchaser cannot possibly be deemed competitive. San Francisco is compelled by State and City laws to support certification and feature requirements that are not being offered by multiple systems. Thus, the City is forced to continue extending a contract that is increasingly expensive, for software that is long overdue for an update.

The cost of this predicament is increasing. Since the contract was written in 2007, San Francisco has paid around $1.1 million per year in operational costs for our current system, provided by Dominion, our current election system provider.[14] The initial contract ran for four years, and has now been extended to eleven. As the contract was expiring at the end of 2018, the Department of Elections was compelled to enter into a new contract with the same provider. In a budgetary analysis in 2017, the DoE noted that the vendor had asked for a "updated pricing model", which turned out to be a four year contract with three one-year extension options at an annual cost of $2 million per year. Accepting this increase in prices was San Francisco's only option, because no alternative systems fit the City's needs.

---

[6] Memorandum from the Director of Elections to the Board of Supervisors, 2006.

[7] San Francisco Board of Supervisors, Resolution No. 654-07

[8] San Francisco Board of Supervisors, Resolution No. 494-11

[9] San Francisco Board of Supervisors, Resolution No. 269-13

[10] San Francisco Board of Supervisors, Resolution No. 005-17

[11] San Francisco Portal for 2015 Bidding Process (Lots of Broken Links)

[12] Overview of Sequoia (now Dominion) Ranked Choice Voting systems in use in California

[13] Request for Proposals for Leasing or Renting a Voting System, San Francisco DoE, February 1st 2018

[14] The contract was originally between San Francisco and Sequoia voting systems, but Sequoia was purchased by Dominion in 2010.

It is challenging to determine whether these prices are reasonable. This is because the national market for election technology is subject to many of the same forces which distort San Francisco's purchasing decision. Vendors have heavily consolidated state marketplaces so that cities and counties have fewer options, and vendor lock-in is common. Moreover, funding sources have changed dramatically over the last ten years because of the rapid injection of Federal money through the Help America Vote Acts (HAVA) of 2002 and 2018. Finally, contracts for voting systems have both operational costs (that are a yearly recurring costs for software licensing and support), and capital investments (larger, one-off payments for machines). Parsing a contract cleanly into operational and investment costs complicates analysis and comparison of election budgets.

Uncertainty about the future of the OSV project makes it impossible to provide robust cost estimates. Projections for the cost to build an open source system have varied between $8 million and $50 million[15]. Even at the high end of that estimate, this would still save San Francisco money on a 15-20 year time horizon. We do not know much about what the world will be like in 20 years, but we know there will still be elections, and that they will largely operate under the same rules that they do today.

The jury concluded that San Francisco necessarily overpays to operate the critical infrastructure of its democracy. Elections are conducted in an environment of necessity, with limited recourse. San Francisco's decision to construct its own voting system seems to provide a way out of the bind it finds itself in today.

Security

In open source software, security depends on scrutiny. Open source software is not fundamentally more or less secure than closed source software (where the source code is not available for inspection). Instead, the security of open source software increases with its use and importance, because the more people that pay attention to it, the more likely it is that a security researcher or engineer will discover and report a vulnerability within the software. Some of the world's most widely used software is open source, including its two most used operating systems (Linux and Android). These systems are secure because hundreds of security experts, organizations and researchers are invested in finding and fixing their vulnerabilities. However, open source software that doesn't receive much attention is more vulnerable than similar closed-source software. This is because attackers have more insight into how the code works and can find vulnerabilities more easily, but no security researchers are finding and fixing the same vulnerabilities.

[15] The jury did not find any convincing justifications for any of the cost projects we have examined, nor do we possess the expertise to effectively evaluate the likely cost.

This leads to a critically important understanding - the best way to make sure that San Francisco's open source election system is secure is to get other counties and states to use it too. If the election security of more jurisdictions is tied together, there will be greater interest in scrutinizing and defending the software the group depends on. Safety in numbers should be a guiding principle of the Open Source Voting project.

We don't have enough information to compare the security of a hypothetical OSV election system against an existing closed source analogue. There is no reason to believe that there are vulnerabilities in the election systems in use today, though we do have evidence to suggest certification is not an effective test of system readiness.[16] However, there is also not strong confidence that such vulnerabilities do not exist–because we don't have broad public access to the source code. The consensus of the information security community is that an open source project, if used broadly (and thus studied broadly) would be more secure against malicious actors than a closed source system.

Virtue

The energy and passion that drives advocates toward an OSVS does not stem from their objective analyses of cost, security or vendor lock-in. Though each of these provide support for the development of an Open Source Voting system, it is the more abstract virtues that have pushed this issue forward in the public and political spheres.

Advocates believe that the OSV project would function as a technological landmark for San Francisco, and be a boon for United States counties and democratic countries around the world. This is typically the type of argument used to drive the project forward, and though these claims are not discussed or evaluated within this report, it is helpful to remember that this project means more to San Francisco than the sum of its projected cost savings and security wins. Public ownership of the infrastructure of democracy provides virtuous, if intangible, benefits.

Organizational Structure and Partnerships

San Francisco does not have a track record of mission-critical software development.[17] Though this is an obstacle for the OSV project, it also opens up a variety of organizational possibilities, many of which involve collaboration outside the boundaries of the City.

Several proposals have circulated that suggest San Francisco's Department of Elections (DoE), or Department of Technology (DoT) should develop the open source election system "in house",

[16] In newsworthy events in the run up to the first implementation of RCV in San Francisco in 2004, there were times when the system (though certified by the SoS) would fail to produce results. This does not inspire confidence in certification as a measure of completeness.

[17] San Francisco does have some measure of experience developing open source software, as the DoT has developed some for internal use in past projects. However, the scope of these projects has been expressly limited to San Francisco, and none of the projects we found were for critical functions of government.

rather than purchasing the software from a vendor.[18] There is ample evidence to suggest that these are not workable recommendations. The DoE does not have experience developing software, nor with managing large technology projects. Building technology is squarely outside of the mandate of that Department to administer elections within San Francisco. Though the DoT does have an active software development team, and an official policy to develop software as open source when possible, the majority of the tools published as open source components are specific to the needs of San Francisco. With few exceptions, they are not in use outside the city, and lack any form of community engagement or development health[19]. The DoT does not have a deep understanding of the project requirements, nor does it have a track record of successfully developing software at scale for external use. The DoE's lack of software experience is equally disqualifying.

Though the jury recommends against the City writing the software for an open source system itself, we do not mean to diminish the importance each department will play within the successful completion of this project. The Department of Elections can be the subject matter expert on product requirements, has the capacity to build connections and partnerships, and has successfully navigated election system certification[20]. The Department of Technology would be a key partner with the DoE to establish product requirements that conform to open source best practices, evaluate vendor performance, and analyze the technical progress and roadblocks for the project.

Non-Profits as Collaborators

Open source software is frequently developed and overseen by non-profit organizations. We have identified several ways in which partnerships with existing non-profit organizations could move the project forward. There are technology non-profits that might be interested in helping to build the software from the ground up, and others that could provide technical and elections guidance. There is a broad spectrum of potential collaborations–from San Francisco as a laboratory for a non-profit software development organization, to a non-profit as a consultant on security and vendor appraisal. In order to clarify the possibilities, the jury sketches out three possible non-profit partnerships to bookend the range of possible collaborations.

18F is a technologically focused branch of the federal General Services Administration (GSA) dedicated to improving the user experience of government. 18F is located around the country, with flagship campuses in DC and San Francisco. They can write software for state and federal

[18] This would mean hiring folks to build the project as employees of each department.

[19] A full list of these projects is available at http://open.innovatesf.com. On each project that the Jury examined, we were not able to find evidence of a robust open source development effort, external users, or strong code review practices.

[20] This capacity was demonstrated during the department's quick work with Dominion in 2017 to patch the "spectre/meltdown" vulnerabilities.

government agencies anywhere that federal funding is used. Their policy is to develop all code as open source (unless it presents a security or privacy risk).[21] Their main criterion in evaluating which projects to tackle is the potential for reuse and impact–dimensions along which an Open Source Election system scores well. So long as some federal funding is being used in the project, 18F would be an excellent collaborator with San Francisco: they have extensive experience building open source infrastructure, a clear mission alignment, and would be interested in building the code in such a way that it could be reused by other jurisdictions.

