Marco De Luca

Enhancing Software Development Processes for Industrial Software Systems

Tutor: Prof. Anna Rita Fasolino co-Tutor: Pasquale Cimmino

Cycle: XXXVII Year: Third

Candidate’s information

- MSc degree in Computer Engineering, University of Naples “Federico II”
- Research group: REvERSE
- PhD start and end dates: 01/11/2021 – 31/10/24
- Scholarship type: company-funded
- Partner company: Micron Semiconductor Italia S.R.L

Summary of study activities

- **Ad hoc PhD courses / schools:**
  - Big Data Architecture and Analytics
  - Impreditorialità Accademica
  - Using Deep Learning Properly
  - Ethics and AI
  - ISSSE 2023 - 16th International Summer School on Software Engineering
- **Courses attended from MSc curricula:**
  - Software Testing
  - Machine Learning e Big Data per la Salute
  - Natural Language Processing
- **Attended Conferences:**
  - **Participation** at the *International Workshop on Quality and Measurement of Software Model-Driven Development* (QUAMES), 2022
  - **Presenting author** at the IEEE 20th *International Conference on Software Architecture Companion* (ICSA), 2023
  - **Presenting author** at the 18th *European Conference on Software Architecture* (ECSA), 2024

Research area(s)

- **Software Development and Documentation process in safety critical domain in compliance with ISO 26262:**
  - The automotive industry is transforming rapidly due to the growing integration of software in modern vehicles
  - Key challenges arise in the automotive software development process

- **Robustness of locators in Template-based Web Application Testing**
  - GUI-based test cases use locators to interact with web elements (e.g., buttons, text fields) for automated testing. Test cases from Capture and Replay (C&R) tools are "fragile" and break easily with minor layout changes, even if the web application's functionally remains unchanged
  - This "fragility" represents a significant problem in maintaining reliable automated testing for evolving web application

Research results

- **Software Development in the automotive domain:**
  - A **Community detection technique** and tool for identifying developer communities within developer social networks to facilitate team formation
  - A **Software Architectural documentation template** that complies with the requirements set by ISO 26262
  - A **tool-based Software Architecture Recovery (SAR)** technique to support documentation practices in industrial setting
  - A **framework to characterize architectural metrics in support of continuous compliance processes** in the industrial domain with respect to safety standards and guidelines

PhD Theis

- **GUI-based testing:**
  - Definition of “**Hook-Based**” **locators**, that exploit an HTML tag attribute, referred as “hook”, to allow the unique identification of each tag of a Web Page to reduced the “fragility” of the test cases.

Research products

[J1]	M. De Luca, A. R. Fasolino, A. Ferraro, V. Moscato, G. Sperlí, P. Tramontana, *A community detection approach based on network representation learning for repository mining,* **International Journal of Expert Systems with Applications,** vol. 231, 2023, DOI: 10.1016/J.ESWA.2023.120597
[J2]	M. De Luca, A. R. Fasolino, P. Tramontana, *Investigating the robustness of locators in template-based Web application testing using a GUI change classification model,* **International Journal of Systems and Software,** vol. 210, 2024, DOI: 10.1016/J.JSS.2023.111932

Research products

[C1]	P. Tramontana, M. De Luca, A. R. Fasolino, *An Approach for Model Based Testing of Augmented Reality Applications,* **RCIS Workshop,** Barcellona, Spain, 2022, Publisher, DOI: https://ceur-ws.org/Vol-3144/QUAMES-paper2.pdf
[C2]	D. Amalfitano, M. De Luca, A. R. Fasolino, *Documenting Software Architecture Design in Compliance with the ISO 26262: a Practical Experience in Industry,* **IEEE 20th International Conference on Software Architecture Companion (ICSA-C),** L'Aquila, Italy, 2023, pp. i-xi, IEEE, DOI: 10.1109/ICSA-C57050.2023.00022
[C3]	D. Amalfitano, M. De Luca, A. R. Fasolino, P. Pelliccione, T. Santilli, *Characterizing Software Architectural Metrics for Continuous Compliance in the Automotive Domain,* **IEEE 21st International Conference on Software Architecture (ICSA),** Hyderabad, India, 2024 pp. 182-193, IEEE Computer Society, DOI: 10.1109/ICSA59870.2024.00025
[C4]	M. De Luca, S. Di Meglio, A. R. Fasolino, L. L. L. Starace, P. Tramontana, *Automatic Assessment of Architectural Anti-patterns and Code Smells in Student Software Projects,* **28th International Conference on Evaluation and Assessment in Software Engineering (EASE '24),** Salerno, Italy, 2023, pp. 565-569, ACM, DOI: 10.1145/3661167.3661290
[C5]	D. Amalfitano, M. De Luca, D.F. De Angelis, A. R. Fasolino, *Automated Architecture Recovery for Embedded Software Systems: An Industrial Case Study,* **18th European Conference on Software Architecture (ECSA),** Luxembourg, Luxembourg, 2024, pp. 55-68, ACM, DOI: 10.1007/978-3-031-70797-1_4

