

| Business Template  **Frameworks & Advanced Selenium** |
| --- |
| Task 1 *Project ‘B’. Duration of 3 years. The scope of the project is the implementation of the Data Platform in the cloud using the on-premises data. The data will be extracted daily and processed by ETL process in the cloud platform. The testing team is planned to be 2 QAs. 1 QA is a technical person and the second is only worked as a manual tester. The Dev team consists of 4 persons. The main activities are data extraction from the on-premises Database (+ETL), setup of the Dashboards with the new logic and complex calculations. There is no automation testing framework available and the tech stack on the project is mostly Python based + many services which are familiar with.*  **Project B requires test automation because the system involves daily ETL processes and complex dashboard calculations, making manual testing time-consuming, repetitive, and error-prone over the planned three-year duration. The testing team consists of two QAs, one technical and one manual, while the development team has four members. Currently, there is no automation framework, and the tech stack is mostly Python with various cloud services, which makes Python-based automation feasible. Automation is necessary to improve reliability, accelerate regression testing, and reduce repetitive human effort, while the manual QA can focus on exploratory testing and edge cases.**  **For this SUT, automation should focus on ETL data validation to ensure extracted on-premises data matches cloud storage, data integrity checks to verify transformations, and testing dashboard calculations with complex formulas and aggregations. Selenium can be used for automating dashboard UI testing, while Python scripts handle ETL and backend validation. Tests should be parameterized so that new scenarios can be added in separate files without affecting existing automation scripts, supporting scalability and maintainability. A framework design will be implemented to organize tests into categories such as unit testing, regression, and UI testing, making implementation, maintenance, and debugging easier.**  **A CI/CD pipeline in Databricks can be used to test each layer as data flows downstream, allowing failures to be detected early with a build-fail approach, reducing time to locate and fix issues. Supporting automation requires version-controlled test scripts, test data management, and continuous monitoring.**  *2. Define and collect information and criteria for calculating the ROI (Return on Investments) value for this SUT. Provide a rationale for your calculation with a description of each step.* *Step 1: Define the Cost of Manual Testing (CM)* **CM represents the total man-hours that would be spent if all testing were done manually.**   * **Manual testing for Project B involves daily ETL validation, data integrity checks, and dashboard verification.** * **Estimated at 20 hours per week for one QA.** * **Duration of the project: 3 years, 52 weeks per year.** * **Calculation:** * **Rationale: Daily data and dashboards require repetitive checking; 20h/week accounts for realistic human workload without overloading the QA.**  *Step 2: Define Investment into Automation (I)* **Investments into automation include all man-hours required to implement and maintain automated tests. These consist of:**   1. **Framework Implementation (FW)**     * **Setting up a Python-based automation framework for ETL, data integrity, and dashboards.**    * **Includes folder structure, test runners, reporting, and logging.**    * **Estimated effort:**    * **Rationale: The technical QA can set up a medium-complexity framework in this time.** 2. **Scenario Creation (S)**     * **Writing automated test cases, parameterizing them for easy addition of future scenarios, and debugging.**    * **Estimated at 30 hours/week for 26 weeks (6 months).**    * **Calculation:**    * **Rationale: Includes time for creating ETL and dashboard tests, including Selenium for UI testing. Parameterization allows new tests to be added in separate files without interfering with existing scripts.** 3. **Automated Test Execution and Result Analysis (E + R)**     * **Running automated tests and reviewing failures.**    * **Estimated at 6 hours/week for 52 weeks × 3 years.**    * **Calculation:**    * **Rationale: Once automated, minimal human intervention is needed to monitor results and fix occasional test failures.**  * **Total Investment (I):**  *Step 3: Apply ROI Formula* **The ROI formula is:** Step 4: Interpretation and Rationale  * **The ROI of 75.7% indicates that the time saved by automation significantly exceeds the effort invested.** * **CM – I = 1,344 man-hours represents the net gain in human effort over three years.** * **This gain justifies the investment in framework, scenario creation, and test execution.**   **Automating repetitive ETL and dashboard checks reduces human errors and allows the manual QA to focus on exploratory testing and edge cases.** |

# Task 2

***1.*** *Use the framework code as a skeleton.*

***2.*** *Integrate your unit and integration tests from the previous modules into your PyTest + Selenium framework.*

***3.*** *Add 2 checks for the Power BI report.*

***4.*** *The following Microsoft Income Statements report (PowerBI tab) will be used.* [*income statements report*](https://www.microsoft.com/en-us/Investor/earnings/FY-2023-Q3/income-statements)

***5.*** *Implement 2 tests for the report like the presence of some buttons or fields, titles, legends, or the presence of some bar, etc.*

***6.*** *Use the framework code which can be found here:* [*DQE\_LAB\_2024*](https://github.com/AntonLazarchik/DQE_LAB_2024)

