|  |  |
| --- | --- |
| D:\PAL\Logo_4_21_15\Logo_4_21_15\Primary Logo\png_files\EPAM_LOGO_Full_Color_RGB.png | L&D Automated Testing Mentoring Program |

Module 1: Introduction to test automation

## home task

As a basis for this home task, take SUT (system under test) of your current project.

1. Think about things below:

* Is it necessary to set up test automation processes for this SUT? Why?
* What should/could be automated for this SUT? Why? How?

Write a short essay answering the questions above.

## Answer:

I am currently a part of the team working for a client, who has been outsourcing their work to EPAM for 12+ years. This means, the applications we develop and support appear to be huge systems, which consist of lots of services. It would be near to impossible to manage the releases without test automation, or would require enormous number of manual testers to do “monkey job” rerunning same test cases over and over again, let alone writing them. That is one of two main reasons we use automation broadly, trying to cover as much as possible, even including edge cases, if we have capacity to do so.

The second main reason we have automation is to have fast feedback on newly developed code. It is a very common situation when our tests catch bugs just after new build of the application becomes available. Moreover, if written clearly, developers might run them locally before even trying to merge their code to master branch. However, I hardly ever seen such devs, they’re like unicorns😊

It is very important to have balance in the Test Automation Framework(TAF), so that there’s not too many and not too little tests. Apart from Unit-tests(which are usually written by developers), the first thing that has to be automated is application’s services or APIs.

First, we need healthchecks, also known as smoke checks. These are fast tests, which run as often as the budget permits. It’s important to use an effective alerting system for them. After that, it is helpful to have some complicated API tests. They will show, if a piece of functionality was broken or impacted. The last chunk of tests we might add to API suits consists of some e2e user scenarios. For example: login->create an event->edit an event-> delete an event -> logout.

Regarding UI tests, the builds to run should have nearly same structure as API. One crucial thing they have to have in common is independency. The more independent the suits are – the better. Nevertheless, we should always remember to mimic end user’s workflow to get relevant fast feedback from our tests.

1. Define and collect information and criteria for calculating ROI (Return of Investments) value for this SUT. Provide rationale of your calculation with description on each step.

As a formula for ROI you may use simplified one (that is based on man-hours approach) as:

,

CM – cost of manual testing (man-hours)

I – investments into automation (man-hours)

,

FW – time spent for implementing framework;

S – time spent for creating TA scenarios;

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

R – time spent for results analyzing.

## Answer:

Let the project be in active development and support for 6 years. Average time required for manual testing is 80 hours per week.

Considering only first 2 years are dedicated to active development and other 4 to change requests, bug fixes and support, we will need 120 man-hours per week for the first half a year to implement TAF, a year and a half to write scripts spending 80 hours per week. It will take 20 hours per week for executing, analyzing results and supporting test for the rest 4 years

This takes us to ROI

Reviewing these calculations, it can be easily understood that there’s a huge benefit of test automation in a long run even if it might be more expensive in the first two years.

## BONUS TASK:

1. Use **ROI calculator from additional materials provided “**AutoTest-ROI-Calculator.xls” for detailed ROI calculation.
2. Create detailed automation plan similar to EPAM test plan template.

## acceptance criteria

1. Answer is given for a question: “Why/why not” it’s necessary to set up automation processes for the project?
2. Answers are given for questions: What should/could be automated? Why? How?
3. ROI is calculated.
4. There should be descriptions provided for each step of ROI calculation.
5. There should be clear conclusion about the results of ROI calculation.
6. Bonus task: detailed plan is done according to the template
7. Bonus task: screenshots of detailed calculations in **Online ROI calculator** view are provided to a mentor.