UCC Java Release Final Report

2019 Summer

7/30/2019

# Team Members Information

|  |  |  |
| --- | --- | --- |
| Team Members | Attended Classes Remotely? | Attended Meetings Remotely? |
| Chong Ren | Never | Never |
| Xi Lan | Never | Never |
| Zhenyu Shou | Never | Never |
| Xiaopeng Han | Never | Never |

# Project Objective

1. Implemented maintainability index functionality on latest version UCC-G\_1.3.1. We implemented maintainability Index calculation based on ucc-j\_2018.05 for Java and Python.
2. Integrated UCC Function Level into latest version UCC-G 1.3.1. Changed several variable problems that comes out due to the version difference.
3. Fix bugs in GUI code for ucc, add some new features and integrate it to ucc.
4. Fix bugs and integrate original UCC release code to UCC-G 1.3.1.

# Project Summary

<This section is for you to describe in greater detail the research you did, the tasks you accomplished, how your work is significant to UCC, and what you learned through the process. >

1. We did a thorough research and have a quick idea about maintainability index. We learned that maintainability index is an important index to indicate how complex the project is. After reviewed the previous relevant code and understand how the index is calculated, we implemented maintainability index functionality on latest version UCC-G\_1.3.1 based on ucc-j\_2018.05.
2. Got familiar with java swing through internet, and understood what the previous group had done in their GUI code. Then analyzed what new features I needed to add in GUI and accomplished the resize and drag&drop features and added one new button for function level. Also I fixed several bugs in the code during this summer. I learned how to use java swing after this project.
3. Learn how the earlier team worked on UCC function level, ucc function level is based on ucc-j 2018.05 but what we need to do is to integrate it into ucc-g 1.3.1 which brought a big problem. What the integration result could do is to count the first file however it couldn’t read all files for now.
4. Several modifications are applied to correct the UCC Release 1.3.1’s output files. Based on the different results from UCC C++ version and Java version, many output files have incorrect results, including physical SLOC, logical SLOC, cyclomatic complexity and etc… Revising language counters and language properties are significant to make UCC more reliable. Debugging is not a simple thing, and it may relate to many files. To reduce the amount time of debugging, we have to think each language and its property carefully.

## Test Summary

Test cases are located in all\_files/admin/dr\_class/Projects/UCC\_in\_Java/2019\_1\_Spring/Test/test\_cases, 34 languages are tested below:

ADA, ASP, Assembly, Bash, Batch, C#, C, coldfusion, coldfusionscript, cpp, cshell, css, fortran, HTML, IDL, java, javascript, jsp, makefile, matlab, nextmidas, pascal and perl, php, python, ruby, scale, sql, vb, vbscript, verilog, VHDL, xmidas and xml.

Some of bugs in mentioned in previous report have been fixed, and the rest of the unsolved bugs are all included in Bug Report.

# Projects’ Decisions

1. For the GUI part, I decided to use GridBagLayout instead of absolute distance to accomplish the resize feature. It would make things easier when we try to make component resize and to add new components.
2. For Maintainability index part, in order to debug I spent a long time to compare code with my eyes. After I adopt comparison software, there is a significant improvement of my work.
3. Web languages like ASP, ColdFusion, HTML, JavaScript will be differentiated and counted separately, it is decided to count <% as HTML codes instead of ASP, this may not be very correct. However, based on works from old teams, right now it should be the best solution. Also, the counting standard between C++ and Java is quite different for these language counters. For C++ version, for example, a ColdFusion file which is combined by ColdFusion, HTML, JavaScript will be counted to generate a single ColdFusion file. However, in UCC-J, different languages will be counted to generate different output files. It is pretty easy to count lines of code for many times or count the wrong keywords.
4. We decided to do our own parts, and integrate the code together. However, we had many conflicts when combining the code because some of us modified the same file at the same time. Instead, it is recommended to merge code every week.

# Known Bugs/Issues

<Describe bugs that exist or that you have found and have not yet been resolved. Or describe the weaknesses of the project with suggestions of how to improve. If the project/task is not complete, describe what still needs to be completed and how. >

1. The maintainability index is implemented based on ucc-j\_2018.05 but there is still a minor difference of the result of maintainability index with ucc-j\_2018.05. One of the probable reasons may be some indexes used when calculating MI has been changed after debug.
2. When called ucc function level it might cause general functions go wrong. The reason might be the difference between ucc-j2018.05 and ucc-g 1.3.1.
3. For language counter bugs, please refer to the Bug Report in the Docs folder.

