Secure Online Voting System

Systems Engineering Project
Oral Presentation

Haijing ‘Henry’ Chen
May 1, 2018
 Agenda

- Introduction
- Proposed System
- Deliverables
  - Requirement Analysis and CONOPS
  - Functional Analysis
  - Trade Study
  - Conceptual Design
  - System Specification
  - Test and Evaluation
  - Risk Management
- Future Developments
- Project Wrap-up
About Me

- **Background**
  - Software Engineer – KEYW Corp. (Current)
  - Software Engineer – Northrop Grumman
  - B.S. in Electrical Engineering – University of Maryland

- **Experience – Full-Stack software developer**
  - Web development
  - Server-side software development
  - Database management
  - Linux system management

- **Interests**
  - Web application technologies
  - Online gaming
Project Proposal

- **Secure Online Voting System (SOVOS)**
- Software application to allow online voting that is accessible, secure, reliable, and auditable.
- Secured from vulnerabilities through rigorous analysis and implementation of modern security technology to protect the software, data, hardware, and network.
- Work in concert with existing voting and voter registration systems in the state of Maryland, including auditing and recounting.
New System for an Online Age

- Evolution of legacy non-digital systems into online software applications
- Availability of mobile, hand-held computing devices
- Availability of Internet connectivity
- Increased industry focus on software security

- Increased voter turnout and participation
- Ease of use
- Less time spent
- Decrease stress on physical polling places and election volunteers
Consider the Limitations

- Voter confidence – counteract ineligible voting, ballot stuffing, integrity of ballots cast

- Security of online systems
  - Existing security technology
  - Misuse or incorrect implementation of security
  - Unknown security threats

- Each state has different voting systems

- Integration with existing voting systems - Maryland
System Engineering Approach

Diagram:
- Need → Proposal
- Proposal → Requirements and CONOPS
- Requirements → Requirements and CONOPS
- Requirements and CONOPS → Physical Analysis Allocation
- Functional Analysis → Requirements
- Physical Analysis Allocation → Evaluation and Decision
- Evaluation and Decision → Potential Solutions
- Potential Solutions → Solution(s)
Requirements and CONOPS

- CONOPS
- Requirement organization
- Methods used to gather requirements
- Operational Requirements
- Requirement metrics
Concept of Operations

Hacker Attacks → STOP → SOVOS

Maryland Voter Registration Database

Polling Place

Voter using any computing device

SOVOS E-ballot

Paper ballot

Election
Requirement Organization

- **Traceability**
  - x.x.x.x.x numbering scheme. 1.1.1 is a child requirement of 1.1.
  - Track the source of requirements to project deliverable

- **Metadata**
  - KPP – Key Performance Parameters
  - Potential trade study
  - Quantitative?
  - Verification Method
    - *Inspection*
    - *Analysis*
    - *Demonstration*
    - *Test*
Requirement Gathering

- Project proposal to CONOPS to operational requirements
- Needs Analysis
  - Technological Opportunity
  - Operational Deficiencies
  - Operational Analysis
- Contact key stakeholders
  - Local politicians
  - Local political parties
  - Contact SMEs
- Subject area research
  - Analysis of existing online voting systems: Australian iVote, Estonia, Voting in Kenya, Helios Voting
## Operational Requirements

<table>
<thead>
<tr>
<th>Req #</th>
<th>KPP</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y</td>
<td>System shall allow voters to vote online using a computer device connected to the internet.</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>System shall be able to simultaneously collect votes from all voters attempting to vote online.</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>System shall tabulate votes collected and produce voting results for online votes.</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>System shall leverage the existing Maryland state voter registration database, which will be the source system of record for voter registration information.</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>System shall work in conjunction with existing voting system in the state of Maryland and shall not exclude voters from using existing voting methods.</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>System shall protect against hacker attacks on all system components.</td>
</tr>
<tr>
<td>7</td>
<td>y</td>
<td>System shall adhere to U.S. federal laws and regulations on voting and elections.</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>System shall adhere to Maryland state laws and regulations on voting and elections.</td>
</tr>
</tbody>
</table>
## Operational Requirements