The Open Source Election Technology (OSET) Institute is perhaps the non-profit organization most clearly aligned with the development of an open source voting system in San Francisco. Based in Palo Alto, the organization is attempting to bring about the actual development and use of an Open Source Election system. OSET claims to be building "ElectOS", an open source election system.[22] In the past they have built voter registration tools used by "Rock the Vote", all of which were developed as open source software.[23] OSET would be another logical partner for the work San Francisco is attempting to accomplish–they have at least some experience developing and building open source software, and have a clear mission alignment.

The Electronic Frontier Foundation (EFF) is a San Francisco based non-profit with a wide portfolio of projects that center on protecting transparency, security, and civil rights in digital spaces. They have experience advocating for legislative change, developing widely used open source software, and scrutinizing the security of open source software. The EFF has even commented on the security of election auditing procedures and election casting systems. The EFF would be an excellent collaborator on an Open Source Voting project as a mentor, able to leverage San Francisco the benefits of decades of activism within the space.

Partner Counties

The proposed Open Source Voting System would be able to operate elections in any number of counties, states, and nations. This sets up a natural set of collaborating jurisdictions who might be willing to commit funding to the project, or to using it if developed. Though this potential seems obvious, it has not yet materialized as concrete action from San Francisco. San Francisco should evaluate the capability and interest of potential partner counties. A good initial boundary for the partnerships to consider is the 58 counties within the state of California, and particularly the four counties using RCV (Alameda, San Francisco, San Leandro, Santa Clara). Many California counties have similar election laws, and this partnership scope would enable San

[21] 18F's Official Open Source Policy, Github

[22] It is important to note that the software for ElectOS was not found by this jury, and this remains a claim of the organization more than a finding of preparedness or completion.

[23] Trust the Vote Project Organization Page, Github

Francisco to seek statewide funding for the development model under a single certification authority. Discussion of why this boundary is outlined in Appendix L.

Additional Funding Sources

Different analyses of how much an Open Source Voting System will cost to build have ranged from $8 million to $50 million dollars. This report was not able to determine a reasonable estimate for the final cost of developing an OSVS, in part because so many factors remain unresolved. Establishing a clear, transparent and predictable cost structure for the project is essential to determining project value, and setting reasonable expectations for users, decision makers, and the public. However, the Jury has found a number of structural choices that could help reduce cost to the city.

In 2002, Congress passed the "Help Americans Vote Act" (HAVA), which allocated matching funds to states to improve their election infrastructure through the Election Assistance Commission (EAC). In the 2018 omnibus spending bill, Congress authorized additional funding to the EAC to continue its work, which focuses on modernizing election technology and providing security guidelines and testing. It remains unclear whether federal funds can be used for this project (as the legislation is geared toward procuring election technologies, rather than developing them). This is obviously a question that the City should attempt to resolve. Though San Francisco has used all of its HAVA funding from the 2002 cycle, it is possible that it could use 2018 funding toward an OSV project.

Over the last decade, various members of the California legislature have expressed strong support an Open Source Voting project, through a variety of avenues. Pursuant to a 2004 request by the California legislature, in 2006 the California's Secretary of State published a report on Open Source Voting (a summary is available in Appendix H).[24] In 2013, the state Senate passed SB-360 to make the certification process more effective for a modular election system.[25] In statements made in 2018, Assemblyman David Chiu and Senator Scott Wiener pledged to request $8 million in matching funds from the State budget for the first California county to develop and certify an open source election system with a copy-left license.[26] [27] If San Francisco is able to develop an effective open source election system operating on commodity hardware, it has the potential to serve a broad range of counties and other jurisdictions, including all California counties under a single set of certification laws. It appears likely that San Francisco

[24] Open Source Software in Voting Systems, California Secretary of State, 2006

[25] California Senate Bill 360, 2013-2014

[26] A Copy-Left license is a particular flavor of open source license that maximizes future collaborative development. More information is available in the Glossary.

[27] The details of this proposal have circulated in pro-OSV camps, but details on it have been hard to find. This effort is being driven by the California Clean Money Campaign.

could receive state financial assistance toward the completion of the project, particularly given the open enthusiasm for the project expressed by San Francisco's representatives within the legislature, and Secretary of State Alex Padilla.[28]

Uniting every one of these funding sources is unlikely, but evaluating each thoroughly and transparently is a good step forward. Unless external funding sources are found and committed, San Francisco's pursuit of this project implies that the city will foot the full bill. Establishing a transparent cost structure is critical to evaluating the project's value to the city, finding development partners, and increasing its probability of success.

Plan for Budget Shortfalls

Finally, there are completion risks associated with funding that make this project (as an all or nothing venture) particularly risky. When funding is tight, political will for ambitious and abstract projects often evaporates. This was the case in Travis County, Texas' attempt to build an Open Source Voting System.[29] The project started in 2009, and had been funded for 7 years before a budget shortfall in 2017 led the county to abandon the project.[30] San Francisco should consider the possibility of funding falling through, and either pre-allocate funding for the project, or pursue the project in a modular manner so that if funding dries up, the project can be paused, and can restart when funding is resecured.

Partnership with the Secretary of State

Certification by the California Secretary of State poses an unscoped complexity to developing an Open Source Election System in San Francisco. As is outlined in Appendix F, the certification process is rigorous, and is based on a frozen version of the software and hardware. There are two challenges that an open source voting project will face due to certification requirements.

The first is the "open source patching problem". Security researchers are constantly discovering problems with software or hardware that leaves pieces of hardware or software vulnerable to attack.[31] Though open source source code can be patched quickly, patching is counter to the "frozen" nature of certification. In the face of security related bugs, recertification imposes significant financial and logistical overhead. This doesn't only apply to the code written for the election system, but poses the same problems for the other pieces of common open source software that the election system would depend upon.

---

[28] Report by Bay Area NBC News, October 4th, 2016

[29] A full discussion of OSV outside of San Francisco is provided in Appendix I.

[30] Travis County ditches plan to craft its own voting system, My Statesman, October 3rd, 2017.

[31] For example, Red-Hat linux (a standard installation of one of the most popular pieces of software in the world) received about 1 vulnerability per week in the last year that they described as "critical".

Even if the core functionality of an election system does not need patches, it is inevitable that the underlying systems that it runs on will, a problem not limited to open source software. Late in 2017, there was a security vulnerability found which impacted a large proportion of the world's computers.[32] This included the computers used by the DoE to conduct elections within San Francisco. To ameliorate the problem, the San Francisco DoE worked with Dominion to recertify the system with the Secretary of State after a patch could be added to the software.

The second problem has to do with the way certification interacts with modular development (which is discussed in detail in Appendix K). Developing the system in independent and interoperable components for use in subsequent elections (as would happen under a modular model) is at odds with the current state of certification of voting systems within California. Each component modification/change out would require a new certification with the Secretary of State, incurring fees and elongating the certification process.[33]

Though a large number of ambiguities still exists for this project, certification is one of the largest unscoped threats to the long term success of the project. Without a strategy for how to work within certification requirements, the project is likely to take more time and more money to complete than expected. Establishing a formal partnership with state certification authorities will be essential to completing a viable project on a reasonable timeframe. This is also critical in narrowing exposure to vulnerabilities when they are discovered. The jury is confident that such a partnership is possible, due to the strong support the project has received from Secretary of State Alex Padilla.

Risk Management

While this report has addressed a number of ways of improving upon the OSV project though concrete dimensions like funding, partnerships, and certification, these actions alone are not sufficient to ensure that the project is completed successfully. An Open Source Voting project is a risky endeavor, and managing this risk is central to delivering a secure, cost-effective and timely system for use in the City. This requires systematically tracking, managing and mitigating project risks. Analyzing the OSV project through the lens of risk management is going to be critical to the full lifecycle of the project, including development, certification, testing, rollout and maintenance.

The largest set of risks introduced by the project stems from areas where the city lacks proficiency. A report by Slalom Consulting, "Open Source Voting System Feasibility

[32] This was the "Spectre/Meltdown" bugs that were associated with flaws in speculative execution logic on intel processors.

[33] LA County has made some progress on this front. Their accessible vote marking system has already been certified for use in the state, despite it not being an end to end solution. This appears to be at odds with the current state of certification regulations that the secretary of state enumerates on their site, but this Jury couldn't surmise how these systems are supposed to interact.

Assessment", performed a comprehensive analysis of the capacities of the City and its agencies' capacities, and in doing so, exposed a series of risks to quality, security, cost and timeline that stem from the City's lack of capacity in some critical abilities.[34] These unsupported faculties include change management, open source community engagement, transparent code review practices, project management, system assembly and validation, technical documentation, hardware maintenance, and supplier validation.

