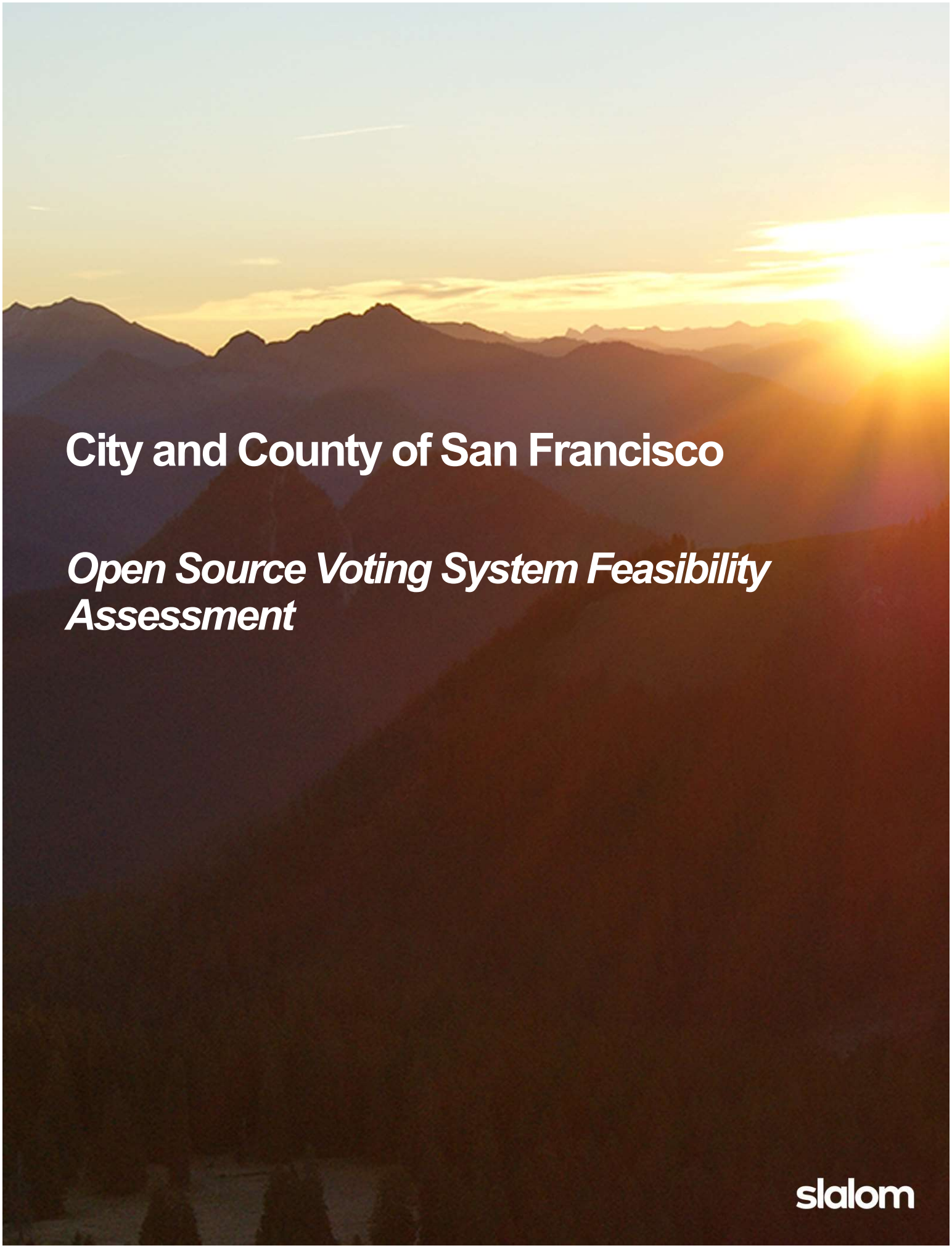


City and County of San Francisco

Open Source Voting System Feasibility Assessment

slalom

0 Table of Contents

1 Introduction

The City and County of San Francisco (City) is considering the feasibility of its options for developing a highly accessible, open source voting system (System), and the costs and time frames associated with those options.

This report evaluates the feasibility of developing an open source voting system and highlights the risks, development options and the capabilities the City has to undertake its development.

Given these considerations, the City will have to evaluate how the opportunity to build this system balances out against the risk and investment it will take to do so. Furthermore, the City will need to evaluate its own willingness and ability to take on the challenges in building such a System.

1.1 Purpose and Organization of this Document

The purpose of this document is to inform the City of the feasibility of its options to create a highly accessible, open source voting system and the costs and time frames associated with the options

This document intends to:

- Lay out the scope of voting system to help frame the assessment and create cost estimates
- Explain the assumptions and considerations relative to Open Source software development, program delivery, system build, certification, run, and maintain phases
- Explain the required capability model required to successfully execute this project
- Confirm the options that were assessed, the methodology of assessment, and the evidence-based approach to evaluating those options
- Summarize the costs, timelines, and risks associated with each option
- Provide a set of actionable next steps for the project sponsor

This report does **not** intend to:

- Be a detailed design document for a voting system
- Make recommendations on specific partners, vendors, or technologies
- Provide a definitive option recommendation to the City and County of San Francisco
- Discuss the possibility of legislative change at the state or federal level

1.2 How to navigate this document

This report has four main ‘chapters’:

- **Section 1** summarizes the objectives, scope and methodology of the feasibility assessment
- **Sections 2 – 6** put the findings of the report front and center and include the summary analysis from the point of view of risk, cost, and time. These are the sections to focus on to understand the data and considerations when making a decision on this project.
- **Sections 7 – 9** are as educational/context setting and explain the considerations from a delivery, system build, and system maintenance perspective.
- **Sections 10 – 13** are the appendices that comprise more detailed information relating to system scope and capability model evaluation

1.3 Objectives of Assessment

This assessment has been created with the principles highlighted in the RFP. The critical goals that the project team had in mind throughout this assessment were as follows:

- Build trust in the accuracy of the election by creating a transparent and auditable end-to-end voting process

- Provide a voting experience that has a high level of accessibility and usability above minimum requirements for voting systems

Additionally, the project team was aware of the following sub-goal: to provide a base level of education to the reader on the complexities and implications of undertaking this project, and provide any suggestions that will make the undertaking of this project smoother.

1.4 Scope and Methodology of Assessment

The complexity of assessing the feasibility of an open source voting system first required an understanding of the high-level capabilities of a voting system to help create a baseline. It is from this high-level understanding that options of an open source voting system were extrapolated and evaluated across the dimensions of: Capability, Cost, Time, Risk.

A total of six (6) options were evaluated across the dimensions of: Capability, Cost, Time, Risk based on the research conducted to understand the high-level capability of a voting system.

This consisted of two phases. First, a review of laws, examination of existing systems and current trends in voting systems was conducted to help the project team define the scope of a 'voting system'. Next, this research was triangulated with the federal Elections Assistance Commission and the California Secretary of State to confirm the scope. This enabled the project team to define what a system would comprise so that time, cost, risk, and required capabilities could be estimated.

Over forty (40) interviews were held with stakeholders within the City's Election Department, technology departments, and special interest groups to understand the various group's ability to deliver a project of this nature and scale.

1.5 Critical Assumptions

In summary, the project assumes the following:

Strategy & Design

- No need to debate the explicit benefits of Open Source vs. Closed Source in this assessment; do need to clarify and explain all the risk and considerations of undertaking an open source strategy
- New voting system must comply with all voting system regulations and accessibility requirements
- When assessing delivery capability of each option, this is defined as a combination of track record, skillset, capacity/resource, and desire to own
- City of SF will continue to be a paper-based jurisdiction (the paper ballots will remain)
- The same number of polling places will continue to be utilized
- There will be ~double the number of Accessible Voting Device systems for use in precincts
- A critical set of 'start criteria' must be met in order to the project to initiate – see section 2.2 for the proposed list of these
- A coordinated discovery design phase will be completed that will confirm the overall architecture of the solution so that subsequent phase addressing the sub-systems will be in harmony.
- Accessibility analysis must be part of the overall program throughout the timeline i.e. during the design, the development, the testing, the rollout, and ongoing enhancements
- There is a requirement for all things to be accessible i.e. not just the system, but also the documentation/test scripts, reports, design docs etc. For example if system documentation is provided to the Open Source community it will need to be accessible via a screen reader for accessibility.

Development

- City of SF to 'own' everything and use Open Source community to increase quality/transparency.
- It is not the intent of the Open Source community to reduce cost or increase velocity of project (at least initially).
- It is not expected that coding will be done by the community during the build phase
- The Audit requirements will be clearly understood and reflected in the system design during the project discovery phase.
- Of the models of Open Source software development, it is assumed to be 'Benevolent Dictator for Life'
- Build by component and in a way, that delivers value soonest
- A 6-month timeline has been assumed for the tender process to execute on any RFPs – this needs to be tested upfront. It is assumed that one RFP would take 6 months per unit, not multiple RFPs per unit. In the case that multiple RFPs are required additional lead time will be required.

Maintenance and Certification

- Decision point at eighteen (18) weeks prior to the next election to make determination if a feature is considered major (or minor, or administrative), since procurement of new consultants to handle testing for those features takes 10 weeks
- Certifying a system (like the one proposed) in pieces has never been done before so some assumptions were made after a conversation with the Secretary of State. End to End Certification Timing Assumption:

Components being Certified	Timing
Full Build of Component	6-12 months
Major Feature	2 months
Minor Feature	1 month
Bug Fix	1-2 days

2 Feasibility Summary

Six different delivery options (described in section 4) were evaluated during this assessment. We believe that the highest likelihood of project success combines parts of multiple options in addition to the mitigation of key project risks.

Given the inherent challenges of this endeavor and based on our research the most feasible approach is to:

- Have this program be owned by the city, specifically the Department of Technology. Hire or reassign the required expertise to administer and guide this project.
- Enlist multiple vendors for the entire build phase and contract them for ongoing operational support.
- Commit to building the Open Source community and only start relying on the community for delivery of new features once it has been proven to be engaged and reliable.
- Partner with LA County that has developed human-centered/accessibly designed furniture and electronic voting devices already[1].

[1] http://vsap.lavote.net/process/

- Partner with a jurisdiction, ideally within California, so that the certification is only with one body - the Secretary of State of California. This will allow for sharing of costs.
- Partner with existing open source voting group(s) like (but not limited to) OSET Institute to learn from and possibly build upon the assets that they have already created.
- Approach the project in an agile manner aiming to provide value as soon as its developed.
- Conduct an in-depth assessment of Open Source licensing models and only proceed with an Open Source license when the implications are understood by key city stakeholders such as IT and the legal department.

2.1 Key Risks and Mitigation Actions

For a project of this nature to be successful, the following risk factors must be considered with regards to developing, sustaining and securing the system.

2.1.1 Change to Current Solution Delivery Model

Building a system like this requires a shift from the City's preference of configuring to coding a new software system to which the City would have to commit to. Even with reliance on outside vendors, there will be a significant need for leadership, decision-making, and product management in house.

The City should:

- Determine if they are willing to make this shift and if so commit to develop this capability (the components of which can be found in the Capability Model in the appendix)
- Position this capability in the Department of Technology
- View this as a capability which will serve this effort first then be refocused on the next custom application

2.1.2 No Specific Requirements for a Voting System

This report does not present a design of the voting system. Although some thinking has been done to scope out what a prototypical voting system would entail, actual system requirements have not been gathered. Therefore, a minimum of **+100% margin** should be applied to both the timeline and costs (both upfront and ongoing).

The City should:

- Commit to a Discovery phase for this project. This engagement would develop a clear picture of what success looks like and how specifically the voter's needs will be met. It also involves a deep dive into the critical success factors such as technical constraints, licensing, program operating model, procurement and possible coordination with other organizations or jurisdictions.
- Use the output of this engagement to issue an RFP with an amount of detail which will increase the likelihood of multiple vendor's responding.

2.1.3 Ability to attract and engage multiple vendors

Travis County TX attempted to find vendors to build an open source voting system called STAR-Vote. In the end they did not find a vendor that was interested in the Open Source model and had to change their approach[2].

[2] http://www.mystatesman.com/news/travis-county-ditches-plan-craft-its-own-voting-system/15GsWZ8FnWntGgUz25L1TL/

The City should:

- Have a backup solution in case this approach does not deliver a reliable voting system.
- Expect that they will likely have to work with many different vendors to deliver and maintain the entire solution.
- Commit to a Discovery phase and use the outputs from this phase to provide a well-defined set of RFPs which will reduce the perceived risk for vendors.

2.1.4 Ability to establish a healthy, functional, and reliable O/S Community

The development of an Open Source community requires time and attention and the City needs to decide if this is a focus that they want to have. The City does not have a track record for building thriving Open Source communities nor is it its focus to do so. There is also not an analogous model to make assumption around since a project like this has never been completed in the United States, at least not at this scale. Without a commitment to building an Open Source community this project will not achieve its full potential. It is possible that this community could form organically but for value to be derived for this project the work of that community still needs to be incorporated into the code stream and put through the certification process which will not be able to be done by the community.

The City should:

- Decide if they want to commit to building this community.
- If they decide to they should they should hire at least two initial resources for this effort. One will be a technical architect who will focus on the technical aspect and impact of incorporating code and input from the community as well as tooling and documentation to empower those who are interested to participate. The second is an Open Source program manager who will initially focus on evangelism to grow the community, communication with the community and community structure and operations.

2.1.5 Certification of the system(s) with the Secretary of State

The State certification process as it operates today is the certification of an entire system end to end and has only been done with commercial vendors. In a more incremental/agile delivery methodology there is a need for a more flexible and adaptable certification process that can triage between administrative, minor, and major changes as the system is being developed

From initial conversation with the Secretary of State they are open to this nimbler approach. This however has not yet been proven and poses a serious risk to the project's ability to deliver in this way, as well as the assumptions and estimates of timing.

The City should:

- Conduct a Discovery phase where this interaction model with the state will be documented and negotiated with the Secretary of State.
- In addition, the Discovery phase should define a specific phased approach to implementation examining the existing system and the constraints it will impose. This will better define the requirements of how the City of SF and the State will need to work together.

2.1.6 Open source licensing

Open source licensing as they exist today may not be sufficient for the purposes of the System being built for elections. Although the Department of Elections expressed a preference for Version 3, GNU General Public License as they believe it provides other users the greatest access to view, modify, and use a System's software code, it is worth evaluating whether this will meet all needs and stages of the product

The City should:

- Align and agree on the license with which the city is most comfortable.
- Engage key project stakeholders and the legal department for the city to do a full analysis of licensing models and the implications of each.
- Consider if that value proposition can be addressed by a non-Open Source method.
- Strongly codify the value proposition and rational of conducting this project as an Open Source project.

The OSET Institute has created their own open source license to address some issues they perceive to be an issue with government engagement. Two specific points of rationale that they have outlined in their document[3] "RATIONALE DOCUMENT—VERSION 2.2" :

> *"**1. Governing Law**. Most government procurement regulations require the application of local state law or federal law to the material terms and conditional s of any contract. Most open source licenses lack law or venue selection provisions. Also, the application of particular law affects the interpretation of a license document as a whole, and therefore, we conclude, is necessarily a modification to all of its terms. Thus, to agree in a separate contract that a particular body of law applies to a license would be an additional restriction on that license.*
>
> ***2. Venue.** Many state and federal procurement regulations require that disputes be resolved in particular venues. Please see our comments above regarding governing law."*

2.1.7 Partnerships effect on delivery timeline

If collaborating with other jurisdictions, there is a risk that the City of SF and their schedule and priorities differ.

The City should:

- Evaluate these partnership options and determine what mitigations can be put in place to maintain control of the requirements and timeline.

2.1.8 Legislative constraints

The City of SF is a paper-based jurisdiction. This is assumed to be the case going forward. This project's expenditure is to replace the existing electronic voting machine with a more accessible solution developed in an open source manner. The plan is to replace the one-per-precinct model, with at least two in each to ameliorate the risk of longer wait times to use the electronic voting machine. This does not replace the paper-based system, and all the expenditure for that system will remain.

If this constraint can be lifted, the City can see a greater use of the electronic voting machine and a replacement of the paper-based system.

The City should:

- Conduct an analysis of the cost and value trade off to change the legislative constraints.

2.1.9 Security Risk

In addition to the scrutiny placed on the development of this system from a state certification perspective, there will be additional public attention and pressure on whomever develops a voting system, especially one developed in such a transparent fashion. If the City develops a new system

---

3 https://static1.squarespace.com/static/528d46a2e4b059766439fa8b/t/55fcc566e4b049b6f6c15194/1442628966530/OSETPublicLicenseRationale_v2.2.pdf

from scratch they need to be comfortable with the transfer of responsibility for security from a shared responsibility (vendor and City) today to a complete responsibility. Alternatively, they need to find a build and run partner who is willing to assume this responsibility with them.

The City should:

- Consider if this risk is one it is willing to assume.
- If not, then look for partners who can share the responsibility and clearly define the liability assumed by each party or parties.

2.1.10 Quantifying value of the program

It is hard to determine the quantified value and benefit of this program in terms of the public good. This is an evaluation that can only be made by the city in light of its other needs.

The City should:

- Conduct an assessment and comparison of the alternatives uses for this capital should be conducted to ensure that this investment is appropriately prioritized against the City's portfolio of needs.

2.1.11 Complexity of souring vendor support and managing multiple concurrent RFPs

The City's procurement via RFP from vendors includes a number of process complexities and requirements that impact the timeline of any project. Running an RFP process can take a considerable amount of time, energy, and cost. It important to be realistic in the overall number of RFPs and the bandwidth of the group that is able to execute on managing them. This feasibility report expects that when soliciting support from vendors, it is expected that a single RFP would be released for the overall discovery and high-level architecture design phase; with subsequent RFPs per each of the seven sub-system elements presented in Section 3 and within the phases described in Section 8. This would put the number of RFPs in the range of 8-10 in total (i.e. one for discovery/design and then approx one per sub-system element).

Additionally, later in **Section 8.2 (Agile vs Waterfall)** the concept of Agile vs. Waterfall project delivery is considered. When it comes to the procurement of services, it is expected that the overall 'Epic-level' user stories will be defined in a waterfall approach up front and that the subsequent build phases, or sprints, would occur in a more agile manner.

The City should:

- Determine which office within the city has the capacity and capability to run the RFP process(es)
- Commit to finding a single vendor for the design phase in order to reduce the overall program risk
- Limit the number of RFPs to a the smallest possible number to decrease complexity. A strategy for doing this should be developed in the discovery phase.
- Evaluate the optionality for vendors to bid on more than one sub-system so as to generate economies of scale in their detailed design and development

2.2 Critical Project Initiation Criteria

There are a number of milestones that must be reached or actions that need to be undertaken before we can assume that the project has officially started and the overall project timeline can commence.