# Development Notes

<In the table below, list the files and functions you added, modified, deleted, and give an explanation – such as, what new functionality did you add, what bugs did you fix, etc. Be sure to specify the class/file before the functions in the list. >

|  |  |  |  |
| --- | --- | --- | --- |
| **General Description of New Feature or Code Modification** | | | |
| fix GUI’s bugs, add the resize and drag&drop features and new buttons | | | |
| **New Files** | **Description** | **New Functions** | **Description** |
| DropTargetListenerImpl | used to monitor the drag&drop behavior | drop | read the location of the file we dropped |
| **Modified Files** | **Description** | **Modified Functions** | **Description** |
| MainGUI.java | the main GUI. add new features and fix old bugs. | countingReset | add new button’s reset |
| main | add code to set split panel’s size |
| differencingReset | add new button’s reset |
| initialize\_CountingPanel | use GridBagLayout and add constraints to adjust the location of component and accomplish resize. And add the dropListener and function level button |
| initialize\_DifferencingPanel | use GridBagLayout and add constraints to adjust the location of component and accomplish resize.And add the dropListener and function level button |
| initialize\_ProcessingPanel | use GridBagLayout and add constraints to adjust the location of component and accomplish resize. |
| initialize\_LanguagePanel | use GridBagLayout and add constraints to adjust the location of component and accomplish resize. |
| initialize\_DownPanel | use GridBagLayout and add constraints to adjust the location of component and accomplish resize. Delete the code to search resource for stop and start button since the former group did not give the resource |
| initialize\_LeftPanel | change the resource path for advantage button since the former group did not give that resource. change the function we overwrite in addMouseListener to fix the bug that sometimes it will not respond. |
| initialize\_Panel | change the panel layout to make them resizable. add componentListener to keep the proportion of split panel during resize. |
| initialize\_Uppane\_panel | change every inner panel’s layout. |
| ValidateThread | add code for function level button. add commitEdit for some component in processing panel to fix the bug that they can not get the new value after the number is changed. change the code to make them add correct command about checking box in basic panel according to counting or differencing. |
| **Deleted Files** | **Description** | **Deleted Functions** | **Description** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **General Description of New Feature or Code Modification** | | | |
| Integrate Function level part and fix bugs | | | |
| **New Files** | **Description** | **New Functions** | **Description** |
|  |  |  |  |
| **Modified Files** | **Description** | **Modified Functions** | **Description** |
| JavaCounter.java | Added 8 functions | CountFuncPSLOC | Counts PSLOC within a block of function definition in the exact same way as CountFilePSLOC does. |
| CountFuncLSLOC | Counts LSLOC within a block of function definition in the exact same way as CountFilePSLOC does. |
| CountBracketsInLine | Return number of curly brackets in a line. A helper function to help determine the end of a function used in CountFuncPSLOC. |
| CopyCmplx | Return a deep copy of an arraylist of CmplxDataType. A helper function. |
| CountFuncKeyword | Count function’s keywords. A helper function used in CountFuncLSLOC. |
| CountFuncCyclCmplx | Count function’s cyclomatic complexity. A helper function used in CountFuncLSLOC. |
| CopyMap | Return a deep copy of tree map. A helper function used in CountFuncLSLOC. |
| CountFuncNumAndWordOperands | Count function’s number and word operands. A helper function used in CountFuncLSLOC. |
| Modified 2 functions | CountFilePSLOC() | Integrated with CountFuncPSLOC to allow function level processing |
| CountFileLSLOC() | Integrated with CountFuncLSLOC to allow function level processing |
| CodeCounter.java | Added an extra variable to store the list of UCCFiles that contain function blocks. | No functions were modified in this file | |
| MainUCC.java |  | ParseUserRequest() | Added cases to the switch to determine if user wants to process source codes function-by-function. |