These risks need centralized management. The project owners will need to synthesize the sea of ink that has been spilled on this project into a series of tangible risk areas. Each area will need to have a risk-owner, and a strategy for managing the risk, through either avoidance, transfer, mitigation, or acceptance. In some cases, structural or architectural decisions around the development of the system could avoid some risks. In others, transferring risks to vendors or partners who are better prepared to address them can help. Some risks can be mitigated through preventative actions, while still others will need to be explicitly accepted. Successful completion of the project will be contingent upon how effectively its many risks are enumerated, tracked, and ameliorated.

Operation within Mandate

Though counterintuitive, building an election system that can serve counties other than San Francisco will ultimately increase the likelihood that the project meets the needs of the City. This is because a wider audience draws the interest and backing of a broader set of collaborators, increases system security, and provides more impetus for regulatory and federal partnerships. This project works best for San Francisco if it is a large and collaborative effort.

However, the mandate of San Francisco's government is to serve local needs. San Francisco's publicly elected officials write and approve budgets that serve San Francisco residents. The Department of Elections operates elections for San Francisco voters. The Elections Commission oversees election administration and policy within San Francisco. The Department of Technology serves to build tooling for internal city needs. None of these organizations has experience with cross jurisdictional partnerships, and working toward goals other than those of San Francisco citizens is outside each of their official mandates.

Thinking and operating on a scale larger than San Francisco is likely to make this project more successful, but the mission of policy and decision makers in San Francisco is to serve local constituents. This disconnect between mandate and requirement is structural, and requires a structural fix. San Francisco's Mayor will need to deliberately consider how to assemble internal resources of this project in order to address this disconnect.

[34] The Slalom report is discussed in detail in Appendix H.

Distributed Knowledge, Responsibility and Ownership

Despite affirmative support from a broad set of stakeholders, San Francisco's Open Source Voting project has been slow to translate political will into concrete action. The jury sought to understand why the Open Source Voting project, which has supporters within a plethora of city departments, decision makers, and external organizations, has failed to gain traction. We found three primary causes.

First, requisite sets of experience within city government are dispersed. The DoE has insight into election administration, system requirements, and certification, but has no experience in rigorous software development, or open source practices. The Department of Technology has experience developing open source software, but is typically doing so for internal city projects without external use. The City's Committee on Information Technology (COIT) has the broader picture of how the OSV project fits into a set of budgetary tradeoffs with other technology projects, but only has experience with purchasing technology, and lacks context on how to build and design it. The Election Commission has broad understanding of both the technical issues around open source software and the general shape of election systems, but lacks context on certification and system requirements.

Second, the City's decision makers are not informed on the issue. The Mayor's office is ultimately responsible for deciding the future of the effort, but lacks an understanding of how feasible and risky the project is. In order to inform that analysis, the Mayor's office has leaned on external reports and COIT, neither of which have been able to offer sufficient guidance. COIT is not equipped to perform analyses of software that is built, rather than licensed, and the Slalom report only led to a recommendation for more research.[35] The Mayor's office is waiting on more information to determine whether the project should be pursued, but the only way to get that information is to begin the planning under clear leadership.

Third, and most critically, there is not a clear project owner, because developing a system is not a clear function of any branch of San Francisco's government. The true momentum for the project stems from the enthusiasm coming from members of the Elections Commission, but the EC does not have the authority to own the project, though they have set up advisory structures to continue moving the needle forward. The EC relies on the Department of Elections to be the project advocate and owner, but software development is a role the department is not structurally suited for. The DoE focuses on its mandate of operating Election Systems, and is reluctant to get into the game of building them. The DoE has largely kicked the can down the road by pointing to the glaring uncertainties of project planning and funding, both of which would need to be resolved before a full project could be committed to.

These misalignments of expertise and decision making capacities have kept the project at a standstill because they have diluted responsibility and knowledge in an environment of risk.

[35] A detailed discussion of the Slalom report is included in Appendix H.

Within this context, previous Mayors have continued to request more information as a way to put off making a definitive decision on the project. To this end, the late Mayor Lee commissioned a report from Slalom Consulting on the project that was delivered in January of 2018.[36] The Slalom report was intended to iron out a clear path forward for the project, but it raised more questions than it resolved, and its largest recommendation was the allocation of $1 million dollars for further analysis. The path of continued indecision is expensive.

The way forward is to establish a clearly responsible project owner who has the enthusiasm and mandate to drive the project onward. Since it is unlikely that a single individual will be able to fully understand and appreciate its complexities, it is advisable that the project owner be structurally supported by the folks across the City with complementary experience. Structural alignment of city agencies is critical for this work to advance, and a well defined owner is a prerequisite for any progress.

[36] This report is discussed in detail in Appendix H.

Conclusion

Developing an Open Source Voting System has the potential to bring the City and County of San Francisco a number of concrete and principled benefits in the long term, including cost savings, increased election security, and public ownership over the critical infrastructure of democracy. It is likely to free the city from the constraints of vendor lock-in, and the accompanying risk of financial exploitation. The value of these long-term benefits needs to be weighed against the security and completion risks that the project must overcome in order to achieve them.

Election system security should be paramount to the design and implementation of the project. The security of the resulting system depends on it being used by multiple jurisdictions, and garnering the attention, respect, and scrutiny of the open source community. That means keeping the finished product in front of as many minds and eyes as possible. Doing this requires deliberate effort to build generic features to support use outside of San Francisco, empower the open source community to engage with the development, and keep the project true to its ideological roots of transparency and availability.

Partnerships will be critical to completing an OSVS successfully. Partnerships with nonprofits could fill in gaps in the city's expertise, or even develop the software in its entirety. Early coordination with regulatory authorities and state officials has the potential to dramatically decrease the cost of the project while shortening the timeline for development and certification. Finally the city should look to extend partnership opportunities to other counties within California to share costs and offer an election platform under a single certification authority.

The complexity of the proposed system and the need for the efforts of a wide range of city stakeholders make it paramount that a small set of responsible persons be brought together to own structural decisions necessary to advance the program, and an explicit role of project owner be assigned to a person who can shepherd the venture forward.

San Francisco clearly stands to benefit if it can develop an open source voting system, but the City is not on track to complete that endeavor. The project does not have a clear advocate nor a logical home within the existing operations of San Francisco government. Excitement for the project is misaligned with authority and funding. All paths that ultimately lead to a successful project require a clarified consolidation of expertise, responsibility and authority, and decisive commitment under strong leadership.

Findings

F1. There is not a clear project owner that is responsible for building an Open Source Voting System in San Francisco, which prevents the project from making any progress.

F2. Progress on the Open Source Voting project has been limited because responsibility has consistently and ambiguously been passed around between organizations without a clear source of funding or a mandate for completion.

F3. Progress on the Open Source Voting project has been slow because of the large number of stakeholders, and the dispersal of their expertise, and the uncertainty each party has about the overall project.

F4. Progress on the Open Source Voting project has been slow because all parties are appropriately concerned about security, and few within San Francisco government have the technical background to accurately evaluate security concerns.

F5. Today, only one company can operate California certified Ranked Choice Voting Elections - Dominion Election Systems. San Francisco has a continuing legal obligation to purchase systems from Dominion, regardless of cost or competitiveness, due to county RCV rules, restrictions on procurement due to LGBT discrimination in other states, and state certification requirements.

F6. The operational cost charged by Dominion Systems increased from 1.1 million per year to 2 million per year between the contracts from 2006 to 2018 and 2018 onward. San Francisco did not have a viable alternative to accepting this price increase.

F7. The California counties that use Ranked Choice Voting are in the same financial predicament as San Francisco when it comes to procuring their voting system software. This makes them ideal partnership candidates, as they face the same set of challenges under the same regulatory authority.

F8. Too many variables remain unresolved to draw confident analysis about completion cost or timeline of the OSV project.

F9. Though certification by the California Secretary of State is an indication that an election system is reasonably secure, certification does not guarantee election system security.

F10. The security of an Open Source Voting System would reflect the ratio of the number of good actors to bad actors that are looking at it to find vulnerabilities, which makes getting the attention of external security experts a top level priority for the OSV project.

F11. If an Open Source Voting system is going to be used only by San Francisco, it is unlikely to attract the requisite attention of security experts and white-hat engineers

necessary to be confident in its security.

F12. The ability to efficiently patch vulnerabilities in open source software is a foundational property of successful and secure open source projects, and certification by the Secretary of State poses an unscoped period of delay to any patch to an OSVS system.

F13. Although patches to open source systems are common, any patch of an election system will necessitate recertification by the California Secretary of State. The timeline and cost of this recertification can vary wildly depending on the size of the fix, and its urgency. There is some evidence that modular certification can be supported by the Secretary of State.

