PhD in Information Technology and Electrical Engineering
Università degli Studi di Napoli Federico II

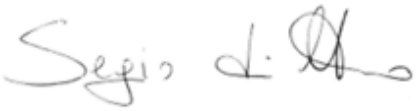
PhD Student: Sergio Di Meglio

Cycle: XXXVIII

Training and Research Activities Report

Year: First

Tutor: prof. Sergio Di Martino



Co-Tutor: Fabio Scippacercola

Date: October 14, 2023

Training and Research Activities Report

PhD in Information Technology and Electrical Engineering

Cycle: XXXVIII **Author: Sergio Di Meglio**

1. Information:

- **PhD student:** Sergio Di Meglio
- **DR number:** DR996632
- **Date of birth:** 07/11/1996
- **Master Science degree:** Computer Science **University:** UNINA
- **Doctoral Cycle: XXXVIII**
- **Scholarship type:** *Fervento srl*
- **Tutor:** Prof. Sergio Di Martino
- **Co-tutor:** Dott. Fabio Scippacercola

2. Study and training activities:

Activity	Type[1]	Hours	Credits	Dates	Organizer	Certificate[2]
Connecting the dots: investigating an APT campaign using Splunk	Seminar	2	0.4	11/11/2022	DIETI	Y
Cybercrime and Information Warfare: National and International Actors	Seminar	2	0.4	19/11/2022	DIETI	Y
Digital Forensics	Seminar	2	0.4	9/12/2022	DIETI	Y
Stabilizer Renyi Entropy and Quantum Complexity	Seminar	1	0.2	2/11/2022	DIETI	Y
Analysis and control of functional brain networks	Seminar	1	0.2	9/03/2023	DIETI	Y
Virtualization technologies and their applications	Course	23	5	30/01/2023 to 17/03/2023	DIETI	Y
How to boost your PhD	Course	16	4	11/01/2023 to 1/03/2023	DIETI	Y
Generation of Mobile EEG System: Hands-on-Experience	Seminar	1	0.2	29/06/2023	DIETI	Y
International Summer School on Software Engineering (ISSSE	Course	24	3	12/06/2023 to 15/06/2023	UNISA	Y

2023)						
Ricerca e formazione nelle società della transizione digitale	Seminar	5	1	22/09/2023	CINI	Y
Big Data Architecture and Analytics	Course	20	5	16/06/2023 to 20/07/2023	DIETI	N (waiting)
Artificial Intelligence and Natural Language Processing	Course	13	3	25/09/2023 to 4/10/2023	DIETI	N (waiting)
Co-supervised the thesis activities of a M.Sc in Computer Science student working on web Testing	Tutorship		0.6	04/2023 to 10/2023	Prof. Sergio Di Martino	
Co-supervised the thesis activities of a B.Sc in Computer Science student working on web Testing	Tutorship		0.6	5/2023 to 10/2023	Prof. Sergio Di Martino	
Presentation of a seminar lectures within the “Software Project Management and Evolution” course held by Prof. Sergio Di Martino for M.Sc degree in Computer Science about Performance Testing	Tutorship	2	0.2	5/2023	Prof. Sergio Di Martino	
Presentation of a seminar lectures within the “Software Project Management and Evolution” course held by Prof. Sergio Di Martino for M.Sc degree in Computer Science about Static Code Review and SonarQube	Tutorship	2	0.2	5/2023	Prof. Sergio Di Martino	

2.1. Study and training activities - credits earned

	Courses	Seminars	Research	Tutorship	Total
Bimonth 1	**0**	**1,4**	**7**	**0**	**8,4**
Bimonth 2	**0**	**0**	**10**	**0**	**10**
Bimonth 3	**9**	**0.2**	**8**	**0**	**17,2**
Bimonth 4	**0**	**0,2**	**7**	**0,4**	**7,6**
Bimonth 5	**3**	**0**	**5**	**0**	**8**
Bimonth 6	**8**	**1**	**5**	**1,2**	**15,2**
Total	**20**	**2,8**	**42**	**1,6**	**66,5**
Expected	**30 - 70**	**10 - 30**	**80 - 140**	**0 – 4.8**	