<table>
<thead>
<tr>
<th>Req #</th>
<th>KPP</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>y</td>
<td>System’s online voting process shall be easy to learn, accessible, and have a time commitment of less than in-person voting.</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>System shall be auditable to verify the integrity of the system functions and requirements.</td>
</tr>
<tr>
<td>11</td>
<td>y</td>
<td>System shall support election recounts.</td>
</tr>
<tr>
<td>12</td>
<td>y</td>
<td>System shall provide evidence to voters and election officials that votes are recorded as they intended and not tampered.</td>
</tr>
<tr>
<td>13</td>
<td>y</td>
<td>System shall protect the privacy of the voter and the anonymity of the vote.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>System shall have redundancy and recovery procedure in place to protect all data produced.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>System shall be able to automatically diagnose and indicate that the system is operating correctly.</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>System shall have complete and comprehensive documentation and training for system operation.</td>
</tr>
</tbody>
</table>
Concept Exploration

- Deep dive into the various subject areas for SOVOS
  - Operational concepts of existing voting systems
  - Maryland election laws and governing bodies
  - Federal election laws and governing bodies
  - Industry standard best practices for software and computer security
  - Data redundancy
  - Protecting system network
  - Vote recount
  - Auditing a software system
  - Anonymity of the vote
  - Encryption
Requirement Metrics

- Total System Requirements: 96
  - Operational Requirements: 16
  - Trade Study Requirements: 6
  - KPP Requirements: 61
  - Quantifiable Requirements: 4

- Inspection Requirements: 32
- Analysis Requirements: 17
- Demonstration Requirements: 33
- Test Requirements: 14
System Engineering Approach

1. Need
2. Problem Definition
3. Functional Analysis
4. Requirements
5. Physical Analysis Allocation
6. Evaluation and Decision
7. Potential Solutions
8. Functional Analysis
9. Solution(s)
Functional Analysis

- Logical generation of subsystems based on grouping system requirements

- Analyze system requirements for functions

- Decompose top level functions into lower level functions
  - Functional flow and sequence
  - Functional hierarchy
  - Functional interfaces

- Functional system context
Subsystems

- Software Security
- Voter Registration
- Vote Reconciliation
- Client Software
- Server Management
- Server Software
- Database Management
- Network Security
Functional Requirements

- Client Software subsystem functional flow
<table>
<thead>
<tr>
<th>Req #</th>
<th>Requirement</th>
<th>Function?</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>System shall allow users to fill out and submit their ballot for an election online.</td>
<td></td>
<td>Client Software</td>
</tr>
<tr>
<td>1.5.1</td>
<td>System client component shall present to the user a ballot user interface that allows user the fill out the ballot that is specified by the SOVOS system.</td>
<td>Yes</td>
<td>Client Software</td>
</tr>
<tr>
<td>1.5.2</td>
<td>System ballot user interface shall accept user inputs for filling out the ballot.</td>
<td>Yes</td>
<td>Client Software</td>
</tr>
<tr>
<td>1.5.3</td>
<td>System ballot user interface shall verify the ballot is filled out correctly.</td>
<td>Yes</td>
<td>Client Software</td>
</tr>
<tr>
<td>1.5.4</td>
<td>System ballot user interface shall allow the user to submit a valid and completed ballot.</td>
<td>Yes</td>
<td>Client Software</td>
</tr>
<tr>
<td>1.5.5</td>
<td>System ballot user interface shall alert the user if the current date is not one of the election dates and prevent user from filling out the ballot or viewing the empty ballot.</td>
<td>Yes</td>
<td>Client Software</td>
</tr>
</tbody>
</table>
## Function Interface Traceability

- Interface directionality; source, destination, data transferred
- Traceability to function and report section

