	Definition of Done		1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual, exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands)							1) All previous regression tests pass	A Regression tests for new functionality fun automatically every build     A Code written in pairs? viaward by other members.	4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 5) manual , exploratory testing has been conducted 5) manual , exploratory testing has been conducted	7) Checkout and build fully reproducible (1-2 commands)							
	Acceptance Criteria Defir		The user can submit a file for input	command can be run to calculate correct IRV election results 2) Rec	receives correct results of the election 3)	all tests pass and system is able to test runtime of 100000 ballots (6)	Functionality of calcResults is maintained, but all functionality that involves reading from files is moved to readinput()		The user can submit a file for input	command can be run to calculate correct IRV election results	receives correct results of the election	all tests pass and system 4 is able to test runtime of 100000 ballots 6)	Functionality of 7) Ch calcResults is maintained, but all functionality that involves reading from files is moved to readInput()		"flename" command does not cause previous input flesnames to be overwritten, and instead adds the supplied flename to a list of	FileHandler no longer keeps track of a single filename. Instead, FileHandler maintains a list of filenames.	FileHandler.setinputfile() no longer exists, and FileHandler.addlnputFile() is created with full functionality. New function checks the supplied filename for if it exists. If it filename for if it exists, the it is added to the list of filenames. Otherwise it is ignored.			
Sprint Backlog	Who's working on it		sean,jasper	sean, Jasper	sean, ann	sean, jasper, ann	sean, ann		Shey, ann	Shey, jasper	Shey, Jasper	Shey, Jasper, ann								
	Status	3.5 Not Started	Not Started	Not Started	0.5 Not Started	Not Started	Not Started	6 Not Started	Not Started	Not Started	0.5 Not Started	Not Started	Not Started	4.75 Not Started	Not Started	Not Started	Not Started			
	Est Time (hours) S	3.5 N	0.5 N	N 9.0	N 0.5	2	υ. Ο	2 0	N 5.0	N 2.1	N 5.0	Z m	Z 9.0	4.75 N	0.25	0.25	Z 0.25			
	PBI/Task E	IRV Winners (#1)	Fix file parsing to allow multiple winners/ensure that file parsing works properly	Fix calcResults() function to allow multiple winners	Ensure that correct winners can be retrieved by user	Unit testing	bonus: split calcResults() into readinput() and calcResults() (low priority)	OPL Winners (#2)	Ensure that file parsing works properly	Fix calcResults() function to calculate the winners correctly	Ensure that correct winners can be retrieved by user	Unit Testing	bonus: split calcResults() into readinput() and calcResults() (low priority)	Multiple Files (#7)	Change input loop so that "filename" always adds the given filename to a list of files, rather than overwriting previous filenames	give the flehandler a list of flenames rather than just one flename	replace FileHandler.setInputFile() with something like FileHandler addInputFile(), this new method should behave similarly to setInputFile but should add valid files to the list of filenames rather than just overwriting the previous filename			

2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual, exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands)							1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual, exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual, exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands)							
System creates a FileHandler object when the program starts up, this object is closed when the program is closed when the program is closed via the "exit" command. The System no longer creates a FileHandler object when the first valid filename is passed in.	FileHandler has a getter function which returns the FileHandler's list of files.	an OPL election can be ran which draws its inputs from more than one input file	an IRV election can be ran which draws its inputs from more than one input file	the user can remove a filename from the list of filenames via a command.	All unit tests for each of the tasks for this PBI pass, along with all previous unit tests.		An IRV election can be fed invalid ballots via input file(s) and the election will automatically ignore and mark the invalid ballots in the relevant audit file	Can log a string to the audit file via some FileHandler method. This method takes a string and logs that string to the audit file.	all tests pass and system is able to test runtime of 100000 ballots		ElectionPO class exists and extends the Election class	ElectionPO.readInputs() exists and can read all candidates/votes in from all supplied input files.	CandidatePO class exists and has all needed functionality which the Candidate class did not have.	all tests pass and system is able to test runtime of 100000 ballots
Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	2 Not Started	Not Started	Not Started	Not Started	4.25 Not Started	Not Started	Not Started	Not Started	Not Started
0.25	0.25	0.5	0.5	0.5	2	2	90	ις. Ο	-	4.25	0.25	51	φ <sub>.</sub> Ο	2
possibly change from creating a FileHandler when the first file is passed in, to creating one when the program starts?	create a getter for FlieHandler. filenames (probably useful for adding multiple file support to electionOPL/electionIRV)	update electionOPL to support reading input from multiple files	update electionIRV to support reading input from multiple files	possibly create function which makes it possible to remove files from the list of input files (low priority)	Unit testing	IRV ballots Validation (#9)	change the code which reads the ballots in so that it checks each ballot for if it is valid. Invalid ballots will be disregarded/added to audit file	update FileHandler to support this new audit file (add a function)	Unit testing	PO CSV handling (#4)	create ElectionPO class	write ElectionPO.readInputs() which reads all supplied files for inputs.	If the default Candidate class is not enough to complete this, then write a CandidatePO class	unit testing