These are as follows:

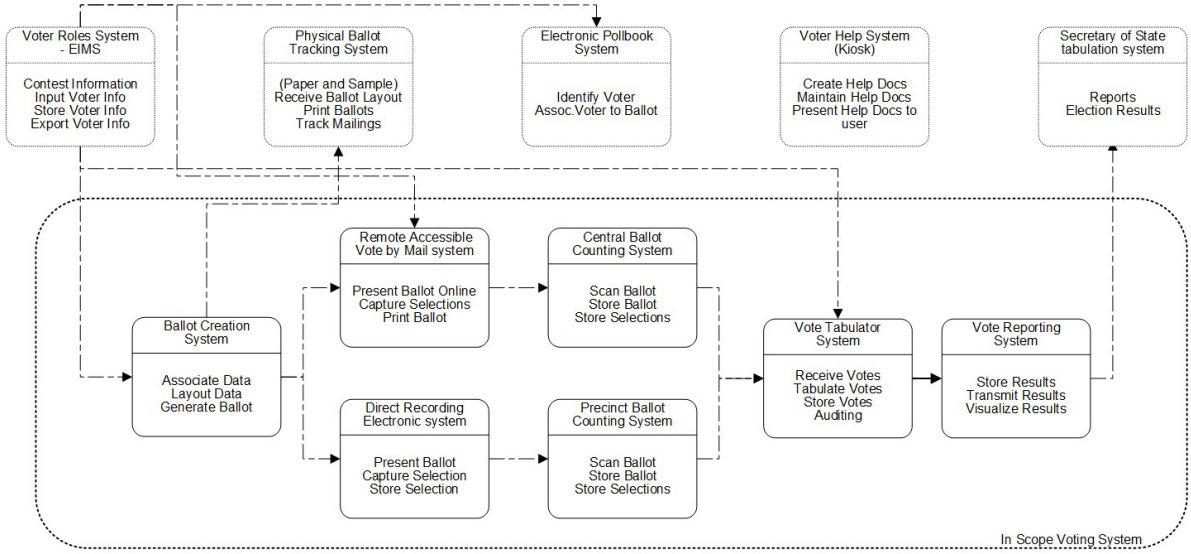
1. Obtain a Memorandum of Understanding from the Secretary of State
    - To confirm that the modular nature and agile production process of the system design and build can be certified in a timely manner and that a service level agreement can be established, or at the very least written assurances of the ability to meet certain timelines for certification given agreed upon criteria.
2. Open Source License Legal Review
    - Complete the legal review of the open source license type (GNU Public License, version 3) that the City has identified as their preference to confirm any issues throughout the product lifecycle
3. Prepare For and Officially Initiate a Project
    - Select a proposed option: Based on the evaluated options, determine which options makes most sense to move forward with project.
    - Determine Project Leadership: Define requirements for leaders (e.g. project owner, product owner, budget manager, program manager).
    - Build Project Operating Model: Understand and build out a model for what the organizational structure will look like to run the project, and how they will deliver value.
    - Define Roles & Responsibilities: Define requirements for what each team member must do to be successful.
4. Finalize Budget and Funding
    - Define Budget: Define total amount of financial resources that needs to be allocated for this project, including YoY costs, capital/operating expenses, etc.
    - Determine Funding Allocation: Allocate funding per department resources needed, and see if there are gaps that need addressing.
    - Plan and Estimate Spending: Within each category of spend, determine the forecast of costs over time and the means by which each will be tracked. Allow for cost contingency on any external contracts, and time contingency on internal spend.

3 System Build Scope

Defining the requirements for an Open Source voting system is not in scope for this feasibility study. This poses a challenge when trying to determine the scope, capabilities needed, cost and timeline to deliver one.

To address this issue, a generalized model of a voting system was created. First, a review of laws, examination of existing systems and current trends in voting systems was conducted to help the project team define the scope of a 'voting system'.

The diagram below shows the components of the model 'voting system' which was used for cost and timeline estimates. For more details on the methodology used, please refer to Appendix B - Proposed Voting System.



4 Description of Delivery Options

A total six different delivery options were identified, and these have been detailed in the table below.

#	Short Name	Description
1	Dept. of Elections Only	*Ownership:* The Department of Elections (Department) will be held accountable for the project's success, and election to run smoothly on Election Days. *Development:* The IT staff within the Department will i) build the source code, ii) purchase hardware, iii) run/maintain the entirety of the project to make sure it will be certified *Open Source Community Engagement:* Department will govern and own the interaction with the community *Support:* All capabilities will lie in the Department of Elections, with no support from other departments in the City or external vendors involved. *External Vendors:* None

#	Short Name	Description
2	The City of SF Only	*Ownership:* The City of SF will assign ownership to specified departments, with one department providing high-level oversight and guidance to ensure project delivery. *Development:* The Technology Departments in the City (e.g. Digital Services) will provide all capabilities to develop the voting system and run/maintain the project through its completion. *Open Source Community Engagement:* The City governs and own the interaction with the community *Support:* All capabilities will lie in the Technology Departments in the City with no external vendors involved. *External Vendors:* None
3	Vendor support without existing assets	*Ownership:* Ownership of the project remains with the City *Development:* A new voting system will be developed with vendors from scratch rather than building those capabilities in-house *Support:* The City or Department will bring in needed capabilities from external vendors via RFP or other contractual vehicles *External Vendors:* Yes
4	Vendor support with existing assets	*Ownership:* Ownership of the project remains with the City *Development:* A new voting system will be developed with vendors by building upon external party's existing assets rather than building from scratch *Support:* The City or Department will bring in capabilities from external parties via RFP or other contractual vehicles *External Vendors:* Yes
5	Collaboration with Jurisdictions within CA	*Ownership:* Jurisdictions within California will share ownership and create a voting system with the rationale being that a system that services Californian jurisdictions will only have to be certified by the California Secretary of State. *Development:* Collaboration with jurisdictions within California to create a voting system that meets requirements for the participating jurisdictions. *Open Source Community Engagement:* Collaboration with jurisdictions within California to manage interactions with the community *Support:* The Technology Departments in the Department of Elections or the City will collaborate with other jurisdictions within California to share resources in order to provide all capabilities to develop the voting system *External Vendors:* None

#	Short Name	Description
6	Collaboration with Jurisdictions outside of CA	*Ownership:* Ownership of the project remains with the City *Development:* Collaboration with jurisdictions outside of California to create the System *Open Source Community Engagement:* Collaboration with jurisdictions outside of California to manage interactions with the community *Support:* The Technology Departments in the Department of Elections or the City will collaborate with other jurisdictions outside of California to share resources in order to provide all capabilities to develop the voting system *External Vendors:* None

(1) The City joining Travis County's project was a listed option in the RFP, but cannot be evaluated as an option anymore as the project has been cancelled.
(2) Each option can be permutated in ways that can involve external vendors via RFP.

5 Costs and Timeline

There are three major areas of cost considerations for this project. These break down into

i) Development Costs which vary upon how the System software is developed,

ii) One-Off Fixed Costs for System hardware and certification, and

iii) Ongoing costs that would be required to maintain the system and carry out elections

To develop the cost model, we used the information from the **System Build Scope** section and assumed the most likely delivery method, which is for the city to own the overall program but use vendors for the development of the system and on-going maintenance. The costs shown in this section reflect that model of delivery. Starting in **Professional Services Build Cost – Per Option** we consider the cost impact of using six different delivery options.

As stated before defining the requirements for the City's Open Source voting system is not in scope for this feasibility study thus a 100% margin should be applied to these costs.

The build timeline estimated for the baseline option is **3-6 years** – further details can be found in **System Build Considerations (Section 8)**

A summary of costs is provided here with a further breakdown in the following sections.

Category	Description	Costs*	Frequency / Time period
Professional Services associated with Development of System (Build Phase)	• Discovery • Ballot Creation System • Remote Accessible Vote By Mail System • Accessible Voting Device System • Precinct Ballot Counting System • Central Ballot Counting System • Vote Tabulator System • Vote Reporting System	**$ 11.5M – $ 27.8M**	Over the course of the build and implementation (~3-6yrs)*
Hardware	• Ballot marking device • Furniture • Peripherals • Scanners	**$ 6.35M – $ 16.4M**	One-off

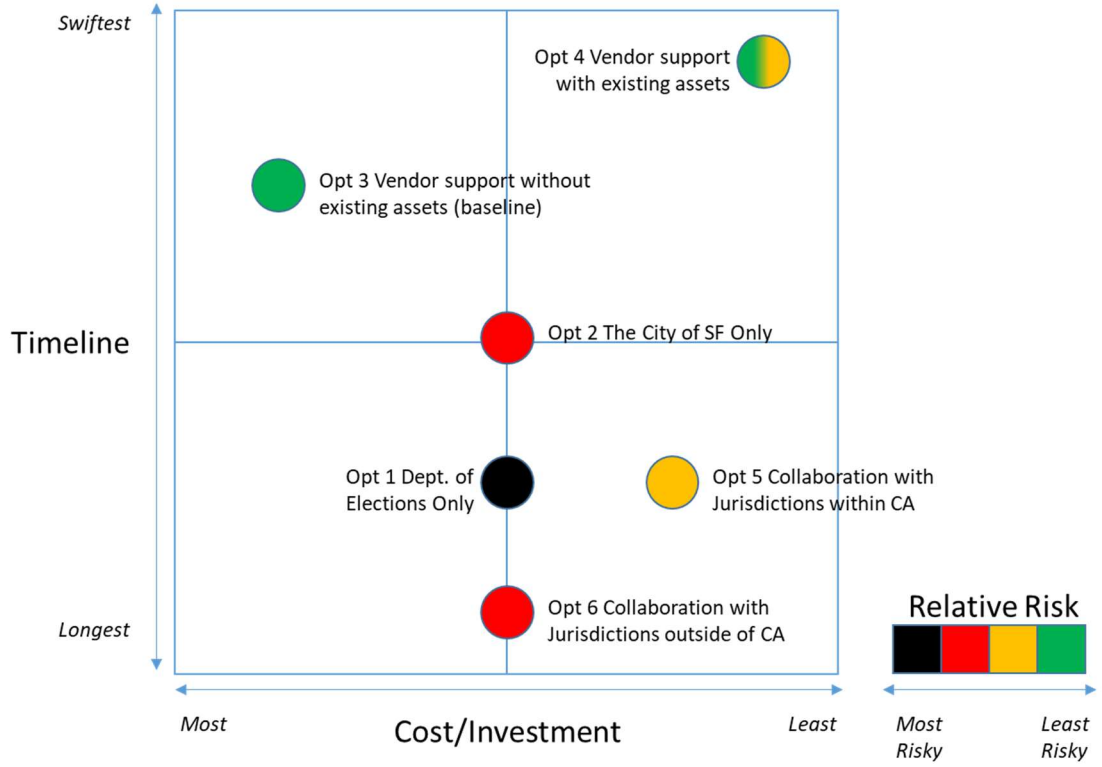
	• Precinct scanners		
Certification	• Deposit	**$ 0.4M**	One-off
On-going	• Hardware Storage (Accessible Voting Device) • Application Hosting • Professional Services Roles	**$ 3.3M-6.6M**	Per year
Per Election	• Support • Paper Ballot Costs • Poll workers for day of election • Poll worker training • Maintenance and Licensing	**$ 3.4M-4.83M**	Per election

**based on a +100% contingency*

5.1 Assessment of Delivery Options

Each of the City's options to deliver a highly accessible open source voting system were evaluated across the dimensions of capability, cost, time and risk. The graph below displays these options evaluated across cost of development and time to completion.

Note that the axis of the graph below is inversed, hence the lowest cost and time to completion option is to the top right of the graph. Additionally, the level of risk corresponds with the color of the circle.



5.2 Build Phase Costs

5.2.1 Professional Services Build Cost - Baseline

There are seven components that were estimated to be in scope for this System and costs of each component was determined by creating a model with the following parameters:

- Skill set required
- Duration of the project
- Utilization each resource required

The numbers below reflect our baseline costs and assume that the City will own the overall program, but vendors will be contracted to do the build. These costs do not include hardware costs those are provided in section 5.1.3.

System Component	Cost	Notes and Assumptions
Discovery	$1.1M-1.2M	Cost basis: 4 month duration with 7 resources (full and partially engaged) at an average rate of ~ $263/hr Develop a clear picture of what success looks like and how specifically the voter's needs will be met. It also involves a deep dive into the critical success factors such as technical constraints, licensing, program operating model, procurement and possible coordination with other organizations or jurisdictions. Since the project is expected to be delivered in an agile way the discovery does not define all requirements. It seeds the development process with the highest value work first and provides a holistic view of the project challenges and proposed approaches. Each of the sub-systems will also include a discovery phase to begin.
Ballot Creation System	$1.2M – $3.0M	Cost basis: Average rate of $237/hr Build -- 5 month duration with 8 resources (full and partially engaged) Certification – 5 month duration with 4 resources (partially engaged only)
Remote Accessible Vote By Mail System	$0.99M – $2.4M	Cost basis: Average rate of $238/hr Build -- 4 month duration with 7 resources (full and partially engaged) Certification – 4 month duration with 4 resources (partially engaged only) Requires Ballot Creation work to be completed first
Accessible Voting Device System	$4.5M – $11.0M	Cost basis: Average rate of $245/hr Build -- 8 month duration with 14 resources (full and partially engaged) Certification – 8 month duration with 7 resources (partially engaged only Largest most complex component
Precinct Ballot Counting System	$1.1M – $2.8M	Cost basis: Average rate of $238/hr Build -- 5 month duration with 7 resources (full and partially engaged) Certification – 4 month duration with 4 resources (partially engaged only

System Component	Cost	Notes and Assumptions
		Slightly less on certification costs past this point since the certification process should be well understood by this point in development.
Central Ballot Counting System	$0.95M – $2.4M	Cost basis: Average rate of $238/hr Build -- 4 month duration with 7 resources (full and partially engaged) Certification – 4 month duration with 4 resources (partially engaged only Assumes Precinct Ballot Counting System is done first.
Vote Tabulator System	$0.85M – $2M	Cost basis: Average rate of $243/hr Build -- 3 month duration with 8 resources (full and partially engaged) Certification – 3 month duration with 4 resources (partially engaged only
Vote Reporting System	$0.86M – $2M	Cost basis: Average rate of $237/hr Build -- 3 month duration with 8 resources (full and partially engaged) Certification – 3 month duration with 4 resources (partially engaged only
Build Total	**$11.55M – $27.8M**	

5.2.2 Professional Services Build Cost – Per Option

The City has options for developing a highly accessible Open Source voting system. The System can be developed by the Department of Elections, the City of San Francisco, with support from external vendors and in collaboration with jurisdictions within and outside the state of California. The description, capabilities, risks and gaps with regards to each option differ and this has been detailed in **Section 6** of this document. In this section, we explore the cost and time variance from the baseline estimated above.

5.2.2.1 Option 1: Department of Elections Only

This option explores if the System is developed by the Department of Elections with no support from external vendors.

The baseline cost as broken down in **Professional Services Build Cost - Baseline** is estimated to be around $11.5M - $27.8M with an estimation of 3-6 years for its build. The deviation from this baseline is detailed below:

	Estimate	Variance	Justification
Cost	**~$8M – 19.5M**	Approximately -30% variance from baseline cost estimate	• FTE costs is less than the contractors rate hence the costs will be lower than baseline

Time	4 -8 years	Approximately +1year variance from baseline time estimate	• Each resource takes a lot longer to hire • With fewer resources, the timeline may increase • Description of skillset can be found in the Capability Model
Risk	BLACK	Significantly higher risk than baseline	• Disruption from purpose of department • Weak Capability Model. Further details can be viewed in **Section 6**

5.2.2.2 Option 2: City of San Francisco Only

This option explores if the System is developed by the City of San Francisco with no support from external vendors.

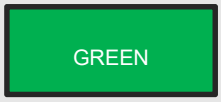
The baseline cost as broken down in **Professional Services Build Cost - Baseline** is estimated to be around $11.5M - $27.8M with an estimation of 3-6 years for its build. The deviation from this baseline is detailed below:

	Estimate	Variance	Justification
Cost	**~$8M – 19.5M**	Approximately -30% variance from baseline cost estimate	• FTE costs is less than the contractors rate hence the costs will be lower than baseline
Time	3.5-7 years	Approximately +0.5 year variance from baseline time estimate	• Need to hire people but the City has more capabilities currently in place to execute the Build
Risk	RED	Higher risk than baseline	• Lower visibility into Department of Elections needs. • Some specialization can be provided by existing resources but complete capability is not currently set. • Capability Model can be viewed in **Section 6**

5.2.2.3 Option 3: City of San Francisco with vendor support without existing assets

This option is the baseline estimate for the cost, risk and time to build the System defined in **Professional Services Build Cost - Baseline** above.

	Estimate	Variance	Justification
Cost	**~$11.5M - $27.8M**	N/A – Baseline	• N/A - Baseline

Time	3 -6 years	N/A - Baseline	• N/A - Baseline
Risk		N/A - Baseline	• Least risky option

5.2.2.4 Option 4: City of San Francisco with vendor support with existing assets

The baseline option for delivery is that the City of San Francisco pools its resources and brings in help for the build from outside vendors. This option differs from the baseline as the System is created with the support of vendors that building upon existing assets.

The baseline cost as broken down in **Professional Services Build Cost - Baseline** is estimated to be around $11.5M - $27.8M with an estimation of 3-6 years for its build. The deviation from this baseline is detailed below:

	Estimate	Variance	Justification
Cost	**~$7M – 16.7M**	Approximately -40% variance from baseline cost estimate	• Building upon existing assets results in a lower cost due to shorter time to completion.
Time	2.5-5 years	Approximately -0.5 year variance from baseline time estimate	• Building upon existing assets results in a shorter time to completion.
Risk		Slightly higher than baseline	• Existing assets may not have been built with SF needs as a priority. This poses some risk as existing assets may need to be re-engineered to fit the needs of the City

5.2.2.5 Option 5: City of San Francisco in collaboration with jurisdictions inside California and without vendor support

The baseline option for delivery is that the City of San Francisco pools its resources and brings in help for the build from outside vendors. This option differs from the baseline as the System is created in collaboration with jurisdictions within the state of California. The rationale being that a system that services Californian jurisdictions will only have to be certified by the California Secretary of State.

The baseline cost as broken down in **Professional Services Build Cost - Baseline** is estimated to be around $11.5M - $27.8M with an estimation of 3-6 years for its build. The deviation from this baseline is detailed below:

	Estimate	Variance	Justification

Cost	**~$7.5M – 18.0M**	Approximately -35% variance from baseline cost estimate	• FTE Cost is less than contractor rate (~50%) • Able to share development and certification costs with other jurisdictions
Time	4-8 years	Approximately +1 year variance from baseline time estimate	• Collaboration results in increased complexity of execution
Risk	AMBER	Moderately higher than baseline	• Lack of control over focus / priorities • Complex solution without a full set of capability coverage. More details on capability assessment can be viewed in **Section 6** • More parties to manage

5.2.2.6 *Option 6: City of San Francisco in collaboration with jurisdictions outside California and without vendor support*

The baseline option for delivery is that the City of San Francisco pools its resources and brings in help for the build from outside vendors. This option differs from the baseline as the System is created in collaboration with jurisdictions outside the state of California.

The baseline cost as broken down in **Professional Services Build Cost - Baseline** is estimated to be around $11.5M - $27.8M with an estimation of 3-6 years for its build. The deviation from this baseline is detailed below:

	Estimate	Variance	Justification
Cost	**~$8M – 19.5M**	Approximately -30% variance from baseline cost estimate	• FTE Cost is less than contractor rate (~50%) • Splitting costs will be lower, but will have to account for the expense to accommodate requirements and regulations across participating states
Time	4.5-9 years	Approximately +1.5 year variance from baseline time estimate	• Collaboration results in increased complexity of execution • Differing certification processes across states since some states can bypass direct federal approval
Risk	RED	Higher risk than baseline	• More states mean even more complexity of regulations and laws • Collaboration across states may be challenging • Certification process may be more complex than just CA

5.2.3 Hardware Costs

There are two major hardware costs to be considered. The first is the per unit cost of the Accessible Voting Device and the second is the scanners required for the solution.

Type	Cost	Notes
Accessible Voting Device units	$3.05M - $ 9.8M	Assuming 1220 units at $2500 per. Includes peripherals for accessibility, scanning/printing and furniture. LA county's project which is ready for manufacturing has a per unit cost of 4k per unit. This forms the higher end of the estimate.
In-Precinct Scanners	$3-6M	Assuming 610 scanners at $4800 per. This is the same as is in use today.
Scanners	$0.3-0.6M	Industrial High Speed Scanners - deliver and setup
Per Election Total	**$6.35M - $16.4M**	

5.2.4 Certification Costs

The State of California certification process represents a significant cost to the project. The professional services cost already includes the consulting costs required (approximately 15% of the professional services costs) to oversee the process.

In addition, there is a deposit cost paid to the State for certification. It is estimated this number will be **$360k** for the overall system/solution.

