**Task 1 ROI** *(Return on Investments)*

Firstly, this task is about gaining a deep understanding of EPAM’s area of activity. By understanding the core of the subject, we can become more competitive in the market. It also allows for clear profit planning and forecasting. For EPAM specialists, it represents an opportunity for career growth and earning bonuses. 😊

**Optional QA to understand subject**

| **№** | **Question** | **Answer** |
| --- | --- | --- |
| 1 | Is it necessary to set up test automation processes for this SUT? Why? | Yes, because the project is long-term (5 years) and complex (ETL, DWH, ML models). Automation will improve quality and speed up testing. |
| 2 | What should/could be automated for this SUT? Why and how? | ETL process validation, dashboard calculations, ML model outputs, and basic UI testing using the existing framework (BDD + Selenium + JUnit). |
| 3 | Which processes or practices need to be set up to use automation effectively? | Set up CI/CD pipelines, structure test cases in BDD format, implement code reviews for automated tests, and regularly update test documentation. |
| 4 | Are there any specific project requirements for testing processes/tools? | Yes, it is preferable to use the existing framework (Gherkin + Selenium + JUnit) maintained by another client team. |
| 5 | What programming language is mostly used by the team? Are there any automated tests already? | The main language is Python. Some automated tests exist but are maintained by a different team. |
| 6 | What is our current project stack? | Python, cloud platforms (cloud ETL, DWH), Selenium + JUnit for automation, Gherkin for test scenarios. |
| 7 | Who will be involved in testing processes? Who will use and maintain the automated tests? | 2 technical QAs and 1 manual tester. The automated tests will be used by the QA team and DevOps. |
| 8 | What is the size of the testing team? | 3 people. |

I have chosen Project “C” because it is a long-term and complex project. Investing in test automation for this project will allow us to significantly reduce manual testing efforts over time, leading to considerable savings in both time and costs during project execution. Additionally, the complexity of ETL processes, dashboard validations, and ML model outputs justifies the need for a stable and reusable automated testing framework.

**Mandatory QA to understand subject**

**1. Why is it necessary to set up automation processes?**  
Automation is necessary because the project is based on building a cloud DWH and developing ML models, which means frequent updates, large data volumes, and the need for reliable regression testing. Manual testing would be too slow, error-prone, and costly over time.

**2. What should/could be automated? Why? How?**  
Critical business flows, main UI interactions, API endpoints, and data validation processes should be automated. This ensures that after each deployment, essential functionalities are checked quickly. Automation can be implemented using the existing Gherkin + Selenium + JUnit framework, integrated with Jenkins for scheduled runs.

**3. ROI Calculation**

The calculation of ROI (Return on Investment) is based on the following:

* **Basic Framework (FW) setup**: 80 hours.
* **Creation of TA scenarios (S)**: 40 hours/week × 6 months ≈ 1120 hours.
* **Time spent on test execution (E + R)**: 8 hours/week.

For each project:

* **Project A**:  
  Manual testing requires 20 hours/week.  
  Investment in automation: 1696 hours.  
  ROI is **-39%**, meaning the investment does not pay off within the observed period.
* **Project B**:  
  Manual testing requires 20 hours/week.  
  Investment in automation: 3048 hours.  
  ROI is **2%**, meaning the investment barely starts to return after the automation phase.
* **Project C**:  
  Manual testing requires 20 hours/week.  
  Investment in automation: 4440 hours.  
  ROI is **17%**, meaning the automation pays off and brings additional efficiency over time.

**4. ROI Calculation Steps**

**Step 1:**  
Estimate manual testing effort:  
e.g., 20 hours/week.

**Step 2:**  
Calculate manual testing time over project duration:  
e.g., 20 hours/week × 52 weeks/year × project duration.

**Step 3:**  
Calculate automation investment:  
FW + S + E + R hours from the table.

**Step 4:**

ROI Formula:

CM – the cost of manual testing (man-hours)

I – investments into automation (man-hours)

FW – time spent implementing the framework.

S – time spent creating TA scenarios.

E – time spent for tests execution (human job);

R – time spent on results analysis.

**Step 5:**  
Interpret the result:

* Negative ROI → automation is not profitable within the project duration.
* Positive ROI → automation brings savings after initial setup.

**5. Conclusion:**  
Automation is necessary for the project to ensure fast, repeatable, and reliable testing, especially considering data volumes and frequent deployments. Although initial ROI is slightly negative, long-term savings and quality improvements fully justify automation setup.

Atachments:

SUT\_choose.xlsx