F14. There are a large number of non-profit organizations that are willing and eager to help develop an OSV system, as both developers and advisors.

F15. Federal agencies specializing in developing reusable Open Source Technologies, such as the USDS and 18F, are ideal partnership candidates for an OSV project, but their involvement would require that some federal funds be used for the project.

F16. No organization within San Francisco government has formed formal partnerships with non-profit organizations to develop, test, or to advise on OSVS best practices.

F17. No organization within San Francisco government has begun formal discussions with the Secretary of State about the potential for partnership.

F18. The Department of Elections has familiarity with the election system certification process, as most recently demonstrated by their work with Dominion in 2017 to get a patch for the “spectre/meltdown” bugs certified by the California Secretary of State.

F19. Developing Election Systems is currently outside of the mandate for San Francisco's Department of Elections.

F20. San Francisco's Department of Elections has no experience developing critical software.

F21. San Francisco's Department of Technology has demonstrated willingness to undertake open source projects.

F22. San Francisco's Department of Technology does not have extensive experience developing open source technology that is in use beyond San Francisco.

Recommendations

The San Francisco Civil Grand Jury:

R1. Recommends that the Mayor include funding in their next budgeting cycle to hire a "Program Manager" dedicated to shepherd the project forward and own the project. Regardless of the department they report to, the Program Manager will be responsible for communicating with collaborating jurisdictions, engaging experts, managing and tracking project risks, and establishing cost and timeline targets. The Program Manager would need qualifications in technology management, design thinking, and procurement. Funding should be allocated for this process in the next budget cycle. (F1, F2, F3, F8)

R2. Recommends the Mayor's Office set up a working group responsible to centralize the expertise relevant for the OSV project and approve structural decisions made by the Program Manager. The working group should contain (at minimum) a representative from the Mayor's office, DoE, OSVTAC, COIT, and DoT.[37] After planning completes, funding requests for the OSVS would be recommended to the working group by the Program Manager, and would then be recommended to the Mayor for inclusion in the city budget. This group should be formally constructed by October 1, 2018, and should begin a hiring process for a Program Manager as soon as funding is allocated. (F2, F3, F4)

R3. Recommends the Election Commission's OSVTAC should organize and maintain a website to serve as an informational portal on the OSV project. This should include links to (and summaries of) all reports written on the subject (including by the SoS, EC, OSVTAC, CGJ, Slalom, BoS). This resource should be completed by October, 1 2018, and be updated consistently. (F2, F3)

R4. Recommends publishing a quarterly summary of the state of the OSV project. The report should include: an estimate of the completion date, current cost projections, and highlight emerging issues. Until a Program Manager is hired, the reports should be authored by the EC, and afterwards, the report should be authored by the program manager. Reports should commence October 1, 2018, and continue at the start of each quarter until project completion. (F2, F3)

R5. Recommends the Office of the Controller set up a process to trigger review of city RFPs that only receive one bidder, and, when feasible, perform a market analysis to determine why the procurement process has not induced participation of additional vendors. This process should be in place by April 1, 2019. (F5, F6)

[37] The DoE would function as the expert on election administration and certification; the DoT as the expert on open source software and technology development; COIT for weighing city funding priorities, and the OSVTAC as the expert on open source election software.

R6. Recommends the Office of the Controller evaluate the premium San Francisco pays for its Voting System compared to (1) the price paid by other California counties that use Ranked Choice Voting, and (2) the price paid by California counties that do not use RCV, and (3) the price paid by cities/counties outside of California who use RCV. This analysis should be published by April 1, 2019. (F5, F6)

R7. Recommends that the DoT not directly build the software for an Open Source Voting system in the near future, because they have not demonstrated the in-house capacity to tackle a software development task of this magnitude. (F21, F22)

R8. Recommends that the DoE not directly build the software for an Open Source Voting system in the near future, because they lack in-house critical faculties and experience in software development. (F19, F20)

R9. Recommends that San Francisco's Elections Commission conduct a systematic evaluation of partner interest in using the OSV system developed in SF. This evaluation should reach out to all Departments of Elections in all counties within California, focusing on potential use and cost sharing. This analysis and reporting should be completed by April 1st, 2019. (F7, F9, F10, F11)

R10. Recommends that the Department of Elections evaluate the possibility of incorporating 2018 HAVA funding into the development of the OSV system, so that federal technology agencies have jurisdiction to help develop the project. The feasibility of this should be formally evaluated and published by the Department of Elections by January 1st, 2019. (F15)

R11. Recommends that the Department of Elections, along with the Election Commission, reach out to 18F and the USDS to evaluate a possible partnership to build the OSV system with them. These communications should be issued by October 1st, 2018, and the results of those inquiries should be made publicly available after discussion concludes. (F14, F15)

R12. Recommends that the Elections Commission establish a coalition of supportive non-profit organizations in a formal structure to support the project. This list of collaborators and contacts should be constructed and published by January 1st, 2019. (F14, F16)

R13. Recommends that the Department of Elections, working with the Elections Commission, establish a Memorandum of Understanding with the California Secretary of State that addresses how the California certification process will accommodate modular development and vulnerability patches, to align the SoS's process with open source best practices. The discussion of this memo should begin by January 1st, 2019. (F7, F12, F13, F17, F18)

Required Responses

Pursuant to Penal Code section 933. The San Francisco Civil Grand Jury requests responses as follows:

From the following individuals:

Mayor of San Francisco
(F1, F2, F3, F5, F6, F7)
(R1, R2)

Director of the San Francisco Department of Elections
(F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20)
(R8, R10, R11, R13)

President of the San Francisco Elections Commission
(F1, F2, F3, F4, F7, F8, F10, F11, F12, F13, F14, F15, F16, F17, F19, F20)
(R3, R4, R9, R11, R12, R13)

Chief Information Officer Department of Technology San Francisco
(F10, F12, F21, F22)
(R7)

Controller, San Francisco Office of the Controller
(R5, R6)

Reports issued by the Grand Jury do not identify individuals interviewed. Penal Code section 929 requires that reports of the Grand Jury not contain the name of any person or facts leading to the identity of any person who provides information to the Grand Jury.

Appendix A: Voting Systems

A voting system is an umbrella term for the hardware, software, and processes used to conduct an election. Usually, voting systems are licensed and sold as a single unit, comprised of a large number of dependent and opaque proprietary components. Most election systems do not include tools for voter registration and management, but have everything beyond that point requisite to cast, collect, tally, report, and recount votes. Election systems are sold and discussed as wholes, but are comprised of an orchestration of software running on hardware, in combination with detailed logistics. Each component needs to be carefully designed to be interoperable with the other components in order to conduct a smooth and secure election for voters and election administrators. California requires counties to have an election system that fits three sets of needs, though the same system can sometimes fulfill more than one set of requirements.

The first need is a "polling place" system, which includes the software that designs and prints ballots (frequently in multiple languages), the paper ballots themselves, the system used in the polling place for marking votes, the receipt generator that creates a permanent paper record of the vote as cast, and an optical scanner that performs initial tabulation of precinct level results. Additionally, the polling place system typically includes computers, ballot boxes, warranties, and manuals that are required to work with other components to operate the system successfully.

The second need California requires its counties to fulfill is a vote by mail (VBM) system. This system includes the software that designs ballots that can be mailed to voters, the actual delivery of the printed ballots, the association of a ballot ID with the registered voter that it is sent to, signature comparison software, tools providing visibility into the ballot as it is processed through the system, and the scanners that process the ballot into a digital format that can be tabulated alongside the polling place results. Vote by mail systems are heavily used in California, and many counties opt to use the VBM system in lieu of a polling place system.

The third need is for an "accessible" voting system, which accommodate a wide array of abilities while still enabling a voter to mark and submit a ballot. These systems tend to be similar to the polling place systems, but have a more flexible vote marking component, which enables those with a wide range of physical disabilities to mark and cast votes.

These three constituent needs are required by California law, but to see how these broad pieces get broken down into itemized costs and technologies, check out a bid submitted by a voting system provider to cover all three systems.[38] It includes things like power cord holders and remote transfer software that folks don't typically think of, but that are essential to conducting a successful election.

[38] A 2017 bid on a complete voting system, submitted by Dominion voting systems (SF's current provider).

Appendix B: Open Source Software

Source code (the text that generates software) is governed by copyright law, just like books, movies, and art. Most source code is accompanied by a license that describes how the code may be used, and who controls it. As with other creative works, most source code licenses are proprietary, aimed at maintaining control and profitability over the work for the owner of the intellectual property. However, software licenses place additional constraints over those we typically think of around other types of creative works.

