NON-CONFORMANCE REPORT

Enron Email Analysis

NAME: Chetane Sookdeb

STUDENT ID: 20918918

Non-Conformance Report

The Problem:

1. The first problem encountered during the assignment was using an incorrect approach and lacking SQL queries. The initial approach taken did not align with the concepts and techniques learned in class. As a result, the assignment did not reflect the understanding and application of the SQL querying skills acquired.

2. The second problem involved the inefficient utilization of code. Multiple lines of code were implemented to accomplish specific tasks, which could have been achieved more effectively using the appropriate modules or techniques. This resulted in higher code complexity and reduced efficiency in achieving the desired outcomes.

3. The final problem encountered during the assignment was the occurrence of numerous errors throughout the testing phase. These errors included common types such as syntax errors, attribute errors, name errors, and operational errors. These errors hindered the smooth execution and functionality of the code, delaying achieving the desired results.

Severity & Impact:

1. The first problem regarding the absence of SQL queries significantly impacted the assignment. On a severity scale of 1 to 10, this error is assigned a severity level of 10. Failing to apply the knowledge of SQL queries resulted in a substantial deviation from the assignment requirements. It required a complete restart of the assignment, causing significant delays and potentially jeopardizing the successful completion of the task.

2. The second problem, which involved the inefficient utilization of code, has a lower severity level. On a severity scale of 1 to 10, this error is assigned a severity level of 3. While the program could still function correctly despite the increased lines of code, the error caused a delay in completing the assignment due to the additional effort required to write and manage the excess code.

3. The last problem, which encompassed various common errors encountered during the testing phase, holds moderate severity. On a severity scale of 1 to 10, this error is assigned a severity level of 5. The impact of this problem was manifested through delays in the assignment's progress. The need for multiple reviews, error identification, correction, and retesting consumed additional time and effort, resulting in a moderate setback.

How it occurred:

1. The first problem occurred due to a misunderstanding the assignment's deliverables. In the beginning, there was a misconception regarding the specific requirements and expectations of the assignment. This misunderstanding persisted until the realization occurred. However, proactive measures were taken to identify and rectify the error before it caused significant consequences.

2. The second problem arose from a habit of writing lengthy and complex lines of code. This approach was utilized as a means to break down the tasks into smaller steps. However, in this instance, it resulted in unnecessary complexity and an increased number of lines of code. The error was eventually resolved through extensive research and exploration of appropriate libraries and techniques.

3. The last problem, involving multiple errors during the testing phase, primarily originated from typographical errors. Despite taking precautions, such errors are often inevitable and can occur intermittently. However, the use of certain tools and resources facilitated the identification and correction of these errors.

How to prevent it from happening again:

1. To prevent the recurrence of the first problem, it is crucial to read and understand the assignment deliverables thoroughly. Take the time to review the requirements and objectives of the assignment in a task-by-task manner. Regularly cross-reference the progress made with the deliverables to ensure they are being fulfilled accurately and in line with the assignment's purpose.

2. The second problem can be avoided through consistent practice and continuous improvement. You can refine your skills and develop efficient coding habits by dedicating regular time to coding practice. Emphasize the importance of writing concise and optimized code to accomplish tasks, exploring different coding techniques and libraries to simplify complex tasks whenever possible.

3. To mitigate the occurrence of typographical errors and other common mistakes, it is essential to adopt careful coding practices. Continuously verify each line of code while writing and pay close attention to details. Utilize code editors or integrated development environments (IDEs) that provide real-time error detection and highlighting. Additionally, leverage tools such as linting and spell-checking to identify and rectify errors proactively.

Plan or time estimate to fix:

1. To address the first problem and ensure the proper application of SQL queries in the assignment, it was estimated that an additional 2 full days would be required. This time would be allocated to reviewing the assignment requirements, studying and practicing SQL querying techniques, and implementing the necessary changes to align with the intended deliverables.

2. For the second error, which involves optimizing the code by using the appropriate modules, a time estimate of approximately 2-3 hours was projected. This timeframe was dedicated to researching and identifying suitable modules, refactoring the existing code, and testing its functionality to ensure the desired results are achieved with improved efficiency.

3. The last error, encompassing the resolution of various common errors during the testing phase, was anticipated to take around 30 minutes. This duration would involve careful inspection and debugging of the code to identify and rectify any remaining syntax, attribute, name, or operational errors. It is essential to conduct thorough testing after each correction to ensure the error-free functionality of the program.