3. Research activity:

During the first year of my Ph.D I focused my research topics on Software Testing , following the business interests of Fervento srl, the company that funded my Ph.D fellowship. The main concept covered was techniques and methodologies for testing web applications, especially at the performance level. In fact, the study focused on the *automatic generation of realistic workloads for performance testing* in the web domain. In the web application domain, workloads are basically sequences of requests produced by a number of concurrent users, executing possibly different use cases, in a given period of time [1]. A crucial activity for a tester is to design an appropriate workload that reflects the scope in which the system will be deployed. After an exhaustive study of the state of art, most literature works aim at generating realistic workloads starting from real user behaviors, inferred by analyzing session logs [2-3]. These works present some drawbacks, limiting the early-stage detection of defects and the productivity of performance testers, indeed, the proposed works typically require the SUT to be actually deployed in order to collect real user behaviors and thus can be applied only after the first release. Lastly, most of these works not support novel protocols such as WebSocket, which is increasingly common in modern web application. After studying the state of the art, an approach was developed for automatically generating the workload of a web application using only the data generated from End-to-End test cases performed in the previous phase of the software life cycle. The intuition behind this approach is to exploit the overlap between end-to-end testing and performance testing. Both test the system from the user's perspective. I am currently working on extending the work by introducing a more refined, AI-based method to *automatically detect correlation between http requests*.

Another concept addressed during this year always concerning industrial performance aspect useful for my partner company. The initial development phase of a project is the most important one, especially the choice of technologies to be used has an impact on both the development time and the quality of the project itself.

With the proliferation of dedicated frameworks, developers and organizations face the challenge of selecting the most appropriate technologies for effectively developing high-performance REST APIs [4,5]. The decision-making scenario is further complicated by the possibility of running the applications using emerging and optimized execution environments, such as GraalVM. Some recent works have also compared different REST API frameworks written in different programming languages, evaluating their

performance under different workloads [4, 5]. These works, however, relied on simplistic REST API implementations, which can hardly be considered representative of the complexity of real-world applications, limiting their generalizability. To the best of my knowledge, no work in the literature has investigated the combined effect of different frameworks, written in different programming languages, and different execution environments in the context of REST APIs. As a first step towards filling this gap, I conducted a first study to investigate the combined performance impact on realistic REST APIs of two of the most popular frameworks (Express, Spring Boot) and different execution environments (Node, OpenJDK, GraalVM). I am currently working to extend this benchmark by implementing the industry topic application with other different frameworks written with other programming languages not previously considered, such as Django, FastAPI, etc.

As part of the main theme of software testing, I also worked on the *prioritization of regression tests and on the GUI testing of web applications*. Regarding the prioritization of regression tests, my and industrial partner's goal was to define a new approach of prioritizing GUI test cases to reduce the time required to run GUI tests, which typically take a long time to execute. The idea of our approach was related to source code changes between two successive versions of a software. During the study of the state of the art, the problem emerged that in the field of "web-GUI test prioritization" there is a lack of proposed approaches and datasets to evaluate them. The work then changed to building a dataset for evaluation of approaches to prioritization of web-gui tests. Hence, I'm looking open-source web applications on GitHub, which have:

- A consistent number of GUI test written in Cypress or Selenium.
- At least a consistent number of subsequent versions.

For each web app found, I deployed each version of it in a single container and performed the test of a version on the subsequent version in order to detect regression issues. The dataset will contain each web app the docker image of each version, the tests report and log and videos.

In the middle of the year I started a collaboration with Prof. Porfirio Tramontana and Prof.ssa Anna Rita Fasolino. The work concerns the *analysis of the quality of student projects*. In detail, after an intensive phase of project collection, I was able to collect 150 comparable projects in terms of complexity and technology from two different courses: Object Orientation a course in Computer Science major, and Software Engineering a course in Computer Engineering major.
Static code review with SonarQube was applied to each project to identify problems (code errors, bugs, and vulnerabilities). In addition, Architectural Software Testing with the ArchUnit library was applied to check for violations of the architectural design patterns covered in the lectures. The work aims to:

- Compare the software quality of students from different degree major and with different programming backgrounds
- Provide information on which course topics to give more attention and which new topics to introduce.
- Make available a dataset of Java project, and their quality assessment with SonarQube and ArchUnit.