Proprietary software licenses usually prohibit any modification of the software and restrict unpaid usage, in order to maximize the creator's control over the use and distribution of their work. We have all encountered these licenses when we have downloaded software, and agreed to their "terms and conditions". Copyright exists to give a creator control over the fruits of their creative labor, ensuring others cannot duplicate, sell, or misappropriate their work. licenses are the way that the terms of this control are established. Open Source Software exists outside this traditional mechanism of creating and disseminating software.

Open Source Software, refers to software governed by one of many licenses that offer liberal use and modification restrictions; enabling the source code to be viewed, copied, and run by the public. By distributing a work under an open source license, authors choose to explicitly forgo most of the controls given by a proprietary license—waiving rights to control the work and how it can be modified, copied, presented, and eventually, run. Typically, open source software is free to all to be inspected, downloaded, changed, used, built upon, and repurposed, though some restrictions still remain depending on the specifics of the license.

In contrast to proprietary software development, where the main motivation for the production of the software is to generate revenue to recoup development costs, and a make a return on the investment of creating the software, the motivations of individuals and organizations to build and contribute to open source projects are not as black and white. Some for-profit corporations choose to open source their software in the hopes that doing so will lead to its proliferation and widespread use. This often leads to industry-wide standardization, giving the original author or owning organization leadership in the future of the industry, and engaging a community of folks to further develop and advance the software. However, many other open source projects are maintained by non-profits and individuals motivated by the idea that software should be ubiquitous, secure, and free.

Open source software is everywhere. The world's most popular operating systems (Linux for servers, and Android for smartphones) are both open source projects. Companies like Amazon, Apple, Google and Facebook open-source components of their core systems to encourage standardization. Government organizations like NASA, the Department of Defense and the CIA strategically open source projects that they hope will get attention and community contributions.

Appendix C: Properties of Open Source Software

There are a number of benefits that typically accompany open source software development, as well as a number of risks. We will give a brief discussion of the properties that are most relevant to the development of an open source voting system.

The first significant benefit to open source software is that software can be reused. Rather than having multiple organizations and communities write code which accomplishes the same basic tasks, an Open Source project that serves some fundamental purpose saves subsequent engineers the necessity of rebuilding the same components. Rather than starting from scratch, software developers are able to spend less time on fundamental systems and more time on the pieces of code that are specific to their task.

A second benefit is the emergence of an ecosystem of interoperable pieces of software. Many of the most used pieces of software in today's world are open source projects that perform well-defined tasks. Over time, this set of software grows, and new systems are built which rely on previous systems. This leads to a long term trend of an expanding set of compatible open source projects.

A third benefit to open source projects is the pattern of "copy-left"—a pun on "copyright". In a copy-left software license, anyone is free to use and modify the source code, but any modifications need to be made available under the same copy-left license: any project based off of a copy-left project must have a copy-left license. This has the benefit that all future features and patches to the software are legally required to be available to everyone, causing a virtuous cycle of development and availability extending beyond the initial software development.

A fourth benefit to open source software is that when an open source project is used by many organizations and individuals, a larger set of folks care about maintaining the health and security of the source code. The more attention given, the more secure the project tends to be. However, for the same reason, a lack of use and attention can be a liability to an open source project. This idea is discussed in depth in Appendix D.

There are also risks to developing software as open source. The primary risk is that the software will not receive sufficient attention and scrutiny, and this makes the software more vulnerable to attack and abuse. This risk is discussed in depth in Appendix D.

Appendix D: Security within Open Source Software

A vulnerability is a defect in a computer system (software or hardware) that weakens the security of the system. A vulnerability does not necessarily provide an attacker a way of controlling the system or seeing what it is doing. However, by weakening the security guarantees of the system, it may allow the system to be manipulated in the future, possibly when exploited in combination with other vulnerabilities. Vulnerabilities are frequently found in both software and hardware, usually by “white-hat” engineers (hackers searching for vulnerabilities to fix them), including folks working in academia, corporations, and governments who inspect source code, hardware, and software for vulnerabilities. Vulnerabilities are incredibly common. There are over a dozen “critical” vulnerabilities found every year in the operating system Linux (the most widespread piece of software in the world).

A patch is a change to source code that fixes any sort of problem within the code. Within the context of security, a patch eliminates or mitigates a vulnerability. When a vulnerability is reported for a piece of open source software, the owner of the project usually writes a patch to mitigate the vulnerability, or incorporates an existing patch from elsewhere into their project.

Since the source code in an open source project is available for anybody to inspect, it becomes easier for anyone to find vulnerabilities in the source code and potentially exploit those vulnerabilities. Source code for proprietary software is difficult to obtain and is less transparent, which adds a hurdle to understanding and manipulating it. Open Source Software is easier for everyone to find vulnerabilities in - both hackers and security researchers. The same openness that enables bad actors to find vulnerabilities in code makes it easier for good actors to find the same vulnerabilities, and patch them before they can be exploited.

When code is universally available to inspect, the competition between exploiters and fixers is won by the side with the larger number of experts. Open source projects that are widely used and monitored are some of the most secure pieces of software in the world. Correspondingly, open source projects that do not receive much attention from security minded engineers are likely some of the most vulnerable in the world to attack. The difference between these two outcomes is entirely based on the rigor with which the open source software is scrutinized for vulnerabilities, and that is typically determined by how many folks use the project, and the criticality of the role it plays when used.

The other thing that determines open source project security is the speed with which patches can be incorporated into the operational software. Once a vulnerability is announced/discovered for a piece of software, all dependent projects are exposed to the vulnerability until they incorporate the patch. Sometimes, project configuration and requirements make it difficult to perform quick patches, and these systems are less secure than their speedily-deployed counterparts because there is a longer period between the notification of a vulnerability and the time when a patch can be deployed.

Understanding and incorporating these findings into the development of an open source election system is critical to the project being sufficiently secure to use as a piece of democratic infrastructure.

Appendix E: Components of an Open Source Voting System

Now that we have covered the basics of open source software (in Appendix D) and voting systems (in Appendix A), and key terms around software development (in the Glossary) we can begin in earnest by describing what an Open Source Voting system would look like, and what it would mean for both San Francisco, and other cities and counties.

An Open Source Voting System would run on “commodity” hardware. Commodity refers to a non-specialized set of computers that are available for anyone to purchase. Commodity hardware is sometimes described as “off-the-shelf” because it is hardware that one could imagine buying at a non-specialized retailer. Software created to run on commodity hardware must be written in a way that it can function on any machine that meets some standard set of specifications. The computers, scanners, and printers necessary to run an election in an OSVS would all be commodity hardware. Because the hardware is broadly available and the software and instructions are free online, anyone could buy these generic machines and run the election system software on them.

Commodity hardware is cheaper and more available than specialized hardware, enabling the voting system to be replaced and replicated easier. This means that under an OSVS, the machines used to conduct an election could be upgraded or replaced without any need to select a different software to run on the new hardware. Writing software without thinking about hardware (called “hardware agnostic” development) is a best practice in modern software development. Rather than tailoring the software for the machine, you build it on top of layers of system “abstraction,” enabling you to assume that any machine running the software has a basic set of properties, freeing your application to be unconcerned with the minutia and idiosyncrasies of your hardware.

In a similar vein to the way it would interact with hardware, an Open Source Voting System would also operate over standard, widely used and tested pieces of foundational software. This would likely mean running on the Linux operating system, an open source operating system used on the majority of the world’s servers, and standard I/O “drivers”, software that allows software to communicate with other hardware such as scanners and printers.

The code that conducts the elections using these standard components would be the “open source voting software” within the open source election system. This software would be a collection of interdependent projects, each of which would accomplish a small, well-defined task. Ballot creation software would generate the layout and content of the ballots for printing, optical scanning software would convert paper ballots into a digital form, tabulation software would convert digitally representations of ballots into coherent results.

It is a common mistake to just look at a voting system as just these two parts - software running on hardware. Equally important is the final component of an open source election system - a comprehensive set of instructions and logistics. This must include details of how to set up the software to run on the machines, how the different pieces of hardware would have to be configured, assembled, tested, integrated, protected, and operated. Though this may sound like a small component of the system, its importance to running a secure, efficient and trustworthy election should not be underestimated. Errors in these procedures (more than errors in software or hardware) expose an election system to abuse, fraud and error.

