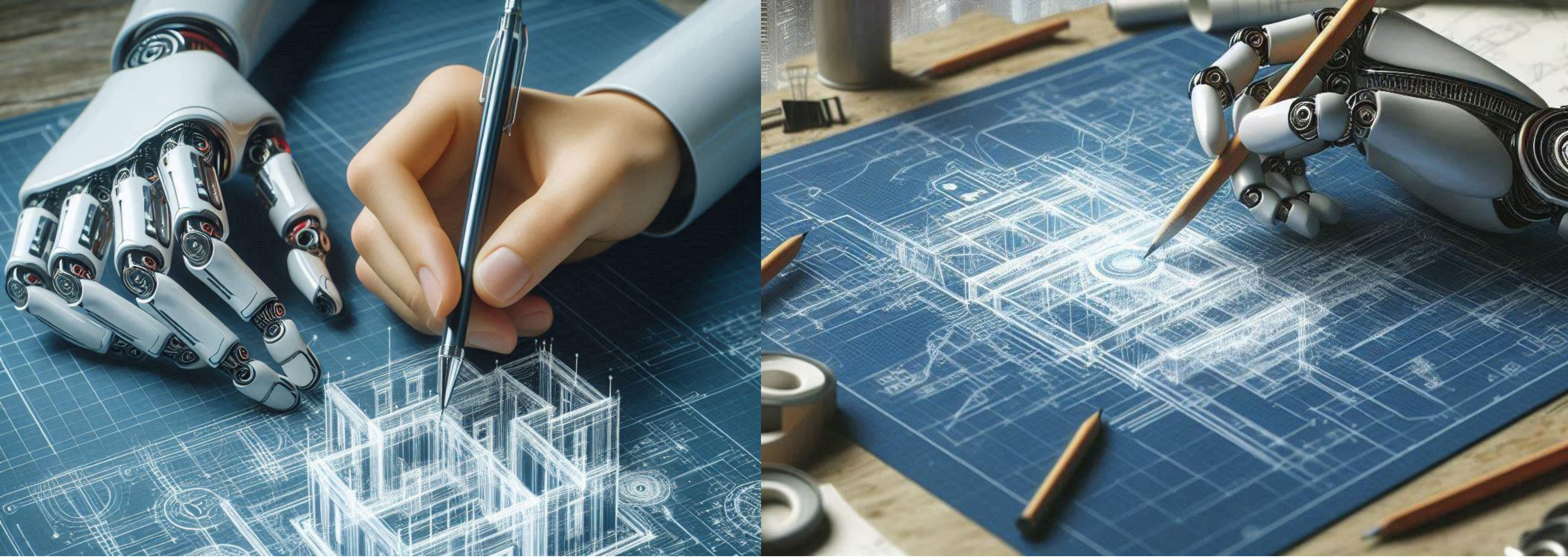


LLMs for Engineering and Architecting Software:

What Works, What's Missing, and What's Next?

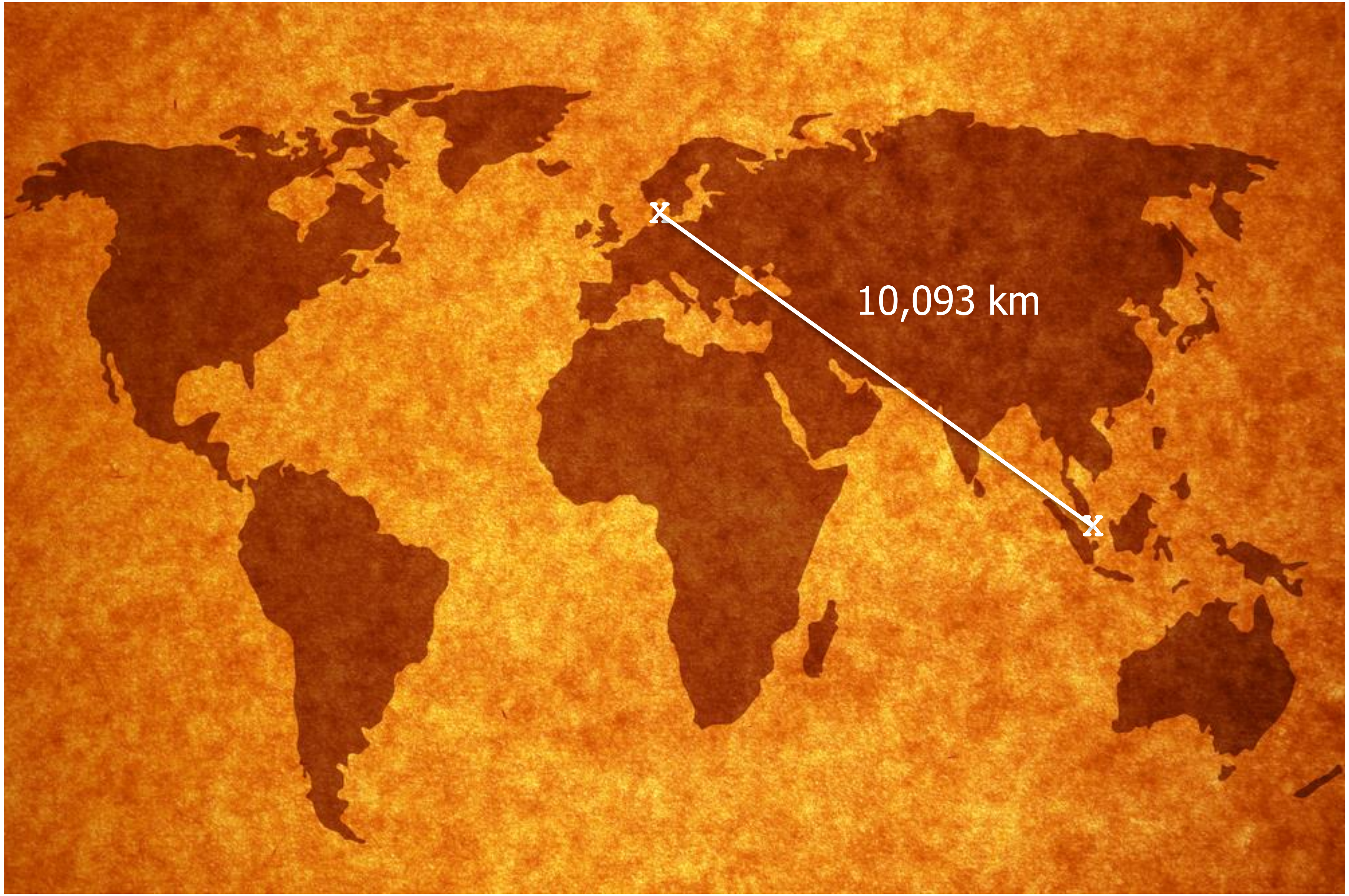
David Lo, FACM, FIEEE

ICSA 2025, Odense, Denmark, April 2025

SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and
Information Systems

Self-Introduction



Self-Introduction

0 5 10 km
0 5 10 mi
103°50'
104°00'
MALAY.
MALAYSIA
Kuala Johor
1°30'
Johore Strait
Yishun New Town
Pulau Ubin
Pulau Tekong
Ang Mo Kio New Town
Bukit Timah
Singapore Changi Airport
1°20'
Tuas
Bedok
Queens-town
SINGAPORE
Jurong Island
Sentosa
Singapore Strait
1°10'
Main Strait
INDONESIA
103°40'
Pulau Batam

Self-Introduction

Self-Introduction

Singapore Management University

- Third university in Singapore
- Number of students:
  - 8000+ (UG)
  - 1800+ (PG)
- Schools:
  - Business
  - Computing
  - Economics
  - Accountancy
  - Law
  - Social Science

CELEBRATING
MEANINGFUL
IMPACT
25
YEARS
AND BEYOND

Center for Research on Intelligent Software Engineering (RISE)

Elsevier JSS’21, Bibliometric Study

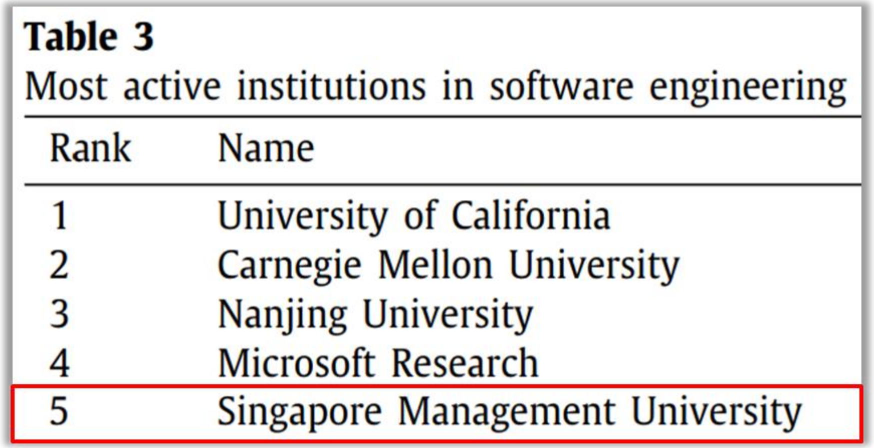
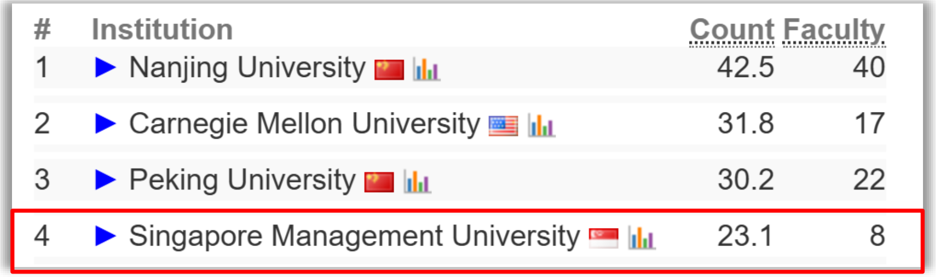


Table 3
Most active institutions in software engineering

Rank	Name
1	University of California
2	Carnegie Mellon University
3	Nanjing University
4	Microsoft Research
5	Singapore Management University

CSRankings, SE, Feb 2025



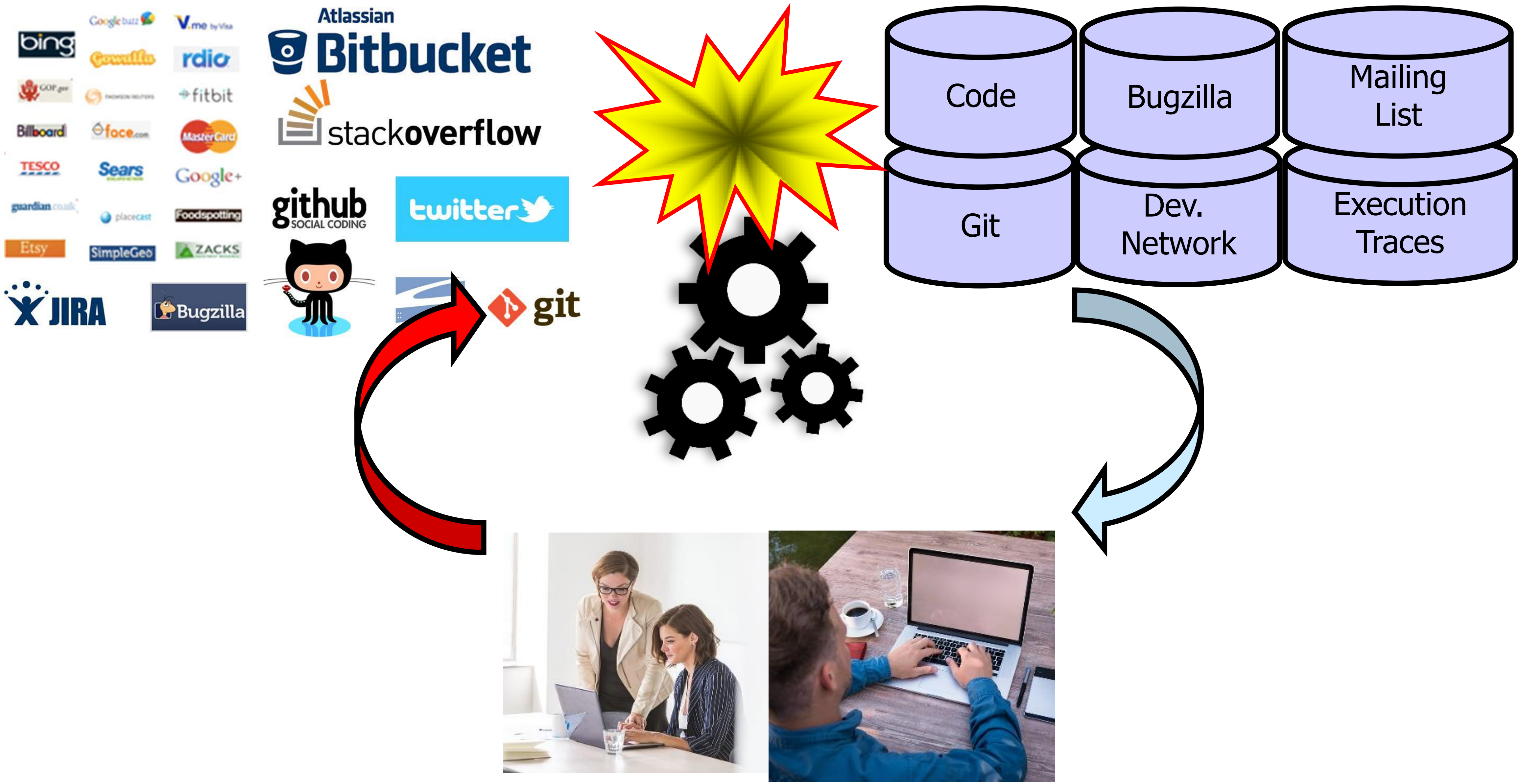
#	Institution	Count	Faculty
1	▶ Nanjing University	42.5	40
2	▶ Carnegie Mellon University	31.8	17
3	▶ Peking University	30.2	22
4	▶ Singapore Management University	23.1	8

SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and
Information Systems

Centre for Research on
Intelligent Software
Engineering

AI for Software Engineering



Code

Bugzilla

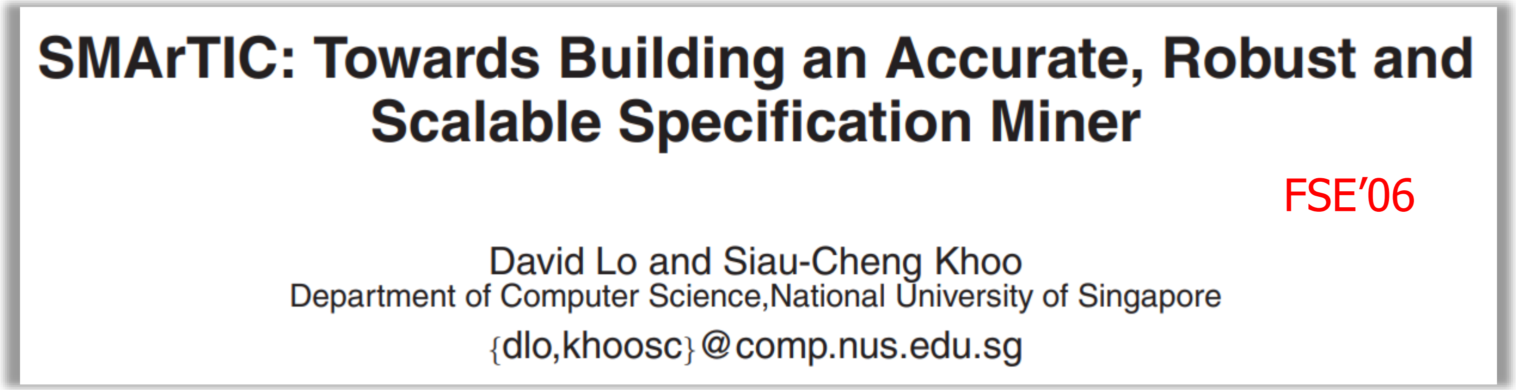
Mailing List

Git

Dev. Network

Execution Traces

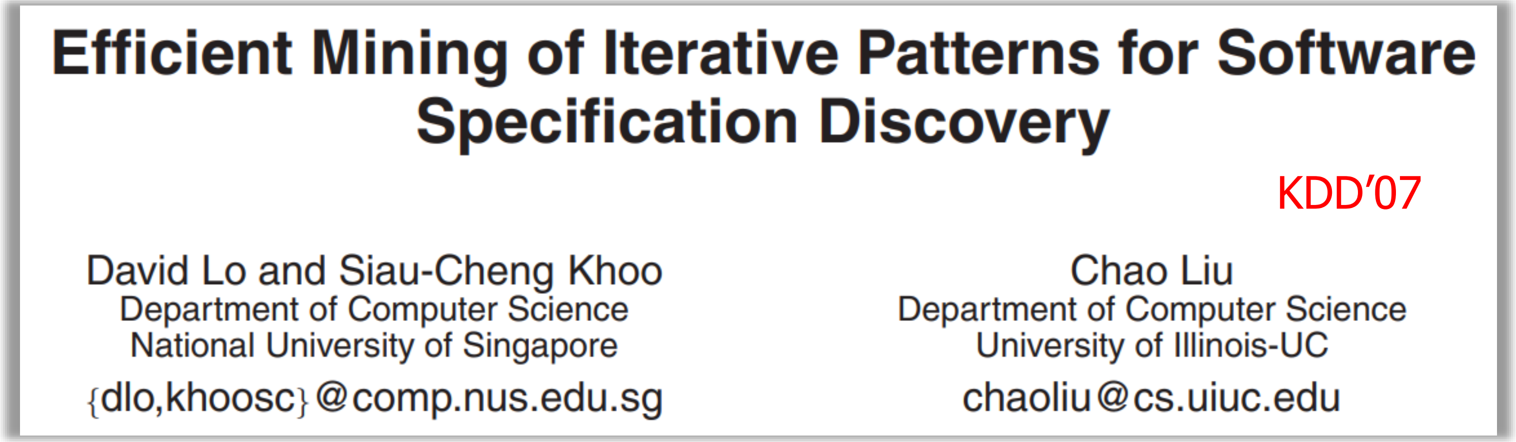
Experience with AI4SE



SMArTIC: Towards Building an Accurate, Robust and Scalable Specification Miner

FSE’06

David Lo and Siau-Cheng Khoo
Department of Computer Science,National University of Singapore
{dlo,khoosc}@comp.nus.edu.sg