<table>
<thead>
<tr>
<th>ID</th>
<th>From Key</th>
<th>From Item Name</th>
<th>To Key</th>
<th>To Item Name</th>
<th>Data transferred</th>
<th>Function(s)</th>
<th>FAR Section</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>SOVOS</td>
<td>Software Security</td>
<td>External</td>
<td>Voters</td>
<td>SOVOS client software</td>
<td>6.5.1</td>
<td>2.1</td>
<td>Software Security</td>
</tr>
<tr>
<td>2002</td>
<td>SOVOS</td>
<td>Software Security</td>
<td>SOVOS</td>
<td>Server Management</td>
<td>SOVOS server software</td>
<td>6.5.2</td>
<td>2.1</td>
<td>Software Security</td>
</tr>
<tr>
<td>2003</td>
<td>External</td>
<td>MD Voter Registration System</td>
<td>SOVOS</td>
<td>Voter Registration</td>
<td>Online voter registration</td>
<td>4.6</td>
<td>2.2</td>
<td>Voter Registration</td>
</tr>
<tr>
<td>2004</td>
<td>SOVOS</td>
<td>Voter Registration</td>
<td>External</td>
<td>Voters</td>
<td>Login authentication information</td>
<td>4.1.1</td>
<td>2.2</td>
<td>Voter Registration</td>
</tr>
</tbody>
</table>
Trade Study

- Analysis of alternatives on system requirements
  - Rejected trade studies
  - Informal trade studies
- Formal trade study
Formal Trade Study

- Methodology
  - Select topic
  - Define alternatives
  - Define criteria
  - Conduct SME interviews
  - Define criteria weights
  - Define criteria utility functions and utility of each alternative
  - Generate results
  - Perform sensitivity analysis
Trade Study

- Topic - Determine best design approach to multifactor authentication for the SOVOS client user interface

- Types of multifactor authentication
  - Basic – username, password, PIN, security question, text message
  - Token – physical or software token with automatically generated login key
  - Biometrics – fingerprint, palm scan, facial recognition, voice recognition
Alternatives

1. “All of the above” “3-factor” authentication – uses all 3 types
2. Simple authentication – uses only basic authentication
3. Basic and biometrics – uses everything except tokens
4. Basic and tokens – uses everything except biometrics
### Criteria

- A. Usability
- B. Privacy Concerns
- C. Security
- D. Accessibility
- E. Implementation Complexity
- F. Maintainability
SME Interviews

- SMEs
  - Computer security expert
  - Senior software engineer
  - Experience software tester

- Questions
  - Rank relative importance of each criteria to each other
    - Used to generate the pair-wise comparison matrix
  - Estimate the effectiveness of each alternative for each criteria
    - Used to generate the utility of alternatives
  - Additional feedback and subject matter research used to generate utility functions
Multifactor authentication using basic authentication and token.

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Criteria A</th>
<th>Criteria B</th>
<th>Criteria C</th>
<th>Criteria D</th>
<th>Criteria E</th>
<th>Criteria F</th>
<th>Criteria Weights</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8000</td>
<td>0.4000</td>
<td>9.5000</td>
<td>6.5000</td>
<td>0.8000</td>
<td>2.5000</td>
<td>0.3262</td>
<td>4.35482</td>
</tr>
<tr>
<td>2</td>
<td>1.0000</td>
<td>1.0000</td>
<td>6.0000</td>
<td>1.0000</td>
<td>9.2000</td>
<td>9.0000</td>
<td>0.1423</td>
<td>3.53356</td>
</tr>
<tr>
<td>3</td>
<td>0.9000</td>
<td>0.4000</td>
<td>9.0000</td>
<td>6.5000</td>
<td>3.0000</td>
<td>4.5000</td>
<td>0.3531</td>
<td>4.41155</td>
</tr>
<tr>
<td>4</td>
<td>0.9000</td>
<td>1.0000</td>
<td>9.0000</td>
<td>8.4000</td>
<td>4.8000</td>
<td>4.5000</td>
<td>0.0836</td>
<td>4.75891 *</td>
</tr>
</tbody>
</table>

* Sensitivity analysis did not affect the final result.

Updates to requirements and system design
System Engineering Approach

1. Need
   - Problem Definition
   - Functional Analysis
   - Physical Analysis Allocation
   - Evaluation and Decision

2. Requirements
   - Potential Solutions
   - Conceptual Design

3. Functions
   - Solution(s)
Conceptual Design

- Initial physical architecture and conceptual design
- Functions translated to physical components
- Functional system context and interfaces translated to physical context and interfaces
- Traceability of physical components to functions and interfaces
- Conceptual design of subsystems based on functional context of subsystems
System Context

