

Bug Report Team T

1. Method Used in Testing

We ran our test suite on the code, and analysed each failing test case, trying to determine if it was failing due to implementation differences or a bug. We decided to use our tests because we had a good suite of functional tests which covered all scenarios and it would be an ideal way to find functional bugs. To avoid finding the same bugs in different scenarios without recognizing them. We started by applying all the unit tests first and only then progressed to integration.

Where it was hard to determine what was going on, or when it was not clear what would be seen by the user, this was complemented by manual testing: we typed commands directly into the application, and observed the external behavior. Throughout most of the process we used a black-box approach, and only looked into and debugged the code in specific issues where it was unclear whether something was a bug.

In certain cases where our test suite made us aware of the problem but did not include a test case designed specifically for it we adapted test cases to make it clearer to diagnose.

We also explored by then looking at the code and understanding if there are any evident bugs (like null checks, planted bugs that weren't fixed) for any structural bugs and any false developer assumptions made for methods, objects etc. One great advantage we got from AutomatedTestingSuite was that running our automated generated tests helped us find 2-3 bugs. But we had to make sure they weren't implementation specific and failing hence it had to be properly examined. Using IntelliJ interpreter, FindBugs tools helped in finding structural possible bugs and allowed us to explore if they can be triggered through user commands. We also ensured that if multiple tests failed due to a single fault, we only reported them once such that resolving each bug resolves the main fault. We provided multiple tests to ensure that other teams can test if all tests pass once the main fault is fixed.

Finally, we put some effort into System Testing by running the Shell and trying different commands in different formats with different functionalities (output, no output, error etc.). This allowed us for testing many real time applications of the System and ensuring they work fine and do manual testing.

To run the failing tests:

Place the `dummyTestFolder` in the same directory level as `src`. The file `TeamUBugs.java` contains all test cases which reveal the bugs. The tests can be run forked by method for best understanding of actual bugs. The JavaDocs for each method state the description of bug and the number in `@DisplayName` can be used to point it to more details in the `BugReports.pdf` file.

2. Bug Report

#	Description	TestCase	Comments (do not fill)
1	Commands with extra spaces around don't execute or give result	E.g. <code>testTokeniseWithQuote()</code> <code>echo " hi " bye</code>	
2	Double backquote nesting results in wrong execution of commands in <code>CommandSubs</code>	<code>testNestedQuotesMultiple()</code> <code>echo abc</code> <code>`echo 1 2 3`xyz`echo 4 5 6`</code>	
3	Command Subs doesn't return the result as list of tokenized arguments for the outer command	<code>paste `ls *.txt`</code> If <code>`ls *.txt`</code> returns two files, the paste result should contain both files but here it's only one first files <code>testPasteWithLs()</code>	
4	<code>CommandSubs</code> within Double quotes trims spaces	<code>echo "This is space:`echo " "`. "</code> Should print a space before the period. <code>testInterpretationOfSpace()</code>	
5	Unnecessary extra lines between <code>ls</code> output of files	<code>ls *.txt</code> If it contains two txt files there shouldn't be any extra space between two files names <code>testGlobbingWithMultipleMatches()</code>	
6	Irrelevant trimming of actual output when command subs	<code>echo "`wc -c hell.a.txt` - `wc -c hell.z.txt`"</code>	

	used	<p>should output</p> <pre>3 hell.a.txt - 3 hellz.txt</pre> <p>And not</p> <pre>3 hell.a.txt - 3 hellz.txt</pre>	
7	Directly calling System.exit(0) makes ExitApplication untestable	Untestable code. Ends test suite process	
8	Wc command recognizes stdin as a null argument	<p>E.g testWcOfStdin()</p> <pre>InputStream inputStream = new FileInputStream(new File(WC_TEST_DIR + StringUtils.fileSeparator() + WC1_FILE)); //NOPMD wcApp.run(null, inputStream, stdout);</pre>	
9	Sort command implementation output differs from what is required for some flags	<p>E.g</p> <pre>testNRFFlagsMixedSort() testNFlagMixedSort() testNFFlagsMixedSort() testNRFlagsMixedSort() testStdinSort() testMoreThanOneFileSort()</pre>	
10	<p>Diff command show different output format than what is expected</p> <p>Diff command has implementation error that differ from expectations</p>	<p>E.g.</p> <pre>testDiffFilesWithDifferentContentUsingFlagQ testDiffFileAndStdinWithSameContentUsingFlagS testDiffFilesWithSameContentUsingFlagSB testDiffFileAndStdinWithDifferentContentUsingFlagQ</pre> <p>testDiffFileAndStdinWithDifferentContent</p>	
11	Cut command output has an extra carriage return	<p>E.g.</p> <pre>testCutOnStdinWithSingleNumberUsingFlagB testCutOnStdinWithSingleNumberUsingFlagC testCutOnStdinWithCommaSeparatedN</pre>	

