Test Automation Mentoring Program: Advanced

TEST AUTOMATION FRAMEWORK DESIGN. HOME TASK

Tech stack: .NET, Java, JavaScript

TABLE OF CONTENTS

INTRODUCTION

You will be working with Report Portal application either deployed locally on your workstation or using rp.epam.com during all the mentoring program.

In this task please use tools you **have been working with**. All new tools will be added in the framework later. The goals of this module are:

- Create version control repository
- Practice with Docker
- Practice building TAF from scratch
- Build the basement for further exercises

MARKS

Points	Mark	Description
1	0.5	Unsatisfactory
2	1	Unsatisfactory
3	1.5	Unsatisfactory
4	2	Satisfactorily
5	2.5	Satisfactorily
6	3	Satisfactorily
7	3.5	Good
8	4	Good
9	4.5	Excellent
10	5	Excellent

Home task is DONE if more than (>=) 4 points received.
Mentor can take away points if Mentee's code quality does not meet his/her requirements.

DEADLINE

It's expected that Mentee completes a task in 2 weeks. The deadline can be adjusted in case of:
1. Vacation
2. Illness
3. Business trip
4. Exceptional case

Before adjusting the deadline please notify and confirm with Coordinator about changes.

TASKS

1. CREATE PROJECT REPOSITORY

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

1. Create Git repository. Use **EPAM Git** or **GitHub** for example.

NOTE: EPAM GitLab has restrictions of working with Jenkins. So if you plan to use Jenkins as your CI tool – choose GitHub/BitBucket/public GitLab at this stage.

2. Grant Your Mentor and Coordinator with the Reporter access to your repository.

DEFINITION OF DONE

Your Mentor and Coordinator are granted the Reporter access to your git repository.

ADDITIONAL INFO

It's recommended to use GitFlow as branching strategy for your work.

2.   CREATE TEST PROJECT

Complexity	Points	Necessity
Intermediate	2	Task is Mandatory

TASK

1.   Using tools and technologies that you know (have experience with, work on your current project, know better) create a Test Automation Framework. The framework will be used for testing the Report Portal. There should be presented the following layers:

   a.   Core

   b.   Business

   c.   Tests

   Use models/business objects where needed.

   [Java Note] Use any build tool you would like to.

2.   Integrate below components into your framework (using correct layer):

   a.   Logger (Use any you would like to)

   b.   Reporter (Use any you would like to)

   c.   Test runner (Add the one which you have most experience with)

   d.   Configuration

   e.   Utilities (Add any you find useful)

3.   Add a configuration to support a web version of the report portal (https://rp.epam.com). You can use your EPAM credentials to log-in. Also adding external user is possible (e.g. your private e-mail) into your personal project

While designing your framework keep in mind that components (Logger, Test runner, etc.) can be changed during the program run, so define the approaches and think about the most appropriate architectural decisions.

DEFINITION OF DONE

Your test project is created and contains all required layers and components.

3.   SETUP CODE REVIEW PROCESS IN YOUR GROUP

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

Pass a code review course from your materials. Consider and choose a code review tool that will be used in your group for further code reviews (Git, Gerrit, VS, etc.). Consult with your group colleagues and Mentor.

Create a default branch policy – code review should be mandatory to complete merges. For group – at least 2 reviews including Mentor, for individual – just Mentor.

Note (for group mentoring) - You will be doing a code review for your group colleagues in scope of all upcoming modules.

DEFENITIONS OF DONE

Code review course passed. Branch policy set up.

4. BASIC CI

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

Set up quality gate for pull/merge requests including:

    a. Build successful
    b. Integration with pull request

DEFENITION OF DONE

Your PRs trigger your TAF solution to build.

5. SONARQUBE

Complexity	Points	Necessity
Intermediate	2	Task is Mandatory

TASK

Set up SonarQube (or any other code analyzer). Steps below are for Sonar:

    c. Add SonarQube into docker-compose file.
    d. Create a project in SonarQube and connect to your existing project.
    e. Run sonar scanner and check issues found.
    f. Fix several issues (discuss with Mentor which exactly) and re-run scanner.
    g. Share results with your Mentor.

DEFENITION OF DONE

The sonar report is shared with your Mentor before your fixes and after.

Ver.	Description of Change	Author	Date
1.0	Initial Version	Yury Karpinski	14-Oct-2020
1.1	Updated: Structure	Yury Karpinski	24-Jan-2020
1.2	Removed task #5	Yury Karpinski	26-March-2022
2.0	Merged with Code review/Code quality	Yury Karpinski	27-July-2022