External Software Delivery Service

Client Computer System

Server Operating System

World Wide Web

Client Software Install File

Compatibility

Operating System Interface

Server Dependency Software

Admins

Admin UI input

Admin UI output

Manual recount ballots

Login attempt

SOVOS System

SOVOS Voting Results

Election Officials

Online voter registration

MD Voter Registration System

SOVOS user guide

Client software installation

Human user challenge

Login attempt

Fill and submit ballot

Client software user interface

username, password, authentication token

Voters
Physical Linkage Context

- External Software Delivery Service
- Client Computer System
- Server Operating System
- Server Dependency Software
- World Wide Web
- Internet
- Function Interface
- Function Interface
- Function Interface
- Data Storage Device
- Election Officials
- MD Voter Registration System
- User Interface
- Data Storage Device
- Mail
- User Interface
- Voter
- Admin
Conceptual Block Diagrams
## Component Traceability

<table>
<thead>
<tr>
<th>ID</th>
<th>Subsystem</th>
<th>Item Name</th>
<th>Item Type</th>
<th>Requirement Traceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1033</td>
<td>Client Software</td>
<td>Client Computer System</td>
<td>External Component</td>
<td></td>
</tr>
<tr>
<td>1034</td>
<td>Client Software</td>
<td>Client Software</td>
<td>Internal Component</td>
<td>7.3.1, 7.3.2, 7.4.1, 7.4.2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.2.1, 9.2.2, 9.2.3, 9.2.4,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13.1.1</td>
</tr>
<tr>
<td>1035</td>
<td>Client Software</td>
<td>Voter</td>
<td>External Component</td>
<td></td>
</tr>
<tr>
<td>1036</td>
<td>Client Software</td>
<td>Internet</td>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>1037</td>
<td>Client Software</td>
<td>User Interface Service</td>
<td>External Component</td>
<td></td>
</tr>
<tr>
<td>1038</td>
<td>Client Software</td>
<td>Network Service</td>
<td>External Component</td>
<td></td>
</tr>
<tr>
<td>1039</td>
<td>Client Software</td>
<td>Function Interface</td>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>1040</td>
<td>Client Software</td>
<td>User Login Service</td>
<td>Internal Component</td>
<td>1.1.1.1, 1.6.1.1, 1.6.1.2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.6.1.3, 1.6.1.4, 1.6.1.5,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.6.1.7, 9.4.1, 10.6.1</td>
</tr>
<tr>
<td>1041</td>
<td>Client Software</td>
<td>Human User Challenge</td>
<td>Internal Component</td>
<td>6.9.1, 6.9.2, 6.9.3, 6.9.4,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service</td>
<td></td>
<td>9.4.2, 10.6.2</td>
</tr>
<tr>
<td>1042</td>
<td>Client Software</td>
<td>Ballot Service</td>
<td>Internal Component</td>
<td>1.5.1, 1.5.2, 1.5.3, 1.5.4,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.5.5, 9.4.3, 10.6.3, 12.2.1</td>
</tr>
<tr>
<td>1043</td>
<td>Client Software</td>
<td>VVPAT Service</td>
<td>Internal Component</td>
<td>12.1.1.1, 12.1.1.2, 12.1.1.3,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.2.2, 12.3.1, 13.1.2</td>
</tr>
<tr>
<td>1044</td>
<td>Client Software</td>
<td>SOVOS User Interface</td>
<td>Internal Component</td>
<td>1.1.1.2, 1.1.1.3, 1.2.1, 1.2.2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.2.3, 1.2.4</td>
</tr>
</tbody>
</table>
System Specification (A-Spec)

- Generated the complete system specification for SOVOS

- Methodology
  - Review traceability of physical components and interfaces to functional requirements and interfaces.
  - Review all existing requirements for clarity, unambiguity, and traceability.
  - Review of all requirement metadata for accuracy and applicability
    - KPPs
    - Trade studies
    - Verification methods
  - Review of Quantitative Requirements for performance metrics
  - Review remaining system concerns
Requirement Metrics

Total Requirements: 251
Operational Requirements: 16
Trade Study Requirements: 6
KPP Requirements: 62
Quantifiable Requirements: 37
Functional Requirements: 139

Inspection Requirements: 36
Analysis Requirements: 45
Demonstration Requirements: 105
Test Requirements: 63
Demonstration and Test: 2