PhD thesis overview – Context

- **Context**:
  - Automotive industry is undergoing a profound transformation, driven by the increasing *integration of software into modern vehicles*
  - From optimizing engine performance to enabling ADAS, software now plays a pivotal role in every aspect of vehicle design and functionality
  - This growing reliance on software demands more effective and efficient development processes
  - The automotive industry is adopting several standards to ensure that software systems are fault-tolerant and meet stringent safety and quality requirements
- **Problem**:
  - Multidisciplinary team development
  - Compliance with *functional safety* standard like ISO 26262
  - Lack of documentation practices in industries

PhD thesis overview - Objective

This thesis addresses the challenges arisen from the increasing integration of software in modern vehicles:

- **Multidisciplinary Team Formation**:
  - Improving processes for building teams that can effectively handle the complexity of automotive software development across multiple disciplines
- **Documentation Practices**:
  - Proposing a *software architecture documentation template* that complies with the requirements set by ISO 26262
  - Introducing *tool-based Software Architecture Recovery* (SAR) approach to automatically generate software architecture documentation from existing projects
- **Compliance with Functional Safety Standard:**
  - Definition of a *multidimensional framework* and *evaluation criteria* for characterizing and assessing software architecture metrics, to be used in the continuous compliance process with respect to safety standards or internal requirements

PhD thesis overview - Methodology

- The following methodologies were applied to enhance software development in the automotive domain:
  - Literature review
  - Experimental evaluation of state-of-the-art solution
  - Industrial Survey
    - (focus group, interview, questionnaire)
  - Industrial Case Study

PhD thesis overview - Contributions

- My PhD thesis presents the following contributions:
  1. A community detection technique and tool for identifying developer communities within developer social networks
  2. Development of a software architectural documentation metamodel that complies with the requirements set by ISO 26262
  3. A tool-based Software Architecture Recovery (SAR) technique to support documentation practices in industrial settings
  4. A framework to characterize architectural metrics in support of continuous compliance processes in the industrial domain with respect to safety standards and guidelines

Contribution 1:
Community Detection Technique

Motivation

- **Modern vehicles depend on integrated mechanical, electrical, and software systems, requiring collaboration among multidisciplinary teams**
  - Poor team dynamics can cause inefficiencies and project failure.
- **Developer Social Networks (DSNs) have recently emerged as an effective tool for the analysis of community in software projects and software ecosystems**:
  - Leveraging the information of this DSNs can support several interesting task like i) Community Detection, ii) Expert Finding, iii) Team Formation
- **Most approaches proposed in the literature employ *homogeneous* graphs that are characterize by single entities as nodes** (i.e. developers) **and a single type of edge** (i.e. commit relationship)

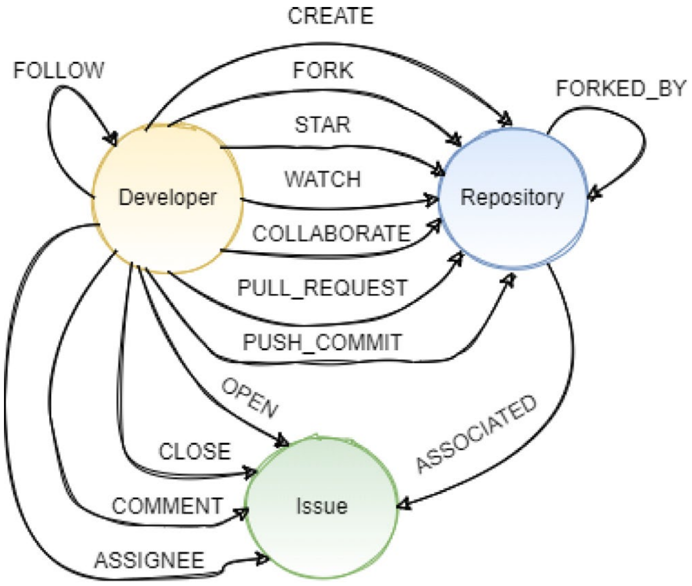
Activities

- Study of the literatures on the use of *homogeneous* and *heterogeneous* graphs for the representation of DSNs

- Definition of our representation model for the information extracted from software repositories