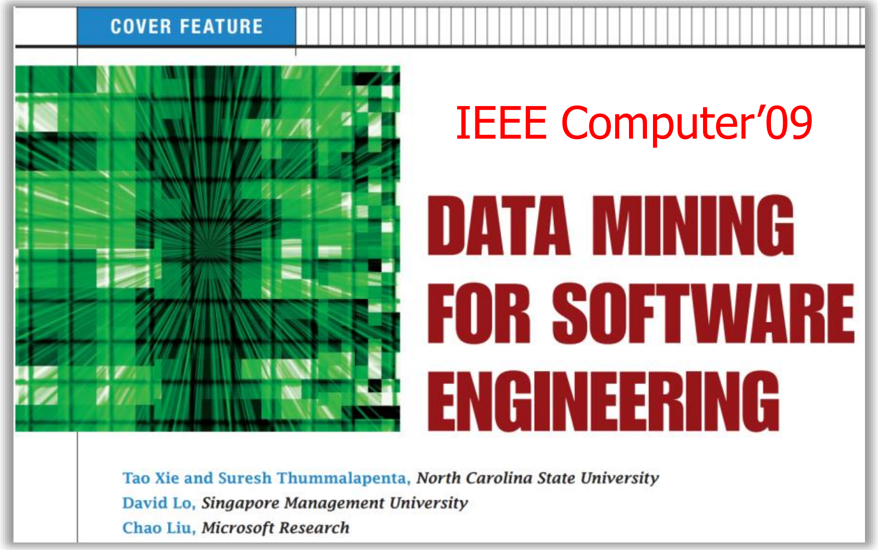


Efficient Mining of Iterative Patterns for Software Specification Discovery

KDD’07

David Lo and Siau-Cheng Khoo
Department of Computer Science
National University of Singapore
{dlo,khoosc}@comp.nus.edu.sg

Chao Liu
Department of Computer Science
University of Illinois-UC
chaoliu@cs.uiuc.edu



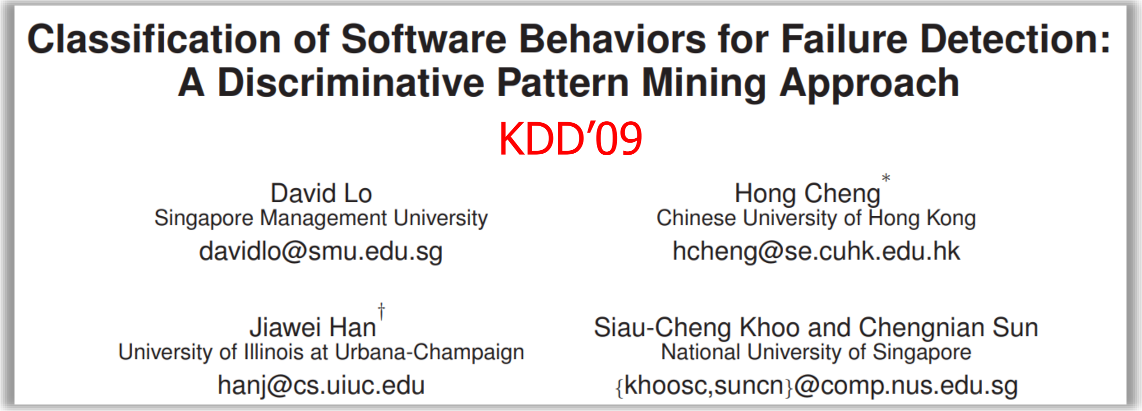
COVER FEATURE

IEEE Computer’09

DATA MINING FOR SOFTWARE ENGINEERING

Tao Xie and Suresh Thummalapenta, *North Carolina State University*
David Lo, *Singapore Management University*
Chao Liu, *Microsoft Research*

Experience with AI4SE



Classification of Software Behaviors for Failure Detection: A Discriminative Pattern Mining Approach

KDD'09

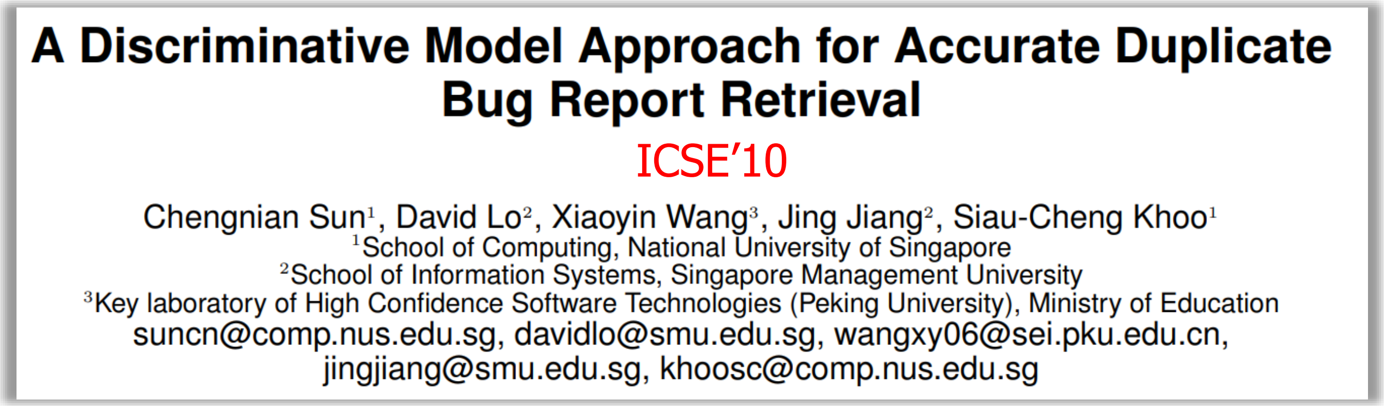
David Lo
Singapore Management University
davidlo@smu.edu.sg

Hong Cheng*
Chinese University of Hong Kong
hcheng@se.cuhk.edu.hk

Jiawei Han†
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Siau-Cheng Khoo and Chengnian Sun
National University of Singapore
{khoosc,suncn}@comp.nus.edu.sg

Test oracle generation

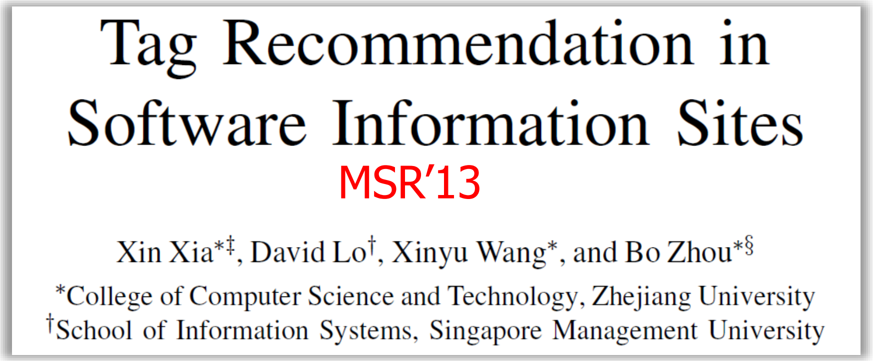


A Discriminative Model Approach for Accurate Duplicate Bug Report Retrieval

ICSE'10

Chengnian Sun[1], David Lo[2], Xiaoyin Wang[3], Jing Jiang[2], Siau-Cheng Khoo[1]
[1]School of Computing, National University of Singapore
[2]School of Information Systems, Singapore Management University
[3]Key laboratory of High Confidence Software Technologies (Peking University), Ministry of Education
suncn@comp.nus.edu.sg, davidlo@smu.edu.sg, wangxy06@sei.pku.edu.cn,
jingjiang@smu.edu.sg, khoosc@comp.nus.edu.sg

Intelligent issue trackers

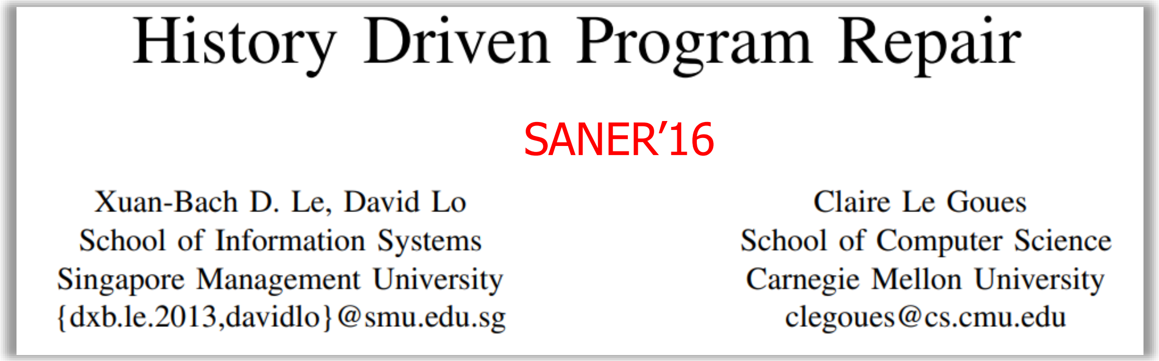


Tag Recommendation in Software Information Sites

MSR'13

Xin Xia*‡, David Lo†, Xinyu Wang*, and Bo Zhou*§
*College of Computer Science and Technology, Zhejiang University
†School of Information Systems, Singapore Management University

Intelligent crowdsourced SE



History Driven Program Repair

SANER'16

Xuan-Bach D. Le, David Lo
School of Information Systems
Singapore Management University
{dxb.le.2013,davidlo}@smu.edu.sg

Claire Le Goues
School of Computer Science
Carnegie Mellon University
clegoues@cs.cmu.edu

Intelligent program repair

"History-driven program repair influence our work, the overall pipeline is similar"

- Facebook Engineers

“If you want to go far, go together” – African Proverb

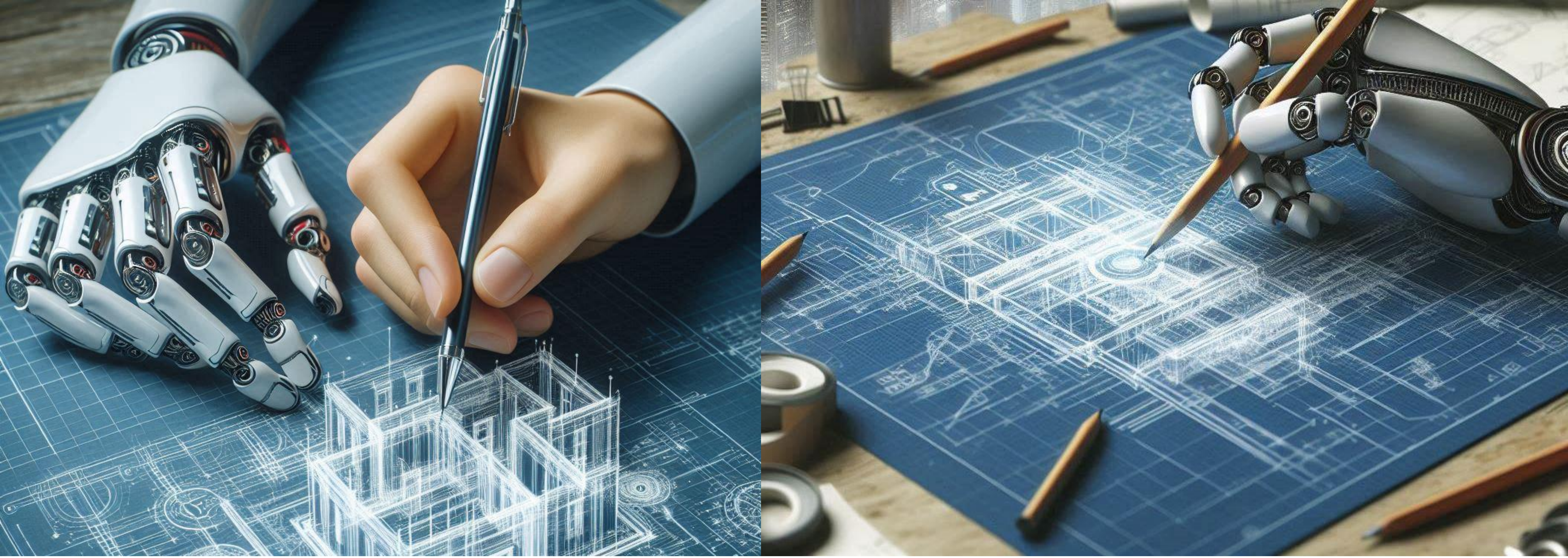
UNIVERSITY OF COPENHAGEN
Mälardalen University
HØGSKOLEN I OSLO OG AKERSHUS
UNIVERSITY OF OULU
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY
Inria
uni.lu UNIVERSITÉ DU LUXEMBOURG
UNIVERSITÀ DEGLI STUDI DI SALERNO

ROYAL HOLLOWAY UNIVERSITY OF LONDON
HUMBOLDT-UNIVERSITÄT ZU BERLIN
Carnegie Mellon University
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
UCSB
UTD
NC STATE UNIVERSITY
THE UNIVERSITY OF CALIFORNIA DAVIS

THE UNIVERSITY OF CALIFORNIA UCLA
THE UNIVERSITY OF CALIFORNIA IRVINE
UNIVERSITY OF HOUSTON FOUNDED 1927
Queen's UNIVERSITY
UBC
UNIVERSITÉ Concordia UNIVERSITY
University of Victoria
UnB
ZHEJIANG UNIVERSITY 1897
PEKING UNIVERSITY 1898
NANJING UNIVERSITY 1902

香港中文大學
The Chinese University of Hong Kong
SUN YAT-SEN UNIVERSITY
CHONGQING UNIVERSITY 1929
THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學
MONASH
Australian National University
THE UNIVERSITY OF MELBOURNE
UNIVERSITEIT STELLENBOSCH UNIVERSITY

NASA
GOVTECH SINGAPORE
Microsoft Research
SAP
Adobe
VERACODE
You change the world, we'll secure it.
Alibaba
Tencent



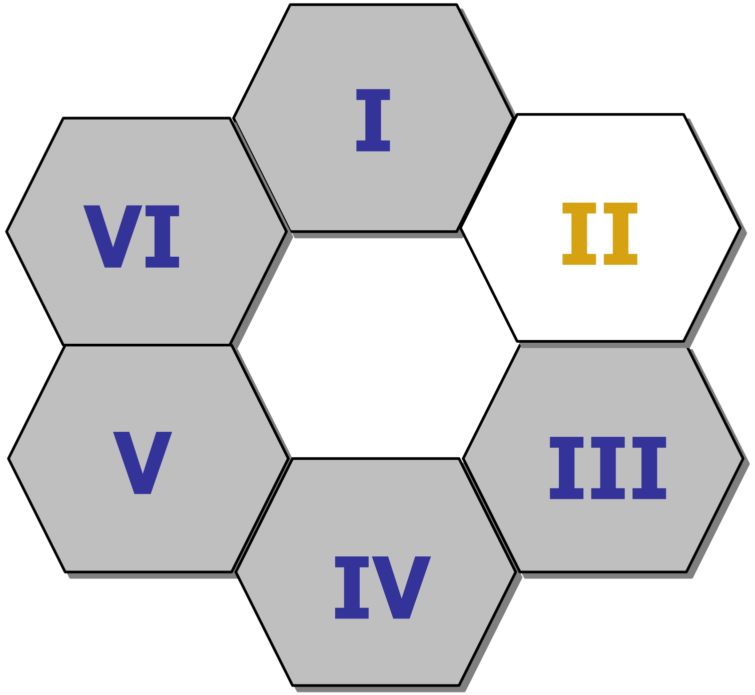
LLMs for Engineering and Architecting Software:

What Works, What's Missing, and What's Next?