<table>
<thead>
<tr>
<th>Requirement</th>
<th>RTM Total</th>
<th>New Requirements</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAR</td>
<td>96</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>FAR</td>
<td>227</td>
<td>131</td>
<td>10</td>
</tr>
<tr>
<td>TSR</td>
<td>233</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>CDR</td>
<td>235</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>A-Spec</td>
<td>251</td>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>
A-Spec Products

- Requirements Traceability Matrix
- Functional Interface Traceability Matrix
- Physical Component Traceability Matrix
System Concerns

- Software performance, failure, and reliability
  - Software response time and user perception
  - Mean time to repair (MTTR)

- Approximation of system usage
  - Server architecture
  - Network architecture

- System assumptions
  - Leverage existing Maryland Voter Registration Database
  - Using US Postal Service

- Cost and budget

- Security concerns
Test and Evaluation (TER)

- Initial TER Scope
  - Client Component
    - Client Software subsystem (whole)
    - Software Security subsystem (part)

- Methodology
  - Analyze System Specification requirements for verifiability.
  - Generate Integration Approach, the approach and plan for how the system will be integrated and tested to verify functionality and performance requirements.
  - Generate Qualification Approach, the approach and plan to subject the completed system to system qualification tests.
  - Generate the Verification Cross-Reference Matrix (VCRM), correlating Integration & Qualification tests to requirements.
Test Approach

- Modular approach to testing, “system builds”

- Integration Approach
  - Functional Test Environment – Test the functions of individual components
  - Integration Test Environment – Test the integration of components

- Qualification Approach
  - System Test Lab – Complete physical implementation in a test environment
  - SOVOS System (Beta Testing) – Production system prior to delivery
# Test Plan Matrices

- Test ID
- Test Name
- Verification method
- Objective and details
- Test environment
- Test input
- Test output
- Requirements tested
- Expected results or pass/fail criteria
VCRM

- Used to cross-reference tests and verification to the requirements.
- Tracks additional changes made to requirements to clarify tests
- Every requirement is tested by at least one test
- Verification method of the requirement should match the verification method of the test
Risk Management

- Defined the Risk Management Plan for SOVOS
- Tracked progress of risk mitigation over the course of the project

<table>
<thead>
<tr>
<th>Risk</th>
<th>Initial Likelihood</th>
<th>Initial Consequence</th>
<th>Final Likelihood</th>
<th>Final Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to prevent known security threats</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Unknown security threats</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Error in tabulating vote results</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Lack of stakeholder confidence</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Lack of stakeholder feedback and SME input</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Lack of domain knowledge</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Unforeseen schedule risk</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Risk Management Results

- Risk 1: Failure to prevent known security threats

- Mitigation:
  - A. Subject matter research
  - B. RAR
  - C. TER
**Risk Management Results**

- **Risk 2: Unknown security threats**
- **Mitigation:**
  - A. RAR
  - B. TER

### Risk Management Results Table

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Begin</th>
<th>A</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>May 2017</td>
<td>Nov 2017</td>
<td>Mar 2018</td>
</tr>
<tr>
<td>Yellow</td>
<td>Jun 2017</td>
<td>Dec 2017</td>
<td>Apr 2018</td>
</tr>
<tr>
<td>Green</td>
<td>Jul 2017</td>
<td>Jan 2018</td>
<td>May 2018</td>
</tr>
</tbody>
</table>

### Likelihood vs. Consequence Matrix

- **Begin** (A)
- **End** (B)
Risk Management Results

- Risk 3: Error in tabulating vote results

- Mitigation:
  - A. RAR
  - B. TSR
  - C. TER
Risk Management Results

- Risk 4: Lack of stakeholder confidence

- Mitigation:
  - A. RAR
  - B. TER
Risk Management Results

- Risk 5: Lack of stakeholder feedback and SME input

- Mitigation:
  - A. Quickly reach out to key stakeholders
  - B. TSR
Risk Management Results

- Risk 6: Lack of domain knowledge

- Mitigation:
  - A. Proposal
  - B. RAR
  - C. TSR
Risk Management Results

- Risk 6: Unforeseen schedule risk

- Mitigation:
  - A. Semester Extension
  - B. Semester Extension #2
  - C. Completion of all deliverables
A security focused system engineering design for a modern online software application that allows voters to quickly and easily vote in state and federal elections.

Complete SOVOS System implementation will need to consider additional security factors:

- Background investigation for employees
- Security analysis of software development process
- More rigorous security testing and feedback loop
  - NSA
  - FBI
  - Professional hackers
SOVOS is Secure!

- Security in people
  - Background check and security vetting for everybody involved in SOVOS

- Security in process
  - Security focused development
  - Software dependency security analysis and management
  - Security analysis of design and implementation

- Security in product
  - Security focused testing and validation
  - System penetration testing with security experts
  - Secure and validated delivery of software to users
SOVOS is Secure!

- Security in registration
  - User registration through existing Maryland Voter Registration system
  - First-time login information physically delivered to user

- Security in login
  - User login using multifactor authentication including physical authentication token

- Security during use
  - End-to-end encryption of communication between client and sever
  - Human verification, challenge questions, CAPTCHA
  - Vote-verified “paper” audit trail (VVPAT)
SOVOS is Secure!

- Security in setup
  - Check list for server setup and configuration
  - Automated diagnostic for server setup

- Security in operation
  - Health and status monitoring
  - Database backups
  - Network intrusion detection and mitigation

- Security in management
  - Background check and security vetting for everybody involved in managing SOVOS servers
  - Two-person rule for admin access
  - Auditing of all admin functions performed
SOVOS is Secure!

- Security of the vote
  - VVPAT
  - Auditing
    - Audit all key SOVOS functions, including client and server functions
    - Audit all admin functions
    - Database backups for audit logs
  - Recount
    - Generate paper ballot from VVPAT to facilitate manual recount
## Schedule Evaluation

<table>
<thead>
<tr>
<th>WBS Item</th>
<th>Task Title</th>
<th>Expected Date</th>
<th>Expected (hours)</th>
<th>Actual Date</th>
<th>Actual (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Conception and Approval</td>
<td>5/14/2017</td>
<td>66</td>
<td>5/14/2017</td>
<td>66</td>
</tr>
<tr>
<td>2.1</td>
<td>Requirements Analysis and CONOP</td>
<td>5/28/2017</td>
<td>60</td>
<td>11/16/2017</td>
<td>80</td>
</tr>
<tr>
<td>2.2</td>
<td>Functional Analysis</td>
<td>6/11/2017</td>
<td>70</td>
<td>12/3/2017</td>
<td>80</td>
</tr>
<tr>
<td>2.3</td>
<td>Trade Study</td>
<td>6/18/2017</td>
<td>45</td>
<td>12/7/2017</td>
<td>65</td>
</tr>
<tr>
<td>2.4</td>
<td>Conceptual Design</td>
<td>7/2/2017</td>
<td>70</td>
<td>1/28/2018</td>
<td>90</td>
</tr>
<tr>
<td>2.5</td>
<td>System Specification</td>
<td>7/16/2017</td>
<td>60</td>
<td>4/12/2018</td>
<td>75</td>
</tr>
<tr>
<td>2.6</td>
<td>Test and Evaluation</td>
<td>7/30/2017</td>
<td>65</td>
<td>4/14/2018</td>
<td>65</td>
</tr>
<tr>
<td>2.7</td>
<td>Risk Management</td>
<td>8/6/2017</td>
<td>40</td>
<td>4/12/2018</td>
<td>55</td>
</tr>
<tr>
<td>3.1</td>
<td>Final Report</td>
<td>8/20/2017</td>
<td>81</td>
<td>4/27/2018</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>Oral Presentation</td>
<td>8/24/2017</td>
<td>32</td>
<td>5/1/2018</td>
<td>22</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>589</strong></td>
<td></td>
<td><strong>658</strong></td>
<td></td>
</tr>
</tbody>
</table>
Lessons Learned

- Importance of traceability of requirements, interfaces, and components
- Importance of staying on schedule
- Proper use of Word
- Importance of subject matter expertise and experience
- Scoping the project early
- Schedule more time for review of deliverables
Program Recommendations

- Earlier stimulation of project ideas
- Better Blackboard notifications
- Better notification for updates to SE project guidelines
Any Questions?