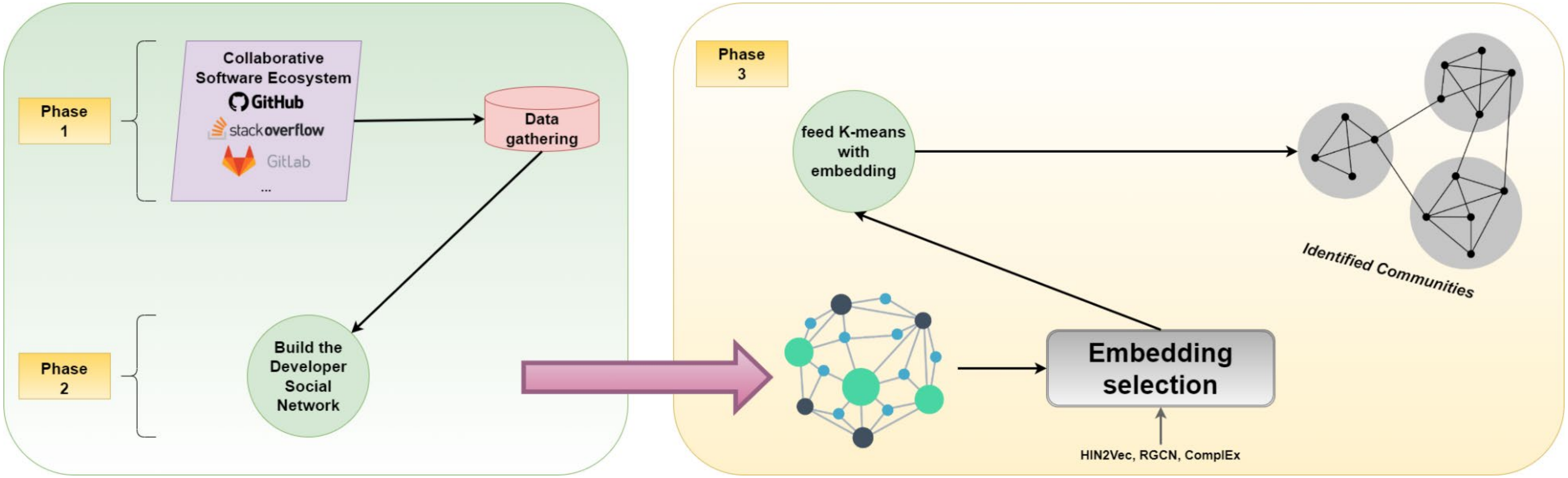
- Definition of a novel *heterogeneous* graph-based approach able to capture and handle in effective way all the complex and strongly-correlated information inside a software repository
  - Heterogeneous graph can have different type of nodes and edges
  - They have a more complex topological structure which make them more information rich

- Evaluation of the proposed heterogeneous graph-based approach with respect of state-of-the-art homogeneous graph-based approach



Proposed Solution

- Our *community detection algorithm* (**BERTO**) is based *on heterogeneous* graph and leverages *graph-embedding* and *K-means*



BERTO overview

Results

Comparison of the average modularity

Modularity comparison						
K	emb_50		emb_150		emb_300	
	BERTO	*ABDCI [1]*	*BERTO*	*ABDCI [1]*	*BERTO*	*ABDCI [1]*
50	77.75%	68.35%	82.64%	86.34%	78.57%	82.31%
150	67.44%	35.38%	77.19%	80.74%	72.49%	76.90%
250	55.18%	25.27%	68.23%	73.49%	64.55%	69.86%

Comparison of the Median and Standard Deviation (STD) of community members

Median and STD Comparison												
K	emb_50				emb_150				emb_300			
	BERTO		*ABDCI [1]*		*BERTO*		*ABDCI [1]*		*BERTO*		*ABDCI [1]*	
	Median	STD	Median	STD	Median	STD	Median	STD	Median	STD	Median	STD
50	44.12	26.08	38.10	130.79	58.08	35.41	54.04	50.48	53.03	50.99	49.60	57.93
150	14.03	73.97	16.10	24.73	18.10	12.96	19.00	15.98	19.09	14.08	18.07	17.39
250	6.53	16.78	10.02	12.57	10.01	9.37	10.00	11.36	11.03	7.50	11.04	8.42

- While our model does not outperform the model proposed in [1] in terms of modularity, focusing on standard deviation reveals a key advantage
- Our model show smaller STD compared to ABDCI [1]
- This indicates that BERTO is able to find a more balanced community division

[1] T Hou, X. Yao, D. Gong, Community detection in software ecosystem by comprehensively evaluating developer cooperation intensity, Information and Software Technology, 2021

Contribution 2:
Software architecture documentation metamodel compliant with ISO 26262 requirements

