**Software Testing Mini Campaign**

**Week 13 Third-Party Code Testing**

**Issac Jose Ignatius (1004999)**

Xu Muzi:

<https://github.com/xl-Mu/50.003-Software-Testing-Mini-Campaign>

*Code Inspection:*

Upon inspecting her code, I have noticed the following:

* A user is unable to select a unique combination of columns to allow for comparison between the 2 Files – This is a requirement stipulated for the Mini Testing Campaign. Hence, the user is only able to perform a full row comparison between the 2 files.
* The CompareCSV Program performs a line-by-line comparison between the 2 Files. This means that it is sensitive to duplicate records and the order of the records.

Based on the above observations, I have identified a set of bugs. The **most critical bug** is that the software program is unable to compare records against a unique combination. This was a functional requirement of the CompareCSV Program.

*Blackbox Testing:*

Then, I tried to do the following –

1. Check the program’s ability to work with duplicate records
2. Check the program’s ability to work with records with different number of records in both files
3. Check the program’s ability to work with different order of columns
4. Check the program’s ability to work with different order of rows

Code and Files Used: xl-Mu-50003-Software-Testing-Mini-Campaign.zip (Refer to issac-j17/SUTD\_ESC\_2022\_Software-Testing-Mini-Campaign GitHub Repository).

Result – All Tests Failed. These 4 additional bugs arise from a Design Flaw in the Program where the comparison is made line-by-line instead of a row-by-row comparison. Hence, the Program is unable to recognise duplicate records, different order of columns and different order of rows. Also, a Boolean AND is used to check if both files have a next line. Hence, the comparison is truncated by the file with fewer number of rows present.

Emmanuel J. Lopez:

<https://github.com/PurpleDice95/ESC-HW>

*Code Inspection:*

Upon inspecting his code, I have noticed the following:

* A user is unable to select a unique combination of columns to allow for comparison between the 2 Files – This is a requirement stipulated for the Mini Testing Campaign. Hence, the user is only able to perform a full row comparison between the 2 files.
* The CompareCSV Program only checks whether the next Line exists only for f1. This means that the if f2 is longer than f1, its contents would not be recorded.
* With reference to the README.md file, it mentions that the 2 files must have a column labelled “Customer ID”. While the code checks for “Customer ID#” (with a hash\*) and it also restricts the comparison due to the Introduction of this new column, the main issue is that it does not mention that the “Customer ID#” must be the first column. This will result in an exception even if both files have the column in a different position.

Based on the above observations, I have identified a set of bugs. The **most critical bug** is that the software program is unable to compare records against a unique combination. This was a functional requirement of the CompareCSV Program.

*Blackbox Testing:*

Then, I tried to do the following –

1. Perform a comparison between 2 Files of Different Record Size. The 1st file has fewer rows than the 2nd file. The output.csv file is expected to generate the records that are present in 2nd file but absent in 1st file.
2. Perform a comparison between 2 Files where the position of “Customer ID#” is fixed but the order of columns is changed for the rest. The output.csv is expected to generate no exceptions.
3. Perform a comparison between 2 Files where the position of “Customer ID#” is changed but the contents are all the same. Also, the program exception “No ID field” should **not** be raised.

Note: The poor documentation of how the user should generate the CSV File especially “Customer ID” in README.md vs “Customer ID#” in code validation can be seen as a bug. This is because the validator is character specific. Also, the position of “Customer ID#” not being mentioned to be the First Column can also be seen as a bug. This is because the code is reliant on the position of Customer ID# when validating the contents.

Code and Files Used: PurpleDice95-ESC-HW.zip (Refer to issac-j17/SUTD\_ESC\_2022\_Software-Testing-Mini-Campaign GitHub Repository).

Result – All Tests Failed. These 3 additional bugs arise from a Design Flaw in the Program where both files are parsed together when each file should be parsed separately. Hence, it is prone to a bug when files have different number of rows. Next, the files are compared without any regard to the order of columns. This would severely impact the comparison as it is no longer doing a row comparison but file comparison instead. Finally, “Customer ID#” requirement is not necessary in the first place. Nonetheless, not explicitly stating that the position of the column “Customer ID#” being the first column also results in bugs.