LOG8371E: Software Quality Engineering (Fall 2023)

TP2 (15% weight)

Performance Efficiency

Notes:

Practical work is an important part of the course and is intended to motivate you to understand software quality assurance practices, design software quality assurance plans, develop verification strategies, and use the various tools available to evaluate the quality characteristics of software. It is recommended that you take this work seriously and use your creativity and critical thinking to make it a success. Collaboration with your colleagues is allowed during and outside the laboratory sessions. However, plagiarism regulations still apply at all times.

I) Objectives:

The objectives of this second TP are to get familiar with:

- How to define software performance objectives.
- Measurement of resource utilization from the point of view of software and systems.
- Verification and optimization of software performance through profiling and load testing.

II) Specifications:

Q1) Update the quality plan with performance considerations (15 points)

Update the quality plan to add the quality characteristics and sub-characteristics for PetClinic's performance efficiency (ISO/IEC 25010). Define two quality goals covering two sub-characteristics of performance efficiency (i.e., one goal for each sub-characteristic). Update your quality goals and quality assurance strategies. Focus on the same major feature you specified in TP1. Include the new version of your quality plan in the report.

Q2) Performance profiling (30 points)

Profile the PetClinic system using a profiler like JProfiler and VisualVM. The purpose is to identify performance bottlenecks of the PetClinic system which may provide performance optimization opportunities. You can run these profilers as either a standalone tool or as a plugin of your favorite Java IDE (e.g., IntelliJ or Eclipse). Run PetClinic (standalone or in IDE) and connect the profiler to the running application. Configure the profiler to profile the system. Focus on CPU profiling in this task. Simulate typical use of the system repetitively (so you can sample sufficient performance data), either

programmatically or manually. Submit a small user manual for profiling your subject application, including the steps for the configuration and execution of the profiler, as well as how you execute the subject application. Report the profiling results and highlight the identified performance hotspots/bottlenecks. Discuss opportunities to optimize the performance of the application.

JProfiler (industry standard, 14-day free trial):
https://www.ej-technologies.com/products/jprofiler/overview.html

VisualVM (lightweight, free): https://visualvm.github.io

Q3) Load testing (30 points)

Prepare and run load tests for PetClinic according to your performance-related quality goals. Prepare at least 4 scenarios of load (different number of concurrent queries): one of reduced load, one of average load, one of increased load, and one of exceptional increased load. You can use the JMeter tool to run the load tests. Submit a small user manual of using JMeter for your load tests, including the steps for the configuration and execution of the tool, the input data (the queries and the load). Report the load testing results (i.e., performance measures: CPU, RAM, and response time) in relation to the different load cases. Discuss your results against your quality goals and report on scenarios where the goals are not possible to achieve.

JMeter: http://jmeter.apache.org/index.html

Q4) Optimize the system and compare the performance between the two versions (25 points)

The goal of this question is to investigate the methods of improving the system performance and verify their impact. Focus on your chosen feature, investigate and implement approaches (e.g., optimization) to improve the performance of the system in terms of the defined quality goals. Describe your improvement. Perform load testing (repeat Q3) on this new version and compare the obtained performance measures (CPU, RAM, and response time) with the results from Q3 (using tables). Discuss the differences (i.e., observed performance improvement or regression) and explain possible causes of the differences.

III) Presentation:

Each team should submit a report **no longer than 25 pages**. The report must contain: 1) a table of contents; 2) an abstract that provides a clear overview; 3) an introduction that describes the subject software, the importance of its quality, the chosen feature, the considered quality characteristics, and an overview of the approaches (you can reuse part of the introduction from TP1); 4) the updated quality plan (including the quality goals and the assurance strategies); 5) the profiling approach

(manual) and results; 6) the load testing approach (manual, load scenarios) and results; 7) description of the approaches for performance improvement; 8) comparison of performance between two versions; and 9) a conclusion that summarizes the key points of the report. Diagrams and tables should be clear and have captions; diagrams and tables are self-explanatory or explained in the text. References should be used appropriately when necessary. The report shall be of high quality and be treated professionally (suppose you will submit it to the managers of a company).

IV) Evaluation:

Your report will be evaluated by both the quality of work and the quality of presentation:

Team Score = Quality of work * sqrt (Quality of presentation) * 15 (Weight of TP1)

For example, if you get 90% for the quality of work and 90% for the quality of presentation, you score will be: 0.9 * sqrt (0.9) * 15 = 12.8

Individual Score = Team Score * Contribution Factor

Contribution Factor (CF): CF = 1 for satisfactory contribution; CF = 0 for no contribution; 0 < CF < 1 for unsatisfactory contribution. CF is determined by optional peer reviews.

Optional peer reviews: Each team member may optionally email a confidential statement of work to the lab instructor within 24 hours after the due date of the assignment. A statement of work first lists in point form the parts of the assignment to which each team member contributed. In addition, the statement of work also describes whether the workload was distributed fairly evenly among the team members (you may describe the percentages of contribution by each team member). A statement of work will be used to determine the contribution factor and the score of a team member who is not contributing sufficiently to the assignment, subject to further investigations. It is not necessary to send a statement of work, if a team distributed the work for the assignment fairly evenly and each team member had satisfactory contribution.

I strongly encourage each team to use a **private GitHub repository** for the collaboration. The repository can not only help the team manage and track the work, but also provide evidence for each individual's contribution in case an investigation is needed.

V) Submission:

The work must be done in a team (same team as TP 1) and must be submitted via Moodle no later than:

➢ **November 2nd before 23:59** (one submission is needed for each team).

Please submit a PDF report that includes the names and student numbers of the team members. The file will have the name:

➢ **log8371E_TP2_[TeamName].pdf**

Late work will be penalized by 10% per day of delay. No work will be accepted after 4 days of delay.