Motivation

- **Documenting the Software Architecture Design (SAD) is a challenging activity in industries for safety critical software**
  - This is amplified when the software development must comply with the guidelines of the ISO 26262
- **Several work in literature have described the challenges encountered while developing the SAD in compliance with ISO 26262**
  - Even if all these works provide valuable indication and suggestion for implementing ISO 26262 compliant SAD, it is not always easy to translate these guidelines into practical solution
- **Industry need well-defined and concrete procedures, artifact and approaches to support the guidelines defined by ISO 26262 in a more practical manner**

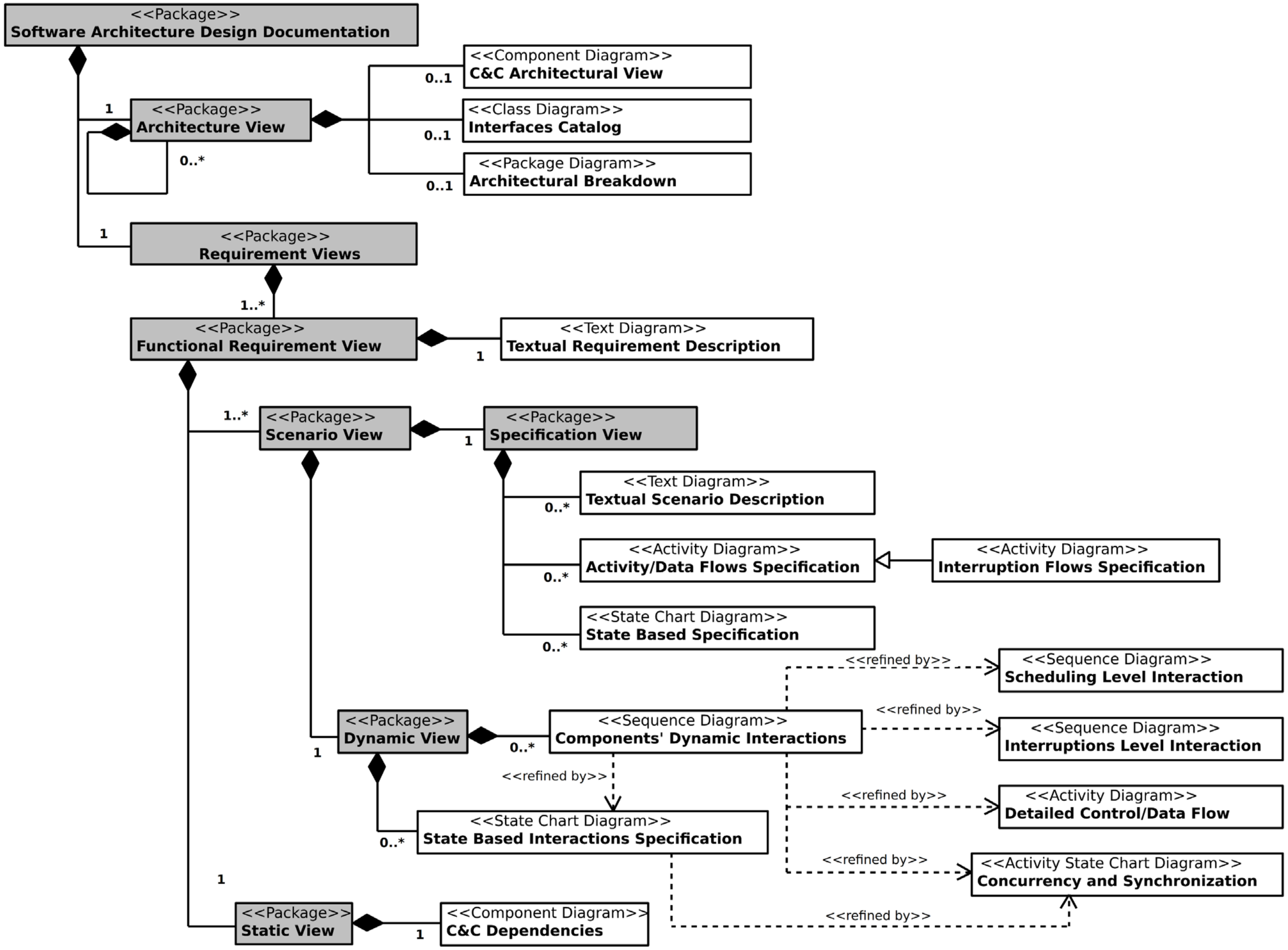
Activities

- **Industrial Survey** to collect **challenges** from automotive safety experts on developing SAD in compliance with ISO 26262
- Based on the survey results, we defined a **Software Architectural Documentation Template** intended to overcome the challenges emerged from the survey
- **Validation** of the proposed template with an **industrial case study** that involved the same experts we enrolled in the survey