| RuntimeParameters.java | Added extra variables to check if function-level processing is turned on | RuntimeParameters() | Initialize the Boolean variable storing function-level processing flag (to false) |
| CSVReporter.java | Modified the counter reporter function to present function-level processing results, if it is toggled on. | GetLangCntrReport() |  |
| ASCIIReporter.java |
| UCCFile.java | Added following flags used in function level counting:    1. boolean IsOutsideFunction;  2. boolean startCounting;  3. int bracketCountForDeterminingFunctionScope;    Added the following parameters used for storing and referencing function level results for a UCC File.  1. ArrayList<UCCFile> funcResults;  2. ArrayList<String> funcDefLine;  3. Map<String, Integer[]> funcBoundMap |  |  |
| **DiffResultType.java** | **Added the following parameters used for differencing and updated all constructers**    **1. String FunctionName;** |  |  |
| ProcessController.java | Added the following parameters to store the results of function level differencing.  1. ArrayList<DiffResultType> DiffFuncResults;  2. ArrayList<DiffResultType>DupDiffFuncResults; | StartProcessController() | Added conditional to check for function level processing all call corresponding ProcessCountReq or ProcessDiffReq functions. |
|  |  | ProcessController() | Added conditional to constructor to initialize function results ArrayLists if they did not exists |
|  |  | ProcessCountReqFunc() | Function called from ProcessController() for Function Level Counting. Provides same functionality as default ProcessCountReq() function just for functional level processing |
|  |  | ProcessDiffBaselinesFuncReq() | Function called from ProcessController() for Function Level Differencing. Provides same functionality as default ProcessDiffBaselinesReq() function just for functional level processing |
| Differencer.java |  | DiffFuncBaselines(ArrayList<UCCFile> cntrResultsA, ArrayList<UCCFile> cntrResultsB,  ArrayList<DiffResultType> diffResults, ArrayList<DiffResultType> dupDiffResults,  ArrayList<DiffResultType> diffFuncResults, ArrayList<DiffResultType> dupDiffFuncResults) | Controller function for processing functional level differencing. Nearly same architecture as standard DiffBaselines function but with additional parameters for functional level processing to be used for storing and generating the results. |
| LSLOCDifferencer.java |  | CompareFiles(ArrayList<DiffResultType> diffResults, DiffResultType diffRes) | New function that is called from the differencing controller for processing function level differencing.    If either Baseline is unmatched it simply counts the files in a similar manner to the standard CompareFiles function.  If the files are matched it will read both files temp LSLOC files into memory. First iterating through FileA’s function definitions and matching them to FileB’s function definitions. Functions will be differenced if there is a match. If no match exists in FileB, the function is considered to be deleted lines. FileB’s function definitions are then iterated through and if a function does not exists in FileA, the function is considered added lines. |
| CSVDiffReporter.Java |  | GenMatchedPairsReport(ArrayList<DiffResultType> diffResults, boolean dupResults) | Added conditional to write the function name to a new column in the report. |
|  |  | GetMatchedPairColHeader() | Added conditional to write “Function” in the column header |
|  |  | GetMainDiffColHeader() | Added conditional to write “Function” in the column header |
|  |  | GenMainDiffReport(ArrayList<DiffResultType> diffResults, boolean dupResults) | Added conditional to write the function name to a new column in the report |
| ASCIIDiffReporter.Java |  | GenMatchedPairsReport(ArrayList<DiffResultType> diffResults, boolean dupResults) | Added conditional to write the function name to a new column in the report. |
|  |  | GetMatchedPairColHeader() | Added conditional to write “Function” in the column header |
|  |  | GetMainDiffColHeader() | Added conditional to write “Function” in the column header |
|  |  | GenMainDiffReport(ArrayList<DiffResultType> diffResults, boolean dupResults) | Added conditional to write the function name to a new column in the report |

