**CS 6356.001: Software Maintenance, Evolution & Re-Engineering**

**Assignment 5: Mining issue trackers and version control data**

**Team 17**

**Team Members:**

* Shivani Talatam
* Zamaan Bawa
* Pritul Dave

## **3. Analyzing issue tracker activity**

Q) How many issues have been reported in the project

1. 5488

Q) How many of these issues are bugs, improvements, new features, tasks, wishes, sub-tasks, and tests? What are their percentages? How many issues of each of these types have been resolved or closed? What are their percentages? Fill in the following table:

| Type of issue | Total | Resolved/Closed |
| --- | --- | --- |

|  | # | % | # | % |
| --- | --- | --- | --- | --- |
| Bug | 3978 | 74.3 | 63 | 1.6 |
| Improvement | 1012 | 18.91 | 59 | 5.83 |
| Task | 139 | 2.6 | 13 | 9.3 |
| Sub-Task | 93 | 1.73 | 15 | 16.12 |
| Wish | 106 | 2 | 0 | 0 |
| Test | 21 | 0.4 | 1 | 4.76 |
| **Total** | 5349 |  | 151 |  |

## **4. Finding co-changed code files**

| No of co-changed files | List of co-changed files | List of commits |
| --- | --- | --- |
| 2 | 1.  pdfbox/src/main/java/org/apache/pdfbox/contentstream/PDFGraphicsStreamEngine.java  pdfbox/src/main/java/org/apache/pdfbox/contentstream/PDFStreamEngine.java  2.  pdfbox/src/main/java/org/apache/pdfbox/pdfparser/BruteForceParser.java  pdfbox/src/main/java/org/apache/pdfbox/pdfparser/COSParser.java  3.  preflight/src/test/java/org/apache/pdfbox/preflight/integration/TestValidFiles.java  preflight/src/test/java/org/apache/pdfbox/preflight/integration/TestInvalidFiles.java  4. pdfbox/src/main/java/org/apache/pdfbox/pdfparser/BaseParser.java  pdfbox/src/main/java/org/apache/pdfbox/pdfparser/COSParser.java | 1. F6be6a8424bcb82af10632dd8ef7ae68947af2b2 2. E5d4688094e3b8f5efc2d287ba265a38488e101d 3. B9273c34bf11968c3bf25122efd79197f04c8964 4. A6a0de7f5605d3919463807ad4936d8da194d92c 5. 68e80e961a352264533b922838a37af64d53b76a 6. 137827ee6e6c81dda86d16f78fbec8106de21e29 7. D3576eef8e429a95e2b49696a8039e4f417a6ad5c 8. 599d33747c5d4bd3392d0f9dcb25cdb8f9638e2c 9. 3a66990e0edf7b60d45fc0ef55ea02a353127990 10. 91bb9040f078f1511e045e2ab93316a55d9e4066 11. 45f11130656f6d0f9d78fcb7753b6541eaec2843 12. D375bb19174ef6ba27c00aa9e62e4995c7657a37 13. ef4b8ef75811124dfc22cc8a852a613b1d5f39e8 |
| 3 | 1.  pdfbox/src/main/java/org/apache/pdfbox/cos/COSDocument.java  pdfbox/src/main/java/org/apache/pdfbox/cos/COSStream.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/PDDocument.java  2.  tools/src/main/java/org/apache/pdfbox/tools/pdfdebugger/treestatus/TreeStatus.java  tools/src/main/java/org/apache/pdfbox/tools/gui/PDFTreeCellRenderer.java  tools/src/main/java/org/apache/pdfbox/tools/gui/PDFTreeModel.java  3.  fontbox/src/main/java/org/apache/fontbox/cmap/CMap.java  fontbox/src/main/java/org/apache/fontbox/cmap/CMapParser.java  fontbox/src/test/java/org/apache/fontbox/cmap/TestCMapParser.java | 1. 16ec5c675c4f03ce78781edb716b6c05d4b4793d 2. e27b7011900a9890cdff62648cfcfc266547ae6e 3. 5cba058f66191cf3f117ef7ee597c7bc84979645 4. 4d4d21b4f71679d850286fa9211b09685967f8d8 5. 5f494c687d441706a7c664a0c175da811e6b7248 6. B410b2b236f163fa706d5700b377e3f259bcb799 7. Bd46b391bd0f662b7fa2ee746efe1e340c7b6190 8. D5de71aef48333947f15c34ca732884569a34532 9. 67596540710770555f41859aa30dd0d0fd7ac621 |
| 4 | 1.  src/org/pdfbox/io/RandomAccess.java  src/org/pdfbox/io/RandomAccessFile.java  src/org/pdfbox/io/RandomAccessFileInputStream.java  src/org/pdfbox/io/RandomAccessFileOutputStream.java  2.  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/interactive/digitalsignature/visible/PDFTemplateStructure.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/interactive/digitalsignature/visible/PDVisibleSigBuilder.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/interactive/digitalsignature/visible/PDVisibleSigProperties.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/interactive/digitalsignature/visible/PDVisibleSignDesigner.java | 1. Bcb9c0059a85347775615e8ad7522f65d3000758 2. 623a0af37723377676f57213cf74b1a909c23272 3. D18984f70b1197cbf53e2e3f6472018b26ced7b5 4. 0043363995339865503811c70dd991b57c06b3cf 5. 4d3263ea6ca045f7dced5899e1885da2e2f8c7e3 6. 2a262cca47ebbdcef1b2f5bb0a2269d3710720e9 7. 18dd7ffab57b66e80879be7edaa92c08058d6e9b |
| 5 | 1.  pdfbox/src/main/java/org/apache/pdfbox/contentstream/PDContentStream.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/PDPage.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/font/PDType3CharProc.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/graphics/form/PDFormXObject.java  pdfbox/src/main/java/org/apache/pdfbox/pdmodel/graphics/pattern/PDTilingPattern.java | 1. D1af0a1cd9e3939e2725b340de10a3747b4e3c58 2. Edf632cd527818ab9de3e3cbfb7c870b17acb352 3. 78e2cac2c9a5235452c0ed2e64a136355f1c2f67 |