Survey Results

ID	ISO REFERENCE	CHALLENGE	DESCRIPTION
C1	Properties, 7.4.1.b: consistency	**Consistency management**	Difficulties in ensuring the consistency between the different artifact produced during the documentation process. One of the most common problem reported by interviewees is for example the difficulties in ensuring consistency between static views and behavior views: occurrence of the same component in different views, but with different names
C2	Properties, 7.4.1.d: verifiability	**Verification of the design principles recommended by the ISO-26262**	Difficulty in finding methods for checking that the SAD adheres to the characteristics defined by the standard (in 7.4.3 Table 3) such as: modularity, maintainability and consistency
C3	Properties 7.4.1.f: abstraction	**Document the SAD with a hierarchical structure**	Abstraction can be supported by using hierarchical structures, grouping schemes or views to cover static, dynamic or deployment aspects of an architectural design
C6	Principle, 7.4.3-3: restricted size of interfaces	**Verifiability of the principle**	Difficulties in documenting software component interfaces to help the verification of the principle
C7	Principle, 7.4.3-4,5: strong cohesion and loose coupling	**Verifiability of the principle**	Difficulties in documenting both static and dynamic relationships among software components for aiding the evaluation of cohesion and coupling metrics to support the adherence to the principle

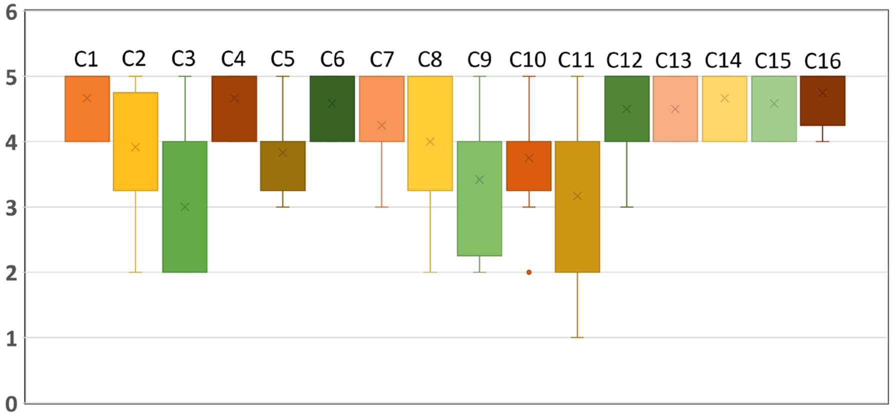
Proposed Solution



Results

- The results were gathered through an **industrial case study** involving the same three experts from four automotive software companies that participated in the initial survey
- The study aimed to answer the following research questions:
  1. *"To what extent is the documentation template accepted by the practitioners to produce SAD compliant with the ISO 26262?"*
  2. *"To what extent does the proposed documentation template overcome the identified safety related challenges?"*

- **Improvements of the proposed documentation w.t.r. the previous one:**
  - Enhanced automated consistency management
  - Interrelated module views of the SAD at different levels of details
- **Simplification in the verifiability of the principles set by ISO 26262:**
  - Stereotype to distinguish between safety and non-safety component
  - Generation of the .xml files for evaluation of complexity metrics
- **Main limitation pointed out:**
  - Lack of traceability link to the requirements that are handled in external tool
  - Lack of mechanism for supporting the multiuser collaborative work



Respondents rated the support of the Documentation Template in overcoming the identified challenges on a 5-point Likert scale