|  |  |  |  |
| --- | --- | --- | --- |
| **General Description of New Feature or Code Modification** | | | |
| Implement MI for Java and Python | | | |
| **New Files** | **Description** | **New Functions** | **Description** |
|  |  |  |  |
| **Modified Files** | **Description** | **Modified Functions** | **Description** |
| JavaCounter.java |  | CountFilePSLOC | move following lines to the end of /\* PREPROCESSING START \*/  line = ro.line; inMultiLineComment = ro.inMultiLineComment; inStringLiteral = ro.inStringLiteral; truncateLinesCount += ro.truncateLinesCount; closingStringLiteral = ro.closingStringLiteral;  Add 2 lines CalculateHalsteadMetrics(cntrResult); CalculateMaintainabilityIndex(cntrResult); |
| JavaCounter.java | Add 4 functions for MI | CalculateHalsteadMetrics  CalculateMaintainabilityIndex  GetUniqueAndTotalOperatorCount  GetUniqueAndTotalOperandCount |  |
| PythonCounter |  | CountFileLSLOC | Add 2 lines CalculateHalsteadMetrics(cntrResult); CalculateMaintainabilityIndex(cntrResult); |
| PythonCounter | Add 4 functions for MI | CalculateHalsteadMetrics  CalculateMaintainabilityIndex  GetUniqueAndTotalOperatorCount  GetUniqueAndTotalOperandCount |  |
| CSVCounterReporter | Add 1 function | GenMaintainabilityIndexCntrReport |  |
|  |  | GenMaintainabilityIndexCntrReport | Add 1 line  GenMaintainabilityIndexCntrReport(cntrResults); |
| **Deleted Files** | **Description** | **Deleted Functions** | **Description** |
|  |  |  |  |