5.3 On-Going Costs

After the build phase has been completed it is expected that a dedicated team will remain focused on the project full time. It is possible that with increased community engagement this team could be reduced over time. As with the build we have assumed a professional services firm will play most roles.

We have assumed that a few of the roles already exist and would therefore be filled by existing the City of San Francisco employees and represent a sunk cost. Those roles are: Executive Oversight, System Administration and Quality Assurance.

Type	Cost	Notes
Hardware Storage (Accessible Voting Devices)	$1-2M	Assumes the continued usage of Pier 48 as is being used today.
Application Hosting	$0.5-1M	It is expected that where possible cloud-based hosting will be used.
Professional Services Roles	$1.8-3.6M	The roles expected from professional services would therefore be: Program Management (1), Product Ownership (1), Open Source Community Management (1), Software Architecture (1) and Software Engineering (1). Hourly costs for these roles are expected to be slightly less than the rates in the build phase due to the full year commitment.
Total	**$3.3-6.6M**	

5.4 Per Election Costs

There are a set of activities and costs required per election. An example of this would be support of the Accessible Voting Devices in case of failure. In addition, there are costs for resourcing, ballots and maintenance. Current costs from the Dominion contract were considered and adjusted based on the change in systems. It is assumed that the Accessible Voting Device count would go from 610 today to 1220 in the new system.

Per Election Costs	Cost	Notes
Support	$0.985-1.97M	Support of the technology and machines used during elections.
Paper Ballot Costs*	$1.4M	Paper ballot costs are not expected to change. Accessible Voting Devices will support the existing paper-based process instead of replacing it.
Poll workers for day of election*	$0.505M	$142 to $195 stipend for 3k workers.
Poll worker training*	$0.015M	10 people by 16 hours at 85/hr rounded up to 15k
Maintenance and Licensing	$0.468-0.936M	Current Dominion system cost for this line item is $483K. The licensing fee is eliminated by an open source system but offset by the maintenance costs of doubling Accessible Voting Devices from 610 today to 1220 in the new system.
Per Election Total	**$3.373-4.83M**	

* These costs are the same as those incurred presently and thus represent no change from the current state. They are included here to confirm that there is no additional costs in these categories.

6 Evaluating the Delivery Options using a Capability Model Framework

6.1.1 Approach

The Capability Model framework was used to clarify what it takes to deliver a successful program and enabled the assessment of options against a comment set of capabilities. Details of how the capability model was created can be found in the Section D of the Appendix.

When evaluating each option solely from a capability model perspective, it became apparent that there were three distinct groupings of options.

Category	Options Combination	Rationale
Dept. of Elections Only	Option 1 – Dept. of Elections	Assessed to analyze whether the dept. could deliver this project on its own.
The City of SF, w/ potential resources from other jurisdictions	Option 2 – City of SF Option 5 – Collaboration with Jurisdictions within CA Option 6 – Collaboration with Jurisdictions outside CA	As the project was not able to assess all possible jurisdictions, it was not possible to discern through this assessment whether additional capabilities could be provided by collaborating with other jurisdictions. The value of collaboration will be captured via the opportunity to share costs and achieve economies of scale.
Vendor Support with / without existing assets	Option 3 – RFP without existing assets Option 4 – RFP with existing assets	Though a thorough vendor analysis was not undertaken, it was apparent that additional capability could be provided through bringing in vendors.

To evaluate each category, the capability model was used as baseline framework for comparison. For each capability, sub-criteria were determined in order to evaluate if that capability is currently feasible to utilize.

The sub-criteria to evaluate each capability is as follows:

- **Track Record** - A record of past performance of this capability as an indicator for likely future performance based on interviews and research
- **Skillset** - A collection of skills and abilities that demonstrate having the capability needed
- **Capacity/Resource** - The ability to perform based on the current resource workload commitments
- **Desire to Own** - Eagerness to be accountable for the outcome based on the capability needed

Based on feedback during many interviews with the City, the scoring metric for each sub-criterion is defined by what was observed or recorded. The scoring metric comprises the following scale with detailed justification in the appendix:

- Not Applicable
- No Evidence
- Minimal Evidence
- Moderate Evidence
- Substantial evidence

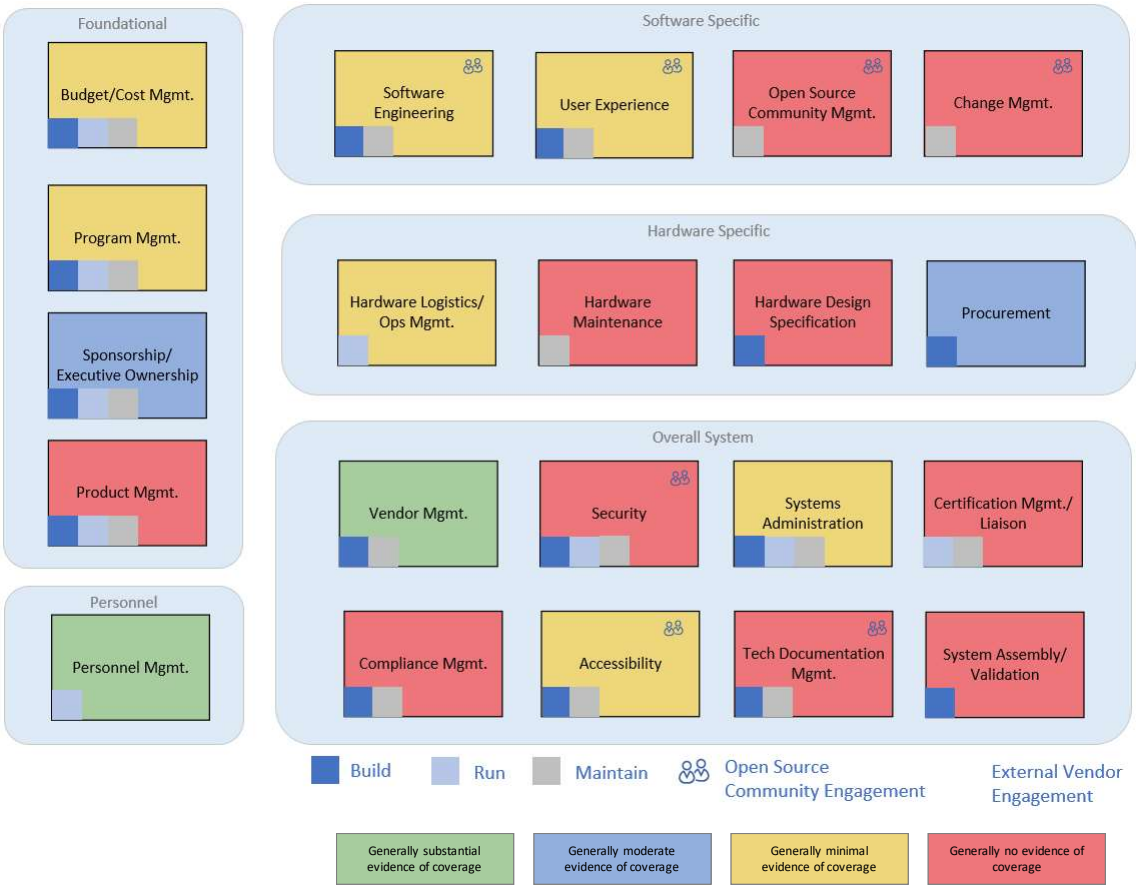
Then, each option's capabilities were summarized to convey the areas of capability coverage and gaps/risks. The summaries are presented below and additional detail can be found in Appendix C.

6.1.2 Summary of capability model findings

From this analysis, it can be seen that Options 3 and 4 have the greatest coverage of the full set of capabilities required to deliver a successful program. Option 1 has the least coverage of all options; and Options 2, 5, and 6 fall somewhere between these others.

It is worth noting, that even in option 3 and 4, there remains a residual risk relating to the design and assembly of voting hardware. Given the limited number of suppliers of COTS hardware that would be applicable it was not possible to discover significant evidence. For that reason it would be worth investigating other jurisdictions that are already some way down the path of developing the hardware.

6.2 Capability Model Findings: Option 1 – Department of Elections Only



6.2.1 Areas of coverage

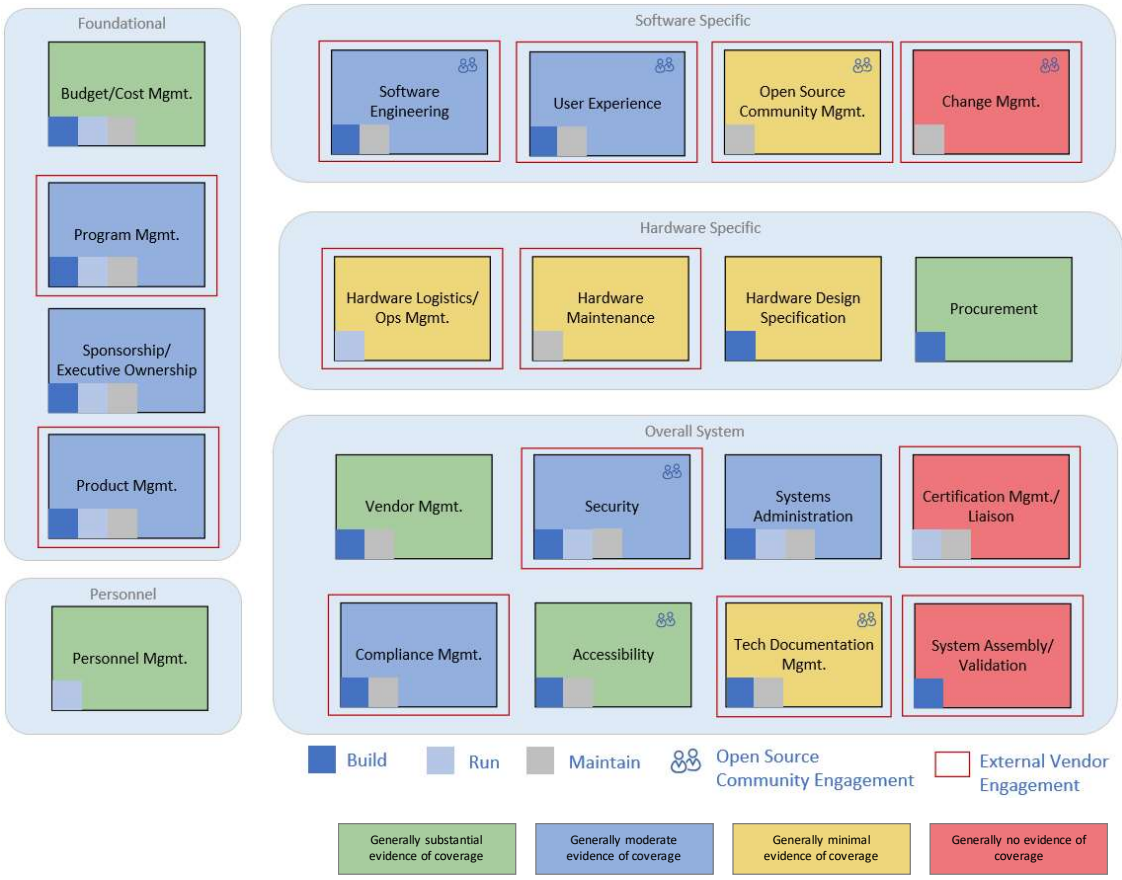
- Personnel Management
  - There is substantial experience in hiring, training, managing poll workers on the election day. However, there is not much of a desire to own this capability in the department - but a mandate for them to own it.
- Vendor Management

- There is substantial evidence for the Department of Elections to handle vendors since most of what they do is working with a chosen vendor that provides voting hardware, software, operations to ensure the voting process goes smoothly for all voters.

6.2.2 Gaps/risks

- Multiple gaps across software, hardware, and overall solution. Notably, no coverage for Open Source projects, hardware maintenance/design/assembly
  - There is no evidence of product management in this department, since they have never run any voting system development projects in the past. Products (e.g. voting machines) in relation to voting has always been procured.
  - There is no evidence of Open Source community management experience in this department. There has been no interaction with the Open Source community in the past.
  - There is no evidence of current change management projects or change leadership within this department as it relates to voting system development.
  - Currently, Dominion staff handle hardware maintenance and keep the Department of Elections informed. Therefore, there are no currently known capabilities around hardware maintenance for voting systems within the City.
  - In the past, design specification was only needed for choosing a voting system vendor. The Department of Elections understands the decision criteria and desired capabilities for a working voting system, and can chose a vender based on the options presented. However, there is no current known capabilities around determining hardware design that is compatible with their custom-made software.
  - There is no evidence of the Department of Elections handling voting system security besides basic server security. This is mainly managed by the vendor.
  - Currently, Dominion informs the Department of Elections of changes that need to be made to the system due to changes in regulation. There is no evidence of compliance management capabilities in the Department of Elections itself.
  - There is no evidence of technical documentation management within the Department of Elections because this is done by the vendor itself. Currently, Dominion provides technical documentation in order to proceed in the certification process.
  - There is no evidence of detailed sub-system system assembly/validation within the Department of Elections because this is done by the vendor itself i.e. the vendor supplies modular units that the employees connect together.

6.3 Capability Model Findings: Options 2, 5, and 6 - The City of SF, w/ potential resources from other jurisdictions



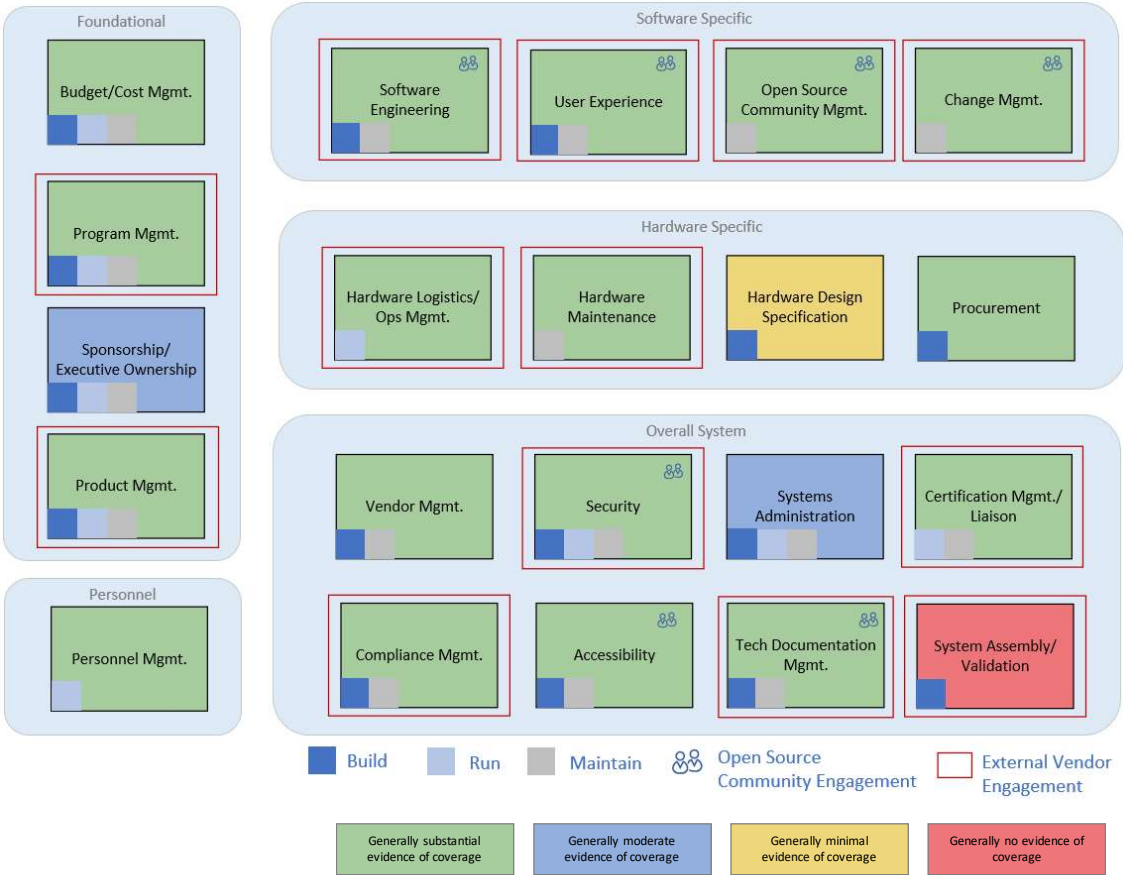
6.3.1 Areas of coverage

- Personnel and Vendor Management as per Option 1
- Budget/Cost Management
  - To our knowledge, the capability to manage budget/cost already exists in the City since there is a department established to control the City's budget. The Budget Director, Melissa Whitehouse, manages budgets for projects like this and decides whether to fund them or not based on a variety of cost factors.
- Procurement
  - The City has an established department called the Office of Contract Administration that manages the procurement process for all consultants, hardware, etc.
- Accessibility
  - There are many accessibility advocates and committees within the City (e.g. Mayor's Office on Disability, Department of Aging and Adult Services, Voting Accessibility Advisory Committee) that have shown interest and want to transform the new voting system to not just meet accessibility requirements, but make it a useable experience for all. They are part of the RFP process to test which system is more useable.

6.3.2 Gaps/risks

- Most Hardware specific capabilities as well as system certification and change management as described in option 1

6.4 Capability Model Findings: Options 3 & 4 Vendor support with / without existing assets



6.4.1 Areas of coverage

Almost all areas of capability model, notably:

- Product Management
  - Given that this option is that of finding a vendor, hiring a product manager for this project would not be difficult because the Bay Area is a booming environment for cultivating innovative products.
- Software Engineering
  - There is a very high likelihood to hire capable software engineers to code this voting system, with additional open source experience since software engineering skills are a very demanded and supplied skill in the Bay Area.
- User Experience
  - Procuring individuals/companies with user experience skills to enhance the voting experience will not be too difficult since design companies like IDEO are headquartered in San Francisco.
- Open Source Community Management
  - There is a big open source community hub in the Bay Area, so finding community managers with expertise in this area will not be too difficult.
- Change Management
  - A lot of consulting firms in the area have change leadership experience, and can be hired for this project to support the strategy around change management for the new voting system.

- Hardware Logistics/Ops Management
  - Currently, the Dominion voting system has Dominion staff managing the hardware operations and logistics. Hardware logistics and operations can still be outsourced to a IT services company in the area.
- Hardware Maintenance
  - Finding a vendor to manage Non-proprietary hardware maintenance will be very likely because no custom expertise is needed.
- Security
  - Expertise in security for open source projects can be found because of the abundance of cyber security skillsets in the Bay Area.
- Certification/Management Liaison
  - In the past, the voting system vendor would ensure their product was certified. Since there is a detailed application process, the skillsets to be a certification liaison must have coordinator experience and may not need specific certification process experience. This skillset will not be hard to find in the Bay Area.
- Technical Documentation Management
  - The need for technical writers is abundant in the Bay Area, and that individual does not have to already specialize in voting system knowledge. This role will not be hard to fill.

6.4.2 Gaps/risks

- Hardware Design Specification
  - In the past, design specification was only needed for choosing a voting system vendor. The Department of Elections understands the decision criteria and desired capabilities for a working voting system, and can chose a vender based on the options presented. However, there is no current known capabilities around determining hardware design that is compatible with the existing custom-made software in the City.
- System Assembly/Validation
  - There must be a physical IT service that provides assembly support and testing validation that can offer their skillsets to help the City ensure their hardware is functional.

7 Delivery Considerations

This section will lay out the specific delivery nuances and concomitant assumptions. It provides a level of education on some key topics so readers of all levels of familiarity with these issues can digest the rest of the report.