		umbersUsingFlagB testCutOnStdinWithCommaSeparatedN umbersUsingFlagC	
12	cp copies a file to the same directory it is in, overwriting itself unnecessarily / Also copies file into itself with similar behavior	testCpFailsSingleFileToSameDir() / testCpFailsSingleFileToItself()	
13	cp with multiple files ignores all following files when there is an error with one of them. Should copy the 'good' files and report error in the end	testCpMultipleFilesOneFails()	
14	paste with a null output stream reports 'No InputStream and no filenames'	testPasteNullOutputStream()	
15	paste with a directory as the only argument reports 'No InputStream and no filenames' Instead of reporting that the file is a directory	testPasteDirOnlyArg()	
16	paste with nonexistent file reports 'No InputStream and no filenames'	testPasteNonexistentFile()	
17	'paste - -' causes input to stdin to be printed twice such that if the user inputs 'a' on a line and 'b' on the following line then terminates the command, the output generated is: a\ta b\tb Instead of a\tb	testPasteTwoStdinArgs()	
18	rm with -r flag will not remove empty directories. Displays 'Permission denied' even though there is drwx permission on the parent directory as required by the assumptions	testRmEmptyFolderRecursive()	

19	'rm -rd .' removes everything in the current directory. This should never be allowed (GNU rm does not allow removing '.' or '..')	testRmfailsDirEndInDot()	
20	Extra unexpected output when running wc command then cut command	WcIntegrationCommandTest.testWcThenCut	
21	Unexpected output when running sort command and then piping output to cut command	SortIntegrationCommandTest.testSortThenCut	
22	Unexpected amount of carriage returns when running sort command and then piping output to cut command	SortIntegrationCommandTest.testSortThenCutNegative	
23	Wrong order of sort command output	SortIntegrationCommandTest.testSortThenFindNegative	
24	Stdin should be printed as '-'	DiffIntegrationCommandTest.testEchoThenDiff DiffIntegrationCommandTest.testEchoThenDiff2 DiffIntegrationCommandTest.testPasteThenDiff2	
25	Extra carriage returns in diff command	DiffIntegrationCommandTest.testPasteThenDiff	
26	Multiple command Subs produces extra empty line at the end	echo `echo `echo hello `` testSameCommandSubsNested()	
27	StringUtils.isBlank() For loop condition is always true defaulting to infinite loop if not an empty string	NOTE: Bug found using test generated from automated testing tool EvoSuite nonEmptyInputToStringUtils()	
28	Throws error that file doesn't exist when it should be creating the directory needed for files position	echo yo > dir1/dir2/out.txt Create a file in non existent directory 2 level down	

		testCreateDirectoryThroughOutputStreamForFile()	
29	<p>Ls Integration with Diff gives no result and understandable error message when there shouldn't be one</p> <p>The tokenization of sub command is done wrong making outer diff to produce wrong result</p>	<p>diff `ls diff*.txt`</p> <p>Produces error as "1" when it should be giving diff</p> <p>testLsWithDiffCommandSubs()</p>	
30	<p>Wrong tokenization of the result of command subs.</p> <p>Only first word retained and everything else removed and not passed into echo</p>	<p>testEchoWithDiffCommandSubs()</p> <p>echo `diff hella.txt hellz.txt`</p> <p>Should not delete everything after first arg of command subs</p>	
31	<p>Command subs deletes first word of commandResult and only passed the rest</p>	<p>Second should also be displayed after First as it is present in one of the files</p> <p>testGrepWithSort()</p> <p>Filters file names present in findInFile.txt and sorts those files. The first output hellz.txt is ignored from result of grep</p> <p>sort `grep hell findInFile.txt` [Directory structure explained in tests</p>	
32	<p>Pipe with cut and echo displays empty lines</p>	<p>Extra empty lines being outputted by cut on a single line to be cut passed through echo</p> <p>testPipeWithEchoAndCut()</p>	
33	<p>Grep with no file provided and no stdin provided</p>	<p>After some period of time it should timeout and give an error</p> <p>testGrepWhenNoFilesProvided()</p>	
34	<p>Cd to a directory with no read</p>	<p>It should throw an exception that</p>	

	permission	directory is not readable	
35	Null App Runner in Shell When App not initialised properly	testFailsNullAppRunner() Fails abruptly	
36	Null Args Resolver in Shell When App not initialised properly	testFailsNullArgResolver() Fails abruptly	
37	Sed with no file provided and no stdin provided	After some period of time it should timeout and give an error testSedWhenNoFileIsProvided()	
38	Bug in argument resolver not being able to get the actual size after resolving an random command involving special characters	testUnexpectedResolvingOfArgs() Gives 1 instead of 8	
39	Irregular parsing of command when there is a sequence inside command subs.	E.g. echo `echo hi ; echo sup` bye Does Not print "sup" when its expected testSequenceInsideCommandSubs()	