SMU
SINGAPORE MANAGEMENT
UNIVERSITY

School of
Computing and
Information Systems

Talk Structure



Self-Intro.

SE 2.0

Software Architecture
(SA) Challenges

LLM4SA
Existing Work

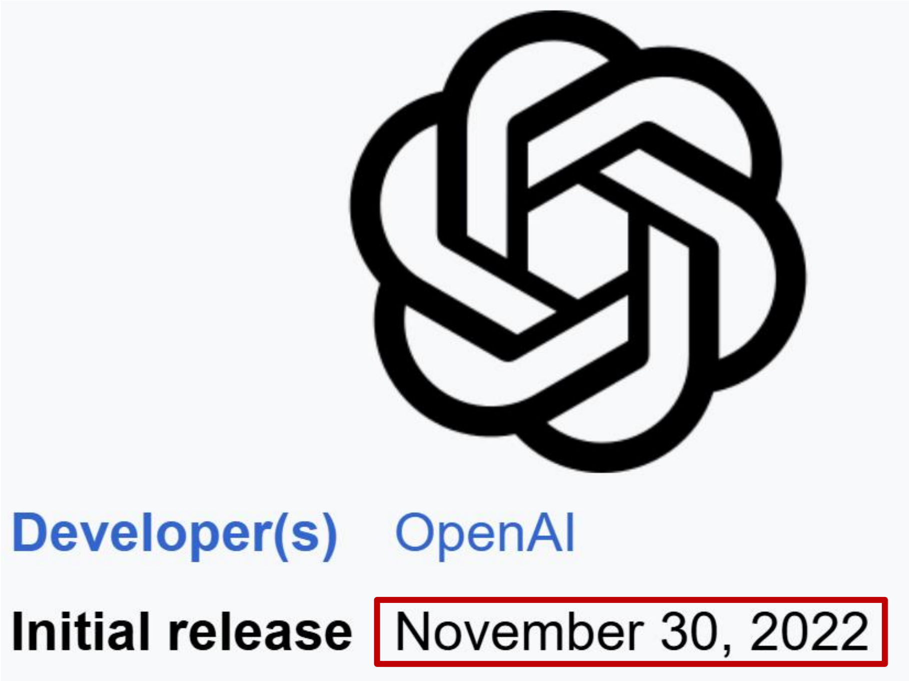
LLM4SA
Open Challenges

Realizing SE 2.0 &
Implications to LLM4SA

Software Engineering 2.0

ICSE2023

Special *Future Of Software Engineering (FOSE)* Track



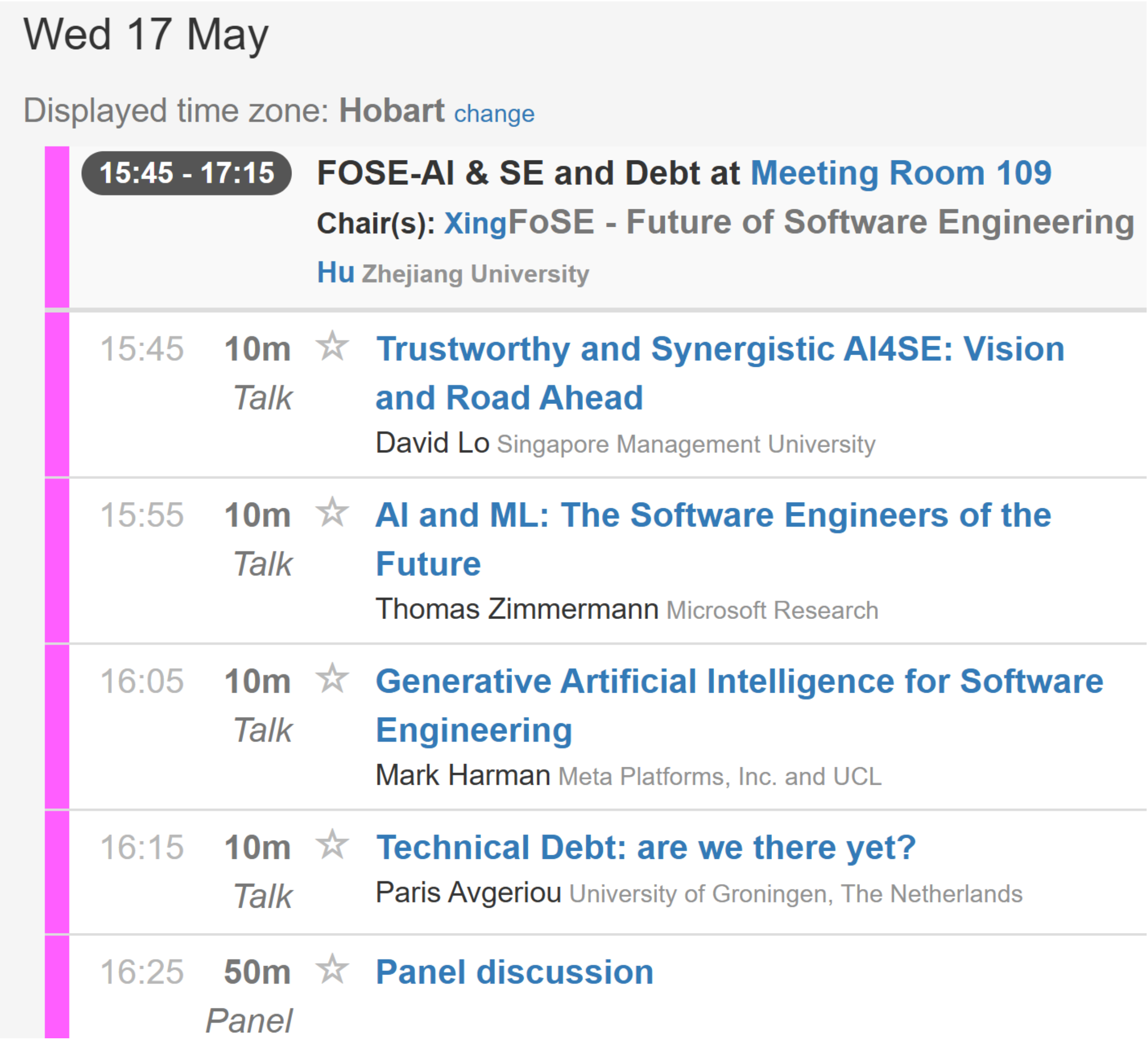
Developer(s) OpenAI

Initial release November 30, 2022

Software Engineering 2.0

ICSE2023

Special *Future Of Software Engineering (FOSE)* Track



Wed 17 May

Displayed time zone: **Hobart** change

15:45 - 17:15 **FOSE-AI & SE and Debt at Meeting Room 109**
Chair(s): Xing FoSE - Future of Software Engineering
Hu Zhejiang University

15:45 **10m** *Talk* ☆ **Trustworthy and Synergistic AI4SE: Vision and Road Ahead**
David Lo Singapore Management University

15:55 **10m** *Talk* ☆ **AI and ML: The Software Engineers of the Future**
Thomas Zimmermann Microsoft Research

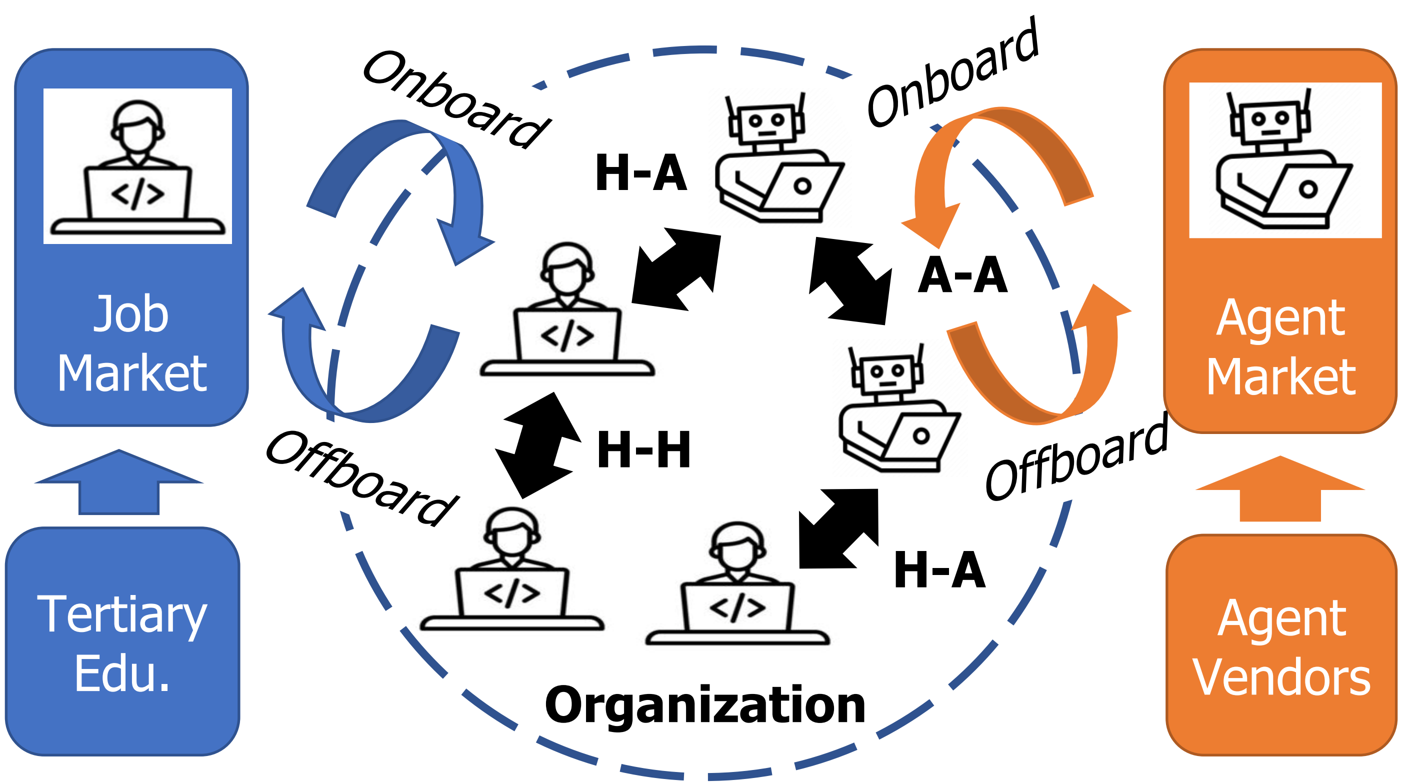
16:05 **10m** *Talk* ☆ **Generative Artificial Intelligence for Software Engineering**
Mark Harman Meta Platforms, Inc. and UCL

16:15 **10m** *Talk* ☆ **Technical Debt: are we there yet?**
Paris Avgeriou University of Groningen, The Netherlands

16:25 **50m** *Panel* ☆ **Panel discussion**

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

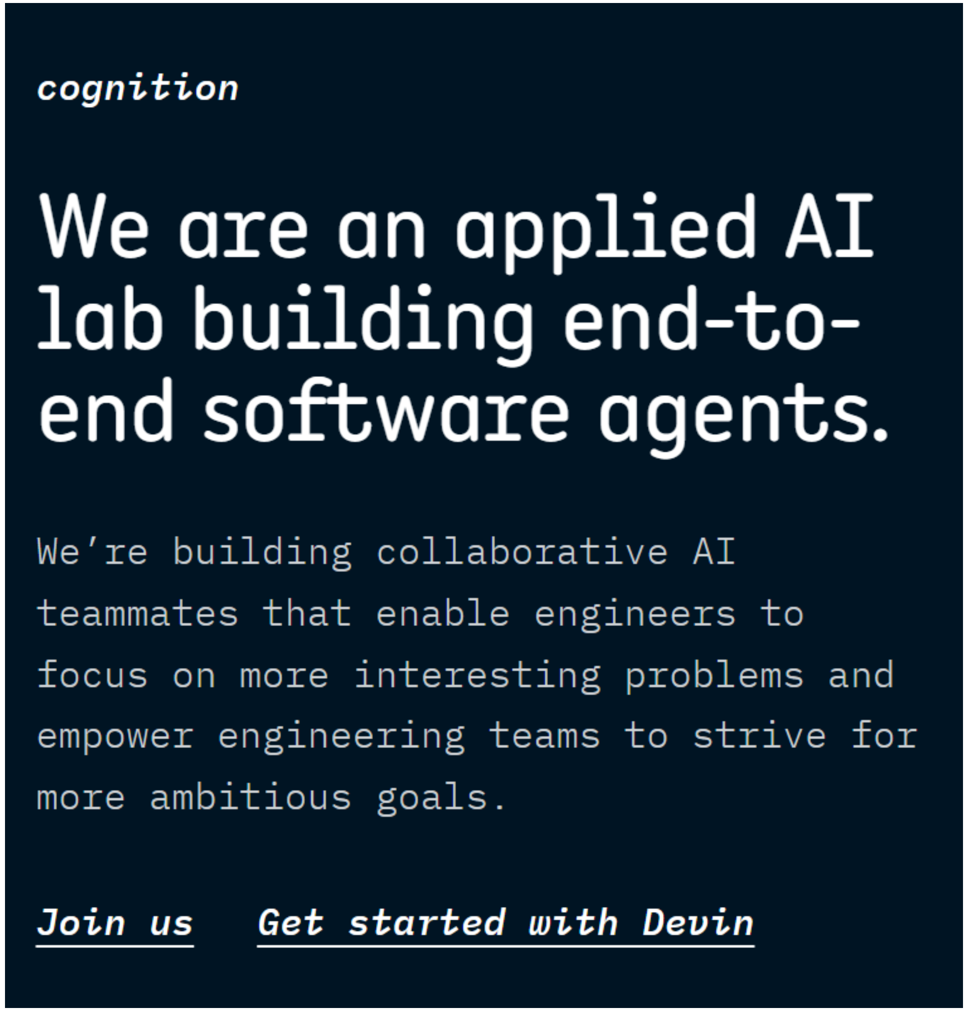
Smart Workmate

Economics

Law

Ethics

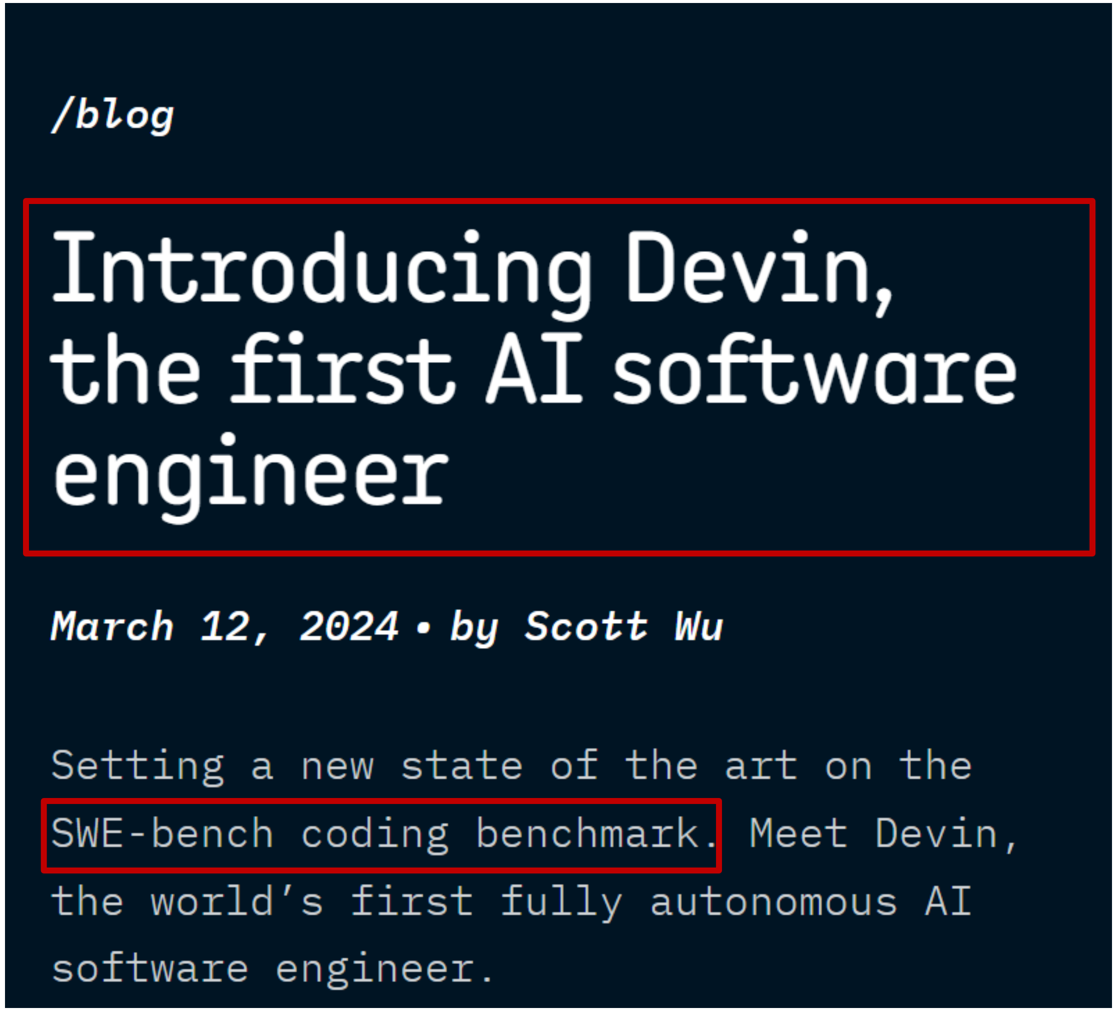
Software Engineering 2.0: A Year After ICSE'23



cognition

We are an applied AI lab building end-to-end software agents.