Contribution 3:
A tool-based Software Architecture Recovery (SAR) technique

Motivation

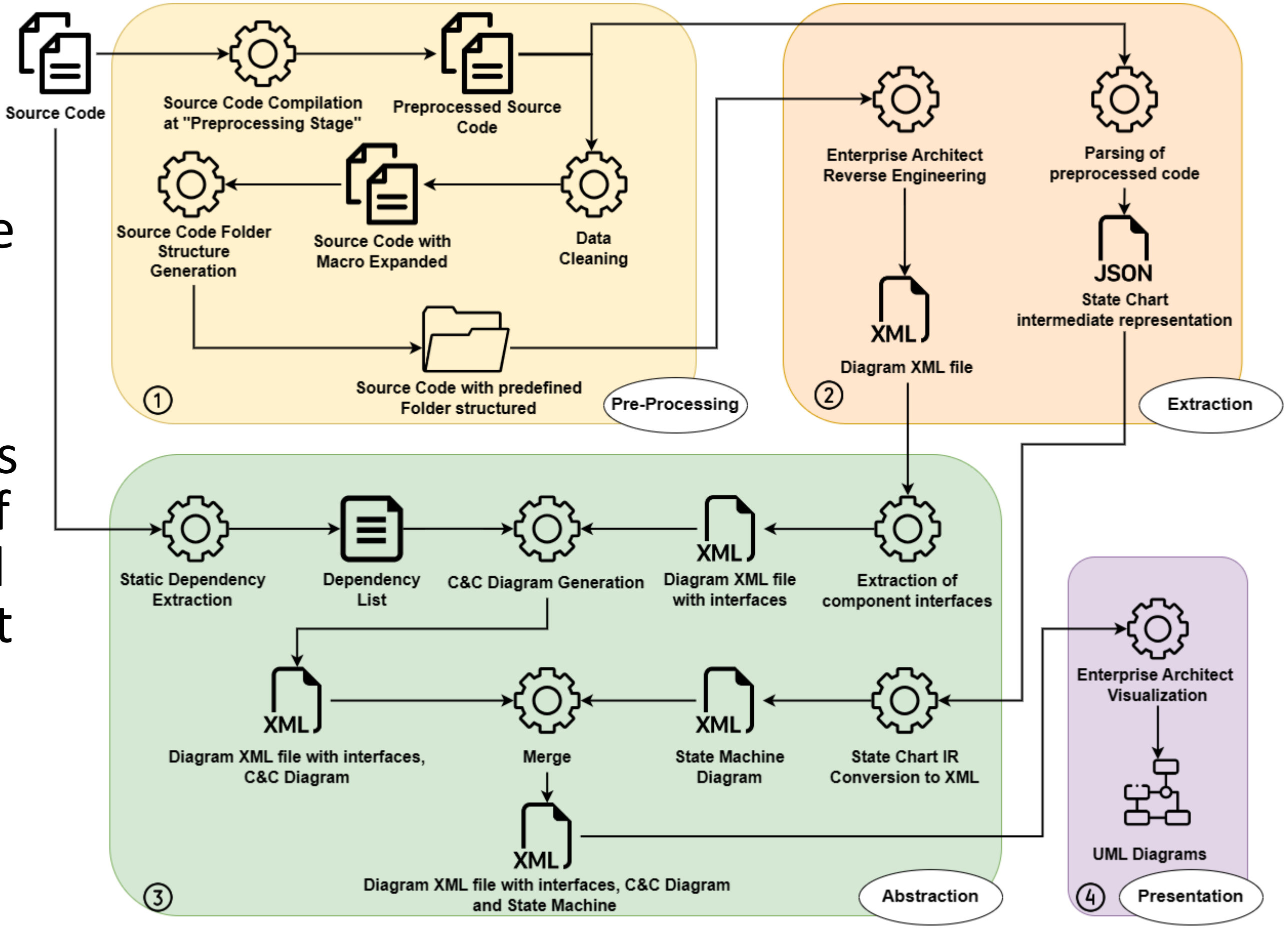
- **Maintaining the documentation accurate, corresponding to the contemporary state of the code is a challenging activity**
  - This is amplified in industrial settings where code rapidly evolves throughout frequent development iterations
- **Several work in literature have introduced reverse engineering techniques for recovering software documentation from traditional software**
  - Only few works focuses on embedded systems. These system are often implemented in low-level coding language posing unique comprehension challenges.
- **Industry usually lack architectural documentation**
  - Due to time and budget constraint

Activities

- Proposal of a SAR approach based on a previously proposed documentation template to reconstruct architecture views that comply with the ISO 26262
- Implementation of the proposed SAR approach in a tool
- Application of the approach in an **industrial case study** in collaboration with Micron
- Validation through an **industrial survey** to assess the effectiveness and accuracy of the proposed tool-supported SAR process

Proposed Solution

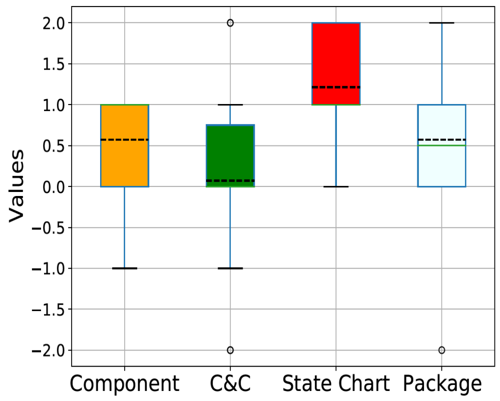
- A tool-supported Software Architecture Recovery (SAR) process to automatically generate UML models from static analysis of C code for embedded system and compliant with the proposed documentation template