If an OSVS is developed by San Francisco, it could be used anywhere. Since all pieces of an open source election system could be easily purchased (hardware) or copied (software and instructions) by any other jurisdiction, other counties, states, and nations could conduct elections using this system at a small cost. Rather than building a similar system of their own, or procuring a closed source system for millions of dollars per year from one of the Voting System vendors, they would be able to use San Francisco's system to conduct their elections at the raw cost of materials and labor necessary to setup and operate the system.

Presently, the election system contract with Dominion systems provides support to the Department of Elections that covers maintenance and servicing of the system, in addition to helping the department with some of the setup of the system. If San Francisco developed its own OSVS, this type of support would likely need to be strategically replaced, a component of the project that has not been evaluated thus far. This is the one component of an OSVS that would not be clearly replicable across jurisdictions.

Appendix F: Certification of a Voting System

Election systems must be certified by the California Secretary of State before they can be used to conduct an election in California.[39] Unlike other states, where certification is closely tied to Federal guidelines, the California Secretary of State considers certification by a federal lab in its application, but does not require it for certification. California's certification process is primarily a checklist of best practices and tests, and is not a comprehensive security or correctness test. Applicants must pay a $400,000 deposit, and wait up to 12 months for the certification result.[40] To perform the certification analysis, California may run a set of undisclosed tests, and may hire an election security expert to inspect the software, hardware, and operations for vulnerabilities.[41] These security experts are the only ones who have the access to audit the software, as the source code of these systems is not otherwise disclosed.

Certification happens at a "snapshot" of software and hardware. This means that the system is certified at a single configuration of the code, processes and devices, and any changes to these components beyond that snapshot require recertification.[42] This property of certification requires discrete "versioning" of software, such that new features, security patches, and performance improvements must be bundled together into a release to be recertified. Because certification is so expensive (and because there are few changes to election systems over time), certification is infrequent, and the major providers of voting systems within California tend to only recertify a new version of their flagship products every 2 to 6 years.

Additionally, the certification process can take a long time to return a result. In order to be certified for use in an election, the software needs to be submitted at least a year in advance, meaning that even slight changes to the software will require a long time to reach the systems in use by voters. However, smaller changes (or changes on a tight deadline) have been given expedient reviews in the past. Large uncertainty and a lack of clear expectations in this domain make it challenging for an OSV project to estimate what a patching model looks like within this uncertain environment.

Certification poses a significant barrier to entry into the voting system market. Only eight vendors have ever certified an election system in California, and only five of those vendors

[39] Current regulations that govern the certification of voting systems in California are provided here.

[40] California Voting System Standards, California Secretary of State, October 2014.

[41] For a discussion of what a vulnerability is, check out the discussion of Open Source Security in Appendix D, or see its definition in the glossary.

[42] Changes or Modifications to a Certified Voting System, Section 20707 of Voting System Certification Regulations.

currently operate election systems within the state.[43] [44] [45] One of those five vendors is Los Angeles County (which has developed its own voting system, and has been certified by the state). Another is Sequoia systems, which was acquired by Dominion Election Systems, meaning that there are only three corporations and one municipality that own certified election systems within the state of California: Dominion, ES&S and Hart, and the County of Los Angeles.

[43] Voting Systems in use for the November 6th 2016 Presidential General Election in California Counties, California Secretary of State, November 6th 2016.

[44] An enumeration of all vendors ever certified with the California SoS is here. Note that many of these vendors have since consolidated under ES&S, Hart, and Dominion Systems.

[45] Voting Technologies Approved for Use In California, May 21st, 2018, California Secretary of State.

Appendix G: Election Systems Vendors

The market for election system technology only has a small number of vendors. This market characteristic largely arises from its product, which is stable, replicable, and security focused. These properties lead to a small number of highly consolidated firms that offer a stagnant and standardized product with limited transparency.

The dominant characteristic of the market for election systems is that security and reliability. In both reputation and certification, these pose a significant barrier to entry for new vendors. Most RFPs request that a prospective vendor have five to ten years of experience conducting elections with their software before it can be considered for purchase. Additionally, new vendors need to have their systems certified by state and federal authorities, requiring significant time and money. Moreover, state certifications frequently rely on national certification labs that have tight ties to the existing vendors and opaque certification processes. These reputation and security considerations pose a prohibitive barrier for new market participants, and results in entrenched market leaders.

Additionally, election software and systems are fairly uniform across jurisdictions that require them. Most elections operate under similar sets of rules and procedures. Once written, software and systems can be deployed and used in any number of jurisdictions at a low marginal cost to the election system provider. This also leads to a natural economy of scale, where only a small number of providers exists, and only a small number of election systems are in use.

Additionally, elections have not changed dramatically over time. Unlike many other technology markets, a simple and time-tested Election System is usually preferable over a cutting edge solution. This is because the fundamental set of problems of election administration has not changed dramatically over time, and changes are viewed as risks to both security and political optics. This lack of innovation limits what new market participants can offer over entrenched market leaders.

Finally, this is a market that has limited potential for expansion or contraction. The number of jurisdictions operating democratic elections changes very little over time, and the zero-sum nature of the market offers limited upside for new election system providers.

These characteristics of the product lead to a market dominated by a small number of privately-owned firms, where long-term contracts licensing legacy software are common. Today, there are three election system providers that have 92% percent of market share for conducting elections within the United States[46]. They are Dominion Election Systems (Dominion), Hart InterCivic (Hart), and Election Systems and Software (ES&S). Each is privately owned and do not disclose

[46] "The Business of Voting, Market Structure and Innovation in the Election Technology Industry", a report out of the Wharton Public Policy Initiative, provided lots of the data within this appendix (Appendix G).

comprehensive financial information. Though the number of system providers has been higher in the past, a series of acquisitions and mergers have consolidated the market participants. Consolidations have tended to happen at the state-market level, such that pre-consolidation a given state might have two participants in the market, and after the consolidation, the state might only have one.

For example, in 2009 there were six vendors with active contracts for election systems in California.[47] However, two of these providers were acquired after the contracts began, bringing the total number of election systems providers for the state to match that of the majority of the country, at three.[48]

This pattern of acquisitions that limit market competition has not gone unnoticed by federal regulators. In 2010, ES&S' acquisition of Diebold/Premier (itself the result of a merger) would have led to such a reduction in the set of available vendors across all states that it was ruled a violation of federal antitrust rules, and ES&S was forced to sell Premier (one half of the acquired company) to Dominion, another giant within election technologies.[49] Even when antitrust law has been invoked, it has not had the effect of supporting a broad ecosystem of vendors. Instead, it has simply distributed market share more evenly among the small number of big providers.

These vendors have not effectively adapted to market needs. Though elections have changed very little over time, occasional innovations like Ranked Choice Voting have popped up which change election system requirements. In the case of RCV, the major providers did not come forward with a plan to support the new system, and instead it was a small independent provider (Sequoia systems) that worked with San Francisco to deliver a certified RCV system in 2007. Sequoia was purchased by Dominion in 2010, and no other RCV systems have had significant market utilization since.

[47] The list of voting systems currently in use in California is provided here.

[48] An overview of these consolidations is on page 17 of the Warton report listed in the footnote above.

[49] US v. Election Systems and Software, Summary Judgement, 2010.

Appendix H: Previous Reports on Open Source Voting

Lots of ink has been spilled on open source voting, much of which has evaluated the project in San Francisco. The large spate of reports track the public interest in the San Francisco Open Source Voting project, and the governmental uncertainty of the project. Though each report that we have reviewed has informed this one to various degrees, four are included here as further reading. These five are covered because they were officially sanctioned by state and county officials, and the broader set (which includes reports from academia, non-profits and think-tanks) tends to offer more opinion and fewer concrete details.

At the urging of the California Assembly in 2004, in 2006 the California Secretary of State produced a report on OSV entitled "Open Source Software in Voting Systems".[50] It focuses on the security of open source software and the current state of both voting systems and open source technologies. One of the more interesting sections is the discussion of the repeatable/iterative problem nature that tends to accompany successful open source software, and the evaluation of claims around market forces adding value through innovation. The report cautions against the mandated development of an OSVS at the state level, but leaves open the possibility that an OSV might be preferable in the future if the overall landscape of software development and security engineering shifts.

In 2011, the San Francisco Voting System Task Force (VSTF) produced a formidable report outlining future options for election systems in San Francisco.[51] The report laid out a wide range of different directions the city could move toward in order to sustainably conduct its elections. These options detail organizational structures within the city and in its relationships with vendors that are insightful and comprehensive. In particular, the analysis given on contrasting open source voting system development and continued procurement informed our findings and recommendations.