We're building collaborative AI teammates that enable engineers to focus on more interesting problems and empower engineering teams to strive for more ambitious goals.



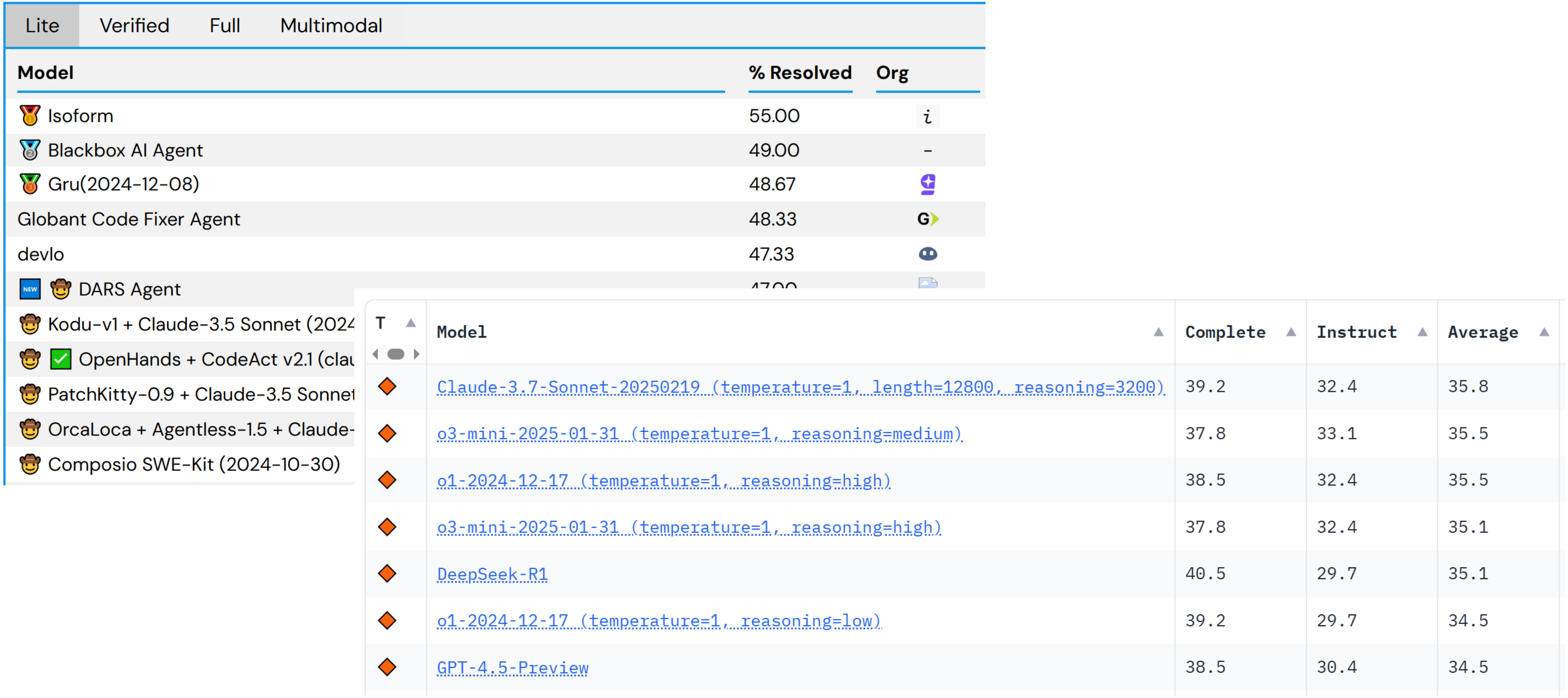
/blog

Introducing Devin, the first AI software engineer

March 12, 2024 • by Scott Wu

Setting a new state of the art on the SWE-bench coding benchmark. Meet Devin, the world's first fully autonomous AI software engineer.

Software Engineering 2.0: Two Years After ICSE’23



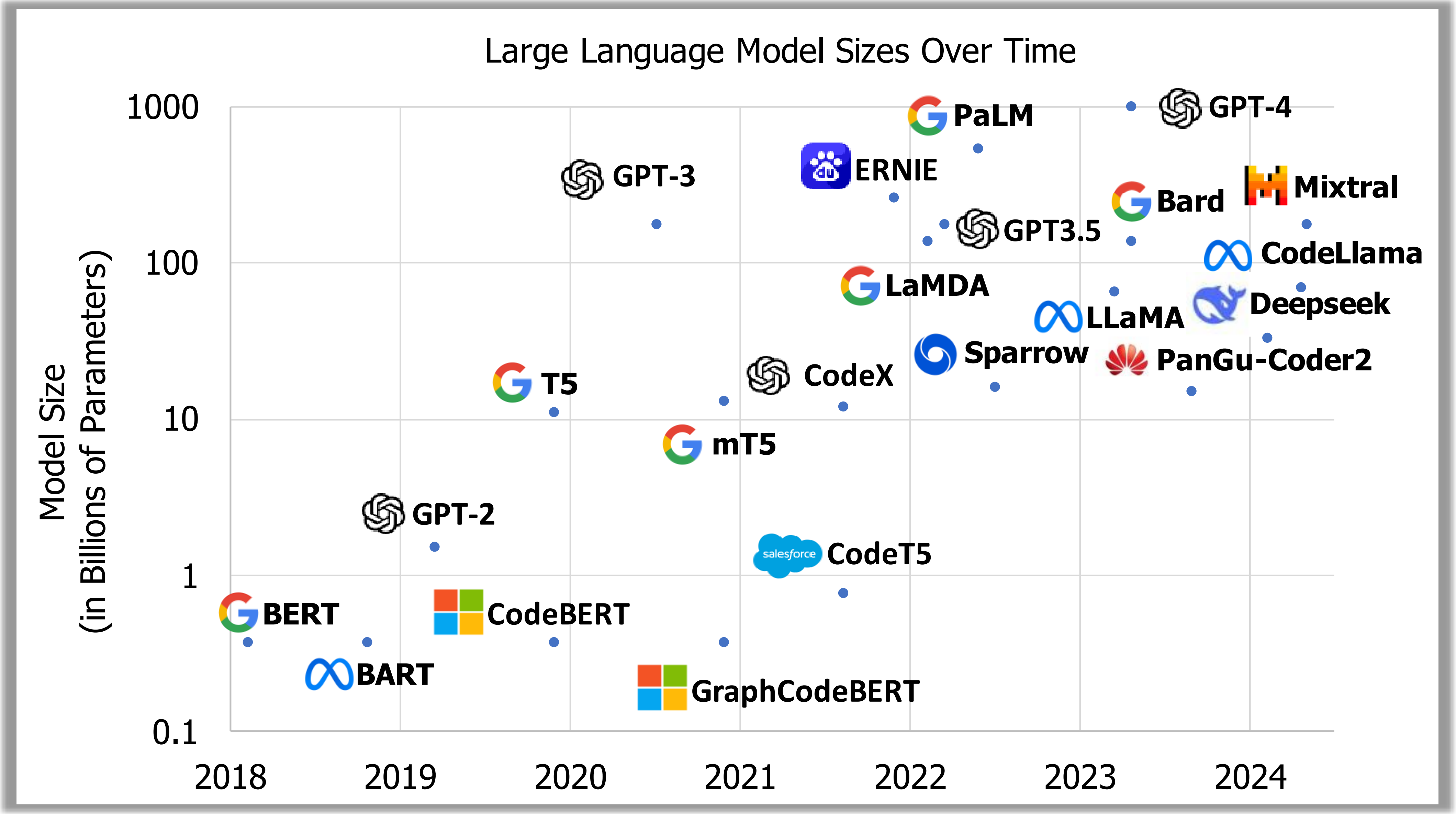
Leaderboard

Lite | Verified | Full | Multimodal

Model	% Resolved	Org
🥇 Isoform	55.00	i
🥈 Blackbox AI Agent	49.00	–
🥉 Gru(2024-12-08)	48.67	
Globant Code Fixer Agent	48.33	G
devlo	47.33	
NEW 🤠 DARS Agent	47.00	
🤠 Kodu-v1 + Claude-3.5 Sonnet (2024		
🤠 ✅ OpenHands + CodeAct v2.1 (clau		
🤠 PatchKitty-0.9 + Claude-3.5 Sonnet		
🤠 OrcaLoca + Agentless-1.5 + Claude-		
🤠 Composio SWE-Kit (2024-10-30)		

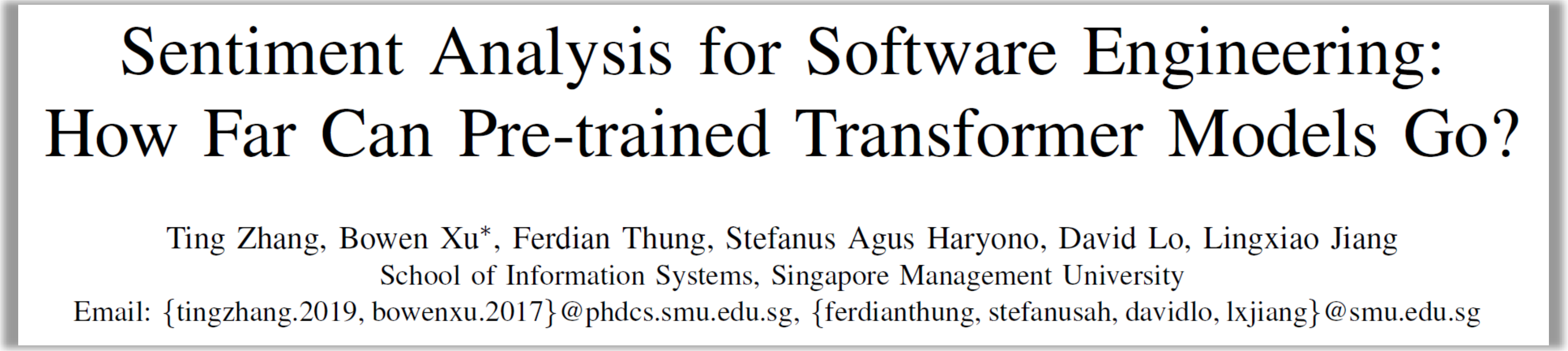
T	Model	Complete	Instruct	Average
◆	Claude-3.7-Sonnet-20250219 (temperature=1, length=12800, reasoning=3200)	39.2	32.4	35.8
◆	o3-mini-2025-01-31 (temperature=1, reasoning=medium)	37.8	33.1	35.5
◆	o1-2024-12-17 (temperature=1, reasoning=high)	38.5	32.4	35.5
◆	o3-mini-2025-01-31 (temperature=1, reasoning=high)	37.8	32.4	35.1
◆	DeepSeek-R1	40.5	29.7	35.1
◆	o1-2024-12-17 (temperature=1, reasoning=low)	39.2	29.7	34.5
◆	GPT-4.5-Preview	38.5	30.4	34.5

Large Language Models (LLMs)



LLM4SE Work @ SMU (often with many other institutions)

ICSME 2020

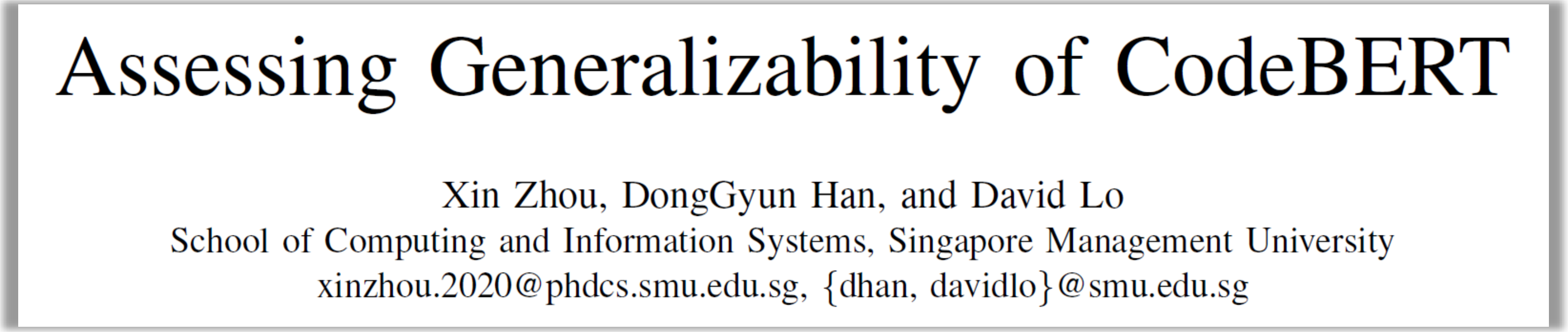


Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?

Ting Zhang, Bowen Xu*, Ferdian Thung, Stefanus Agus Haryono, David Lo, Lingxiao Jiang
School of Information Systems, Singapore Management University
Email: {tingzhang.2019, bowenxu.2017}@phdcs.smu.edu.sg, {ferdianthung, stefanusah, davidlo, lxjiang}@smu.edu.sg

Early work on LLM4SE (requirement analysis)
Most cited paper of ICSME 2020

ICSME 2021



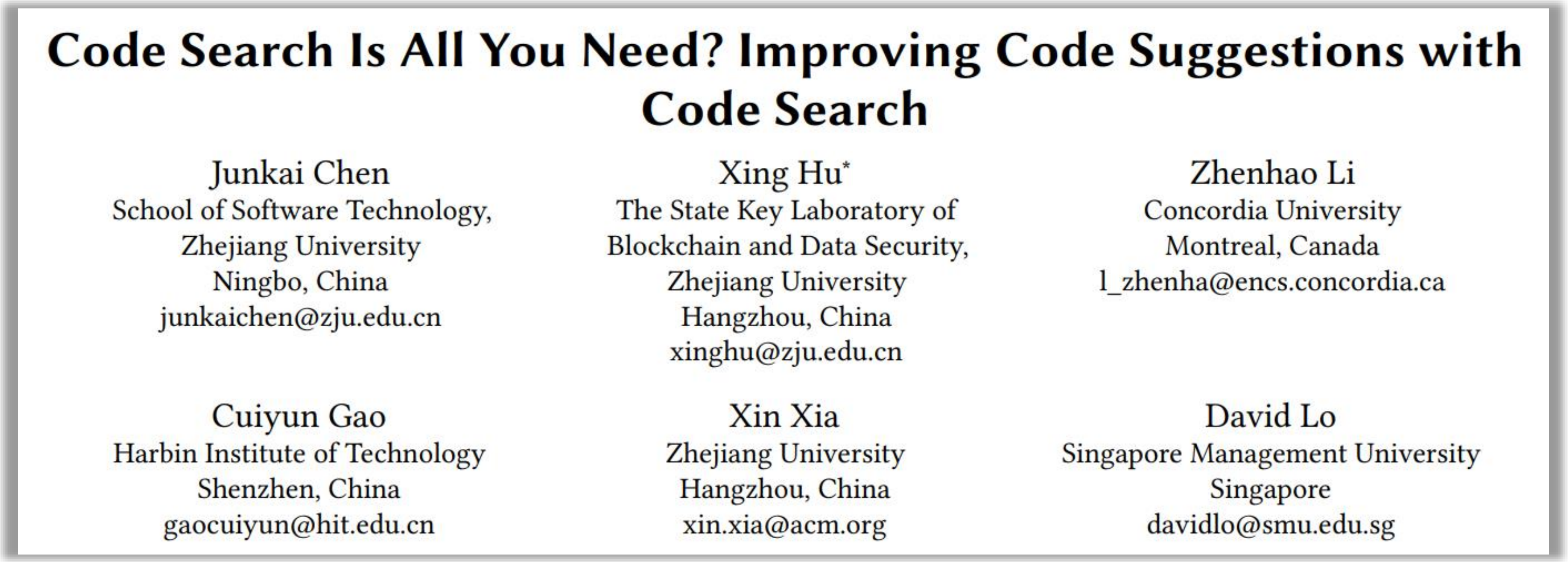
Assessing Generalizability of CodeBERT

Xin Zhou, DongGyun Han, and David Lo
School of Computing and Information Systems, Singapore Management University
xinzhou.2020@phdcs.smu.edu.sg, {dhan, davidlo}@smu.edu.sg