A detailed report for modification below is in Bugs\_fixed\_Report in the same folder. Also, modification for language properties are in Bugs\_fixed\_Report as well.

|  |  |  |  |
| --- | --- | --- | --- |
| **General Description of New Feature or Code Modification** | | | |
| Debug different language counters, fix the SLOC counting if the counter counts SLOC incorrectly, and fix incorrect results for cylcomatic complexity. | | | |
| **New Files** | **Description** | **New Functions** | **Description** |
|  |  |  |  |
| **Modified Files** | **Description** | **Modified Functions** | **Description** |
| src/ucc/main/ProcessController.java | import sun.awt.windows.ThemeReader will show error message, and it is not used in the whole project | //import sun.awt.windows.ThemeReader | Silence the import code |
| JavaCounter.java |  | getFunctionName | Change if (line.equals("{") || line.contains("=") || line.equals("({")) to:  if (line.equals("{") || line.contains("=") || line.equals("({") || line.contains("new ")) |
| PythonCounter.java |  | GetTrueCyclomaticComplexity | Comment out the line: tempCount++;  Now it becomes:  // Increment the count for the def  // No need to increase 1 cause for each function we increase count by 1 in CalculateCyclomaticComplexity()  // tempCount++; |
| PythonCounter.java |  | CalculateCyclomaticComplexity | // Remove this if statement to save the CC1 results by function of the lines that's not in any function and contains no cc keyword  // If the first level count is 0, but there are other levels, remove it  // if (cntrResult.CyclCmplxCnts.size() > 1 && cntrResult.CyclCmplxCnts.get(0).Count == 0)  // {  // cntrResult.CyclCmplxCnts.remove(0);  // } |
| PythonCounter.java |  | GetTrueCyclomaticComplexity | Comment out the line that removes the object:  if (tempCount >= 1)  {  cntrResult.CyclCmplxCnts.get(0).Count = tempCount;  }  else // Otherwise delete the file level item  {  // Remove this line to save the CC1 results by function of the lines that's not in any function and contains no cc keyword  // cntrResult.CyclCmplxCnts.remove(0);  } |
| PythonLanguageProperties.java |  |  | Before:  SetCyclCmplexKeywords(new ArrayList<String>(Arrays.asList("if", "elif", "case", "while", "for", "catch")));  Change into:  SetCyclCmplexKeywords(new ArrayList<String>(Arrays.asList("if", "elif", "case", "while", "for", "except"))); |
| PythonCounter.java |  | CountFilePSLOC | Before, it only uses one String to save the line:  line = ro.line;  Add another tempLine to save the original line:  // Add a tempLine to save the indentation from pre-processing  String tempLine = line;  line = ro.line;  Add leading space before writing to BufferedWriter that passes lines to CountFileLSLOC():  // Add the indentation for countint the cyclomatic complexity  line = tempLine.substring(0, tempLine.indexOf(tempLine.trim())) + line; |
| PythonCounter.java |  | GetTrueCyclomaticComplexity | Add a stack to save the outer function kndx and tempCount, push when meets a new def and pop when meets first line that has no greater indentation:  int kndx = 0;  int tempCount = 0;  // Using stack to save the outer function  Stack<Integer> kndxStack = new Stack<>();  Stack<Integer> tempCountStack = new Stack<>();  (adds line 1036-1038)  Change the if statement in line 1035 from:  if (cyclomaticComplexityObj.keyword.get(k).trim().startsWith("def"))  Into:  // If there is a function or return from inner function  if ((cyclomaticComplexityObj.keyword.get(k).trim().startsWith("def")  && cyclomaticComplexityObj.indentation.get(k) != -999) || tempCountStack.size() < kndxStack.size())  (adds line 1043-1045)  Instead of getting kndx from k, resume kndx when necessary:  // Save the def index  kndx = k;  // Resume the kndx if there is any  if (tempCountStack.size() < kndxStack.size())  {  kndx = kndxStack.pop();  }  (line 1050-1055)  End the loop when there is another def:  // If there is function in function, end the loop if (cyclomaticComplexityObj.keyword.get(k).trim().startsWith("def"))  {  break;  }  // Increment the count for the def  tempCount++;  (line 1068-1073)  Push current kndx and tempCount into stack:  Before:  // Handle last entry if it meets the criteria  if (cyclomaticComplexityObj.indentation.get(kndx) < cyclomaticComplexityObj.indentation.get(k))  {  // Increment the count for the def  tempCount++;  // Set the indentation to -999 as we go so we know which ones we've counted  cyclomaticComplexityObj.indentation.set(k, -999);  }  After:  // Handle last entry if it meets the criteria  if (cyclomaticComplexityObj.indentation.get(kndx) < cyclomaticComplexityObj.indentation.get(k))  {  // If there is function in function, save current kndx and tempCount in stack  if (cyclomaticComplexityObj.keyword.get(k).trim().startsWith("def"))  {  kndxStack.push(kndx);  tempCountStack.push(tempCount);  }  else  {  // Increment the count for the def  tempCount++;  // Set the indentation to -999 as we go so we know which ones we've counted  cyclomaticComplexityObj.indentation.set(k, -999);  }  }  (line 1084-1101)  Pop tempCount out when ends a function:  // Reset our index  k = kndx;  // Set the indentation to -999 as we go so we know which ones we've counted  cyclomaticComplexityObj.indentation.set(k, -999);  // If there is an outer funtion, resume the tempCount  if (!kndxStack.isEmpty() && kndxStack.peek() != kndx)  {  tempCount = tempCountStack.pop();  k = kndxStack.peek();  }  (line 1048-1053) |
| MultiLanguageHandler.java |  | SetupVariables() | Silence line 179 and 182 will solve the problem  //Modified by 2019 UCC-J Summer Team  // ASPStartTags.add("<%");  ASPStartTags.add("<script runat=");  ASPStartTags.add("<asp:");  // ASPEndTags.add("%>");  ASPEndTags.add("</asp:"); |
| ColdFusionUtil.java |  | CreateDeleteItemList | remove ‘ + 1’ after the endPosition |
| **Deleted Files** | **Description** | **Deleted Functions** | **Description** |
|  |  |  |  |