Most of the literature research is limited to evaluating the software quality of non-degree students belonging to online programming courses or a specific degree program. In addition, they merely evaluate software quality by looking at metrics generated by static analysis tools while ignoring problems arising from project architecture [6,7].

Finally, after taking the Big Data Architecture and Analysis course, I am analyzing the state of the art of using LLM (Large Language Model) in the travel industry, specifically I am working on the implementation of a system for generating attractive hotel deplaints describing points of interest near the hotel. The system will be based on ChatGPT, georeferenced data provided by openstreetmap, and information provided by the wikyvoyage dataset

Research products:

- *E. Battista, S. D. Martino, **S. Di Meglio**, F. Scippacercola and L. L. L Starace, "E2E-Loader: A Framework to Support Performance Testing of Web Applications," 2023 IEEE Conference on Software Testing, Verification and Validation (ICST), Dublin, Ireland, 2023, pp. 351-361, doi: 10.1109/ICST57152.2023.00040.*
  status: published.

- ***S. Di Meglio**, L.L.L Starace, S.Di Martino "Starting a new REST API project? A performance benchmark of frameworks and execution environments" 2023, International Conference on Software Process and Product Measurement (MENSURA), Rome, Italy.*
  status: accepted and presented.

- ***S. Di Meglio,** M. De Luca, L.L. L Starace, S. Di Martino, P. Tramontana, A. Fasolino "Analysis of Software Project Quality: A Comparative Study of Computer Science and Computer Engineering Degree Courses",2023.*
  status: final stages of writing.

5. Conferences and seminars attended

- *17th International Conference on Software Process and Product Measurement (MENSURA), September 14-15 2023 in Rome, Italy. I attended this conference as presenting author for the paper "Starting a new REST API project? A performance benchmark of frameworks and execution environments"*

- *16th IEEE International Conference on Software Testing, Verification and Validation (ICST), April 16-20 2023 Dublin, Ireland. I attended this conference as author for the paper "E2E-Loader: A Framework to Support Performance Testing of Web Applications".*

6. Activity abroad:

None.

7. Tutorship

- Two two-hours lectures within the "*Software Project Management and Evolution*" M.Sc. course held by Prof. Sergio Di Martino. The lectures had the following topics: "*Performance Testing*" , *"Static code review and introduction to the use of SonarQube".*

- Co-supervised of thesis activities of a M.Sc. in Computer Science student working on web testing.

- Co-supervised of thesis activities of a B.Sc in Computer Science student working on web testing.

8. References

[1] K. Yorkston, Performance Testing Tasks. Berkeley, CA: Apress,2021, pp. 195–354. [Online]. Available: https://doi.org/10.1007/978-1-4842-7255-8 4

[2] C. Vogele, A. van Hoorn, E. Schulz, W. Hasselbring, and H. Krc- ¨mar, "Wessbas: extraction of probabilistic workload specifications for load testing and performance prediction—a model-driven approach for session-based application systems," Software & Systems Modeling, vol. 17, no. 2, pp. 443–477, 2018.

[3] Z. M. Jiang and A. E. Hassan, "A survey on load testing of large-scale software systems," IEEE Transactions on Software Engineering, vol. 41,no. 11, pp. 1091–1118, 2015.

[4] E. Kemer, R. Samli, Performance comparison of scalable rest application programming

interfaces in different platforms, Computer Standards & Interfaces 66 (2019) 103355.

[5] L. R. Abbade, M. A. da Cruz, J. J. Rodrigues, P. Lorenz, R. A. Rabelo, J. Al-Muhtadi, Performance comparison of programming languages for internet of things middleware, Transactions on Emerging Telecommunications Technologies 31 (2020) e3891.

[6] Arthur-Jozsef Molnar, Simona Motogna, and Cristina Vlad. 2020. Using static analysis tools to assist student project evaluation. EASEAI 2020 - Proceedings of the 2nd ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, Co-located with ESEC/FSE 2020 (2020), 7 – 12. https://doi.org/10.1145/3412453.3423195

[7] Julian Jansen, Ana Oprescu, and Magiel Bruntink. 2017. The impact of automated code quality feedback in programming education. CEUR Workshop Proceedings 2070 (2017).