Early work on LLM4SE (code search and defect prediction)
Most cited paper of ICSME 2021

LLM4SE Work @ SMU (often with many other institutions)

ICSE 2024



Code Search Is All You Need? Improving Code Suggestions with Code Search

Junkai Chen
School of Software Technology,
Zhejiang University
Ningbo, China
junkaichen@zju.edu.cn

Xing Hu*
The State Key Laboratory of
Blockchain and Data Security,
Zhejiang University
Hangzhou, China
xinghu@zju.edu.cn

Zhenhao Li
Concordia University
Montreal, Canada
l_zhenha@encs.concordia.ca

Cuiyun Gao
Harbin Institute of Technology
Shenzhen, China
gaocuiyun@hit.edu.cn

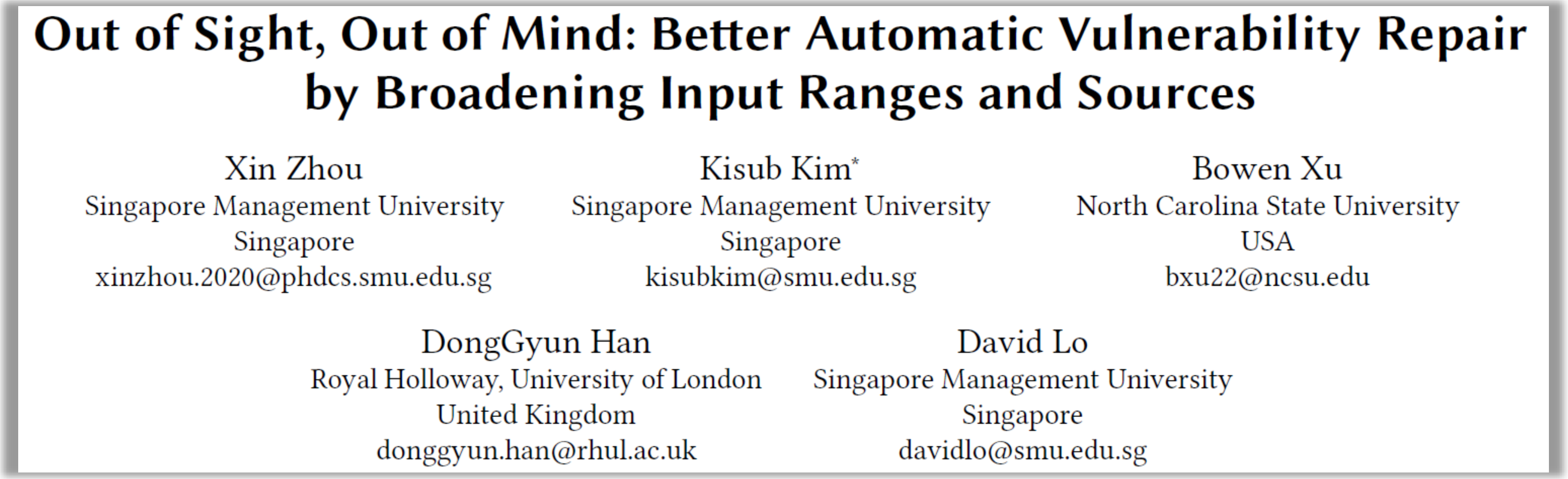
Xin Xia
Zhejiang University
Hangzhou, China
xin.xia@acm.org

David Lo
Singapore Management University
Singapore
davidlo@smu.edu.sg

First plug-and-play, composable RAG method for code LLM
Won an ACM SIGSOFT Distinguished Paper Award

LLM4SE Work @ SMU (often with many other institutions)

ICSE 2024



Out of Sight, Out of Mind: Better Automatic Vulnerability Repair by Broadening Input Ranges and Sources

Xin Zhou
Singapore Management University
Singapore
xinzhou.2020@phdcs.smu.edu.sg

Kisub Kim*
Singapore Management University
Singapore
kisubkim@smu.edu.sg

Bowen Xu
North Carolina State University
USA
bxu22@ncsu.edu

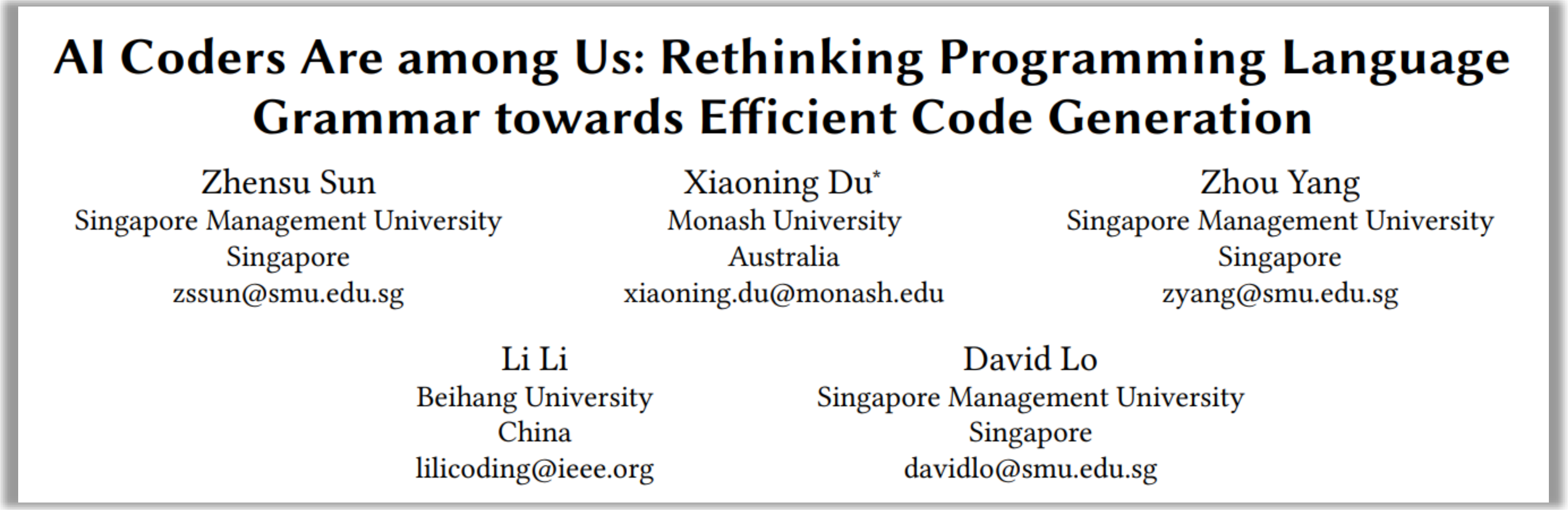
DongGyun Han
Royal Holloway, University of London
United Kingdom
donggyun.han@rhul.ac.uk

David Lo
Singapore Management University
Singapore
davidlo@smu.edu.sg

*First multi-LLM collaboration mimicking human learning for vulnerability repair; lightweight model, achieving **2x** efficacy*

LLM4SE Work @ SMU (often with many other institutions)

ISSTA 2024



AI Coders Are among Us: Rethinking Programming Language Grammar towards Efficient Code Generation

Zhensu Sun
Singapore Management University
Singapore
zssun@smu.edu.sg

Xiaoning Du*
Monash University
Australia
xiaoning.du@monash.edu

Zhou Yang
Singapore Management University
Singapore
zyang@smu.edu.sg

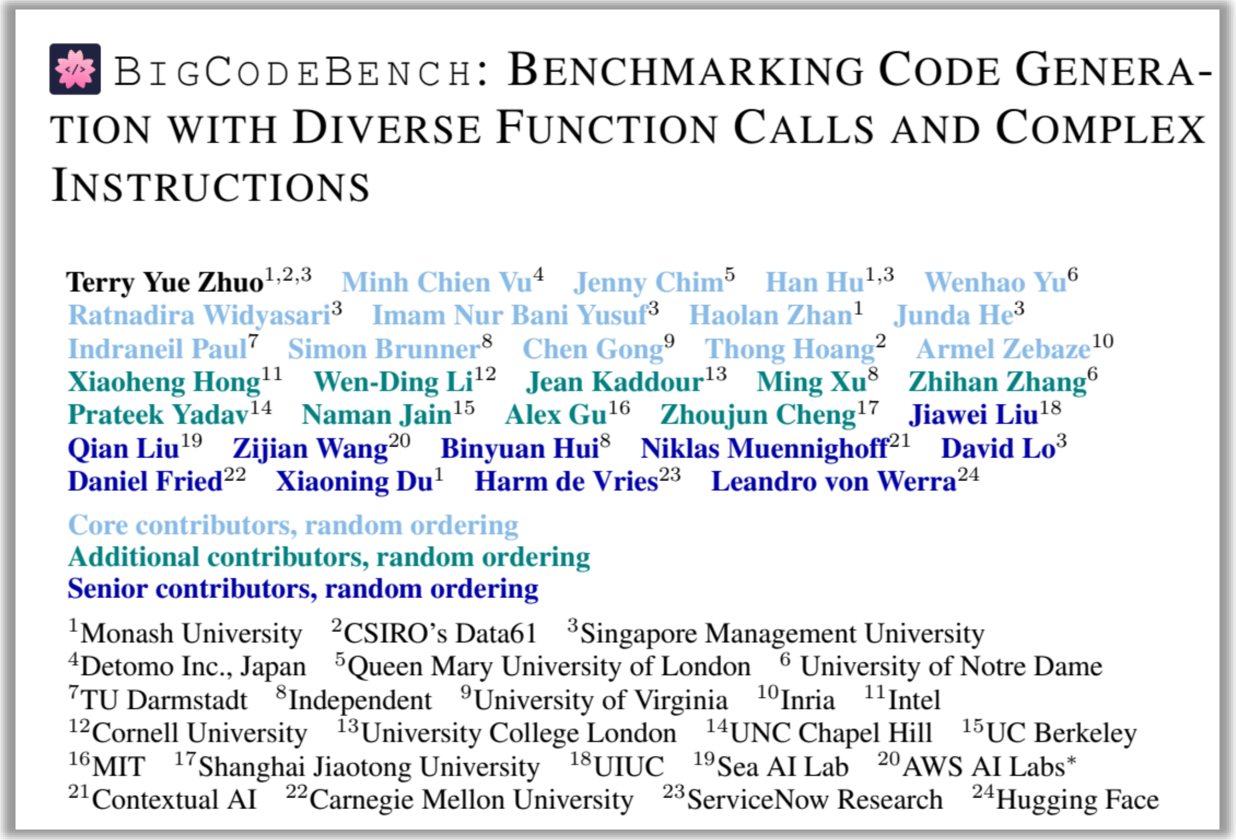
Li Li
Beihang University
China
lilicoding@ieee.org

David Lo
Singapore Management University
Singapore
davidlo@smu.edu.sg

First programming language for LLM agents
Won an ACM SIGSOFT Distinguished Paper Award

LLM4SE Work @ SMU (often with many other institutions)

ICLR 2025



BIGCODEBENCH: BENCHMARKING CODE GENERATION WITH DIVERSE FUNCTION CALLS AND COMPLEX INSTRUCTIONS

Terry Yue Zhuo[1,2,3] **Minh Chien Vu**[4] **Jenny Chim**[5] **Han Hu**[1,3] **Wenhao Yu**[6]
Ratnadira Widyasari[3] **Imam Nur Bani Yusuf**[3] **Haolan Zhan**[1] **Junda He**[3]
Indraneil Paul[7] **Simon Brunner**[8] **Chen Gong**[9] **Thong Hoang**[2] **Armel Zebaze**[10]
Xiaoheng Hong[11] **Wen-Ding Li**[12] **Jean Kaddour**[13] **Ming Xu**[8] **Zhihan Zhang**[6]
Prateek Yadav[14] **Naman Jain**[15] **Alex Gu**[16] **Zhoujun Cheng**[17] **Jiawei Liu**[18]
Qian Liu[19] **Zijian Wang**[20] **Binyuan Hui**[8] **Niklas Muennighoff**[21] **David Lo**[3]
Daniel Fried[22] **Xiaoning Du**[1] **Harm de Vries**[23] **Leandro von Werra**[24]

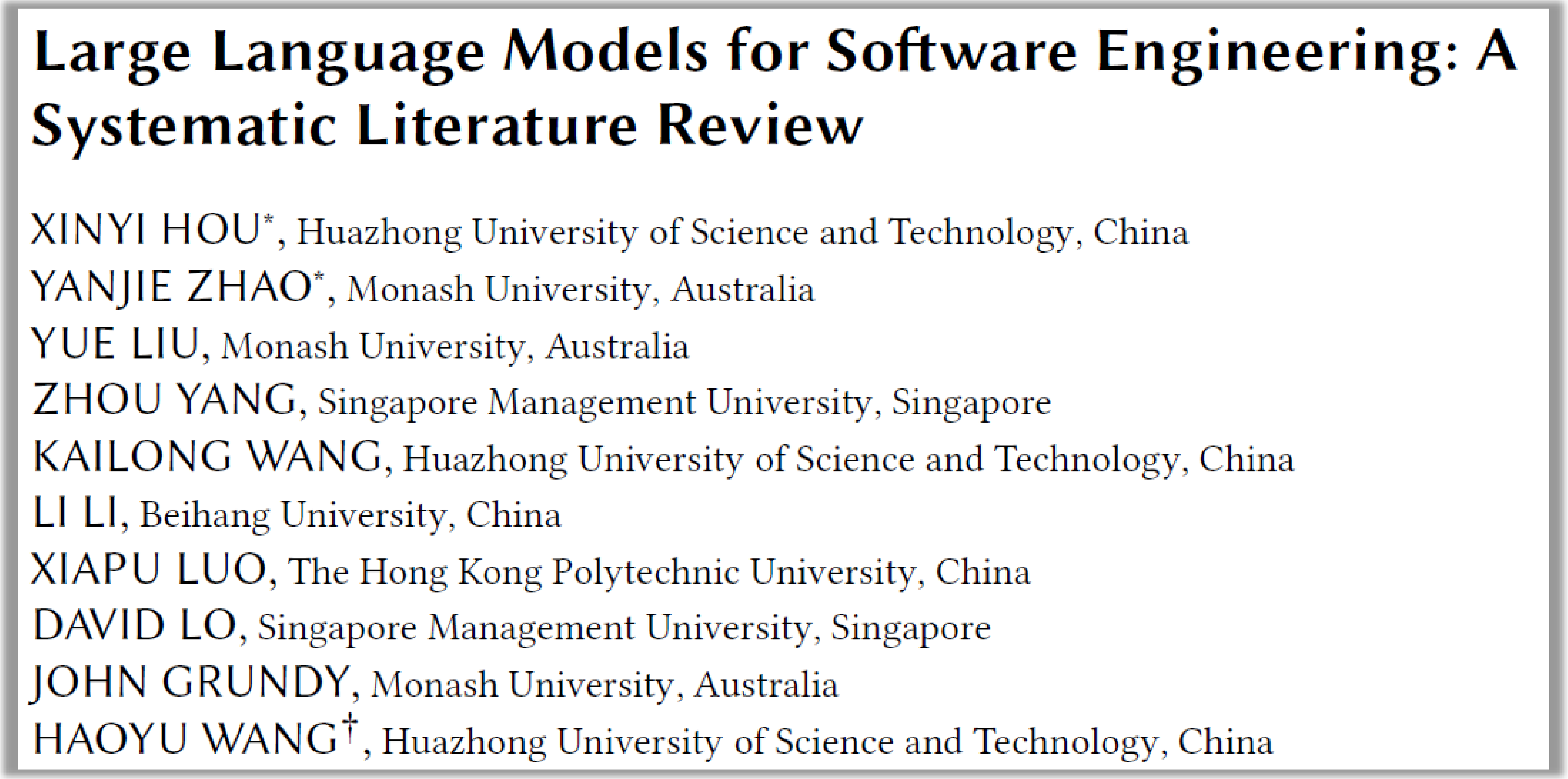
Core contributors, random ordering
Additional contributors, random ordering
Senior contributors, random ordering

[1]Monash University [2]CSIRO's Data61 [3]Singapore Management University
[4]Detomo Inc., Japan [5]Queen Mary University of London [6] University of Notre Dame
[7]TU Darmstadt [8]Independent [9]University of Virginia [10]Inria [11]Intel
[12]Cornell University [13]University College London [14]UNC Chapel Hill [15]UC Berkeley
[16]MIT [17]Shanghai Jiaotong University [18]UIUC [19]Sea AI Lab [20]AWS AI Labs*
[21]Contextual AI [22]Carnegie Mellon University [23]ServiceNow Research [24]Hugging Face