7.1 Open Source Strategy

This assessment was carried out with the explicit request that Open Source be the licensing model for the City of San Francisco's voting system. It is therefore important to understand what Open Source means. For the purposes of this assessment, we define Open Source as follows:

Most people are familiar with programs like Microsoft Word. Word is an example of proprietary or closed source software. The source code used to make the program run is not disclosed to the public and is tightly guarded since it represents a huge investment and prevents competitors from building another application based on Microsoft's intellectual property. Users of the application are required to agree to certain usage restrictions or not use the software. In addition, they are expected to pay for the application's usage.

Open Source software operates in the opposite way. Its source code is made publicly available and other are encouraged to use, copy, learn from, share, and alter it. Open Source software is also

published with a license to define the way that the users can distribute and alter it. Making the source code available in a public place like GitHub does not make it Open Source. If an Open Source license is not employed, but the code is made available, that code can be viewed but the viewer does not have any rights to use or alter it in any way. Open Source projects typically engage a volunteer-based community and therefore can represent an opportunity for cost savings to a project like this one.

A key principle of Open Source for this project is the balance of 'control' and 'engagement' of the community. Too strict in terms of 'control' and the City of San Francisco runs the risk of a lack of interest and freedom of programmers to contribute; too relaxed and the City may not be able to focus and direct the efforts as required and intended.

7.2 Open Source Licensing

Open source licenses can be significantly different. The scope of this assessment does not include a deep dive into each licensing type and its implications nor recommendations for which license to use. Completion of this licensing discovery is vital when the decision to move forward with this project is made (and has been included in the key next steps). The chosen license has far-reaching implications around procurement since some of the tenants of the existing Open Source licenses may not be agreeable to the City of San Francisco and they are bound to it when accepted. With that said the City of San Francisco prefers that the system utilize software using the GNU Public License, version 3 if possible. When considering existing software to incorporate into the System, the City of San Francisco would select software that provides other users the greatest access to view, modify, and use a system's software code. The City prefers that any potential system's Open Source software would apply copyleft characteristics so that anyone would be permitted to freely use and change the system's software but, on the condition, that all subsequent uses and changes would also apply copyleft permissions.

An Open Source license reduces the friction involved in others wanting to use the project. So, for instance in a closed source system the individual or organization who would like to use or alter it would be required to contact the creator and ask for permission to study or use that code and would require specific instructions on how to do so. Since an Open Source project is intended to be shared and used by others from day one, and the license sets out its usage and distribution parameters, other are free to clone or fork the code and do what they like within the Open Source license restrictions.

Creating a voting system under an Open Source license does however have implications to the way that the project is built, maintained, and implemented; this is explored below.

7.3 Open Source Community Engagement

One of the benefits of an Open Source model is the contributions that the community can provide. The community can range from interested individuals, nonprofit organizations, university research labs, or other government organizations.

Just because a project can be used by others due to its Open Source nature does not mean it will be without concerted effort on their part. If the City of San Francisco decides to employ an Open Source approach for this project and expects contributions from the community, it will need to approach this project with a clear Governance model and a focus on enablement of those outside of the city. Enabling those outside of the project means that documentation will need to be a priority so that others can make use of the code. In addition, a focus needs to be placed on building and serving the community of contributors. This will include evangelism, education, communication, and organization. This cannot be taken lightly if the City of San Francisco expects any significant contributions from the community.

In our research it is clear there is a passionate community of people who want to contribute to this potential Open Source project. The size of this group and dedication to the project cannot be measured at this time. It is therefore expected that the contributions of the community during the initial requirements and build phase are primarily for user research input, advisory and limited user acceptance. It is not expected that coding will be done by the community during the build phase.

The involvement of the community after initial development is expected to mature along the lines of how the Apache Community[4], a very well established and successful Open Source community, maturity model works as shown in the table below.

Level	Description
1	The project has a well-known homepage that points to all the information required to operate according to this maturity model.
2	The community welcomes contributions from anyone who acts in good faith and in a respectful manner and adds value to the project.
3	Contributions include not only source code, but also documentation, constructive bug reports, constructive discussions, marketing and generally anything that adds value to the project.
4	The community is meritocratic and over time aims to give more rights and responsibilities to contributors who add value to the project.
5	The way in which contributors can be granted more rights such as commit access or decision power is clearly documented and is the same for all contributors.
6	The community operates based on consensus of its members but with the City of San Francisco community manager having final decision-making power. * *Amended from original Apache maturity model to reflect the City of San Francisco approach.*
7	The project strives to answer user questions in a timely manner.

Increasing Maturity

If a focus on enabling and involving the community is not a priority from the start the City of San Francisco runs the risk of another group taking the code, forking it (creating their own copy), and then creating a whole new project and community which evolves away from the needs of the City of San Francisco. An argument can be made that this separate project could be incorporated into the City of San Francisco voting system to improve it, but this would require extensive effort to reintegrate that code in a way that could pass the certification requirements for the State of California.

7.4 Open Source Governance Model

Open Source makes possible, and implies, that people outside of the original creators will be involved with the project. This being the case a proper governance model for the project is essential.

There are three primary governance models for Open Source projects like this one that were evaluated[5].

- ***"Benevolent Dictator for Life" (BDFL)****: Under this structure, one person or body (usually the initial author or creators of the project) has final say on all major project decisions. Python is a classic example. Smaller projects are probably BDFL by default, because there are only one or two maintainers. A project that originated at a company might also fall into the BDFL category.*
- ***Meritocracy:*** *(Note: the term "meritocracy" carries negative connotations for some communities and has a complex social and political history.) Under a meritocracy, active project contributors (those who demonstrate "merit") are given a formal decision-making role. Decisions are usually made based on pure*

[4] https://community.apache.org/apache-way/apache-project-maturity-model.html
[5] https://opensource.guide/leadership-and-governance/

voting consensus. The meritocracy concept was pioneered by the Apache Foundation; all Apache projects are meritocracies. Contributions can only be made by individuals representing themselves, not by a company.

- ***Liberal contribution:*** *Under a liberal contribution model, the people who do the most work are recognized as most influential, but this is based on current work and not historic contributions. Major project decisions are made based on a consensus seeking process (discuss major grievances) rather than pure vote, and strive to include as many community perspectives as possible. Popular examples of projects that use a liberal contribution model include Node.js and Rust.*

The City of San Francisco will be making a significant investment in this project to get it off the ground. It is also responsible for the very involved and specific certification process that the State requires. There is also the possibility that foreign players or those who wish to taint the process could become involved as part of the Open Source community. For these reasons, and for the purposes of this assessment, it has been assumed that the City of San Francisco maintains a position of final authority and therefore the Benevolent Dictator for Life model is best suited.

7.5 Open Source Governance – Key Roles

This section explains the key roles[6] within the model for Open Source development

7.5.1 Benevolent Dictator – Project Lead Role

The role of the Project Lead is to be the guiding hand on the overall project. This involves setting the strategic direction for the project while balancing the sometimes-competing view of the community. This role defines what level of control and authority those in the community can and should have. This is a very important role since if the community does not agree with decisions being made or the method of decision making they can fork the code and build a separate community. For this reason, the role is very focused on pleasing the community while keeping the project focus clear and on track.

In smaller projects this person is also the evangelist for the project and works hard to engage more and more folks in the project. For the City of San Francisco project this role is vital in building awareness and growing the community size. It is expected that evangelism and fostering engagement will become a near full time role and that this Project Lead role could be played by two individuals. One focused on community engagement and one administering the project direction and execution.

This role is likely to be fulfilled by an employee or employees of the City of San Francisco for the foreseeable future.

7.5.2 Committers

Committers are trusted individuals who have shown a commitment to the project either assigned or earned. They are typically programmers who have a deep understanding of all or parts of the application so that they can review code of others and commit code to the repository. Committers can sometimes manage the contributions of others not related to programming. An example of this would be around the production of documentation or feature and bug management.

For the City of San Francisco it is expected that these folks will initially be full time employees or vetted contractors since there is a security implication and a need for predictability of their contributions to meet election-based deadlines. They also need to understand the implications of the State certification process.

[6] Additional information: http://oss-watch.ac.uk/resources/benevolentdictatorgovernancemodel

7.5.3 Contributors

Contributors are those who contribute to the project, but are not or do not want to be Committers. They don't have access to commit code directly but can contribute through other tools like email and typically contribute things like bug reports, community administration and graphic design. As Contributors show commitment and an understanding of the project they can be considered for the Committer role.

7.5.4 Users

Users are those who have a need for the project. In this case users are the City of San Francisco and potentially other jurisdictions.

7.6 Software Management Tools

To work effectively with an Open Source community, effective tooling is required. These considerations will be most applicable for the later phases of the project once the initial build is created. However, it is prudent to start the project knowing that this will happen later in the lifecycle.

7.6.1 Code Management

The written code needs to be easily and securely accessible to those who may want to contribute. It should not require a lot of effort to take the code, modify it and then submit it back to the project. In addition, it needs to be robust and allow for many versions of the same code to be in flight at any one time possibly by many different people. It must also handle the bug and new feature submission, creation and release in a transparent and robust way.

7.6.2 Documentation Repository

For people outside of the project to be able to act the documentation needs to be robust and accessible. As with code the tooling should allow anyone to submit documentation in a controlled manor. It is often the case that community members may not have the skills to contribute code but can contribute by creating documentation. In many open source software projects, the source files for the documentation is stored in the same repository as the code itself.

7.6.3 Project Management

The tooling needs to effectively manage communication with the community, the documentation creation workflow and scheduling of code releases and release notes.

The most commonly used tool that covers all the needs listed above is GitHub (https://github.com/). It is assumed that this platform will be used for the City of San Francisco project, but vendor selection may change this assumption.

7.7 Programming Language

There are a variety of Open Source programming languages which could be used to create this project. The scope of this assessment is not to evaluate or determine that programming language. Explained here are three main considerations when picking a programming language. It is important to note that it is expected that more than one programming language will be employed based on the specialization required. For instance, interfacing with a scanner might require a language like C where general web development might be handled by a language like Java.

7.7.1 Community Support

When creating a new application there are many building blocks that already exist and do not need to be created again from scratch. An example of this is optical character recognition (OCR) engine. OCR turns text and images on paper into data in a computer. This software is time consuming and complex to create but already exists to enable different programming languages. A programming language with a vibrant community will likely have more of these reusable assets from which to choose.

7.7.2 Skillset Availability

The larger the community of people familiar with the programming language the more than can potentially engage on the project.

7.7.3 Maturity and Expected Longevity

City of San Francisco should employ a programming language which has a solid track record in large complex projects and is expected to be viable for many years, if not decades, into the future.

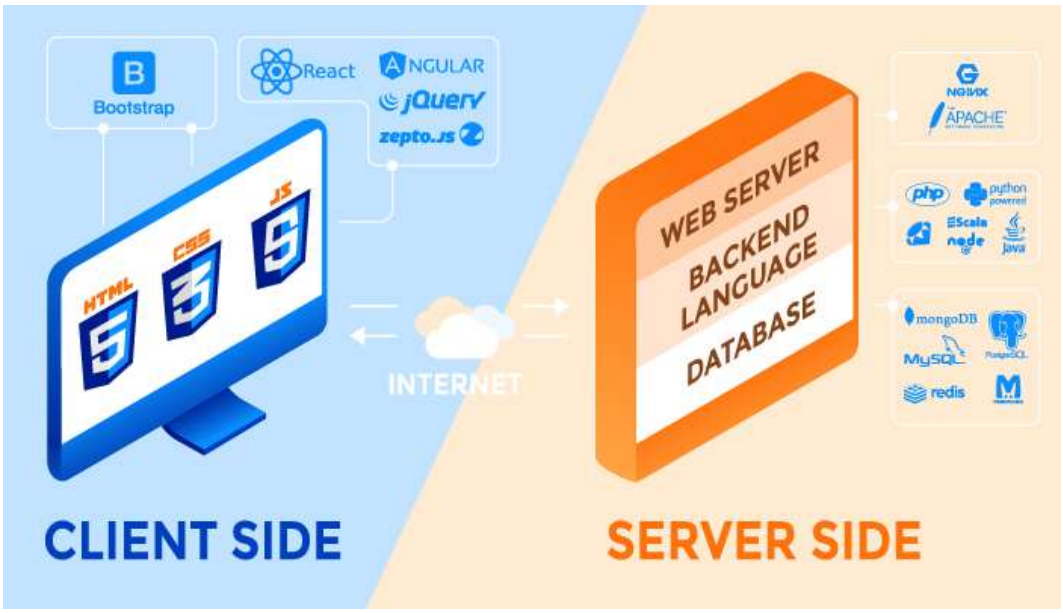
In a design phase of the project a deeper evaluation will need to be done to define not only the programming language but the complete application stack (section 4.8.1).

7.8 Infrastructure

From hardware to software there are a lot of different parts involved in the voting system. In a discovery phase of the project the reference architecture and specifics will be define.

7.8.1 Application Stack

An application stack represents all the software and server system components required to provide application functionality. Other than the Accessible Voting Devices, which will be assessed below, the application interface is expected to be presented via a web browser with a secure internet connection. Below is a representation of a typical stack[7].



[7] https://rubygarage.org/uploads/article_image/file/709/technology-stack-diagram.jpg

Not shown in this diagram above is the operating system which runs on the client and server and forms the interface to the physical computer hardware. To be a pure open source system the entire stack for this project needs to be Open Source including the operating system.

7.8.2 Hosting

In a client-server model, the server hosting can work two ways. The first is that the server is running in a private data center run by the City of San Francisco or a trusted vendor. The second alternative is that it can be hosted by a cloud provider who is responsible for the hardware and potentially the entire stack. This scenario is called public cloud because services are provided to many organizations from the same data centers. The leading cloud services providers have hosting options specifically addressing the increased security needs of government[8] [9] [10].

7.8.3 Accessible Voting Devices Application Stack

The Accessible Voting Devices needs to function as a stand-alone system due to the legislatively imposed restrictions on network connectivity. It will need to function *like* a tablet disconnected from a network. This means that the entire stack from operating system to storage must run independent of a server.

The Accessible Voting Devices will have some unique considerations due to its required interface to peripherals and local storage. These are not typical use cases for a web application. This means that the selection of an operating system will be an important one. The operating system is the part of the application stack where storage and peripherals are integrated. Depending on the hardware selected the stack could be a mobile development stack like Android or a typical web development stack. From a development stand point the web development stack makes leveraging code and skills sets across the other parts of the system easier. A mobile development stack may make peripheral access and storage more seamless.

7.8.4 Peripherals

The Accessible Voting Devices will need to provide reliable and secure connections to a set of peripherals for accessibility as well as a printer and scanner. The expected workflow today is that a user will go to an Accessible Voting Device, scan something to provision the proper ballot, make and validate selections on the device, and then print out a ballot to be scanned separately into a tabulation system. This analog-to-digital-to-analog process requires the Accessible Voting Device to have reliable and robust connections to the peripherals.

In addition to the Accessible Voting Device the voting system will also rely on scanners to extract votes from ballots. They are expected in precinct as well as centrally for mail in ballot tabulation. These devices are extremely important and must be reliable, secure and support flexible interoperability with the selected operating system.

7.9 Security

Generally, in an Open Source project, more people have access to view the code, which can lead to the discovery of vulnerabilities in the code sooner. When implementing an Open Source model where contributions come from the public, there needs to be a proper governance model to assess and approve code to ensure it is secure.

---

[8] https://aws.amazon.com/government-education/government/
[9] https://azure.microsoft.com/en-us/overview/clouds/government/
[10] https://www.ibm.com/cloud/federal

A secure Open Source project depends on active community engagement to continuously find bugs in the code. However, since the code would be free for anyone to use, the "tragedy of the commons" may occur, which means the community may take advantage of the shared resource, but rely on others to fix bugs out of their own self-interest. If this becomes the case, and the community loses activity over time, it can be possible for vulnerabilities to go unnoticed and possibly even exploited.

Another consideration for security is the nature of a pure Open Source solution. This project will be made up of newly written code along with existing Open Source projects, frameworks or libraries. For instance, to create the user interface a technology "stack" ***might*** include:

- Linux for operating system functionality
- Postgres for data storage
- Tomcat for the application server
- Java as the programming language

Each of these components of the technology stack are Open Source. This means that a community is busy enhancing and ensuring that each is secure. For active project like the ones above the code is being patched often. If consistent patching, maintenance and compliance checks are not done on the overall City of San Francisco Open Source voting project, security vulnerabilities can be introduced by any of the components of the technology stack. In a recent study by Black Duck[11] 67% of the Open Source applications they analyzed had vulnerabilities in the components used. A regular patching protocol must be defined and each change to the system will require a review by the Secretary of State.

Although there are many security benefits of Open Source platforms, there is always a non-zero risk of the system being compromised, no matter how rigorous you are in maintaining these systems. As an added security layer and integrity check, paper processes should be used in tandem with the electronic voting submissions to verify that the tallies have not been affected by system tampering. These include random manual audits conducted by hand, like the "one percent manual tally" required by California law (see Sec. 15360 of the California Elections Code).

Another important security consideration is the integrity of the supply chain for the hardware employed in the voting solution. The majority if not all the hardware likely to be employed in this project is manufactured in another country. Actors in those countries may have a desire to destabilize the election process in the United States to further their causes. This project is intended to use COTS (common off the shelf) hardware. The motivation for using COTS hardware is often to save money but it also might have the added benefit of obscuring the use cases from a foreign actor. There are reasons that COTS hardware may not be used in this project. For instance, California has laws that prevent the use of Wi-Fi in Accessible Voting Devices. Depending on Accessible Voting Device order volume it may make sense to have a manufacturer make a standard device without that chip present. This would not technically be COTS hardware anymore and extra scrutiny for supply chain would be advised.

The City of San Francisco must take several measures to ensure that the code and hardware is secure. Firstly, they must market and ensure an active Open Source community that is committed to ensuring the code is secure and rid of vulnerabilities. There must be a clear bug disclosure process, and a possible reward incentive for those who find bugs. The bugs must be frequently monitored, and be remediated in a timely manner to ensure the defects in the code or other Open Source projects are fixed. Additionally, there must be thorough quality assurance testing done on the code itself by running appropriate test scripts or hiring an external vendor to run security checks. This testing should have a documented timeline approach– e.g. a freeze period every so often for code to be reviewed and tested thoroughly, before new code is put into the code base. Also, special attention and

[11] https://www.blackducksoftware.com/Open Source-security-risk-analysis-2017

assurances should be in place to ensure the integrity of the supply chain for vital hardware used in the solution.

7.10 Change Management

It is assumed that any solution will be designed and developed with a user centric approach and should deliver an intuitive design. This means that the training needs for the system will be minimized.

There will however be a need to address change management at the start of this project and on an ongoing basis. It's very likely that after the first version of the system (or sub-systems) is implemented there will be subsequent enhancements / new features.

It is expected that through the development and certification of the systems documentation and training materials will be created. A process will need to be put in place to review the changes that are implemented version to version and then review the training materials and activities.

7.11 Procurement Timeline

Procurement is a major consideration in the project timeline and should not be overlooked. It is likely that much of the build process will go out for RFP. This process takes time. It should also be expected that the Open Source nature of the project will slow down normal procurement timelines. Since a project of this type and size has never been completed by the City before, and therefore has its challenges, it may be difficult to find a large number of vendors who will respond to RFP unless they are broken into many smaller RFPs – adding additional time. In other words, there may be plenty of available resources with the skills to support this initiative in the Bay Area from multiple vendors rather than one, so the RFP process may require a breakout into smaller RFPs in order for procurement process to run smoothly.

7.12 Certification Timeline

Certification is also a key factor in the overall build timeline. Vendors like Dominion are well versed in the certification process and therefore bring an expertise that is unlikely to be found in the market. It is therefore expected that the certification process will need constant attention. We have therefore assumed a Certification Project Manager role which will be constantly involved in shepherding the process. It is expected that this role can manage two certification components at one time. In addition, we have assumed partial engagement a technical team to tweak revise and remediate issues that arise.