In 2015 the San Francisco Local Agency Formation Commission (LAFCo) produced a summary report of the broad technical and security claims made by proponents and opponents of an Open Source Voting System.[52] The report approaches the subject matter impartially through the lens of evaluating the claims on both sides of the debate. Many of the findings in this report are informed through this objective analysis of the structure of the problem, and the report fairly

[50] Open Source Software in Voting Systems, California Secretary of State, 2006

[51] Recommendations on Voting Systems for the City and County of San Francisco. San Francisco Voting Systems Task Force, June 2011.

[52] Study on Open Source Voting Systems. Jason Friend and Angela Lee, San Francisco Local Agency Formation Commission (LAFCO), May 13th, 2015.

evaluates competing narratives and ideologies in a way that gives the reader a better sense of the general problems around OSV.

In 2017, the Department of Elections received funding from the Board of Supervisors to request an outside consulting firm construct a "feasibility assessment" or "business case" for the Open Source Voting project. This request was undertaken at the direct request of the Mayor. The resulting report, created by Slalom Consulting and released in January of 2018, is one of the most helpful and insightful documents we saw on the topic.[53] The document details how organizational structures of the city's interaction with external vendors is likely to shape the outcome of the project, and performs a comprehensive analysis of existing city competencies and capabilities. Basing recommendations off of existing capabilities is an effective lens to view the project through, and the report as a whole lays out some critical ideas around city capabilities and competencies. However we found two flaws with the Slalom report. The first is the pervasive treatment of the city as a cohesive and isolated unit, without consideration of how internal structures and external partnerships can shape the outcomes of the project. The second is that the report was intended (in the RFP) to propose a way forward for the city, but it introduced more questions than it answered. The main recommendation of the report was to do more research, which is of only marginal assistance to the city. With these caveats, the Slalom report is the most important document thus far written about the OSV project.

The Election Commission's OSVTAC maintains a set of recommendations for the development of an open source voting system, which are consistently updated on the OSVTAC's website (which is itself an open source document).[54] These recommendations are a fabulous look into the technical side of the open source voting project, and they do a good job analyzing the implementation paths and details without considering the complications of politics or funding. The recommendations are, as far as the Jury could establish, well reasoned and largely congruent with best practices of open source software development.

[53] City and County of San Francisco, Open Source Voting System Feasibility Assessment. Slalom Consulting, January 2017

[54] Open Source Voting Technical Advisory Committee Recommendations are here: https://osvtac.github.io/recommendations.

Appendix I: History of OSV outside of San Francisco

San Francisco is not the first municipality to attempt to implement an Open Source Voting system. In our research we have explored the history of the handful of attempts made to develop publicly owned voting systems within the United States, and from each such project we can learn about potential pitfalls that San Francisco may face over the course of this project.

Los Angeles County is in the process of developing its own election system, which started off as an Open Source project. As the project continued on for years without tangible progress, county leaders found it easier to pitch the project to citizens as an investment that would pay dividends through licensing, over a benevolent system with abstract benefits. During development, LA County lost the political will to give away their work for free. The system appears to be on track for use in LA county elections, and has been certified by the secretary of state for use as an accessible polling place system. However its potential outside of the county is limited. LA has publicly said there is the potential for external licensing of the system[55], but shying away from the open source model has cost them the support of open source communities and security experts, who are now significantly less likely to participate in the project. The source code for the project is not available for inspection, and as such, does not have the security benefits the accompany public review.

Travis County, Texas, which contains the City of Austin, had also pursued the idea of developing an open source voting system. However, as the project was beginning its development phase in mid-2017, the project was scrapped because of short term budget shortfall. Notably the project had a long history of building action, a broad base of support from local politicians, but faced similar risk, compliance and funding challenges that the San Francisco project faces. When a budgetary shortfall hit the county in 2017, the OSV project was sidelined in favor of more immediate needs of Travis County citizens. San Francisco should learn from this example and appropriately weigh the decision to develop an OSVS against other budgetary priorities.

The only Open Source Voting system that has been successfully deployed in the United States is “Prime III”, a project out of the University of Florida under the review of Dr. Juan Gilbert. A deployment of Prime III called “one4all” was used as an accessible voting system in New Hampshire during the 2016 presidential primaries.[56] The code for the Prime III system is indeed

[55] However, San Francisco could not license the system as it currently stands because it is only a ballot casting system, and doesn’t integrate with the other components of Sequoia software the city uses. Additionally, the system is not currently equipped for RCV.

[56] Prime III, Verified Voting Project.

open source, and available on Github.[57] [58] However the system's inadequacies and red-flags are clear to even the least informed software engineers. The system doesn't appear to be rigorously developed, reviewed or tested. The project on Github has only had three contributors (the PhD Candidate working on it, the doctor himself, and an unclear third party), and its commits are largely an initial check-in and typo corrections. The system is built in Javascript (which is a notoriously buggy and inscrutable programming language).[59] Moreover, the system doesn't have tests, which should concern anyone with a rudimentary software background. Additionally, Prime III only consists of the software to display and cast ballots, and lacks details around logistics, hardware, implementation and security. This is not a voting system, but a component within one. That the system was certified for use in New Hampshire is an unexplained mystery to the Civil Grand Jury. The Civil Grand Jury sought out information about the future of the system, and how it was developed and certified. Unfortunately, repeated requests for interviews with involved parties did not return responses.

From the New Hampshire project, there is a critical lesson about legitimacy. Unless software is developed through a deliberate process, with development checkpoints, testing, requirements, and review, the system is unlikely to be accepted as legitimate by the open source community and the broader world. Source code that is made public after development does not carry the same trust that a deliberate open source development process engenders. Transparency, engagement, and review are critical to the successful completion of this project.

[57] Github repository for Prime III.

[58] Prime III is open source, but one4all was branched off of Prime III before it was used in New Hampshire, and one4all does not appear to be open source.

[59] The theology of Javascript.

Appendix J: Ranked Choice Voting Election Systems

In 2002, San Franciscans voted to amend the city's charter to use Ranked Choice Voting (RCV) in all city and county elections.[60] This systemic change has presented the San Francisco Department of Elections with a restrictive requirement when procuring an election system, as there are very few providers of RCV-capable election systems. Because few counties in the United States use RCV to conduct their elections, none of the major three providers of election system software actively develop or support RCV in their flagship systems: [61]

- ES&S claims that they could operate RCV elections in California, but this systems would require "modifications" in order to support RCV, and RCV is not offered as a standard product, and their solution is not yet certified.
- Hart claims that they have a complete RCV solution, but it is not clear whether any counties in the United States currently use it.
- Dominion does not support Ranked Choice Voting in their flagship offerings. Instead, they support it through legacy software that they obtained through their acquisition of Sequoia Voting Systems. Sequoia/Dominion is San Francisco's current Election System provider, and the provider for all counties within California that use RCV.

This lack of first-order support and active development makes it unlikely that San Francisco will receive truly competitive bids from these three election system providers. The lack of competition among election system providers was one of the many reasons Governor Jerry Brown worked against a statewide mandate for RCV.[62]

An additional complication is that not all Ranked Choice Voting rules are the same. For example, jurisdictions often vary in the number of candidates a voter can select, in the treatment of write-in candidates, and in the elimination strategies used to tabulate instant runoff results. This adds an additional complexity in either tailoring a RCV solution to San Francisco (in the Case of ES&S or Hart), or using legacy products (in the case of Dominion/Sequoia), by limiting the cohesion of market demand.

---

[60] An overview of San Francisco's history with Ranked Choice Voting is available on the DoE's website: http://sfgov.org/elections/ranked-choice-voting.

[61] Voting Systems and Ranked Choice Voting, FairVote.

[62] Brown vetoes bill to broaden ranked-choice voting in California. September 30th, 2016, SFGATE.

Appendix K: Benefits of a Modular Software Development Model

In software development, modular procurement and delivery (simply called modular development) is a methodology for producing software where a small number of project planners design an overall system architecture, and split up the components of the system into small pieces called modules that have well defined boundaries and expectations. Each of these pieces can be built independently by different organizations or vendors. The implementation and delivery of the modular components can happen in sequence - or on some overlapping timeline. This contrasts with the typical "unitary" model of software development, where a contract specifies product requirements that govern expectations about an end-to-end piece of software, and the vendor (once selected) can produce the software in any way that conforms to the overall product requirements.

We have found consistent support for a modular procurement model from software development experts and practitioners, a recommendation also proffered by the Elections Commission's Open Source Voting Technical Advisory Committee. The justifications we give for this recommendation are:

Provide Quality Checkpoints - modular development enables the project owners (the City) many intermediate checkpoints to evaluate the quality of a developer's work. This is critical for source code that will likely be inspected by both the Secretary of State, the public, security researchers and bad actors. Review of past work would be a critical factor in evaluating vendors for new contracts for new modules.