First human-LLM constructed large-scale benchmark for open-ended API-centric programming problems
Oral presentation (top 1% of 11k+ submissions) (top-10 highest-rated paper)

LLMs Seem to Win for Many SE Scenarios

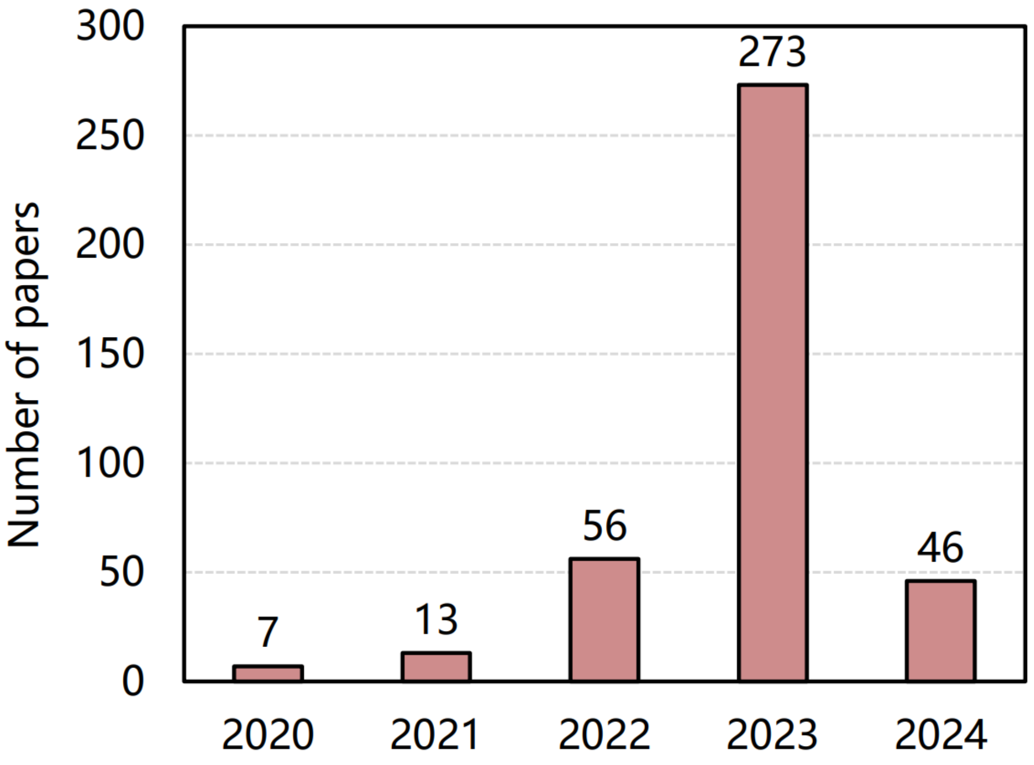
TOSEM 2024

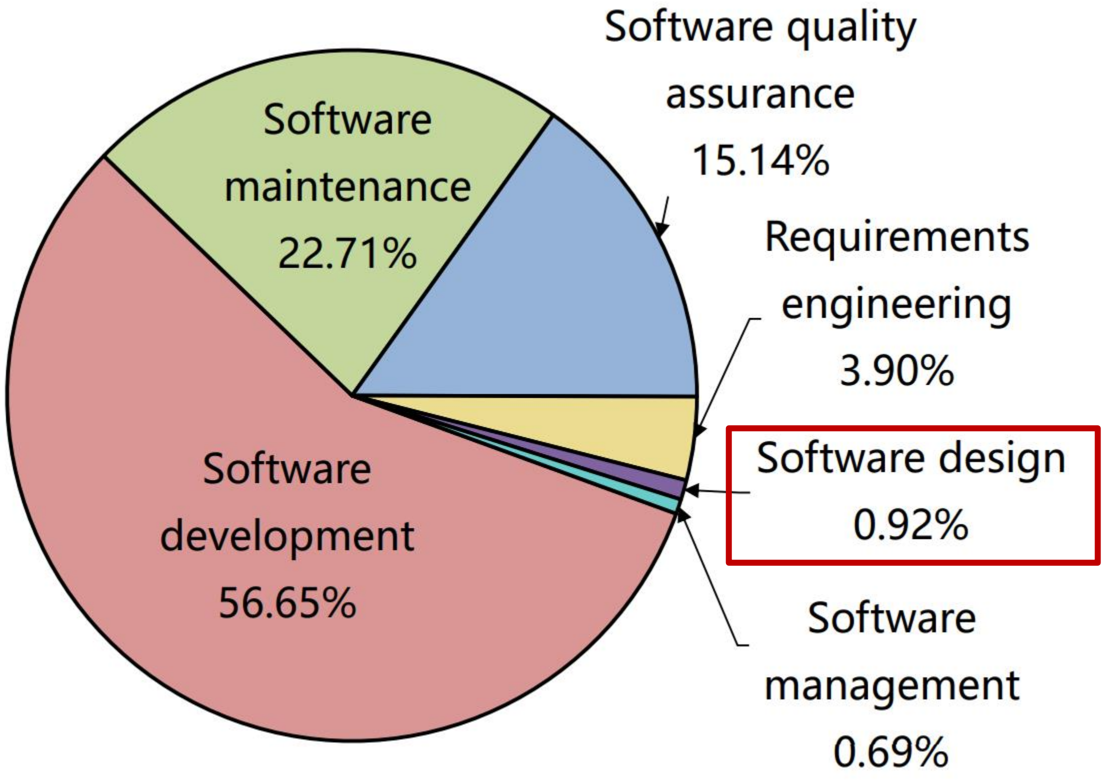


Large Language Models for Software Engineering: A Systematic Literature Review

XINYI HOU*, Huazhong University of Science and Technology, China
YANJIE ZHAO*, Monash University, Australia
YUE LIU, Monash University, Australia
ZHOU YANG, Singapore Management University, Singapore
KAILONG WANG, Huazhong University of Science and Technology, China
LI LI, Beihang University, China
XIAPU LUO, The Hong Kong Polytechnic University, China
DAVID LO, Singapore Management University, Singapore
JOHN GRUNDY, Monash University, Australia
HAOYU WANG†, Huazhong University of Science and Technology, China

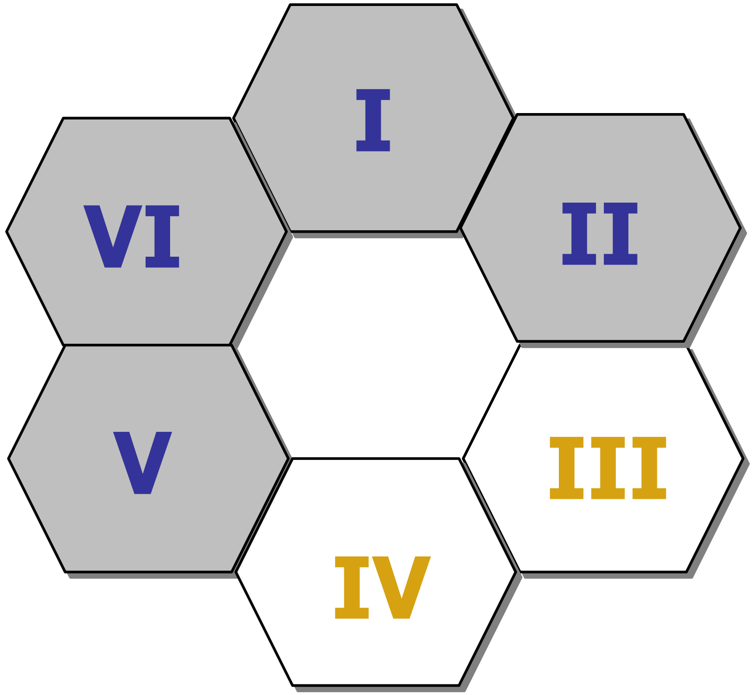
LLMs Seem to Win for Many SE Scenarios





Could LLM help address challenges in architecting software?

Talk Structure



Self-Intro.

SE 2.0

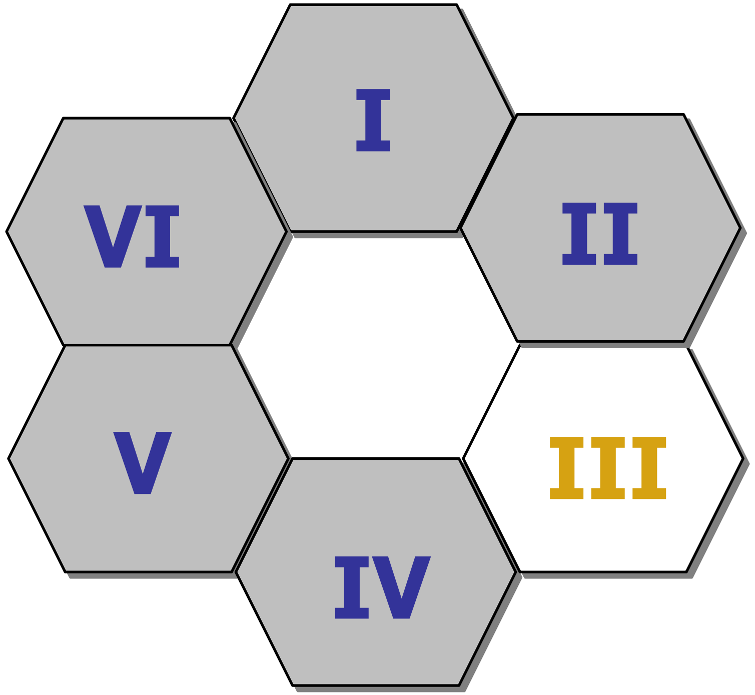
Software Architecture (SA) Challenges

LLM4SA Existing Work

LLM4SA
Open Challenges

Realizing SE 2.0 &
Implications to LLM4SA

Talk Structure



Self-Intro.

SE 2.0

Software Architecture (SA) Challenges

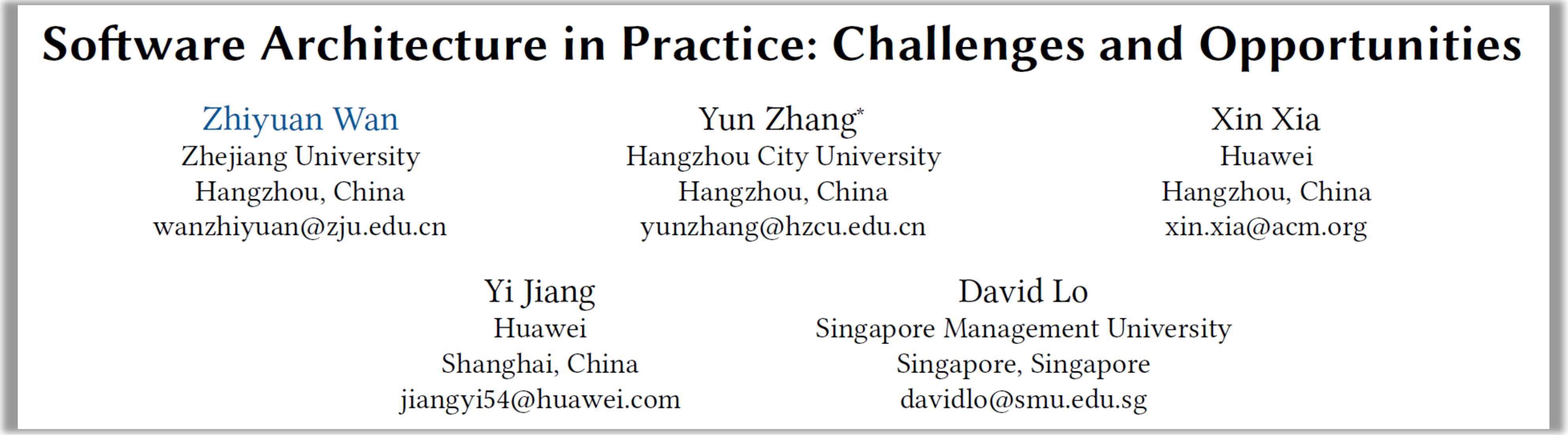
LLM4SA
Existing Work

LLM4SA
Open Challenges

Realizing SE 2.0 &
Implications to LLM4SA

Challenges in Architecting Software Systems

ESEC/FSE 2023



Software Architecture in Practice: Challenges and Opportunities

Zhiyuan Wan
Zhejiang University
Hangzhou, China
wanzhiyuan@zju.edu.cn

Yun Zhang*
Hangzhou City University
Hangzhou, China
yunzhang@hzcu.edu.cn

Xin Xia
Huawei
Hangzhou, China
xin.xia@acm.org

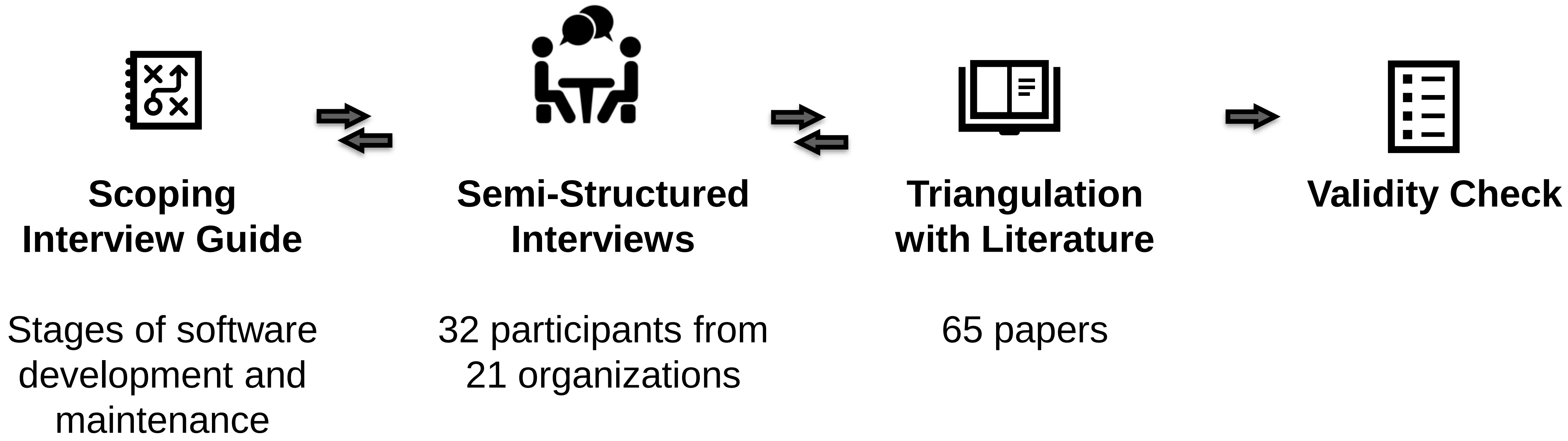
Yi Jiang
Huawei
Shanghai, China
jiangyi54@huawei.com

David Lo
Singapore Management University
Singapore, Singapore
davidlo@smu.edu.sg

Methodologies

- Present **challenges software architects face**, based on an **interview study** and **literature review**.

We adopted a qualitative research strategy based on Straussian Grounded Theory.



