COMPUTER AND CONTROL ENGINEERING

Vodafone/DAUIN - Bridging Human Expertise and Generative AI in Software Engineering

Funded By	Dipartimento DAUIN Vodafone Servizi e Tecnologie SrL [P.iva/CF:05288010969]
Supervisor	MORISIO MAURIZIO - maurizio.morisio@polito.it
Contact	COPPOLA RICCARDO - riccardo.coppola@polito.it
Context of the research activity	In collaboration with Vodafone Digital and the ZTO team in Network Operations, the PhD project aims to define a framework to generate code from functional requirements, fostering synergy between human developers and AI-based actors. The project will involve a systematization of metrics and methodologies to evaluate the correctness of the generated code and requirements performed by the generative AI components to increase the effectiveness and gain trust in the outputs of generative AI.
	Research objectives The main objectives of the PhD programme are the following:The identification of generative AI mechanisms that can aid in code generation from software requirements. The development and assessment of methods for the evaluation of the correctness and the dependability of the application of generative AI to code development;The conduction of formal experiments to evaluate how code generated by AI Compares to human-written code in both functional and non-functional terms. Outline of the research work plan Task 1: Preliminary evaluation of state-of-the-art solutions (M1-M3) The task involves a comprehensive assessment of current solutions in the domain. The objective is to evaluate existing methodologies, technologies, and frameworks relevant to the research context. This preliminary analysis will be conducted systematically by applying Kitchenham's guidelines for conducting Systematic Literature Reviews in the Software Engineering research field. The systematic literature review will also consider grey literature sources (i.e., non-peer-reviewed sources available on various internet sources) to cope with the high novelty of the generative AI research field. The systematic evaluation of the state of the art will be complemented with open and structured interviews with practitioners and developers to understand their main needs and most common practices.

Objectives	Task 2: Selection and Integration of Generative AIs for Code Generation (M4-M18) This task focuses on selecting, customizing, integrating, and training a Generative AI, specifically a Large Language or Foundation Model, to generate code from formal requirements. It includes understanding various use case and formal requirement languages and creating modules for translating natural language requirements into structured notations like Use Case Diagrams. The process involves preprocessing data -collecting, cleaning, and structuring use case language datasets- and training the AI to understand these scenarios. Ongoing evaluation and refinement of the AI are crucial for accuracy. The main goal is to develop a solution that translates use case specifications into high-quality code, with evaluations based on the development effort, error rates, and requirement alignment. The task will use Software Repository Mining (MSR) techniques for diverse dataset collection. The implementation phase of this research will follow the Agile Software Development practices, streamlining the entire software development lifecycle to assess the efficacy of AI-generated code against existing tools and manually written code for both front-end and back-end applications. Furthermore, the research is dedicated to pioneering methods for automatically generating synchronized documentation and unit testing. It will also investigate strategies for conducting code quality reviews, monitoring resource usage efficiently, and evaluating the software's business impact, thereby tailoring the development process to meet the demands of network operations. Task 3: Definition of assessment methods for Generative AI-based code development (M13-M24) The task focuses on defining robust methods to assess Generative AI-based code development. This entails the definition of structured procedures to assess the accuracy, reliability, and compliance with the requirements of the generated code. The goal is to establish a rigorous framework for ensuring the quality of code produced by Generative AI, thus advancing the state of the art in Generative AI code development. Task 3 will involve applying systematic techniques to build taxonomies in Software Engineering (ref. Paul Ralph) through the Straussian Grounded Theory technique (ref. Strauss). Task 4: Analysis of the non-functional implications of Generative AI-based code development (M22-36) The task focuses on a comprehensive analysis of the non-functional implications inherent to Generative AI-based code development. This includes scrutinizing factors such as scalability, performance, readability and maintainability of the generated code. The objective is to discern and mitigate any adverse effects of integrating Generative AI into the code development process. The task will involve conducting empirical experiments over statistically significant samples to compare non-functional properties of software generated by human developers, software obtained through Generative AI, and software obtained through a synergistic interaction between human developers and generative AI tools. List of possible venues for publications

	The target for the PhD research includes a set of conferences in the general area of software engineering (ICSE, ESEM, EASE, ASE, ICSME) as well as in the specific area of testing (ICST, ISSTA). More mature results will be published in software engineering journals, mainly IEEE Transactions on Software Engineering, ACM TOSEM, Empirical Software Engineering, Journal of Systems and Software, and Information and Software Technologies.

Skills and competencies for the development of the activity	The main skills required by the candidate are the following: General knowledge about Large Language Models and application of AI-based algorithms to software development; Experience in software development with object-oriented languages (e.g., Java or C#) and knowledge of the web and/or mobile application domain; Knowledge of software verification and validation techniques (e.g., scripted unit and integration testing, end-2-end testing, Graphical User Interface testing).