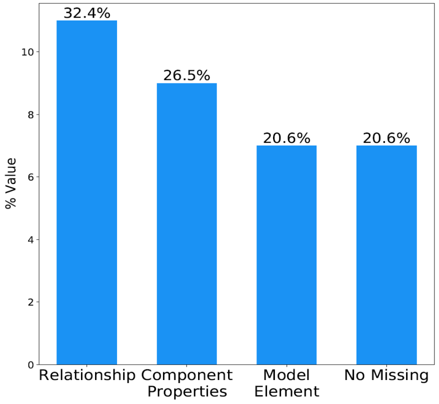
Source Code
Source Code Compilation at "Preprocessing Stage"
Preprocessed Source Code
Source Code Folder Structure Generation
Source Code with Macro Expanded
Data Cleaning
Source Code with predefined Folder structured
1
Pre-Processing
Enterprise Architect Reverse Engineering
Parsing of preprocessed code
XML
Diagram XML file
JSON
State Chart intermediate representation
2
Extraction
Static Dependency Extraction
Dependency List
C&C Diagram Generation
XML
Diagram XML file with interfaces
Extraction of component interfaces
XML
Diagram XML file with interfaces, C&C Diagram
Merge
XML
State Machine Diagram
State Chart IR Conversion to XML
XML
Diagram XML file with interfaces, C&C Diagram and State Machine
3
Abstraction
Enterprise Architect Visualization
UML Diagrams
4
Presentation

Results

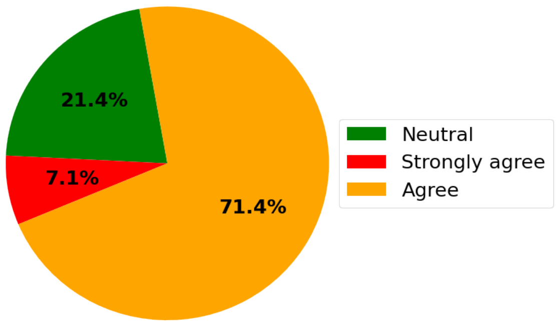
- The results were gathered through an **industrial survey** involving 14 software engineering from Micron
- The questionnaire aimed to answer the following research questions:
  1. *"To what extent the recovered software architecture documentation is considered accurate by the practitioners?"*
  2. *"If important information is missing, what kind of information is it?"*
  3. *"To what extent have the practitioners found the proposed software architecture recovery (SAR) tool useful?"*



Respondents rated the accuracy of each automatically reconstructed diagram on a 5-point Likert scale, comparing it to the previously manually generated diagram



The chart highlights the areas where our tool needs improvement



Pie chart on the closed-ended question: "Based on my experience in using the tool, I find it useful in supporting the comprehension of the system."

Contribution 4:

Framework to characterize architectural metrics in support of continuous compliance processes in the industrial domain

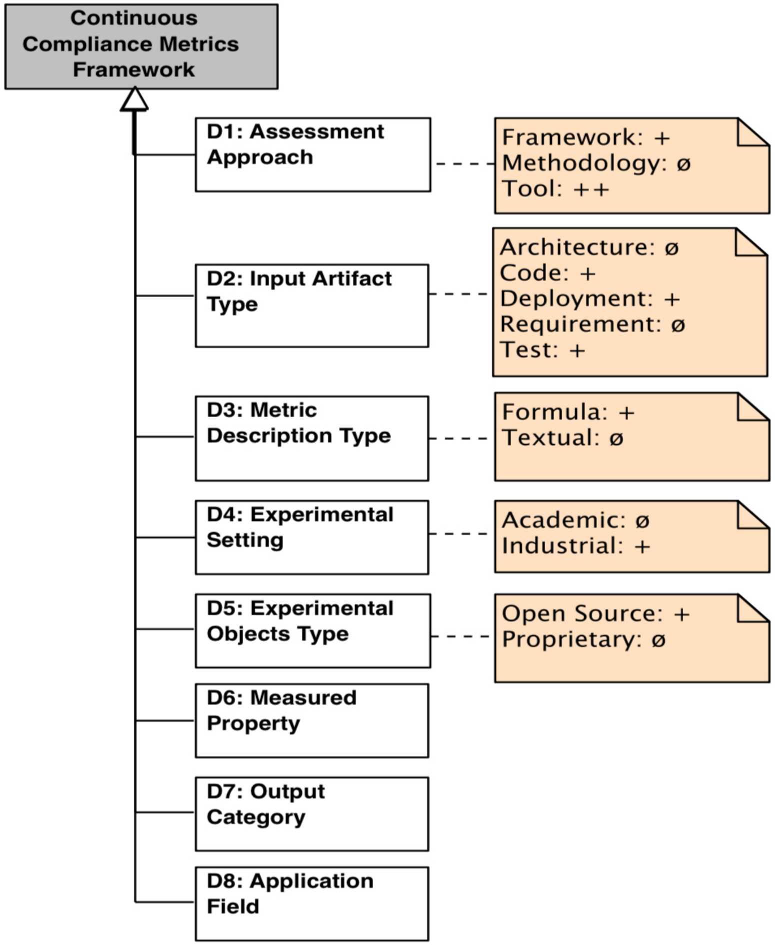
Motivation

- The most advanced original equipment manufacturers (OEMs) often define themselves as software companies
- The possibility of performing over-the-air (OTA) updates after production and when the systems are already used in the operations environment poses questions on the software certification process
- Industry needs **approaches** and **techniques** to ensure compliance with standards throughout the entire system life cycle on a continuous way
- Industries need to ensure compliance with standards in iterative development environments and for new releases of software products