Scoping Interview Guide

Stages of software development and maintenance

Semi-Structured Interviews

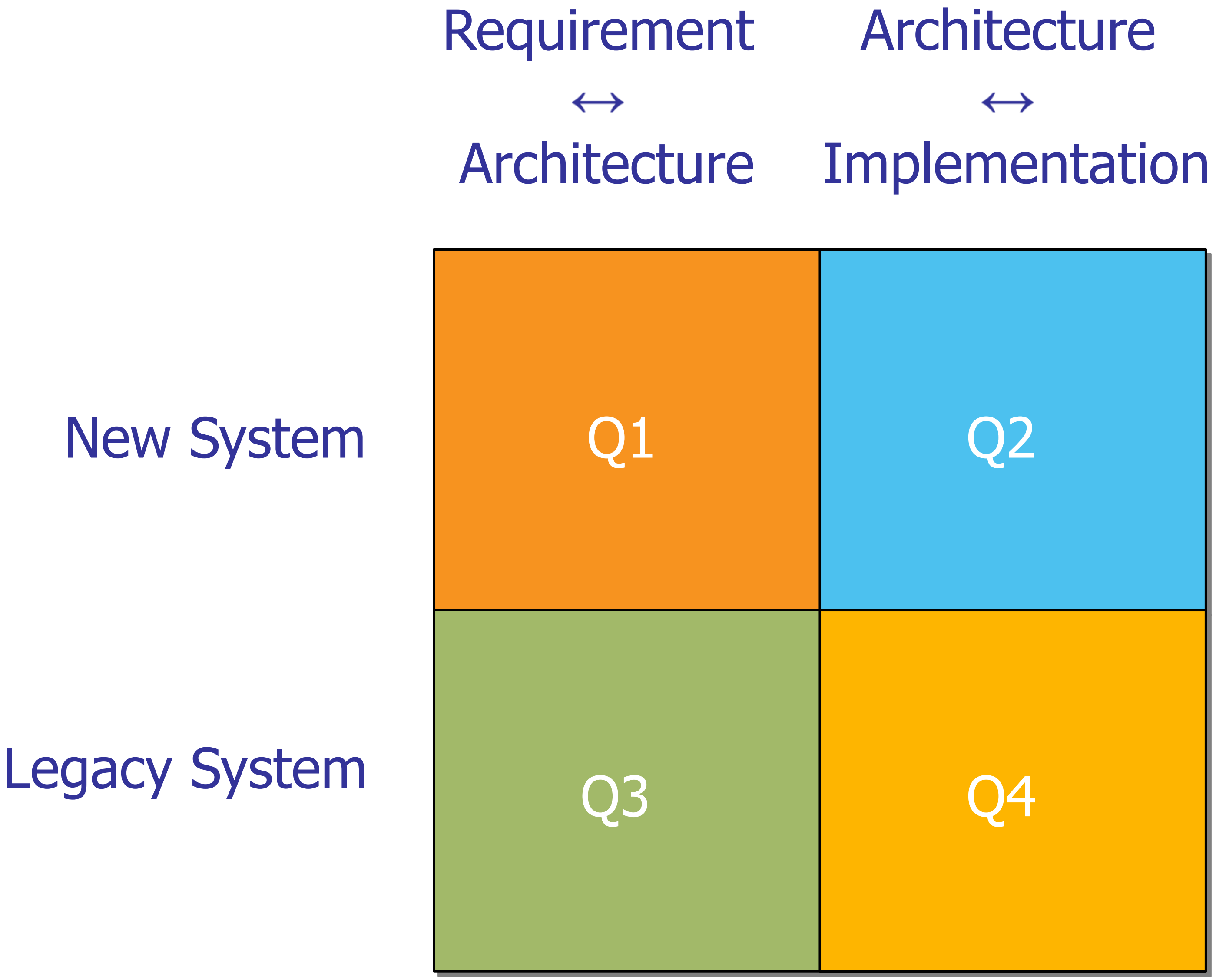
32 participants from 21 organizations

Triangulation with Literature

65 papers

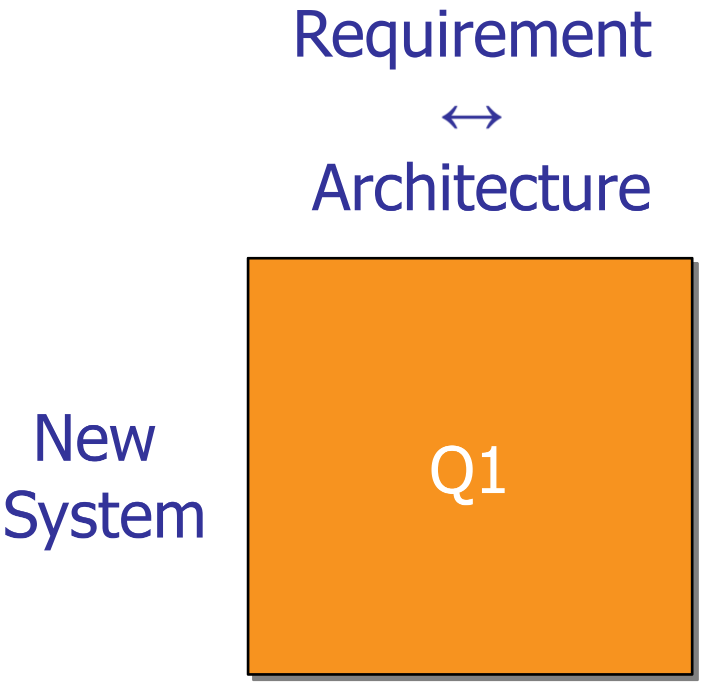
Validity Check

Rehashing Identified Challenges into Four Quadrants



	Requirement ↔ Architecture	Architecture ↔ Implementation
New System	Q1	Q2
Legacy System	Q3	Q4

Quadrant 1: Challenges



[C1] Completing architecture documentation

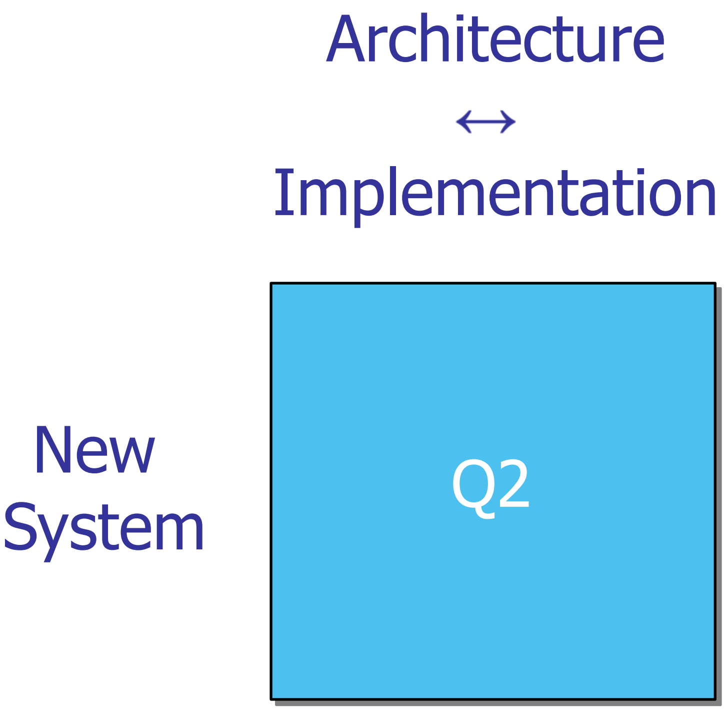
[C2] Anticipating future requirements and trade-offs

"I cannot find the relevant information in architecture documentation"

"we had to think about scaling up the capacity, handling more users, and dealing with increased concurrency in our architecture design ... looking ahead … 2 to 3 years,"

"Architects cannot always foresee future requirements when they design architecture"

Quadrant 2: Challenges



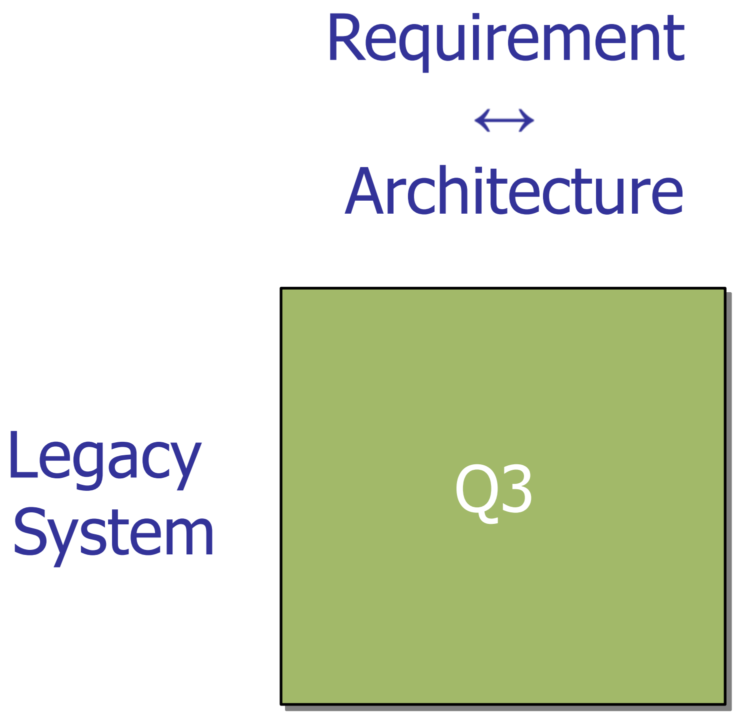
[C3] Documenting traceability links between architecture and code

[C4] Ensuring architecture documentation aligns with implementation

"no standard process or tool support to build trace links between design decisions and their implementation"

"Documentation-code inconsistency sometimes confuses me when I implement new features"

Quadrant 3: Challenges


Requirement
↔
Architecture

Legacy System

Q3

[C5] Consolidating architecture knowledge from scattered sources

[C6] Mapping and propagating business requirement changes to architectural changes

"we simply use Microsoft Office Word to capture our design documents"

"business requirements change over time in different frequencies ... a component tend to be highly coupled with others if it is responsible for both frequently and rarely changed requirements."

Quadrant 4: Challenges

Architecture
↔
Implementation

Legacy System

Q4

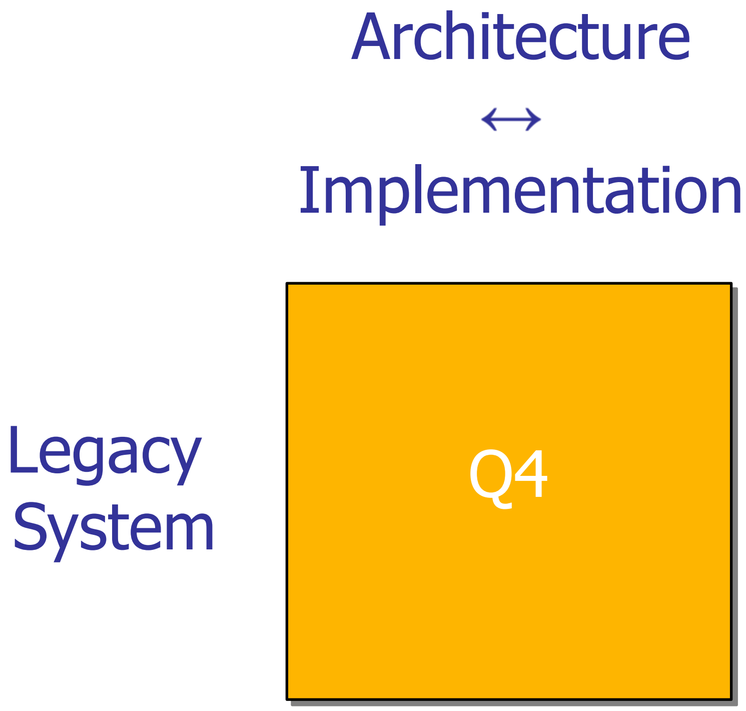
[C7] Maintaining up-to-date architecture documentation

[C8] Detecting and addressing architecture erosion

"developers tend to forget updating architecture documentation when evolving software architectures because of deadline pressures."

"we've got tools to help us catch duplicate code and other code smells, but it's not clear how the results actually help us identify architecture problems."

Quadrant 4: Challenges



Architecture
↔
Implementation

Legacy System

Q4

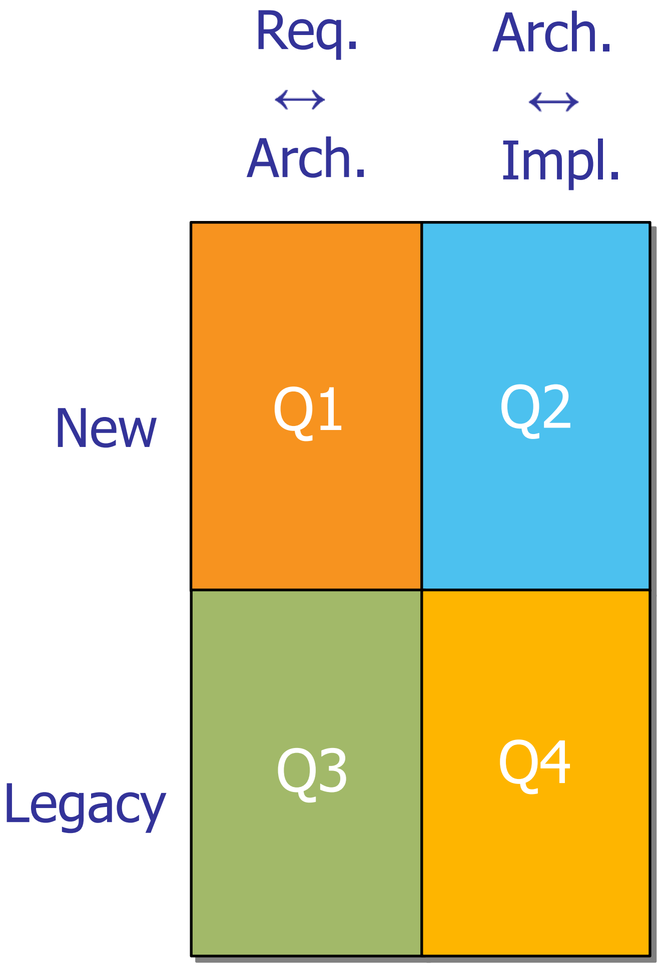
[C9] Analyzing the impact of architectural changes

[C10] Refactoring architecture in a cost-effective manner

"impact analysis of architecture refactoring becomes even challenging for an aging system due to its increasing complexity and staff turnover, because no one in the team could be capable of performing accurate impact analysis"

"given the potentially high anticipated cost of architecture refactoring, the senior management level would like to see clear quantifiable value for the organization."

Ten Challenges Software Architects Face



Req. ↔ Arch.

Arch. ↔ Impl.

New

Legacy

Q1

Q2

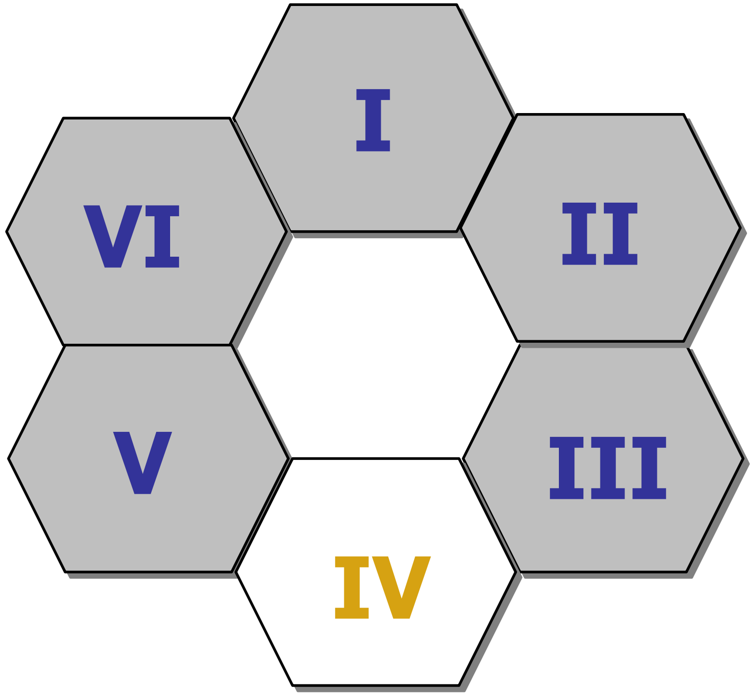
Q3

Q4

Challenges (32 Participants, 21 Organizations, 60+ Papers)
[C1] Completing architecture documentation
[C2] Anticipating future requirements and trade-offs
[C3] Documenting traceability links between architecture and code
[C4] Ensuring architecture documentation aligns with implementation
[C5] Consolidating architecture knowledge from scattered sources
[C6] Mapping and propagating business requirement changes to architectural changes
[C7] Maintaining up-to-date architecture documentation
[C8] Detecting and addressing architecture erosion
[C9] Analyzing the impact of architectural changes
[C10] Refactoring architecture in a cost-effective manner

Given these challenges,
can LLMs help?

Talk Structure



Self-Intro.