# Set Up and Improve Instructions

<Provide instructions on how to set up and compile your code. If you have special instructions on how to run your code to get expected results, describe them here. If you set up a new framework or similar, explain how to set it up and run it here. Any special instructions to help future teams with the project you worked on or began, provide in great detail here. >

Use MainGUI as the main function to compile the whole code.

# Final Project Plan

<Give a week-by-week summary of project timeline. Be sure to specify when you began development, testing, research, analyses, and other activities. >

|  |  |
| --- | --- |
| Week | Goals |
| Week 1 | Project plan, Basic idea of the project, Basic UCC function usage(Counting, Differencing, Extfile, Cyclomatic Complexity, Maintainability Index) -rc |
|  | Do the assignment 1 and learn several function of UCC and how to use the existing UCC code through reading the document and the source code.--zs |
|  | Finish assignment 1 and learn how to use ucc in command line. Reading document and source code.--xl |
| Week 2 | Learn how to use SVN and try to find an efficient software. Find differences between the latest version and the old version. Build up GUI environment and research about previous work--rc |
|  | Meet with each other at class and discuss about the project plan. Learn how to use svn.--zs |
|  | Group meeting, make a plan for the project. Set up SVN and learn how to use SVN.--xl |
|  | Joined team in this week. Plan to finish assignment 1, learn how to use SVN and study the source code. Catch up with team’s project plan. --xiaopeng |
| Week 3 | Started implementing MI for Java from ucc-j\_2018.05 to ucc-g1.3.1--rc |
|  | Integrated the java GUI code to the UCC-g 1.3.1 and try to understand what the former group did with java swing.--zs |
|  | Start to integrate codes, read different version codes.--xl |
|  | Debug for Java and Python, also integrate code or changes from 2019 Spring Team’s integration report. --xiaopeng |
| Week 4 | Finished implementing MI for Java--rc |
|  | Implemented the drag and drop feature that can fill the file path automatically in java GUI.--zs |
|  | Working on integration, found there are quite a lot difference between ucc function level and ucc-g 1.3.1. |
|  | Take a close look at the integration report and Change Log, compare the code in Release 1.3.1, 2019 Spring Team’s code, and 2018.05 code, and finish integrate code mentioned above. --xiaopeng |
| Week 5 | Corrected results of MI implemented last week and Fixed bugs in MI --rc |
|  | Had some research on how to implement resizing in java swing and finnally decided to use GridBagLayout to implement this feature.--zs |
|  | Finished integration, but because there are quite a lot differences between ucc-java 2018.05 and ucc-g 1.3.1. Even integration is finished, it could not work out. --xl |
|  | Read Bugs Found Report from 2019 Spring team, check the bugs they’ve fixed, and modify them in Release 1.3.1. Finish integrate all bugs in Bugs Found Report. --xiaopeng |
| Week 6 | Working on MI for Python, adjust code formates and fixed bugs--rc |
|  | This week I have completed changes on basic panel and I found one bug in their code that the advanced button did not appear on the frame. And I figure out that they used getResource function but did not put the resource on svn, so it can not find that image.--zs |
|  | Learning about the bugs, the main bug found is that it could not read all files. --xl |
|  | Fix bugs in Bugs Found Report that have not been fixed yet, also fix new bugs come out due to the newer version. --xiaopeng |
| Week 7 | Continue working on MI for Python and fixing bugs--rc |
|  | This week I have completed the resize of GUI. I found one bug that sometimes the advanced and basic button could not respond to my click, which is caused by the use of mouseClicked. With this function the button will not work when you have some little move during the press and release operation. The solution is to change the mouseClicked to mouseReleased.--zs |
|  | Looking for the reason for bugs, the main reason is on reading file system. One of the possible reasons is operating system difference.--xl |
|  | Reading counting standard of UCC in C++ version. Fix bugs for web language counters, including HTML, JavaScript, CSS, ASP, ColdFusion and ColdFusionScript. --xiaopeng |
| Week 8 | Finished MI for python--rc |
|  | Fix bugs that gui can not pass the right value to ucc main.--zs |
|  | Finding the reason for bugs, but because I have little knowledge about system and command line. This week I am mainly checking if there was a problem with the command line I used. --xl |
|  | Fix bugs for ColdFusion and ColdFusionScript. Fix bugs for the two languages to get correct cyclomatic complexity after generating correct output files. --xiaopeng |
| Week 9 | Fixed bugs in Mi for Java and now the results are almost same with ucc-j\_2018.05--rc |
|  | Fix bugs that some commands are not passed to uccMain correctly. Add new buttons in GUIto support function level. Try to integrate the whole team’s code together with bitbucket. Learn the git command.--zs |
|  | Get contact with earlier team member, and learned that there is no problem with the command line I used. So I paid more attention on if there is a problem with my integration. However, I barely found any problem. And worked on integrating with other group members. -- xl |
|  | Study Git commands, merge codes together with team members. Fix conflicts and bugs after the integration. Study bugs in Bugs Found Report and fix unsolved bugs. --xiaopeng |
| Week 10 | Integrate code with team work and write final report.--rc |
|  | Write the final report and related documents, submit everything required.--zs |
|  | Writing final report, submit files required. --xl |
|  | Revise code with conflict after integration. Write final report. Write Bugs Fixed Report to show detailed information about modifications. Write Bugs Report to indicate unsolved bugs. --xiaopeng |

# Team’s Strengths and Weaknesses

<Describe team members’ strengths and weaknesses, as well as how team worked together and the team’s strengths and weaknesses. >

Good team communication and cooperation.

# Final Deliverables Checklist

* Completed Final Report
* Copy of original/baseline source code (before modifications) – be sure it is the code you used to make modifications to
* Final source code with modifications
* Final version of Test Cases
* Final version of Defect Log
* Final version of all other documents
* Other. Explain: