

| Committed Product Backlog Item | Tasks | NOT STARTED | IN PROGRESS | COMPLETED |
|--------------------------------|-------|-------------|-------------|-----------|
|--------------------------------|-------|-------------|-------------|-----------|

| | | | | |
|--|--|--|---|--|
| PBI #1: As the product owner, I want IRV and OPLV to work properly so that the software can be used to determine real election results. | | | - Fix the OPLV -1 rank bug - <i>Joel</i> | |
| Acceptance criteria: - Software calculates the results and prints the audit file for the IRV and OPL type elections | | | - Fix the time format on the file names for the Audit and Media summary files - <i>Bat-Ider</i> | |
| Definition done: - Results and audit reports are stored - Eliminate bug list - Unit tests pass | | | | |
| Effort: Medium | | | | |
| PBI Author: Joel Alfveby | | | | |
| | | | | |

| Committed Product Backlog Item | Tasks | NOT STARTED | IN PROGRESS | COMPLETED |
|---|-------|--|-------------|-----------|
| <p>PBI #2: As an election official, I want the software to take multiple election files so that all ballots are properly accounted for from the different voting locations.</p> <p>Acceptance criteria: - Software must be able to accept multiple ballots files for all types of elections</p> <p>Definition done: - Software accepts and can handle one or more than one csv file at a time - Software accepts files from more than one voting location -Results and audit report for each election file is stored in the current directory by the software - All unit tests pass per the acceptance criteria</p> <p>Effort: Small</p> <p>PBI Author: Kanoog Moua</p> | | <ul style="list-style-type: none"> - Create a string vector to store different csv files in order -<i>Bat-ider</i> - Make the software run in a loop until all files are accounted for -<i>Ramanish</i> - Sequentially process the csv files in order and produce output for each csv file separately <i>Kanoog</i> - Write Unit tests - <i>Joel</i> | | |

| Committed Product Backlog Item | Tasks | NOT STARTED | IN PROGRESS | COMPLETED |
|--|-------|--|-------------|-----------|
| <p>PBI #3: As an election official, I want the IRV ballots to have at least half of the candidates ranked and invalidated ballots marked so that a fair election is guaranteed per State Election guidelines.</p> <p>Acceptance Criteria: - Invalidated ballots are removed from the election -Way to store the invalidated ballots in one place</p> <p>Definition done: - Software takes in IRV ballots - Software ranks candidates rounded from 0.5 or above to the next higher integer value for the ballot to be valid - Software removed invalidated ballots - All unit tests pass per the acceptance criteria</p> <p>Effort: Medium</p> <p>PBI Author: Kanoog Moua</p> | | <ul style="list-style-type: none"> - Do not add ballots with <50% of candidates ranked <i>-Ramanish</i> - Create a file that stores the invalidated votes called invalidated_dateofelection.xxx - Write Unit Tests | | |

| Committed Product Backlog Item | Tasks | NOT STARTED | IN PROGRESS | COMPLETED |
|--|-------|---|-------------|-----------|
| <p>PBI #4: As an election official, I want the program to take in PO ballots so that the program can determine the results of the election.</p> <p>Acceptance criteria: - The software takes in PO ballots and displays the winner and number of votes</p> <p>Definition done: - All unit tests pass per the acceptance criteria -The program takes in the csv and correctly parses it -Program displays the winner correctly with the number of votes</p> <p>Effort: Large</p> <p>PBI Author:Ramanish Singh, Joel Alfveby</p> | | <ul style="list-style-type: none"> -Take in ballots via csv - Design and architecture of the PO election type - Create PO election class - Process ballot data into data structure - Find winner of election - Write Unit tests - Percentage of votes for each candidates -Random choice for tiebreaker - Write calculations to audit file (?) - Summary file (?) | | |