Activities

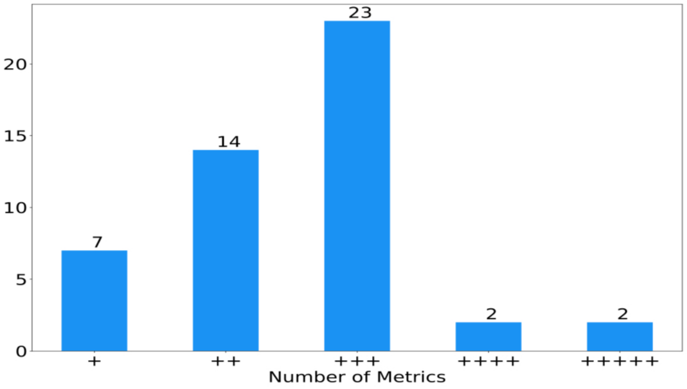
- Literature Review
  - to gather architectural metrics
- Semi-structured interview with safety and security expert to:
  - Identify the key characteristic of a metric for the Continuous Compliance to define the framework's dimension
  - Defining the set of values that can be assigned to each dimension of the framework
  - Define an evaluation criteria, based on the above-defined dimension, for assessing the suitability of a metric to be used in the context of Continuous Compliance

Proposed Solution

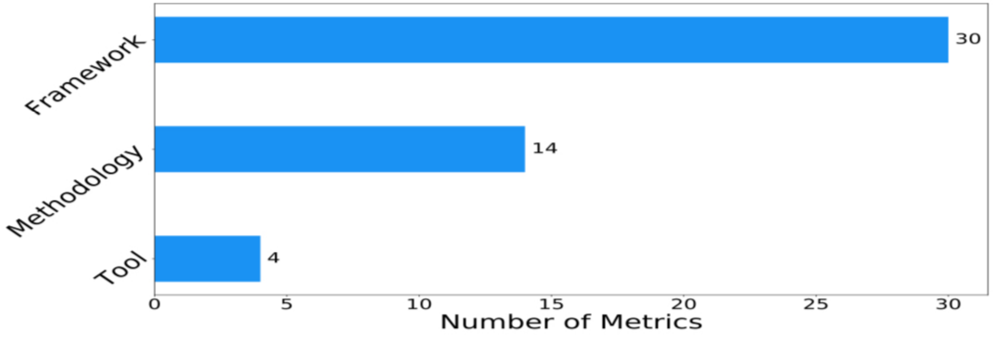


Results

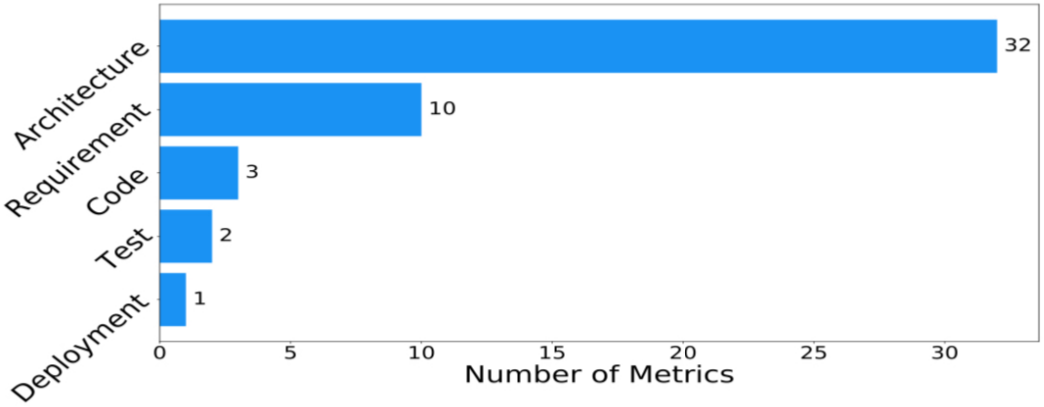
Continuous Compliance Score



Assessment Approach



Input Artifact



Conclusion

- This thesis addressed key challenges related to the increasing integration of software in modern vehicles
- The primary goal was to improve the software development process for automotive industries needing to comply with the functional safety standard ISO 26262
- To achieve this, four main contributions were proposed, focused on enhancing:
  - Team formation
  - Documentation Practices
  - Compliance with safety standard

Thank you for the attention!

Contact:
marco.deluca2@unina.it