**3 Co-Changed files:**

Looking at the set of 3 co-changed files that occur the most, we have the first set which included:

* pdfbox/src/main/java/org/apache/pdfbox/cos/COSDocument.java
* pdfbox/src/main/java/org/apache/pdfbox/cos/COSStream.java
* pdfbox/src/main/java/org/apache/pdfbox/pdmodel/PDDocument.java

These files changed together 12 different times. Many of the commits feature these files alongside other files that are relevant to the cos directory. It makes sense that these files are being changed together since they all share a similar functionality. These three classes revolve around representing the memory of the PDF document.

We used MetricsReloaded for IntellliJ to evaluate the coupling for these files. Here is a table with our results for the CBO (Coupling Between Objects) and RFC (Reference For Class).

| Class Name | CBO | RFC |
| --- | --- | --- |
| * pdfbox/src/main/java/org/apache/pdfbox/cos/COSDocument.java | 61 | 73 |
| * pdfbox/src/main/java/org/apache/pdfbox/cos/COSStream.java | 125 | 68 |
| * pdfbox/src/main/java/org/apache/pdfbox/pdmodel/PDDocument.java | 330 | 212 |

The coupling is rather high between these classes. The PDDcoument class is coupled to 330 classes which is very high. The two COS classes are also pretty high, but the PDDocument is very dependent on other parts of the project. This is not great coding practice since it will be very difficult to maintain and resolve. This is most likely why these classes are always being co-changed alongside many other classes. They are all dependent on each other which forces the developer to do more work.

Looking at the RFC for these classes, they are also high. Each class has many responsibilities which will also cause difficulties for the developer. Ideally, these values should be much lower. PDDocument handles many responsibilities with 212 as the value. We could look to extract this class into smaller subclasses to avoid the complexity.

**2 Co-Changed files:**

The below java files are a set of 2 co-changed files that occur the most number of times compared to the other sets:

* pdfbox/src/main/java/org/apache/pdfbox/pdfparser/BaseParser.java
* pdfbox/src/main/java/org/apache/pdfbox/pdfparser/COSParser.java

These co-changed files occur in the ‘pdfparser’ directory. These files are related because the class ‘COSParser’ extends ‘BaseParser’, this means that changes to the BaseParser class can affect the behavior of the COSParser class, and vice versa. The changes occur together 19 times.

| Class Name | CBO | RFC |
| --- | --- | --- |
| * pdfbox/src/main/java/org/apache/pdfbox/pdfparser/BaseParser.java | 22 | 108 |
| * pdfbox/src/main/java/org/apache/pdfbox/pdfparser/COSParser.java | 32 | 196 |

Based on the provided data, we can infer that the COSParser.java file has a higher degree of dependence on other classes than the BaseParser.java file, as indicated by its higher CBO value of 32 compared to 22 for BaseParser.java. This means that COSParser.java is likely to be more complex and may have more interactions with other classes in the codebase.

The COSParser.java file has a higher RFC value of 196, indicating that it has a larger number of unique methods that can be executed in response to a message compared to BaseParser.java, which has an RFC value of 108. This suggests that COSParser.java may have more functionality and be more complex than BaseParser.java.

Overall, the data suggests that COSParser.java is a more complex and dependent class than BaseParser.java, which may be expected since COSParser.java extends BaseParser.java. However, it's important to note that these metrics alone do not provide a complete picture of the quality or performance of the code, and other factors should also be considered when evaluating the codebase.

## **5. Linking changes and issues**

* The script **“change\_issue\_link.py”** is prepared to link changes and issues
* The csv file **“final\_output.csv”** is prepared, containing the issue-id, commits, source files added, source files modified, source files deleted.
* It is also observed that there are certain issues in the issue-tracker whose commits are not recorded

**Conclusion:**

* The average number of modified files are 17.14
* The average number of added files are 2.54
* The average number of deleted files are 2.32
* Looking at the statistics, we can see that when we have a higher number of commits, there will be a higher number of changes in the source code files.

*[Average is calculated using inbuilt formula in excel]*

**Team Contributions:**

* Shivani Talatam
  + Worked on the co-changed files table
  + Analyzed the set of 2 co-changed files which occurred the most.
  + Effort: 30%
* Zamaan Bawa
  + Worked on co-changed files table
  + Analyzed the set of 3 co-changed files which occurred the most.
  + Effort: 30%
* Pritul Dave
  + Analyzed the issue tracker and completed the issue report
  + Worked on linking changes and issues and developed the change\_issue\_link.py script
  + Effort: 40%