The certification process is something that will slow ongoing development in addition to just the build process. The State of California will maintain an exact duplicate (shadow) of the currently certified system. Any changes to that system must be applied to the shadow version that the State has. This means that all changes must be documented and instruct the State how to patch or change the shadow system. This imparts a significant burden on an agile development process and means a less productive team as compared to a project this does not require this step.

8 System Build Considerations

This section explains the considerations for the system build. These have influenced how the plan for project has been designed and sequenced.

8.1 Speed to Value

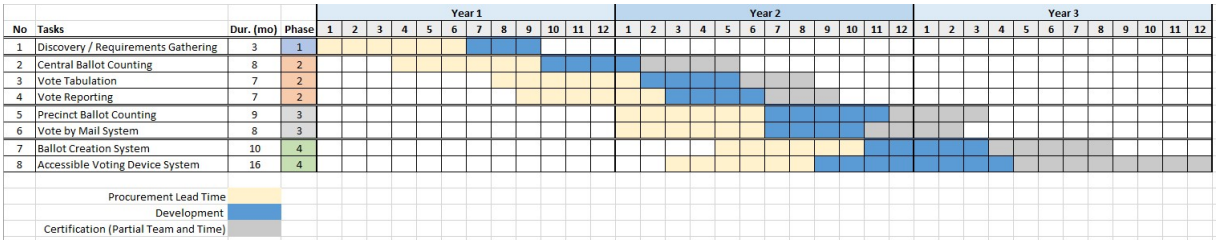
When building a system with many different constituent sub systems there are generally two approaches to roll out the system and therefore realize value. The first is a “big bang” approach. This means that the entire system will be completed and tested as a single system before implementing any parts. The major drawback is that the entire system needs to be complete which can mean that no value is realized during a multi-year build process. Over a long build process, it’s likely that each of the sub systems will be better understood and will therefore require refactoring. This can lead to a project which is delayed or never released due to trying to hit a moving target. It also means that the major challenges inherent in rolling out a complex system are not experienced until the end when the effort to remedy would be the greatest.

The second approach, which is preferable, is to realize value from sub systems as they are available and can add value.

The City of San Francisco currently uses a proprietary system supplied by an outside vendor. This means that interoperability with the existing system will be difficult and the City of San Francisco must look for synergy between sub systems which are sufficiently decoupled from other proprietary system parts.

Further investigation into integration with the proprietary system is needed during discovery but a good example of where this decoupling can work is with Central Ballot scanning. The input to this system is the ballot format already in use. This is something that can be analyzed, understood and supported by the new system. A well-designed ballot scanning system should not be strongly tied to a ballot format as well. Additionally, a Vote Tabulation system should be able to be decoupled if the existing vendor system can output the ballot counts in a standard machine-readable format which should be the case. This is phase of work is described as Phase 2 below and would constitute the first development effort.

With this focus on speed to value we see a development schedule like the following. A much more specific timeline with clear dependencies will be created during the discovery phase (Phase 1).



In addition to the *Phase I - Discovery* phase each of the sub system will also incorporate a short discovery phase with a more focused scope. This timeline assumes that the entity completing the discovery and development will be the same. If only *Phase – 1 Discovery* is undertaken and a different entity is used to create complete the subsequent phases, then this timeline will be longer.

The logical grouping and sequence of system build phases are as follows:

Phase 1	Discovery	Phase 1 is to develop a clear picture of what success looks like and how specifically the voter’s needs will be met. It also involves a deep dive into the critical success factors such as

		technical constraints, licensing, program operating model, procurement and possible coordination with other organizations or jurisdictions. Most significantly it will create the high level requirements and start to build out the near-term feature backlog.
Phase 2	Central Ballot Scanning, Vote Tabulation, Reporting	Phase 2 will involve building the system that scans the ballot, Vote tabulation and Vote reporting. These logically build on each other and provide a unit of work that should be able to work with an existing vendor solution.
Phase 3	Precinct Scanning and Remote Accessible Vote by Mail System	Phase 3 will focus on Precinct Scanning and Accessible Vote by Mail system. Precinct scanning will build upon the work that was done for the Central Ballot Scanning System. The accessible Vote by Mail System can be created separate from an existing vendor system if it can be made to print a ballot in the format of the current vendor ballot.
Phase 4	Ballot Creation and Accessible Voting Device	Phase 4 will be large, but the systems are closely tied together since the ballot format is closely tied to the display of that ballot in the Accessible voting device.

With this approach we believe value can be realized at the end of each of these phases.

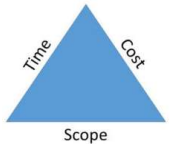
8.2 Agile vs Waterfall

The proposed approach assumes an agile project approach. Agile project delivery means different things to different people. For this report's purposes Agile is defined as:

> *relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.*[12]

In the past Waterfall delivery was common. Waterfall projects aim to clearly define the project requirements, cost and timeline before coding. It is done by having extensive discovery phases and robust requirements documentation. It typically involved a lot of interaction with the project sponsors and user up front to "lock down" the requirements. For large projects specifically in highly regulated industries like government this was desirable due to the need for well understood expenditures as well as the ability to clearly outline what would be delivered for that outlay.

Time, cost and scope were laid out and success was defined by implementing all the requirements in the time and costs outlined at the project start. Waterfall tries to fix the three sides of the feasibility triangle[13].



This approach creates a change averse process where deviations from the original specification are discouraged and disruptive. In addition, it puts a lot of weight on what was known at the project inception not what is discovered during the build process.

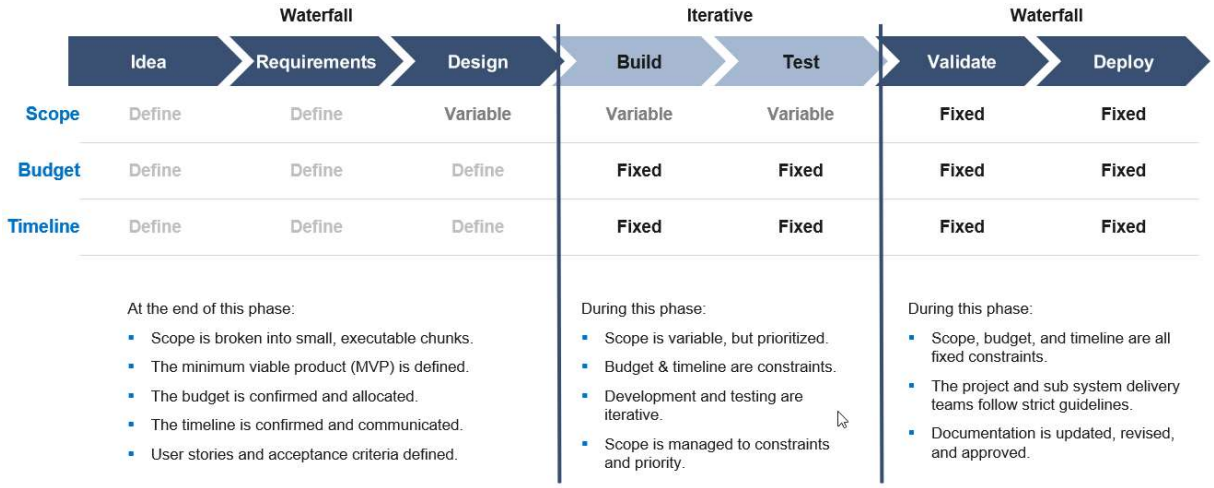
Agile delivery takes a different approach. It starts with the premise that a lot of learning will be done along the way and that change is inevitable and valuable. For this reason, it is highly iterative with the goal to constantly be demonstrating progress and the readjusting at the direction of the project stakeholders. Typically, the scope side of the feasibility triangle is loosely defined and is understood

[12] https://en.oxforddictionaries.com/definition/agile
[13] Source: https://edgeforscholars.org/wp-content/uploads/2017/03/Feasibility-triangle-1-300x231.jpg

to NOT be fixed. The focus is not on delivering all the features but constantly delivering the features which provide the most value.

Successful implementation of an agile approach requires an understanding of this trade off and a highly committed and constantly involved set of stakeholders who will constantly guide the development to the highest value features. It also involves the ability to fund projects without a completely defined set of features and functionality.

This project has hard constraints imposed by the State of California certification process and immovable dates around elections. For this reason, it is likely that the project will employee a hybrid agile process like the following.



8.3 Hardware

The premise for this project is that the design of a System would allow for the City to utilize Commercial off-the-shelf (COTS) hardware components rather than build System-specific hardware.

8.4 Certification Process

Any new voting system must go through a certification process with the Secretary of State to be approved and used in an election. The goals of this certification process are as follows:

- Ensure accuracy in recording and processing voting data
- Identify operational failures, in relation to storage, operation, transportation, and maintenance environments
- Test system performance and function under normal/abnormal conditions
- Ensure completeness and accuracy of system documentation and configuration management

California does not require federal testing of the new voting system since the Secretary of State adopts testing standards that meet or exceed the federal voluntary standards set by the Federal Elections Assistance Commission (EAC).

To approve a new voting system, and its components:

Step	Description
1	The City must fill out a Voting Technology Application and provide sufficient technical documentation and submit to the Secretary of State • Part 1: Request for Approval & System ID • Part 2: Ownership & Control • Part 3: References

	• Part 4: Index of System Tech Documentation • Part 5: System Change Log
2	The City must submit a deposit to the Secretary of State for approximately $360,000 deposit to hire external consultants for testing purposes, in addition to costs for obtaining hardware for 2 cycles for testing purposes, printing ballots ($25,000), subscribing to testing platforms, etc. • If less than $360,000 is used, then the remaining funds will be returned to the City
3	The City must develop testing plan with State Approved Testing Agencies (S-ATA) and schedule the testing sessions
4	The Secretary of State must ensure that the following testing has been completed: • Functionality Testing – test basic core functionality, system capabilities, and use of specific technology or configuration • Hardware Testing – test equipment operability and data accuracy (non-operational testing in environmental facility, operational testing in part environmental facility and part standard lab/ shop facility) including bench handling, temperature testing, running for long durations, etc. • Software Testing – code review test programming completeness, consistency, correctness, modifiability, structure, and traceability • System Integration – test systems integration of all processes and system components - include auditing checks, witness of build, security testing, volume testing, etc. • Documentation – review all documentation for completeness and accuracy around manufacturer practices of QA and Configuration Management
5	S-ATA will present test results back to the Secretary State in a staff report and ensure compliance with voting system regulations
6	A public hearing is held for people to express their views on the proposed system
7	Secretary of State makes final decision on whether to approve the voting system

Timing for approving a new voting system may take from 6 months – 1 year, plus an additional 10 weeks for procurement of consultant services to support testing procedures.

Components of the voting system can be separately submitted for approval, but must prove interop-ability with current systems (e.g. providing detailed plan for testing) for that component to be tested with the rest of the current voting system to ensure the end-to-end system works. With an agile approach, the system may be continuously changing. Modifications to the approved system must still be submitted to the Secretary of State and go through an 'Administrative Approval' process.

To approve a *modification* to approved system:

Step	Description
1	The City must fill out only "Administrative Approval" of Section 1 of the Voting Technology Application to justify need for change, and provide thorough documentation
2	Secretary of State reviews changes, and determines scope of re-testing. Minor modifications ((e.g. bug fixes) to voting system can be subject to limited testing if it does not affect compliance for: • Performance of voting system functions • Voting system security/privacy • Flow of system control • Way ballots are defined, interpreted, or processed
3	Testing is completed by S-ATA. Note: cost for testing modification does not tend to be substantial since most is done in-house by Secretary of State
4	Test results are reviewed, and Secretary of State decides whether to approve the system modification

Timing of modification approval is dependent on the workload of the Secretary of State and nature of the change. It can be a short as one day, or much longer depending on the complexity. There is a

risk that if a system does not pass initial and secondary rounds of testing, the approval process may have to halt entirely because there may not be enough resources to compete in the timeline. An application must be resubmitted once all has been fixed, which will delay the process of certification.

Additionally, there is concern that the current certification process may not be the most efficient for handling agile projects. Because all public requests and code changes that are incorporated into the system will need to be certified with the Secretary of State, certification may be delayed unless there are legislative changes to the current process. With the current process in place, there will need to be a well-defined schedule for when to submit the modified changes to the Secretary of State based on consistency of bug fixes or added new features from the Open Source community for it to run smoothly. However, it is the hope that the certification process for a new or modified voting system will generally be faster because of the robust documentation that comes with the nature of an Open Source project.

8.5 Accessibility and Usability

For in-precinct and remote accessible vote-by-mail systems to be accessible, the design must allow for both direct access without assistance, and indirect access with assistive technologies for persons with disabilities to ensure every individual has the ability to cast a private and independent vote.

A new voting system must meet state and federal accessibility requirements. However, following the accessibility guidelines may not account for usability for all persons or disabled types. The state/federal accessibility requirements includes:

- Title II of the Americans with Disabilities Act
  - https://www.ada.gov/cguide.htm
- Voting Rights Act
  - http://library.clerk.house.gov/reference-files/PPL_VotingRightsAct_1965.pdf
- Voting Accessibility for the Elderly and Handicapped Act
  - http://codi.tamucc.edu/archives/legislation/voting.access.htm
- National Voter Registration Act
  - https://www.justice.gov/crt/title-42-public-health-and-welfare-chapter-20-elective-franchise-subchapter-i-h-national-voter
- Help America Vote Act
  - http://elections.cdn.sos.ca.gov/hava.pdf
- Voluntary Voting System Guidelines 2.0*
  - https://www.eac.gov/assets/1/6/TGDC_Recommended_VVSG2.0_P_Gs.pdf

* Currently a work in progress and not a mandatory requirement, but can be used for testing to determine if the systems provide all of the basic accessibility capabilities required of these systems

Voter input should be considered at every stage of development of this new voting system. For example, feedback should involve the disabled community and all features should be tested by a sample of disabled community.

During the design phase of the project tablets will be evaluated as an option for COTS voting system hardware, currently they are seen as non-compliant with VVSG standards by current accessibility groups. It does not allow for those with motor control and dexterity limitations to use the system. The only way for a tablet to be used is if there are additional accessories that can be plugged into the tablet.

According to the Disability Rights California, hardware must be chosen very carefully for this new voting system. Instead of having one accessible machine at every precinct, their recommendation is to have all machines be accessible at every precinct to ensure voters (including those voters who may not want to disclose they have a disability) can use any machine. Additionally, having all machines be accessible gives voters with disabilities the same opportunities as other voters do. There are many desirable attributes to a voting system that can lead to accessibility and usability for all.

The complication with this is that the City and County of San Francisco is currently a 'paper-based' jurisdiction, and so legislative change may be required to have an entirely electronic voting system. In addition to federal and state accessibility requirements mentioned above, the following are considerations that should be included to create the most independently usable system:

- **Sip & puff:** The mouth-controlled input provides users who cannot move their arms with a simple and effective way to use their breath to control a device, such as their power wheelchair or computer.
- **Keyboard for write-in votes:** Many people with disabilities are unable to type in names of write-in candidates using the touchscreen either because they can hit one large button to cast their ballot by using a big part of their hand or even face to choose a candidate on the ballot, but cannot type on a touchscreen keypad. A manual keyboard should be readily available.
- **Voice Activated:** Voice input for voters who have difficulty using their hands
- **Synchronized audio & visual display:** When synchronized speech and audio are engaged, a voice reads each word as it is displayed. Adjustments to change the volume and tempo should be available to assist voters with intellectual and developmental disabilities, voters with learning disabilities, voter who had traumatic brain injuries and voters who had a stroke.
- **Joystick:** Some voters with disabilities may need to use a joystick to navigate the touchscreen component if they cannot operate the touchscreen because they are unable to raise their hand or cannot accurately hit their selection due to fine motor control limitations or involuntary movements.
- **Tecla switch compatibility:** The Tecla Switch is a wireless device that lets a person with limited to no hand movement control electronic devices, such as a smartphone, tablet or computer, and the driving controls of their power wheelchair using their external switches.
- **Tactile buttons:** An access feature provided as an alternative to touch screen input. It provides key/controls that can be felt in contrast to a touch screen, which provides no mechanism to "feel" the difference between selections.
- **Remote Accessible Vote-By-Mail:** An electronic ballot in accessible format that is either e-mailed to a voter with a disability or downloadable from a secure website that a voter can read and mark using their familiar assistive device (e.g. screen reading software, mouse keys, etc.), print it and mail it back to the county elections officials. Although this is a requirement regulated by state law, format to the e-ballot can change to be made for a more useable experience.

Contributors from the Open Source community may not be aware of mandated accessibility requirements for hardware and software voting platforms when submitting bug fixes, new features, or any platform change. Therefore, if the City elects to utilize Open Source software to maintain the code, it is critical to ensure that there is proper governance around managing the submissions from the Open Source community to ensure that all submissions go through appropriate regression testing and accessibility checks before merging their code back into the source code. The assigned community manager must work with an entity qualified to provide technical assistance in and vetting of Federal and State electronic and physical access compliance as part of this facilitation, along with appropriate user testing.

9 System Maintenance Considerations

Once the system is built and certified there will be activities required to support elections as well as maintain the software, implement new features, do bug fixes and maintain the equipment. This section explains the needs of that phase of the product lifecycle. The implementation of new features would be at City of San Francisco's discretion. After each election, the City of San Francisco would have a choice as to— (1) add no new features, (2) add minor enhancements or bug fixes, or (3) add major enhancements.

9.1 On-going Operations

Although the system is tested and implemented before every election, it must endure ongoing maintenance to ensure that it performs accurately and can be changed to adapt to new regulations and standards that may be put into place. There will still be continuous oversight of on-going community engagement to maintain bug fixes and to add new features. The community manager and team that oversees ongoing community contributions and monitors the testing of systems must remain fully engaged even during any non-election year. A systemized process must be in place to ensure the voting system will be ready for the upcoming election. Timing of these annual maintenance routines will vary based on the complexity of modifications to the hardware/software.

9.2 Physical Storage Space

Since COTS hardware is purchased, physical storage space must continue to house all equipment when it is not in use. The project will double the number of accessible voting devices that are currently operated today with the purpose of reducing the wait time to use an accessible voting device. Based on observation of the current storage facilities, the space seems sufficient to house the expected number of voting machines needed for future elections – so no additional costs will need to be considered.

9.3 Per-Election Operations

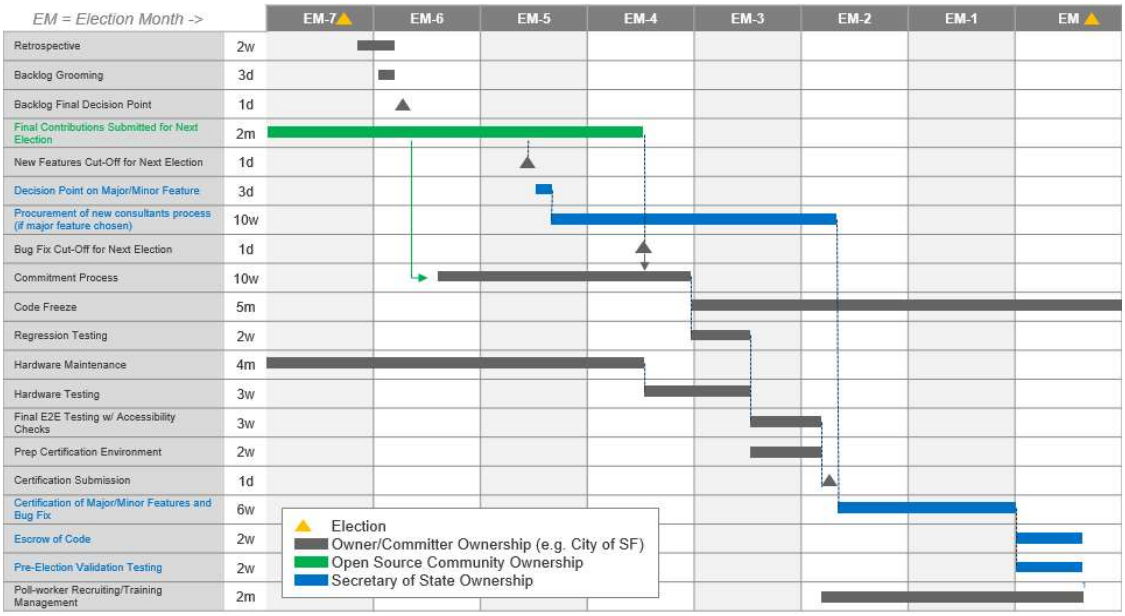
In between election cycles, there must be a scheduled order of activities to maintain voting system operations and to improve the voting system before the next election occurs. The following diagram and descriptions that follow explain those tasks and also the critical path that would have to happen between cycle if, for example, there was two elections per year.

As per the following diagram, after an election, a retrospective must occur in order to have an open discussion on what could be learned and enhanced from the previous election. The management team must gather key takeaways from that discussion, and update/prioritize the backlog with a list of items of which they believe are important to have the Open Source community work on.

As contributions are submitted, a final cut-off date for new features and bug fixes must be set in order for those enhancements to be considered for incorporation into the upcoming election. This cut-off date is also to ensure there will be sufficient time for these potential enhancements to go through all testing measures and certification approval procedures before the next election.

After this cut-off date, a code freeze occurs, and the submissions are reviewed by the Secretary of State to determine if they will be considered a "major feature". If deemed major, then it's assumed that additional staff will be contracted to manage the certification testing process. According to Deputy Secretary of State, the procurement of these consultants alone can take up to 10 weeks. Meanwhile, all necessary testing (e.g. regression testing, hardware testing, accessibility testing, etc.) must be complete before applying for certification to the Secretary of State.

Once the certification application is sent to the Secretary of State with the requested modifications, the certification process and testing is underway, and the Secretary of State decides on whether to approve the changes. Additionally, poll-worker training and recruiting occurs simultaneously. After changes are approved, the code sits in escrow and additional validation testing occurs prior to the election to ensure the voting system is functioning properly and ready for use.



Process Step descriptions

Activity	Ownership	Description
Retrospective	City of SF	Understand lessons learned from previous election cycle to enhance process.
Backlog Grooming	City of SF	Review items in the backlog and re-prioritize if needed.
Backlog Final Decision Point	City of SF	Final decision on prioritization of backlog.
Final Contributions Submitted for Next Election	Open Source Community	Last contributions from the Open Source community that will be considered for the upcoming election. Contributions can continue coming in after this date, but will be considered for the subsequent elections.
New Features Cut-Off for Next Election	City of SF	Last date for submission of new feature contributions to be considered for the upcoming election.
Decision Point on Major/Minor Feature	Secretary of State	City of SF works with the Secretary of State to determine if an included feature is considered major or minor to determine resources for testing during the certification process.
Procurement of new consultants process (if major feature chosen)	Secretary of State	Secretary of state will procure new consultants for testing major features of voting system.
Bug Fix Cut-Off for Next Election	City of SF	Last date for submission of bug fix contributions to be considered for the upcoming election.
Commitment Process	City of SF	Contributions will be reviewed and decisions about whether that contribution will be merged back into the source code will be made during this process.
Code Freeze	City of SF	The source code will not be changed or added to during this time for testing purposes.
Regression Testing	City of SF	Testing changes to the code to make sure old code still works with new changes, and core functionalities are not affected.
Hardware Maintenance	City of SF	Ensure hardware is working properly for testing purposes and for election day.

Activity	Ownership	Description
Hardware Testing	City of SF	Test hardware to ensure it complies with regulation and functions properly.
Final E2E Testing w/ Accessibility Checks	City of SF	Test system as a whole to ensure it complies with regulation (including accessibility requirements), and functions as a whole.
Prep Certification Environment	City of SF	Create separate environment for certification testing.
Certification Submission	City of SF	Date to submit voting technology application for certification to Secretary of State.
Certification of Major / Minor Features and Bug Fix	Secretary of State	Process to certify components of voting system that have upgraded/changed including testing with S-ATA Labs for approval.
Escrow of Code	Secretary of State	Code sits in a siloed platform.
Pre-Election Validation Testing	Secretary of State	Mirror election environment to validate testing of new voting system to ensure it functions on election day.
Poll-worker Recruiting / Training Management	City of SF	Recruit and create training materials for poll-workers on new system features, logistical process, and how to support voters on election day.

10 Appendix A - Acknowledgements

Without the support from these individuals, the completion of this project would not have been possible. Sincere thanks and deep appreciation go out to all for their efforts.

Stakeholder	Department, Title
John Arntz	Dept. of Elections, Director
Ryan Macias	Elections Assistance Commission in D.C.
Nicole Bohn	Mayor's Office on Disability, Director
Melissa Whitehouse	Mayor's Budget Director
Chris Jerdonek	SF Elections Commission, President,
Fred Nisen and Bill Hershon	Disability Rights California, Lead Attorneys
Jackson Muhirwe	Dept. of Technology, Director of Cybersecurity Services
Eddie Gardner	Dept. of Technology, Cyber Security Architect
Saul Melara	Dept. of Technology, Former CTO
Linda Gerull	Dept. of Technology, Chief Information Officer
Erin Lauridsen	Lighthouse for the Blind, Director for Accessible Technology
Carrie Bishop	Dept. of Technology, Chief Digital Services Officer
Matthias Jaime	Commissions of Information Technology, Director
Christina Lutz-Hatfield	Dept. of Technology, Web Accessibility Consultant
Jane Gong	Dept. of Technology, Deputy Chief Digital Services Officer
Henry Jiang	Dept. of Technology, Senior Programmer
Ken Bukowski	Deputy City Administrator – Chief Financial Officer
Todd Rydstrom	Deputy City Controller
Susan Lapsley	Deputy Secretary of State
Aaron Nevarez	LA County Registrar-Recorder/County Clerk, Division Manager of Government and Legislative Affairs
Tim McNamara	LA County Registrar-Recorder/County Clerk
Kenneth Bennett	LA County Registrar-Recorder/County Clerk
Jason Lally	Data Service Manager
Jay Nath	Chief Innovation Officer
Neil McClure	Contractor involved in Travis County Project
Jaci Fong	Office of Contract Administration, Director and Purchaser
Valerie Coleman	Dept. of Aging and Adult Services
Tonia Lediju	Controller's Office, Audit Division
Larry Bafundo	18F
Brent Turner	California Association Voting Officials
Brian Fox	California Association Voting Officials
Tim Mayer	California Association Voting Officials
Gregory Miller, John Sebes	OSET Institute, COO
Josh White	Deputy City Attorney - Elections

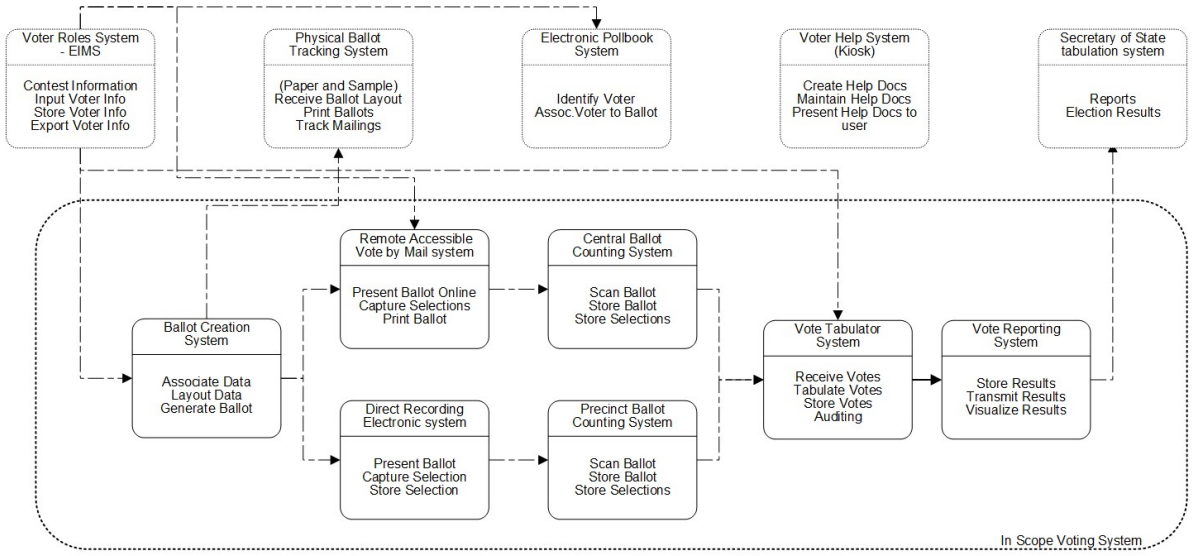
11 Appendix B - Proposed Voting System

11.1 Scope of Voting System

Based on Section 301(b) in the Help America Vote Act (HAVA), a voting system must be able to define the ballot, cast and count votes, report and display results, and maintain/produce an audit trail. Defining the requirements for an Open Source voting system is not in scope for this feasibility study. This poses a challenge when trying to determine the scope, capabilities needed, cost and timeline to deliver one. To address this, we created a generalized model of a voting system. This is based off many different inputs including but not limited to the following sources:

- San Francisco Open Source Voting System Technical Advisory Committee (OSVTAC) https://osvtac.github.io/recommendations/
- US Elections Assistance Commission - VOLUNTARY VOTING SYSTEM GUIDELINES - https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines/
- San Francisco Local Agency Formation Commission Study on Open Source Voting Systems – Final Report http://sfgov.org/lafco/Modules/ShowDocument.aspx?documentid=52577
- California Secretary of State's – California Voting System Standards - http://elections.cdn.sos.ca.gov/pdfs/california-voting-system-standards.pdf
- Travis County Texas STAR vote based system - http://sfgov.org/electionscommission/sites/default/files/Documents/meetings/2017/2017-02-15-commission/STAR-Vote%20Statement%20of%20Intent.pdf
- California Voting System Standards - http://elections.cdn.sos.ca.gov/pdfs/california-voting-system-standards.pdf
- California Elections Code - http://leginfo.legislature.ca.gov/faces/codesTOCSelected.xhtml?tocCode=ELEC&tocTitle=+Elections+Code+-+ELEC
- Uniform Vote Counting Standards - http://www.sos.ca.gov/elections/uniform-vote-counting-standards/
- Voting Modernization Board - http://www.sos.ca.gov/elections/voting-systems/laws-and-standards/voting-modernization/
- Upwards of 40 meetings with special interest groups, the City of San Francisco departments, other jurisdictions.
- OSET Institute - http://www.osetfoundation.org
- Free and Fair - http://freeandfair.us/
- Democracy Live - http://democracylive.com/
- LA County VSAP - http://vsap.lavote.net/process/

Please see attached diagram for what components of the voting system is in scope and out of scope for the purposes of this assessment.



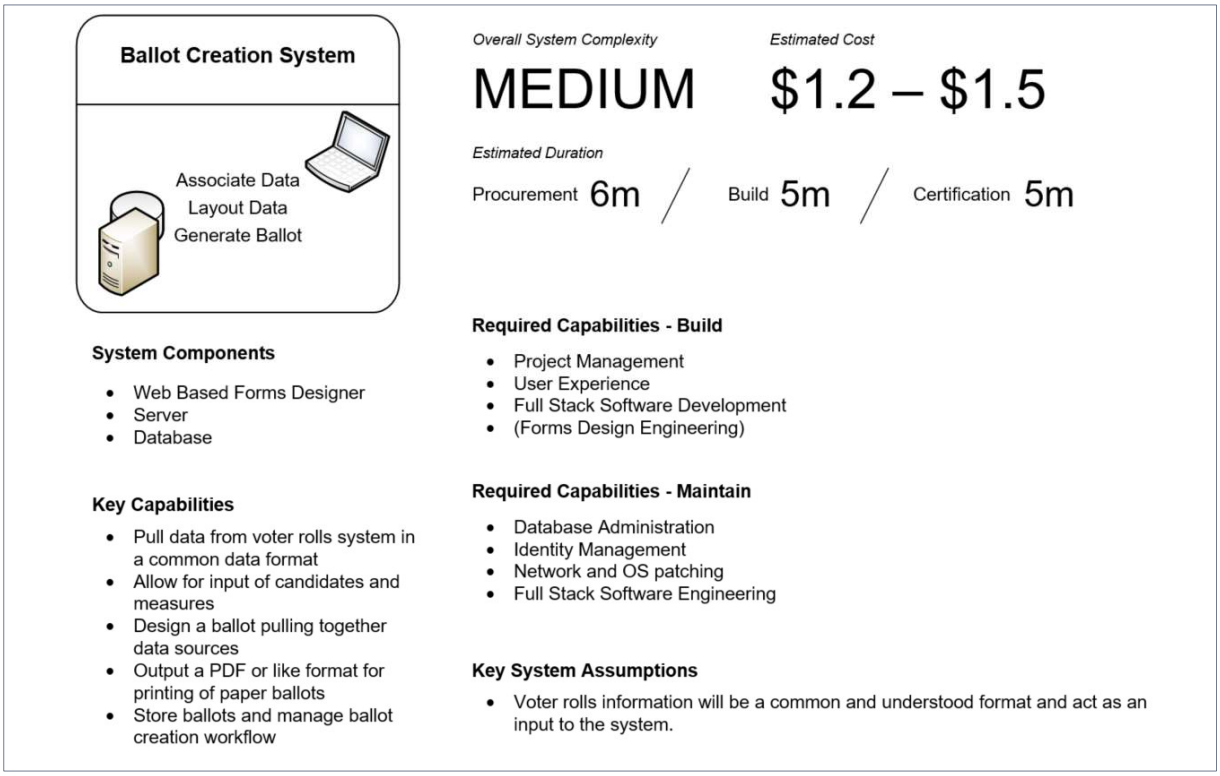
11.2 Voting System Sub-systems

As stated previously the requirements gathering phase of this project has not been complete. As such the following system overviews were created as part of this assessment and provide a high level view of what the sub system that make up the entire voting system look like. This information is supplemental and matches up with the sub systems described in section 3.2 and shown in the diagram on 3.1.

The diagram above is a logical representation of the sub systems involved in a full voting system. Each subsystem is made up of the hardware and software required. Those systems inside the dotted line are considered in scope for this assessment. Below is the information about the in-scope and out-of-scope systems.

11.2.1 Ballot Creation System

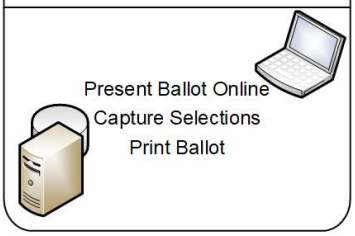
This system will store information about the issues to be voted on and. In addition, it will define the layout of the ballot both electronically as well as for paper ballots. This will feed ballot information into the Remote Accessible Vote by Mail System and the Accessible Voting Device system.



11.2.2 Remote Accessible Vote by Mail System

A remote accessible vote-by-mail system provides a digital ballot to military and overseas voters and to voters with disabilities. The user will identify themselves and this information will be validated against voter role information and then they will be presented a ballot to vote. When they have voted they will print a ballot and mail that ballot into the City of San Francisco.

Remote Accessible Vote By Mail System



Overall System Complexity

MEDIUM

Estimated Cost

$985K – $1.2M

Estimated Duration

Procurement 6m / Build 4m / Certification 4m

System Components

- Web Interface
- Server
- Database

Key Capabilities

- Store Ballot Layouts
- Identify user and present proper Ballot layout
- Present Ballots in multiple languages and in variable resolutions
- Allow user to make selections
- Allow user to verify selections
- Print Ballot with selections

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack Software Development
- Security

Required Capabilities - Maintain

- Network and OS patching
- Full Stack Software Engineering
- Security

Key System Assumptions

- Website which allows people to make selections via a web browser and then print this out and mail it in.
- Can potentially share a lot of code with the Accessible Voting Device System. Assumes both projects leverage the same layout and accessibility work.

11.2.3 Accessible Voting Device System

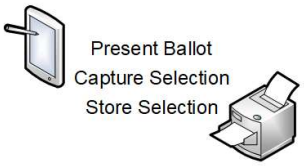
This system will be the hardware, software and furniture required to allow a person to cast a vote accessibly in a polling location. It will have extensive considerations to support accessibilities by creating a flexible, usable experience for the user. This system will augment, not replace, existing paper ballot processes within precinct voting locations.

The expected process it will support is as follows:

- User checks in at a polling location and is provided a piece of paper which represents the ballot that they should be presented
- User approaches an Accessible Voting Device and inputs the paper to a scanner device
- The appropriate ballot is provisioned and presented to the user
- User makes selections and is asked to verify choices
- When the user indicates that they are done with the process the printer attached to the Accessible Voting Device prints out a completed ballot.
- The user takes this printed ballot to a scanning device which records their vote to the tabulator staged in each precinct location.

This is the most complicated and costly system based on the high number of devices, interface between the accessibility and input output peripherals and required furniture. It also requires an extreme focus on usability and security.

Accessible Voting Device System



Overall System Complexity

HIGH

Estimated Cost

$4.5 – $5.5

Estimated Duration

Procurement 6m / Build 8m / Certification 8m

System Components

- Display Device with OS and local storage
- Special Needs Peripherals
- Printer
- Vote Marking Application
- Stand for device with privacy protection

Key Capabilities

- Store Ballot Layouts
- Identify user and present proper Ballot layout
- Present Ballots in multiple languages and in variable resolutions
- Allow user to make selections
- Allow user to verify selections
- Store vote selections
- Print Ballot with selections

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack/Mobile Software Development
- (Peripheral Design and Interface)
- (Device / Software Security)
- (Furniture Design)

Required Capabilities - Maintain

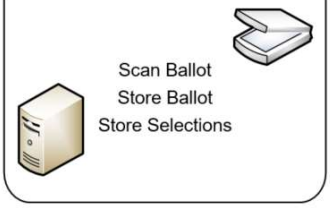
- Device / Furniture and Peripheral Maintenance
- OS patching
- Full Stack/Mobile Software Engineering

Key System Assumptions

- Accessible Voting Device System will use attached scanner to provision the ballot dynamically
- Accessible Voting Device System will not have network connectivity but will store a record of each vote locally for later verification
- Printer will output in a format that the Ballot Counting System can read clearly

11.2.4 Central Ballot Counting System

This system is used to scan paper ballots from the standard vote by mail process. It will use industrial scanners to extract the selections from ballots, store an image of the ballot in addition to the selections, and format them into a format compatible with the tabulation system. In addition, it will collect the paper ballots for audit purposes.



Overall System Complexity

MEDIUM

Estimated Cost

$950K – 1.2M

Estimated Duration

Procurement 6m / Build 4m / Certification 4m

System Components

- Scanner
- Scanning software
- Physical ballot storage
- Computer with redundant storage

Key Capabilities

- Scan ballots to extract selections
- Physically Store ballots
- Redundantly store ballot selections
- Transmit in process results to central voting system
- Format results for ingestion into a central tabulation system

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack Software Development
- (Scanner Software Interface)

Required Capabilities - Maintain

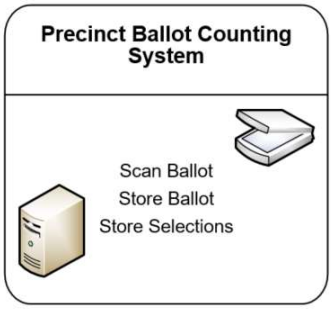
- Network and OS patching
- Full Stack Software Engineering
- (Hardware Maintenance)

Key System Assumptions

- Costing assumes that this project is done reusing work from the precinct ballot counting system

11.2.5 Precinct Ballot Counting System

This system will scan paper ballots from both the standard and Accessible Voting Device. It will use a scanner to extract the selections from ballots, store an image of the ballot in addition to the selections, and format them into a format compatible with the tabulation system. In addition, it will collect the paper ballots for audit purposes. It will have functionality to transport the results securely and redundantly back to the City of San Francisco for input into the Vote Tabulator System.



Overall System Complexity

MEDIUM

Estimated Cost

$1.1K – 1.4M

Estimated Duration

Procurement 6m / Build 5m / Certification 4m

System Components

- Scanner
- Scanning software
- Physical ballot storage
- Server with redundant storage

Key Capabilities

- Scan ballots to extract selections
- Physically Store ballots
- Redundantly store ballot selections
- Format results for ingestion into a central tabulation system

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack Software Development
- (Scanner Software Interface)

Required Capabilities - Maintain

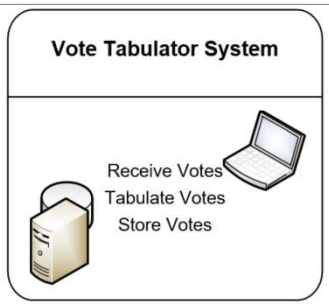
- Network and OS patching
- Full Stack Software Engineering
- (Hardware Maintenance)

Key System Assumptions

- The system will be simple enough that someone with 2 hours training can effectively use it.

11.2.6 Vote Tabulator System

This system will take vote selections from the Ballot Counting systems in a common format. It will tabulate results as well as put the data into long term storage formats.



Overall System Complexity

LOW

Estimated Cost

$850K – 1M

Estimated Duration

Procurement 6m / Build 3m / Certification 3m

System Components

- Web Interface
- Server
- Database

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack Software Development
- Database Engineer

Key Capabilities

- Receive votes from ballet counting systems in a common format
- Transform votes into a common and scalable data format
- Store votes
- Provide basic vote count information once stored
- Ingest information from the voter roles system and match to vote information to provide useful information for analysis in the reporting system

Required Capabilities - Maintain

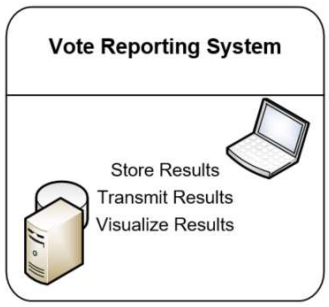
- Database Administration
- Identity Management
- Network and OS patching
- Full Stack Software Engineering

Key System Assumptions

- System will be designed in a highly fault tolerant and redundant way.

11.2.7 Vote Reporting System

This system will ingest vote data and provide a variety of reports and visualization to understand the vote data. It will also apply any required data retention policies. It will provide an interface for sending data to State or Federal vote reporting systems.



Overall System Complexity

LOW

Estimated Cost

$860K – 1M

Estimated Duration

Procurement 6m / Build 3m / Certification 3m

System Components

- Web Interface
- Server
- Database

Required Capabilities - Build

- Project Management
- User Experience
- Full Stack Software Development
- Database Engineer

Key Capabilities

- Ingest Vote Information
- Store votes and manage any data retention policies
- Provide a variety of reports and data visualization interfaces to understand voting data
- Send data to other vote tallying systems like the national vote tally systems.

Required Capabilities - Maintain

- Database Administration
- Identity Management
- Network and OS patching
- Full Stack Software Engineering

Key System Assumptions

- System will manage the votes information for many years so security with be crucial.

11.2.8 Voter Roles System – EIMS – Out of Scope

This is an input to the voting system but not in scope.

The Voter Roles system contains information about those who have registered to vote and information about them to determine what they are eligible to vote on. Information from this system feeds the Ballot Creation System and the Remote Accessible Vote by Mail system. It is also expected to interface with the Vote Tabulator system to allow for vote information to be analyzed by demographic.

11.2.9 Physical Ballot Tracking System – Out of Scope

This system is out of scope.

This system will manage the processes related to delivering the paper ballots and sample ballots to all voters requiring them.

11.2.10 Electronic Pollbook System – Out of Scope

This system is out of scope.

An electronic pollbook system would automate the process of people checking in at a voting location. These systems can potentially also automate the provisioning of the Accessible Voting Devices.

It is assumed that the current paper-based system will continue in use therefore this is out of scope.

11.2.11 Voter Help System – Out of Scope

This system is out of scope.

This system would reside in a polling location and can provide a help kiosk to those with questions on vote related topics. This would allow a large precinct to require less resources to support an election.

11.2.12 Secretary of State Tabulation system – Out of Scope

This system is out of scope.

This system is the one from which reporting for the State or Federal elections is done. The Vote Reporting system will provide information for this system.

12 Appendix C - Capability Model – Overview

This section introduces the capability model concept. This framework was used to clarify what it takes to deliver a successful program and also enable the assessment of options against a common set of capabilities. The structure of the model is in 5 parts:

- **Foundational:** Similar to most major projects, it is important to have budget management, product management, program management, and executive sponsorship skills on the to oversee the project progression.
- **Software Specific:** There are software-specific skills needed to develop the voting system software, including key software engineering and Open Source community management experience.
- **Hardware Specific:** Additionally, hardware-specific needs are mainly around planning logistics and operations to support the COTS hardware purchased.
- **Overall System:** are overall system skills that are needed in order make sure the management of the project runs smoothly and passes certification.
- **Personnel:** Personnel management, more specifically poll-worker training and recruiting, must also be taken into consideration in order to make sure voters have support at the precincts on Election Day.

Explanation of key:

- Denoted on each capability is phases of build, run, and maintain that it is relevant for.
- Capabilities that are highlighted by an icon in the diagram to show that it can be supported by the Open Source community.
- After observation from the City of SF's capabilities, certain capabilities are outlined in red to show what can be procured from external vendors rather than relying on the City to build those capabilities on their own.

Each capability is described in detail below.

Capability	Description
Foundational	
Budget/Cost Management	The ability to plan and control the project budget throughout the project lifecycle to help reduce the chance to go over budget.
Program Management	The ability to manage several related projects to end-to-end build and deployment of the new voting system.
Sponsorship/Executive Ownership	The ability to own the voting system on behalf of the City, including defining the project being accountable for the success of the project, including providing the direction and funding for the project.
Product Management	The ability to be the voice of the consumer of the product and being responsible for the strategy, roadmap, and prioritization of the backlog.
Software	
Software Engineering	The ability to develop the features of the voting system, and to test the system.
User Experience	The ability to design the user interface for the ballot-marking portion of the voting system to make a good experience for the voter.
Open Source Community Management	The ability to facilitate and foster the community of users on an Open Source platform to engage in source code building and ongoing bug fixing. Additionally, the platform must be marketed properly, and communication between users much be managed through the appropriate channels.
Change Management	The ability to prepare, equip, and support the City to successfully adopt change from current voting system to the new voting system and its processes to drive a streamlined voting process.
Hardware Specific	
Hardware Logistics/Operations Management	The ability to work with key vendors to manage hardware resources, such as planning logistics around storage and transportation to precincts for election day.
Hardware Maintenance	The ability to ensure hardware is working properly for testing purposes and for election day.
Hardware Design Specification	The ability to create design requirements to understand hardware needs before choosing a vendor.
Procurement	The ability to manage the process of acquiring contracting services and buying hardware from external resources.
Overall System	
Vendor Management:	The ability to manage potential vendors that provide COTS hardware or contractual relationships with consultants that may support the software engineering division on this project.
Security:	The ability to support security measures to assure mitigated risk around Open Source software and hardware.
Systems Administration:	The ability to support the maintenance, configuration, and operation of the computer systems/servers.
Certification Management/Liaison:	The ability to act as a liaison between the City and the Secretary of State to manage the certification process, including providing documentation and application needed to certify the new voting system.
Compliance Management:	The ability to ensure the new voting system is in accordance with established guidelines, requirements, and legal regulations.

Capability	Description
Accessibility:	The ability to act as a liaison to make sure the voting system meets Federal and State accessibility compliance, and to involve key stakeholders from the disability community in the development process.
Tech Documentation Management:	The ability to write/document technical communication to convey information to another person or party. This capability is necessary to support the certification liaison.
System Assembly/Validation:	The ability to assemble the hardware pieces together and test parts of the system to ensure each part is functional.
Personnel	
Personnel Management	The ability to hire, train, and manage poll-workers for all precincts in the City.

13 Appendix D - Capability Model Assessment Details

Option 1 – Dept of Election Only

Capability Categories	Capabilities	Feasibility Criteria	Degree of Evidence Found	Option Critical Capability?	Justification
Foundational	Budget/Cost Management	Track Record	Minimal Evidence		The Department of Elections has experience managing budgets for contracts they procure, but not for projects specifically. They have never handled managing project budgets throughout its development lifecycle. There is currently one person in the Department that works on managing budgets for contracts.
		Skillset	Minimal Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	No Evidence		
	Program Management	Track Record	Minimal Evidence		The Department of Elections runs elections, which requires big program management skills. However, they have never ran a program that is a technical endeavor. Most of the program management is around adhering to regulations in a compliance-based process, but no new creation of processes has been done outside of handling the known voting process.
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Moderate Evidence		
	Sponsorship / Executive Ownership	Track Record	Moderate Evidence	Yes	The Department of Elections is a sponsor/executive owner for specific initiatives related to voting processes, but not for larger scale projects.
		Skillset	Moderate Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Moderate Evidence		
	Product Management	Track Record	No Evidence		There is no evidence of product management in this department, since they have never run any innovative projects in the past. Products (e.g. voting machines) n relation to voting has always been procured.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
Personnel	Personnel Management	Track Record	Substantial Evidence		There is substantial experience in hiring, training, managing poll workers on the election day. However, there is not much of a desire to own this capability in the department - but a mandate for them to own it.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Software Specific	Software Engineering	Track Record	Minimal Evidence	Yes	The extent of software engineering skills lies in making web application/tools for internal use in the department. There has been nothing larger created in the past, but the skillset may still exist in the department with additional support.
		Skillset	Minimal Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	No Evidence		
	User Experience	Track Record	Minimal Evidence	Yes	For internal web application creation, there is some user experience design that is needed. However, there has been no record of user experience research that has been done for projects within this department, or with any external groups.
		Skillset	Minimal Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Open Source Community Management	Track Record	No Evidence	Yes	There is no evidence of open source community management experience in this department. There has been no interaction with the open source community in the past.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Change Management	Track Record	No Evidence		There is no evidence of current change management projects or change leadership within this department.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
Hardware Specific	Hardware Logistics/Ops Management	Track Record	Minimal Evidence		Currently, the Dominion voting system has Dominion staff managing the hardware operations and logistics. There is high-level oversight from the Department of Elections around storage facilities and election day-of coordination, but most is done by Dominion staff. Therefore, the City has very limited capabilities that exist in this space.
		Skillset	Minimal Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Hardware Maintenance	Track Record	No Evidence		Currently, Dominion staff handle hardware maintenance and keep the Department of Elections informed. Therefore, there are no currently known capabilities around hardware maintenance for voting systems within the City.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Hardware Design Specification	Track Record	No Evidence		In the past, design specification was only needed for choosing a voting system vendor. The Department of Elections knows all requirements needed for a working voting system, and chose a vender based on the options presented. However, there is no current known capabilities around determining hardware design that is compatible with their custom made software.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Procurement	Track Record	Moderate Evidence		In the past, the Department of Elections created 1 RFP/1 Contract to procure the voting hardware/software and chooses vendor. Although there is experience with the procurement process, this process is only done once every 10 years for voting system vendor renewals.
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
Overall System	Vendor Management	Track Record	Substantial Evidence		There is substantial evidence for the Department of Elections to handle vendors since most of what they do is working with a chosen vendor that provides voting hardware, software, operations to ensure the voting process goes smoothly for all voters.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Security	Track Record	No Evidence		There is no evidence of the Department of Elections handling voting system security besides basic server security. This is mainly managed by the vendor.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Systems Administration	Track Record	Minimal Evidence		There is no evidence of system administration capabilities for support the voting system or servers involved. Software that the vendor creates sits in escrow with the Secretary of State before election day, and is not touched by the Department of Elections.
		Skillset	Minimal Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Certification / Management Liaison	Track Record	Minimal Evidence	Yes	There is no evidence of certification capabilities since this is handled by the vendor itself. The vendor usually works with the Secretary of State to ensure their system is certified.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Compliance Management	Track Record	No Evidence		Currently, Dominion informs the Department of Elections of changes that need to be made to the system due to changes in regulation. There is no evidence of compliance management capabilities in the Department of Elections itself.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Accessibility	Track Record	Minimal Evidence	Yes	Currently, accessibility checks on the voting system are not done with the Department of Elections. Dominion does accessibility checks as part of the certification process with the State. The Voting Accessibility Advisory Committee in the City is led by the Department of Elections as a means to provide input and testing in the RFP process - especially going forward.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	Minimal Evidence		
	Technical Documentation Management	Track Record	No Evidence		There is no evidence of technical documentation management within the Department of Elections because this is done by the vendor itself. Currently, Dominion provides technical documentation in order to proceed in the certification process.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	System Assembly / Validation	Track Record	No Evidence		There is no evidence of system assembly/validation within the Department of Elections because this is done by the vendor itself. Currently, Dominion partners with a storage/transportation company to provide assembly for all voting hardware in the voting machines at the precincts.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		

Options 2, 5, and 6 – City of SF only and collaboration with jurisdictions in and outside of CA

Capability Categories	Capabilities	Feasibility Criteria	Degree of Evidence Found	Option Critical Capability?	Justification *Note for Option 5/6, there may be additional skillset/resources housed in other jurisdictions' government that can be leveraged
Foundational	Budget/Cost Management	Track Record	Substantial Evidence		To our knowledge, the capability to manage budget/cost already exists in the City since there is a department established to control the City's budget. The Budget Director, Melissa Whitehouse, manages budgets for projects like this and decides whether to fund them or not based on a variety of cost factors.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Program Management	Track Record	Substantial Evidence	Yes	Skills for program management exists in City Government based on multiple projects run by City officials (e.g. finance system upgrade), but there is no evidence of resource availability or desire to own the program for the development of the new voting system.
		Skillset	Substantial Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Sponsorship / Executive Ownership	Track Record	Substantial Evidence	Yes	Based on our observations, departments drive their own projects. Therefore, there are sponsorship skills from each department to provide direction for project related to that function's goals. In this case, the Department of Elections will have to own this project as the expertise on voting systems lie in that department. However, there is no clear direction if this department will want to own this.
		Skillset	Substantial Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Product Management	Track Record	Moderate Evidence		There is some evidence for product management skills that are being made by departments in the City (e.g. DataSF platform), but not many product innovations that we are aware of. There is no clear department that would want to manage this product.
		Skillset	Moderate Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Minimal Evidence		
Personnel	Personnel Management	Track Record	Substantial Evidence		Currently, the Department of Elections hires and trains all poll-workers for election day every year. This means that there is substantial evidence that those capabilities exist in the City, and no additional resources will be needed.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Software Specific	Software Engineering	Track Record	Substantial Evidence		There are currently software engineers in the Department of Technology, and in other departments in the City. However, the City does not have much experience in custom application development, agile methodology, and open source projects that we are aware of. There is also no evidence for capacity to support this initiative unless resources are hired.
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Moderate Evidence		
	User Experience	Track Record	Moderate Evidence		There are user experience skills that exist in the City that supported the build of the City of SF website. However, it is unclear how staffed their resources are for these skills for a project like this to be built out.
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Moderate Evidence		
	Open Source Community Management	Track Record	Minimal Evidence	Yes	There is currently no open source community management experience that exists within the City. There is one open source portal created by the City called Open Data Explorer, and is residing in GitHub. However, there are no contributing community members, so there has been no open source governance required.
		Skillset	Minimal Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Change Management	Track Record	No Evidence		There is no evidence of current change management projects or change leadership within departments.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
Hardware Specific	Hardware Logistics / Ops Management	Track Record	Minimal Evidence		Currently, the Dominion voting system has Dominion staff managing the hardware operations and logistics. There is high-level oversight from the Department of Elections around storage facilities and election day-of coordination, but most is done by Dominion staff. However, there must be Ops IT staff that help support internal hardware used by City employees.
		Skillset	Minimal Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	No Evidence		
	Hardware Maintenance	Track Record	Minimal Evidence		For internal hardware usage (e.g. computers), a maintenance staff exists in the City to make sure those tools are upgraded and properly working. Currently, Dominion staff handle hardware maintenance and keep the Department of Elections informed. There is no existing capabilities in the City around hardware maintenance for voting systems specifically.
		Skillset	Minimal Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Hardware Design Specification	Track Record	Minimal Evidence		In the past, design specification was only needed for choosing a voting system vendor. The Department of Elections knows all requirements needed for a working voting system, and chose a vender based on the options presented. However, there is no current known capabilities around determining hardware design that is compatible with their custom made software in the City.
		Skillset	Minimal Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Procurement	Track Record	Substantial Evidence		The City has an established department called the Office of Contract Administration that manages the procurement process for all consultants, hardware, etc.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Overall System	Vendor Management	Track Record	Substantial Evidence		The City has substantial vendor management experience due to hired vendors for specific projects that are run by the City. Specifically, Dominion is a hired vendor by the City that has supported their election process the past several years.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Minimal Evidence		
	Security	Track Record	Substantial Evidence		The City has a well established cybersecurity subdivision in their Department of Technology. However, none are well versed in managing security for open source projects.
		Skillset	Moderate Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Minimal Evidence		
	Systems Administration	Track Record	Moderate Evidence		
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Certification / Management Liaison	Track Record	Minimal Evidence		There is no evidence for certification expertise in the City since the vendor - Dominion - manages the certification process with the Secretary of State directly.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Compliance Management	Track Record	Substantial Evidence		There is an audit team that manages compliance checks and hires contractors to provide penetration testing for projects run by the City. However, no compliance checks have been done for open source projects so far.
		Skillset	Substantial Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Minimal Evidence		
	Accessibility	Track Record	Substantial Evidence	Yes	There are many accessibility advocates and committees within the City (e.g., Mayor's Office on Disability, Department of Aging and Adult Services, Voting Accessibility Advisory Committee) that have shown interest and want to transform the new voting system to not just meet accessibility requirements, but make it a useable experience for all. They are part of the RFP process to test which system is more useable.
		Skillset	Substantial Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Substantial Evidence		
	Technical Documentation Management	Track Record	Moderate Evidence		For auditing purposes, procedures must be documented thoroughly. Therefore, there are technical writers that currently document work for their projects within the City. However, for an open source project his extensive, the auditing team has not seen documentation for a project of this scale and with involvement of contributors from an open source community.
		Skillset	Moderate Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	System Assembly / Validation	Track Record	No Evidence		There is no evidence that the City has capabilities to assemble the hardware for all voting machines, and test each part since that was all done by Dominion - the current voting system vendor.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		

Options 3 & 4 – engaging with external vendors building from scratch or upon existing assets

Capability Categories	Capabilities	Feasibility Criteria	Degree of Evidence Found	Option Critical Capability?	Justification
Foundational	Budget/Cost Management	Track Record	Substantial Evidence		To our knowledge, the capability to manage budget/cost already exists in the City since there is a department established to control the City's budget. The Budget Director, Melissa Whitehouse, manages budgets for projects like this and decides whether to fund them or not based on a variety of cost factors.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Program Management	Track Record	Substantial Evidence		*External Vendor Engagement* Hiring a program manager for this project would not be difficult because it would require skills to manage an agile project, which is prevalent in the tech industry across the Bay Area.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Sponsorship / Executive Ownership	Track Record	Substantial Evidence	Yes	Based on our observations, departments drive their own projects. Therefore, there are sponsorship skills from each department to provide direction for project related to that function's goals. In this case, the Department of Elections will have to own this project as the expertise on voting systems lie in that department. However, there is no clear direction if this department will want to own this.
		Skillset	Substantial Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Product Management	Track Record	Substantial Evidence		*External Vendor Engagement* Hiring a product manager for this project would not be difficult because the Bay Area is a booming environment for cultivating innovative products.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Personnel	Personnel Management	Track Record	Substantial Evidence		Currently, the Department of Elections hires and trains all poll-workers for election day every year. This means that there is substantial evidence that those capabilities exist in the City, and no additional resources will be needed.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Software Specific	Software Engineering	Track Record	Substantial Evidence		*External Vendor Engagement* There is a very high likelihood to hire capable software engineers to code this voting system, with additional open source experience since software engineering skills are a very demanded and supplied skill in the bay area.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	User Experience	Track Record	Substantial Evidence		*External Vendor Engagement* Procuring individuals/companies with user experience skills to enhance the voting experience will not be too difficult since design companies like IDEO are headquartered in San Francisco.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Open Source Community Management	Track Record	Substantial Evidence	Yes	*External Vendor Engagement* There is a big open source community hub in the Bay Area, so finding community managers with expertise in this area will not be too difficult.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Change Management	Track Record	Substantial Evidence		*External Vendor Engagement* A lot of consulting firms in the area have change leadership experience, and can be hired for this project to support the strategy around change management for the new voting system.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Hardware Specific	Hardware Logistics / Ops Management	Track Record	Substantial Evidence		*External Vendor Engagement* Currently, the Dominion voting system has Dominion staff managing the hardware operations and logistics. Hardware logistics and operations can still be outsourced to a IT services company in the area.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Hardware Maintenance	Track Record	Substantial Evidence		*External Vendor Engagement* Finding a vendor to manage Non-proprietary hardware maintenance will be very likely because no custom expertise is needed.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Hardware Design Specification	Track Record	Minimal Evidence		In the past, design specification was only needed for choosing a voting system vendor. The Department of Elections knows all requirements needed for a working voting system, and chose a vender based on the options presented. However, there is no current known capabilities around determining hardware design that is compatible with their custom made software in the City.
		Skillset	Minimal Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		
	Procurement	Track Record	Substantial Evidence		The City has an established department called the Office of Contract Administration that manages the procurement process for all consultants, hardware, etc.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
Overall System	Vendor Management	Track Record	Substantial Evidence	Yes	The City has substantial vendor management experience due to hired vendors for specific projects that are run by the City. Specifically, Dominion is a hired vendor by the City that has supported their election process the past several years.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Minimal Evidence		
	Security	Track Record	Substantial Evidence		*External Vendor Engagement* Expertise in security for open source projects can be found because of the abundance of cyber security skillsets in the Bay Area.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Systems Administration	Track Record	Moderate Evidence		
		Skillset	Moderate Evidence		
		Capacity/Resource	Minimal Evidence		
		Desire to Own	Minimal Evidence		
	Certification / Management Liaison	Track Record	Substantial Evidence		*External Vendor Engagement* In the past, the voting system vendor would ensure their product was certified. Since there is a detailed application process, the skillsets to be a certification liaison must have coordinator experience and may not need specific certification process experience. This skillset will not be hard to find in the Bay Area.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Compliance Management	Track Record	Substantial Evidence		*External Vendor Engagement* Risk management and compliance officer skillsets will not be hard to find in this area, as a large number of companies need this skillset to run their business.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	Accessibility	Track Record	Substantial Evidence	Yes	There are many accessibility advocates and committees within the City (e.g. Mayor's Office on Disability, Department of Aging and Adult Services, Voting Accessibility Advisory Committee) that have shown interest and want to transform the new voting system to not just meet accessibility requirements, but make it a useable experience for all.
		Skillset	Substantial Evidence		
		Capacity/Resource	Moderate Evidence		
		Desire to Own	Substantial Evidence		
	Technical Documentation Management	Track Record	Substantial Evidence		*External Vendor Engagement* The need for technical writers is abundant in the Bay Area, and that individual does not have to already specialize in voting system knowledge. This role will not be hard to fill.
		Skillset	Substantial Evidence		
		Capacity/Resource	Substantial Evidence		
		Desire to Own	Substantial Evidence		
	System Assembly / Validation	Track Record	No Evidence		*External Vendor Engagement* There must be a physical IT service that provides assembly support and testing validation that can offer their skillsets to help the City ensure their hardware is functional.
		Skillset	No Evidence		
		Capacity/Resource	No Evidence		
		Desire to Own	No Evidence		

14 Appendix E - Findings from LA County Project

Presented below are the responses (indented and in blue text) from the LA County project team to questions posed by the Slalom project team in late December 2017. More details on their process can be found on their website: http://vsap.lavote.net/process/

1. **Scope, and cost, of what's been done to date**

   It sounds like a very large discovery around the Ballot Marking device has been done. You also mentioned that you have an end to end prototype. Can you describe the project phases to date and scope of each with costs?

   Completed phases and approximate costs
   Phase 1: Public Opinion Research ($1 million)
   Phase 2: Process Assessment ($500K)
   Phase 3: System Design ($15 million)

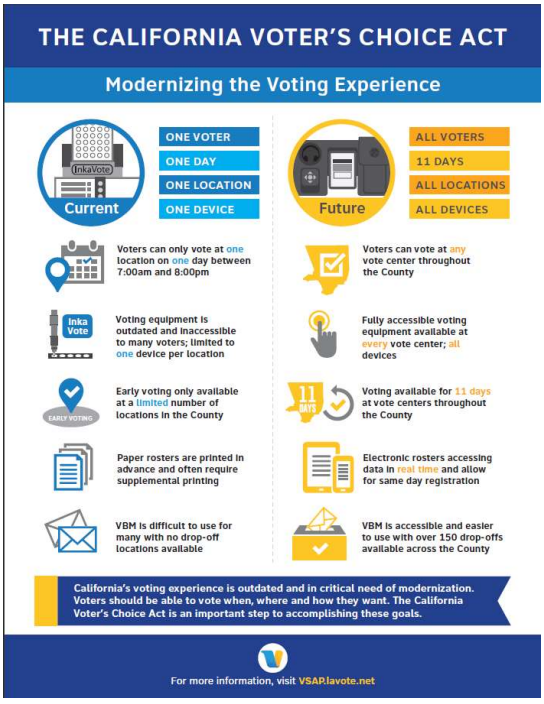
   We are currently in Phase 4 Manufacturing and Certification. This phase involves implementation/procurement of the following system components:

   - Tally (software and COTS scanners)
   - Vote By Mail (ballot design and layout)
   - Ballot Marking Device (BMD) (hardware and software)
   - BMD Manager (software)
   - Interactive Sample Ballot with RAVBM and UOCAVA support (software)
   - Electronic Pollbooks (hardware and software)

   At this time we are in the solicitation and contracting process for these system components, and cannot provide cost figures for them.

2. **Will there be fewer voting locations? i.e. is the plan to have an equal number of new election systems in all locations that historically just had one device and a number of paper-based ballot marking booths**

   No, the Voter's Choice Act authorizes counties to employ vote centers, which is a very different model of voting than the traditional precinct-based model. At vote centers, voters can vote anywhere, instead of at an assigned precinct. Although the VCA prescribes a minimum number of vote centers based on a formula, we plan to have significantly more than the minimum required. At this time, we anticipate deploying approximately 28,000 BMDs over approximately 1,000 vote centers of varying sizes and number of BMD units deployed. Please see attachment.



3. **What programing language is being used for the custom code?**

   The custom code in the Tally solution will be developed in Google Go. Proposed programming environment for the BMD is Android UI Toolkit. For the ISB, consideration has been given to Angular2/CSS/HTML5. These may change once the Prime Contractor is on board.

4. **What operating system is your Ballot Marking Device running?**

   Android is the proposed OS.

5. **We understand that OSET was a consideration when you were beginning your process. Where was their system and/or model lacking?**

   Our intent was always to carry out a County-led voter-centered design process. We had some conversations with them, but they did not submit a formal proposal.

6. **What would be the approximate range of unit cost/price of the ballot marking device (i.e. the physical hardware) if another jurisdiction e.g. SF were to collaborate with you?**

   Based on a preliminary bill of materials developed as part of Phase 3 Design, and also based on input from vendors through our recent RFI, the BMD unit cost is currently estimated to be around $4,000. However, this number should be treated with caution, since the final bill of materials is likely to change, and the high volume production process has the potential to drive down unit costs. Adoption of the VSAP by other jurisdictions could also have the positive effect of lowering cost through volume orders.

15 Appendix F - Comparable data of timelines and costs

The following is the written confirmation of some data that was gathered through the project from a number of meetings with OSET (Open Source Election Technology) Institute. Whilst the project team's inclusion of this information and its source is believed to be valuable it should be stated that this does not represent a direct endorsement or bias towards OSET. It is included here because of their perspective helps triangulate the estimates of budget that the Slalom team prepared and discussed in the main body of this report.

Letter from Thursday, 01.March 2018

Thanks for your continued interest our work and your request for some more information for your upcoming meeting with San Francisco city government officials. You asked three (3) additional questions which I make a best effort at answering hereunder. However, at least one of those questions would significantly benefit from a direct presentation by our CTO or one of our senior technical staff to walk you through more specifics of development status for our primary public technology undertaking: ElectOS.

Please bear with me (*at the mandate of our Legal team*) as I reiterate a couple of important points (*that I apologize for doing because I have explained to your colleagues before, which means I am not successfully communicating the following elements given some elements of your questions below*).

Thus, whenever you're discussing OSET Institute work with clients or others please bear these two points in mind:

1. The OSET Institute is a 501.c.3 purpose-based, tax-exempt, election technology research, development, and education public benefit corporation. A mouthful I know that is, the characterization is very important to us as a consequence of our obtaining our tax-exempt status some five years ago. The short point is: the OSET Institute is in *no way*, *shape*, or *form* a "commercial vendor" of election technology. We cannot be characterized as a "*vendor*" and do not "compete" or "bid" in procurement processes for vendor contract awards. It's a disclaimer of sorts that we often must make in these conversations. That observed, we can (*and do*) serve in a supportive (*or dare I imply subservient*) manner to a "primary" contracting vendor or bidder as a technology resource and domain expert.

2. We do *not* produce "products" although that term occasionally is bandied about much to the chagrin of our Chief Legal Officer. (*Very much*) like the Linux Foundation or Mozilla Foundation, we produce publicly available (*open source in terms of licensing*) government technology. Our principal "work product" or "technology" will be ElectOS™. Similarly, while we extensively provide technical support to the extent our philanthropic funding can budget, we do not have a "call center" or any "customer service" or any of the capabilities one would expect of a commercial technology vendor.

OK, with those disclaimers out of the way, let's move to answering the questions.

Question 1: Progress of OSET Institute Solution

1.1. Where are OSET with what they are building? What activities have been completed?

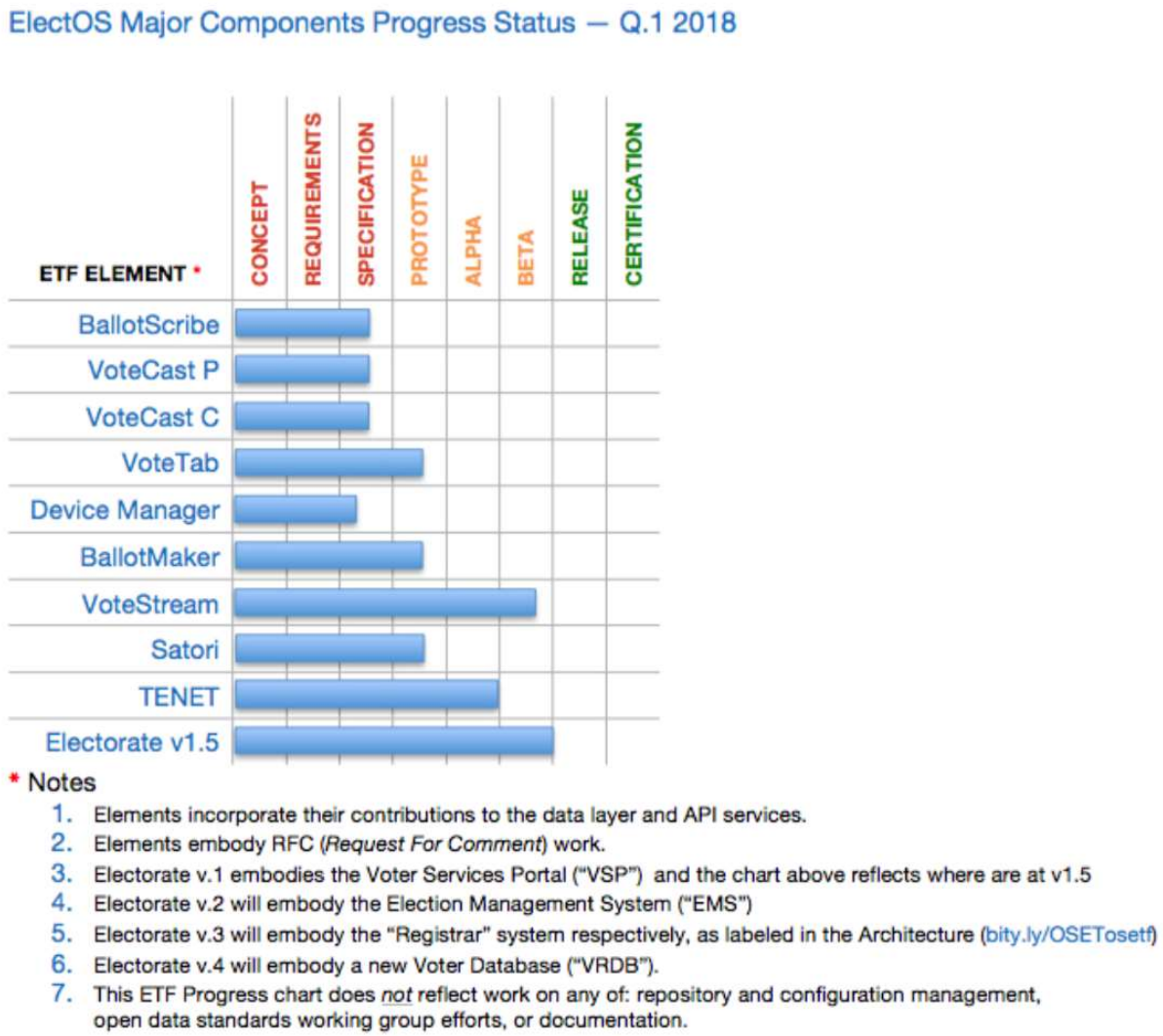
The OSET Institute's principal and fiscally sponsored initiative is the "TrustTheVote™ Project." This is where all of our election technology research and development work is contained. The primary outcome of the Project is what we call ElectOS™. ElectOS is a software layer to an entire election administration ecosystem, which can be best viewed in this somewhat dated, but still technically accurate high-level visual walk-through (HERE[14]).

A major subsystem of ElectOS is the voting system, which is based on the principal of a machine-marked, OpScan counted, paper ballot of record casting and counting process. ElectOS architecture supports rank-choice voting processes.

[14] https://trustthevote.org/wp-content/uploads/2017/05/electos_architecturetour.pdf

The best way to ascertain, short of an extensive document here, the development status of ElectOS voting system is a walk-through by one of our technical team, ideally our CTO, John Sebes. Several of our system components are in some state of design, prototype, engineering, development, or testing. Below is a chart of approximate progress on several components of ElectOS, all of which would comprise the Voting system portion, excluding three: VoteStream™, Satori™, and TENET™.

Again, it would be valuable to have us walk through the specific progress of the voting system components, but here is a first order approximation as of January.



1.2. What's a rough estimate of what it takes (*in terms of work*) to get an MVP? (*i.e., what are the outstanding steps*)?

First, we're a bit struck by the term "MVP," which in our parlance means "*minimally viable product*." In this setting any voting system to be "minimally viable" would have to be at least ready and qualified for State or Federal certification, or in fact "certified" for deployment in a public election.

With that in mind, we are nineteen (19) months from the completion of the ElectOS voting system ready for certification, assuming some very important conditions:

- A continuous full time effort to focus on the completion of: 1] the casting and counting components (BallotScribe, VoteCast, VoteTab); 2] a ballot layout tool—BallotMaker™ (*supporting Rank-Choice*); 3] the required election management system (EMS) components (Electorate™); and 4] a necessary mechanism for hardware and software validity testing ("Device Manager").

- The development project has a full funding commitment of $8.2M USD partitioned into four (4) milestone-based allocations of $2.0M each, and $2.2M for the final installment.
- The engineering and development teams are in place, or ready to on-board over the course of the project schedule.

Question 2: Budget and Benchmarks

2.1. How much has OSET spent to date on their product and what do they estimate it will take to get to an MVP?

The OSET Institute has, per FASB reporting rules for tax and audit purposes, capitalized $1,293,005 investment in the design, engineering, and development of ElectOS as of January 2018. Per our best estimation, forecasting, and planning, there remains $8,200,000 to complete the ElectOS voting system. Some other cloud-based administration subsystems are not included in this cost estimate. Nor does this include the cost of State or Federal certification. Therefore, our CFO and CTO stand by the total capital cost of $9,493,005. (*My earlier assertion of ~$3M capitalized to date was misinformed; I apologize*).

2.2. Can OSET confirm in writing what they have seen of other attempts by other parties? It would be useful to get a sense of whether the number we have as an estimate is reasonable.

We are only aware of two other systems in design or some state of development to look to for comparative analysis, and we believe you are already fully informed on those: Travis County, TX STARVote™ system and L.A. County, CA VSAP system. We were not closely involved with the costing of the LA County project although we've extensively participated in the VSAP process. However, we did participate in extensive cost estimating of the cost-to-implement the STARVote system. We estimated that would require $11.0M. We are also aware of one other software engineering firm that was involved with the STARVote specification development and provided detailed cost estimates to Travis County, TX in a similar amount of approximately $11.0M.

Question 3: Licensing

3.1. When we met, you spoke of the need you discovered to create your own open source license and that even the GNU General Public License v3 was not sufficient for your needs. Please reiterate the motivation and rationale for a new open source license.

I note for the written record that it was never our desire to develop a new open source license. This was *not* a cost (~$150,000) of legal services to design, develop, and obtain OSI accreditation that we ever anticipated, desired, or intended to expend. The development of the OPL (*OSET Public License*) was strictly the result of discovering that in many state and county government procurement settings, where there is a contractual acquisition of goods and services incorporating publicly available software subject to a copy-left license, many procurement and contracting regulations have provisions (*terms and conditions*) for which **a**] the government agency in question requires, and **b**] for which the popular public licenses today, primarily the GPL, do not legally support. So to be clear, this had nothing to do with our desires or needs (*the last thing we believed the world needed was yet another license*), but rather born of the need to provide a sufficient licensing mechanism for public software that was an integral part of a larger procurement in which related goods and services would be purchased with taxpayer dollars. This is an important point: when an I.T. shop decides to acquire open source software, such amounts to a click-wrapper download transaction (*i.e., the licensee click-accepts a license agreement and subsequently downloads the software*.) In fact, in many cases, a click-wrapper is not even present and the mere act of downloading the software is acceptance of the terms of the copy-left license. However, the acquisition (*procurement at a cost of taxpayer dollars*) of systems predominantly dependent on a systems integrator to adapt and deploy on purchased or leased hardware becomes a single contract by regulations in many States, and as we learned, all related legal documents including licenses are taken as a single set of binding agreements. Once the contract administrators engage, the terms of the open source license can often become an issue.

The need for an open source license that could address the objections was the professional legal opinion of our licensing counsel—an internationally recognized expert on intellectual property licensing and specifically open source, who was also our licensing lawyer during our tenure at Netscape Communications in the 90s (*and subsequently the Mozilla Foundation*), which of course, was one of the several pioneers in open source software development and distribution.

Our goal (*or* "*need*") for a license that can address these issues was to ensure we could remove objections to accepting open source technology as an alternative to other commercial products. Candidly, we were aware that those who see public technology as a challenge rather than an opportunity would seek ways to lobby against its use. The terms of the GPL license became a target. The OPL license neutralizes that objection.

All of this is detailed in supporting documents that can be found at:
www.osetfoundation.org/public-license

I hope these answers help your work, and please feel free to follow-up with us in this regard.
Respectfully,

Gregory Miller | Co-Founder, COO | OSET Institute