Provide Timeline Checkpoints - A modular model places less risk on the timeline for the project, because incremental milestones are evaluated against a proposed schedule. This means the City will know further in advance if there is going to be a timeline slippage, and will not have to rush to figure out how to run an election without a functional system if it is not going to be delivered on schedule, enabling a non-emergency contract extension with the existing election system provider.

Parallel Development - Because each of the components within a modular procurement framework has clearly defined and independent product requirements, multiple vendors can work on different components at the same time. This completes the work in parallel, and makes it possible for a functioning system to be delivered faster than if the components were delivered in sequence by the same vendor.

Increasing Vendor Choice - The delivery of a large contract to a new or unproven vendor adds a risk that the vendor will not have the requisite skills and knowledge to deliver the product at all. This is a common but unfortunate pattern in government technology acquisition, most recently highlighted by the Healthcare.Gov fiasco, where an unproven Canadian firm was given an

enormous contract that they were not capable of completing without external assistance. In an incremental development model, this is less concerning, because the implications of one component not being delivered have a lower risk. The project planners can just re-offer the contract to other bidders without every piece of the software development being delayed by vendor failure. This enables newer vendors to enter the bidding process, and enables the project owner the freedom to consider less qualified bids.

Mitigate Certification Risk - An incremental development and feedback cycle enables the project planners to be more confident that the overall system will be certifiable by the secretary of state because there is more time to analyze software, fix mistakes, and choose vendors in between the development of each module. This would ideally be orchestrated with the Secretary of State to figure out an appropriate certification protocol for a modular system.

Appendix L: Advantageous Partnerships for San Francisco

Software is infinitely reproducible. Since source code can be copied just like any other digital file, software can be run on any number of systems when not restricted by a license. This poses an important question for San Francisco as it moves to create its own piece of Open Source Software - what other jurisdictions should the city consider as partners in funding, developing, or using the software? At two ends of the spectrum, it could be designed and built for San Francisco alone, and at the other it could be a cross national effort with partners around the globe. There are tradeoffs to the scope of the partnerships that San Francisco considers.

For one, the broader the target audience of the project, the larger the number of potential contributors, collaborating jurisdictions, and security experts that would consider working on the project, checking it for vulnerabilities or contributing to it financially. As was discussed in Appendix D, the number of folks looking at and using a piece of open source software is a rough analogue for the security of that software, and a proxy for the financial resources a project can glean from interested parties.

However, a broad target audience requires a larger set of features. As an example, San Francisco uses Ranked Choice Voting (something only a handful of other US counties use). A target audience that includes non-ranked choice voting jurisdictions would require building modules that San Francisco will not use in its elections. Additionally, consensus is hard to attain among large groups of stakeholders, which would place a logistical challenge on working with a large number of partners.

In general, if developing software for a broader audience, San Francisco would need to take the peculiarities of other jurisdictions elections operations and certification requirements into account when designing software for the City’s purposes. The two major forces working against scale are compliance requirements and local election law.

One natural boundary for a target audience is one that extends to all of California’s elections. In addition to sharing a certification process, and a potential source of funding (in the California legislature), California has four counties that have some incarnation of Ranked Choice Voting. Any target audience that includes any counties outside of California would require the unenviable task of certifying the system in a variety of jurisdictions. This could mean a single patch to the software would result in a set of many compliance audits across state (and perhaps even country) boundaries. Though this might be the future of the software, a single certification process is certainly preferable while the project is in its initial development.

Appendix M: Unstated Excellence

The Civil Grand Jury initially investigated significantly more topics within the realm of Elections than are laid out in this report. We dove into a range of concerns including signature mismatch invalidations, language and accessibility criteria, voter education efforts, data reporting and consistency issues, and ranked choice voting challenges. On each, we worked to understand the issues deeply, both within San Francisco and without. We gained insight into how the Department of Elections operates through interviews and reams of internal documents. We approached each concern with a deeply skeptical eye, keen on making sure that the DoE was faithfully serving the interests of San Francisco citizens.

This report omits these topics because the Department of Elections is doing a fabulous job. The problems that they face are challenging, and demand versatility and fluency in technological, logistical and interpersonal skills. Our inquiries into almost every element of their operations around elections administration consistently illuminated a department functioning effectively to fulfill the needs of its citizens. The folks we interviewed within the DoE were consistently knowledgeable, experienced, diligent and delightful. They appear to be consistently executing the many challenging roles that they hold with grace and efficiency.

Moreover, the Department’s focus is very clearly on the needs of the voter. Their work to reduce the frequency of votes that don’t count is commendable and should be a model for others to follow. Their education efforts show a clear focus on serving the historically disenfranchised. Their policies are crafted to increase access to voting, and assumes the best intent of the voter.

Finally, the members of the Department show clear drive to always be getting better. In almost every conversation we were blown away by the level of innovation and creative programming that was bubbling up from employees in every role. This gives us further confidence in the future of the Department.

We came into this investigation with a spate of questions and concerns about elections in San Francisco, and over the course of a year we found that most were already mitigated. This is largely because San Francisco has great people conducting our elections.

Glossary

Hardware is an umbrella term for the physical component of computers and other digital devices. Hardware includes things like a laptop or a scanner, but would not include the applications running on the laptop, or the drivers used to operate the scanner.

Software is a term for the non-physical components (such as applications and programs) that run on physical devices. Software includes things like operating systems (Windows or Mac OS), but does not describe the physical device that a user would use to interact with them.

Source Code is the text that can create software: Source code "generates" software, software "runs on" hardware. Because source code is just like any other digital file, it is possible to copy it and run it on multiple machines. This infinite reproducibility means source code is regulated much like other creative works (books, movies, etc.) - it can be copyrighted and its use can be governed by the license that it is released under.

A Version Control System is a mechanism to allow multiple folks to work on the same source code for a piece of software. It enables multiple people to change files simultaneously, and gives a history of how the software was modified over time. Version Control Systems are common in Open Source Software Development because they enable outsiders to see both the history of the code, and potentially contribute new changes.

A Commit is a unified set of changes to Source Code, usually adding one feature, fixing one bug, or any other incremental, well encapsulated piece of work. A Version Control System usually has tools around authoring, editing, reviewing, and approving commits. Commits require the approval of a project owner or moderator.

A Software License is a legal document governing how software and source code can be used, distributed, modified, inspected and repurposed. Software licenses are usually located alongside a piece of source code, and almost all pieces of software have a well defined license.

An Open Source license is an umbrella term for software that is licensed with some combination of characteristics. The universal feature of open source software is that the source code is available for anyone to inspect. Other common, but non-universal, characteristics include the ability to download the source code and run it for free, the ability to modify the code and redistribute it, and the distributed ownership of the software among a large set of contributors and editors.

Open Source Software is a general term for source code and software that is licensed under an Open Source license.

Disclosed Source Software is an umbrella term that includes all software where the source code is free for anyone to inspect. This includes software that is not free for anyone to use, and

software that is not free to modify. Disclosed Source Code helps users of an application have confidence in its correctness, and can have security benefits and drawbacks, just like open source code.

Copy-Left is a property of a software license that requires that any derived works of the original software are accompanied and provided under the same (or a similar) license. This means that if project A has a copy-left provision in its software license, and project B takes Project A and makes changes to it, that project B must publish its source code available under the same license that project A was licensed under.

A Vulnerability is a defect in a computer system (software or hardware) that weakens the security guarantees about that computer system. A Vulnerability does not necessarily provide an attacker a way of controlling the system or seeing what it is doing, but it leaves open "vectors of attack" through which flaws might potentially be exploited. Finding vulnerabilities in both software and hardware is common.

A Patch to source code is a change (or commit). Within the context we are discussing them, a patch closes a vulnerability in software.

A Voting System (also sometimes called an Election System) is any collection of hardware and software that can be used to operate an election. This usually includes many pieces of hardware (ballot creation systems, ballot marking tools, ballot scanners, and the computers that run each component), and pieces of software (voter registration systems, ballot tabulation software, optical character recognition for write ins), in addition to logistical instructions + safeguards that prevent tampering. Folks usually refer to Voting Systems as a whole because these components (hardware, software, and operational support) are typically sold and licensed in large purchasing agreements between election system providers and counties. An example of a bid for a voting system contract is shown here.[63]

Election System Providers are the small number of corporations that sell and license voting systems to governments at all levels of government within the United States.

[63] Dominion Bid for an Accessible Voting System in New York State. August 28th, 2017.