SE 2.0

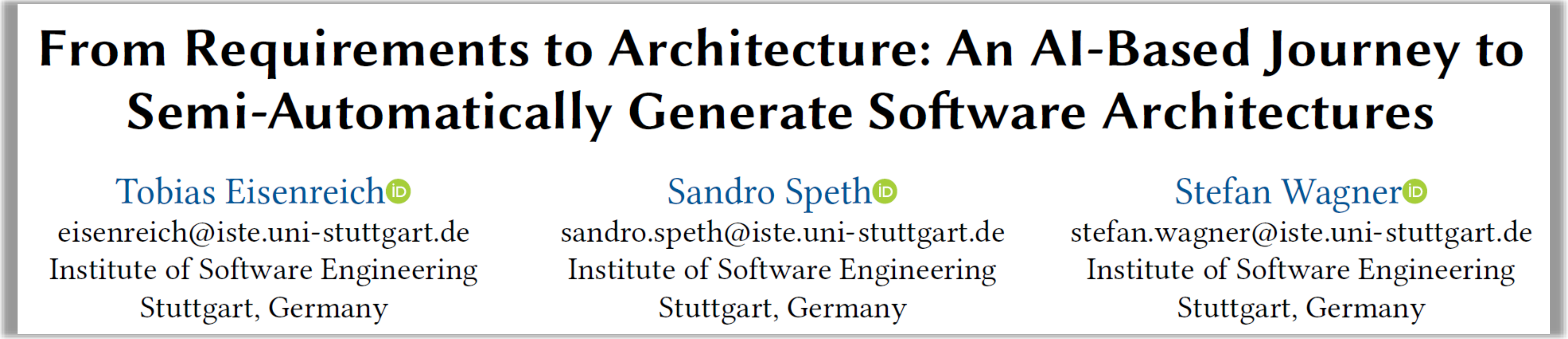
Software Architecture
(SA) Challenges

LLM4SA
Existing Work

LLM4SA
Open Challenges

Realizing SE 2.0 &
Implications to LLM4SA

Existing Work I – Generating Domain Model



From Requirements to Architecture: An AI-Based Journey to Semi-Automatically Generate Software Architectures

Tobias Eisenreich
eisenreich@iste.uni-stuttgart.de
Institute of Software Engineering
Stuttgart, Germany

Sandro Speth
sandro.speth@iste.uni-stuttgart.de
Institute of Software Engineering
Stuttgart, Germany

Stefan Wagner
stefan.wagner@iste.uni-stuttgart.de
Institute of Software Engineering
Stuttgart, Germany

Designing Workshop 2024
(Position Paper)

Existing Work I - Generating Domain Model

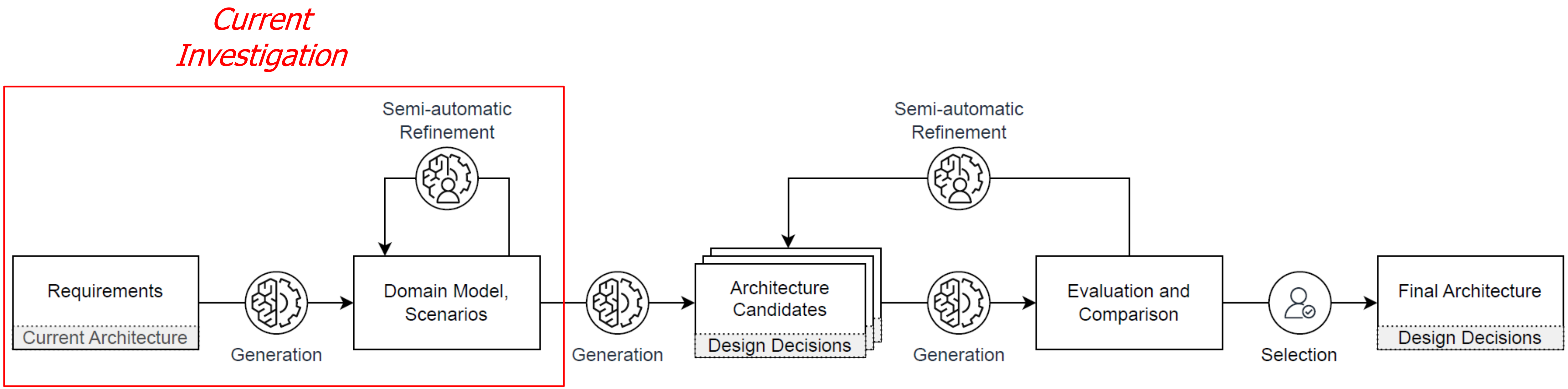
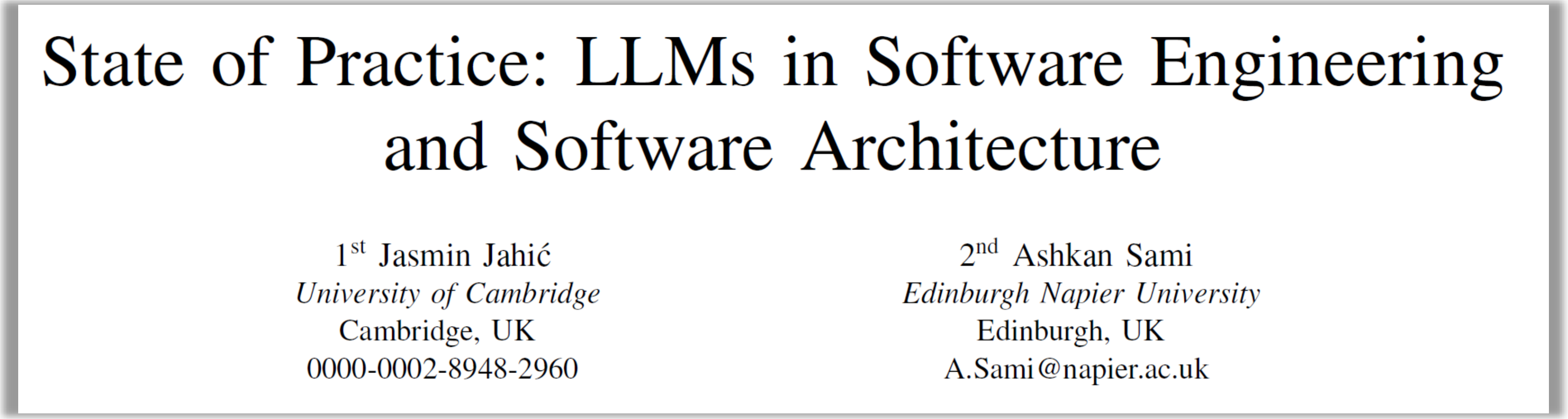


Figure 1: Semi-automatic architecture creation process.

Existing Work I - Generating Domain Model

- Exploratory analysis with chat versions of LLaMA2 70-B and GPT-3.5
    - Task: Generate a PlantUML domain model
    - Input: 91 requirements from the MobSTr-dataset
- Findings: Both models could identify concepts from the requirements
- Limitations:
    - Instead of modeling the domain, the LLMs modeled the system itself
    - LLaMA did not create relations between the concepts

Existing Work II – Generating C4 Container Solution Diagram



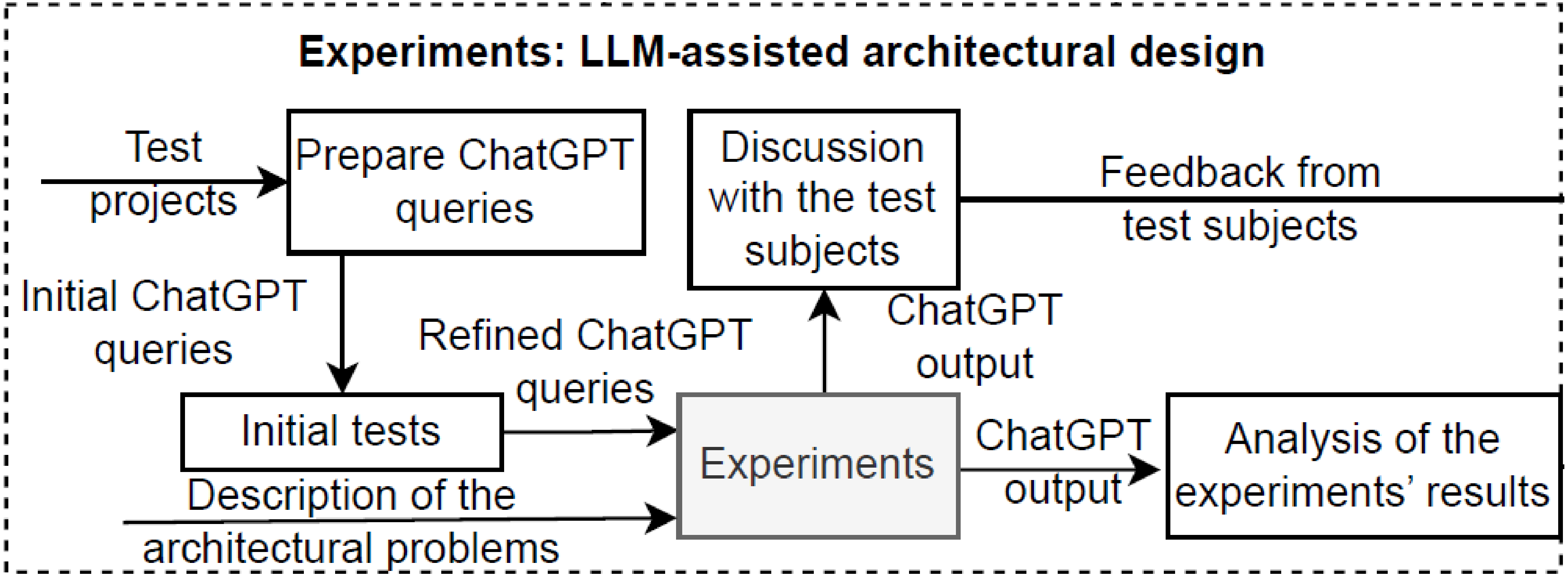
State of Practice: LLMs in Software Engineering and Software Architecture

1st Jasmin Jahić
University of Cambridge
Cambridge, UK
0000-0002-8948-2960

2nd Ashkan Sami
Edinburgh Napier University
Edinburgh, UK
A.Sami@napier.ac.uk

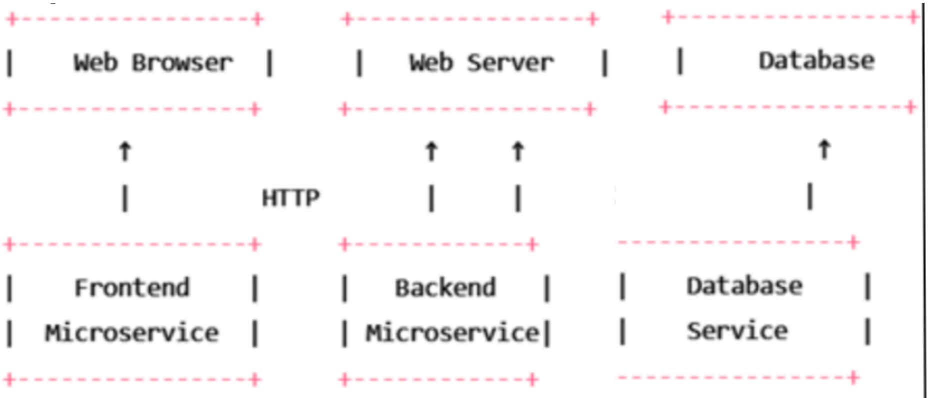
SAML 2024

Existing Work II – Generating C4 Container Solution Diagram

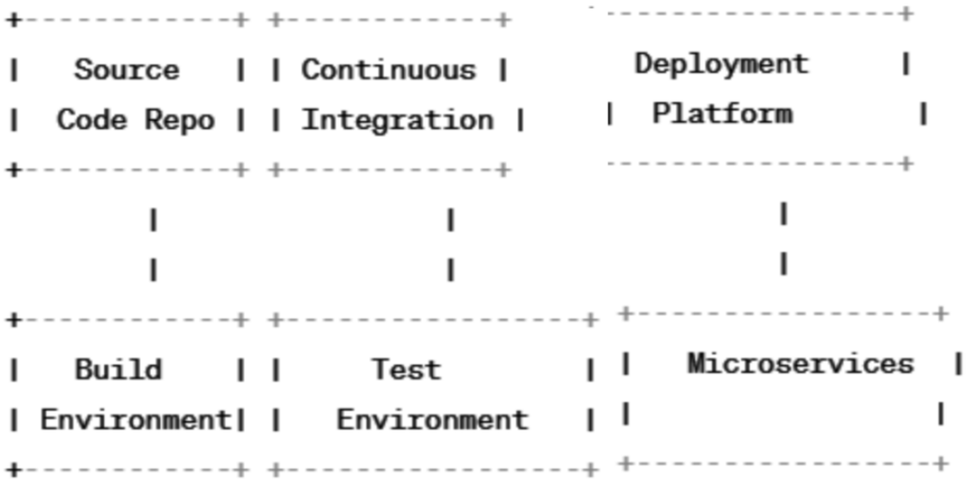

Experiments: LLM-assisted architectural design
Test projects
Prepare ChatGPT queries
Discussion with the test subjects
Feedback from test subjects
Initial ChatGPT queries
Refined ChatGPT queries
ChatGPT output
Initial tests
Experiments
ChatGPT output
Analysis of the experiments' results
Description of the architectural problems

Existing Work II – Generating C4 Container Solution Diagram

- Draw C4 container solution diagrams
    - Several test projects from Master theses
- GPT3.5 with the right prompts can produce results:



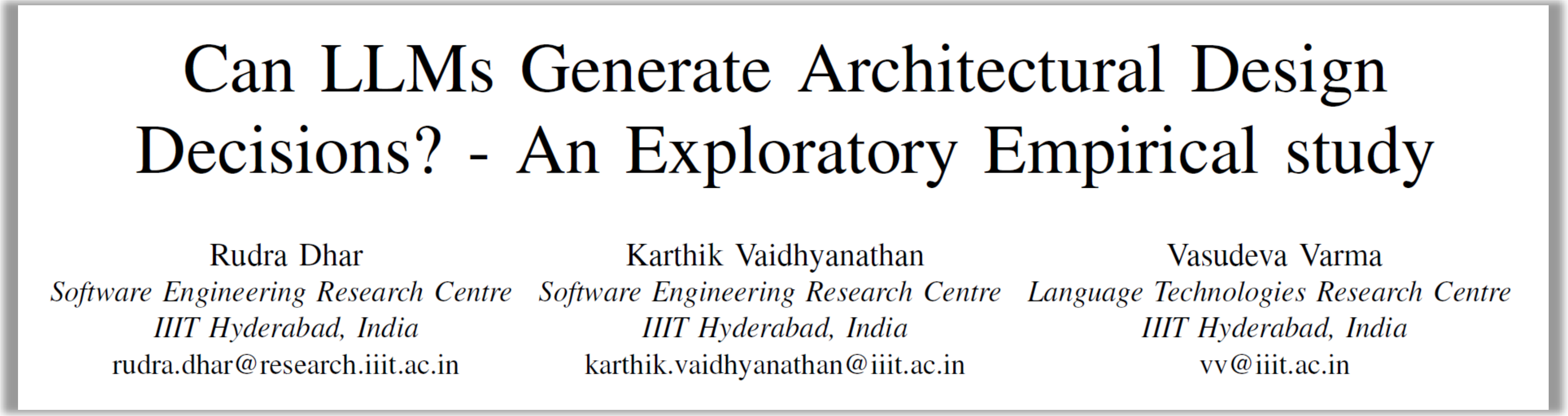
```
+----------------+    +----------------+    +----------------+
|   Web Browser  |    |   Web Server   |    |       Database
+----------------+    +----------------+    +----------------+
       ↑                    ↑     ↑                      ↑
       |          HTTP      |     |                      |
+----------------+    +-------------+    -----------------+
|   Frontend     |    |   Backend    |    |   Database     |
|  Microservice  |    | Microservice|    |   Service      |
+----------------+    +-------------+    -----------------+
```



```
+-----------+ +-----------+     ----------------+
|  Source   | | Continuous |       Deployment    |
| Code Repo | | Integration |  |   Platform       |
+-----------+ +-----------+     ----------------+
      |              |                |
      |              |                |
+-----------+ +----------------+ +---------------+
|  Build    | |      Test      | |  Microservices  |
|Environment| |   Environment  | |               |
+-----------+ +----------------+ +---------------+
```

- Limitations: reproducibility, mixture of high-low level concepts, system boundaries and interactions are not well defined

Existing Work III – Completion of ADRs



Can LLMs Generate Architectural Design Decisions? - An Exploratory Empirical study

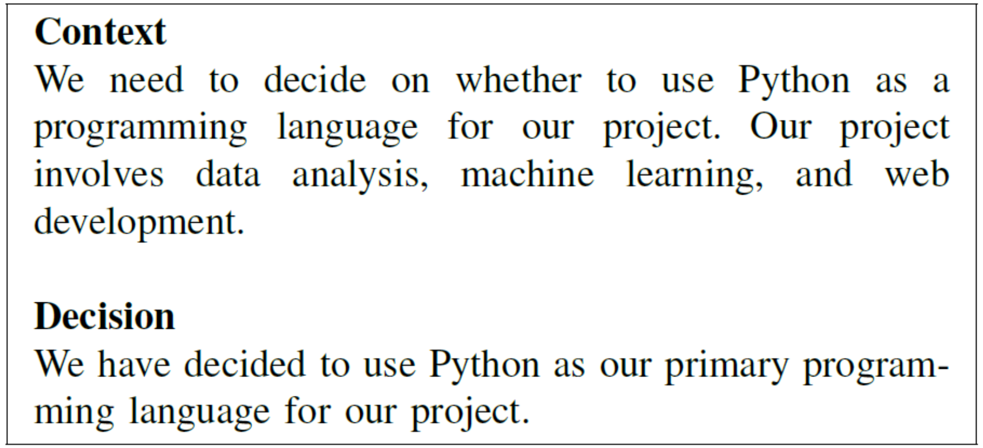
Rudra Dhar
Software Engineering Research Centre
IIIT Hyderabad, India
rudra.dhar@research.iiit.ac.in

Karthik Vaidhyanathan
Software Engineering Research Centre
IIIT Hyderabad, India
karthik.vaidhyanathan@iiit.ac.in

Vasudeva Varma
Language Technologies Research Centre
IIIT Hyderabad, India
vv@iiit.ac.in

ICSA 2024

Existing Work III – Completion of ADRs



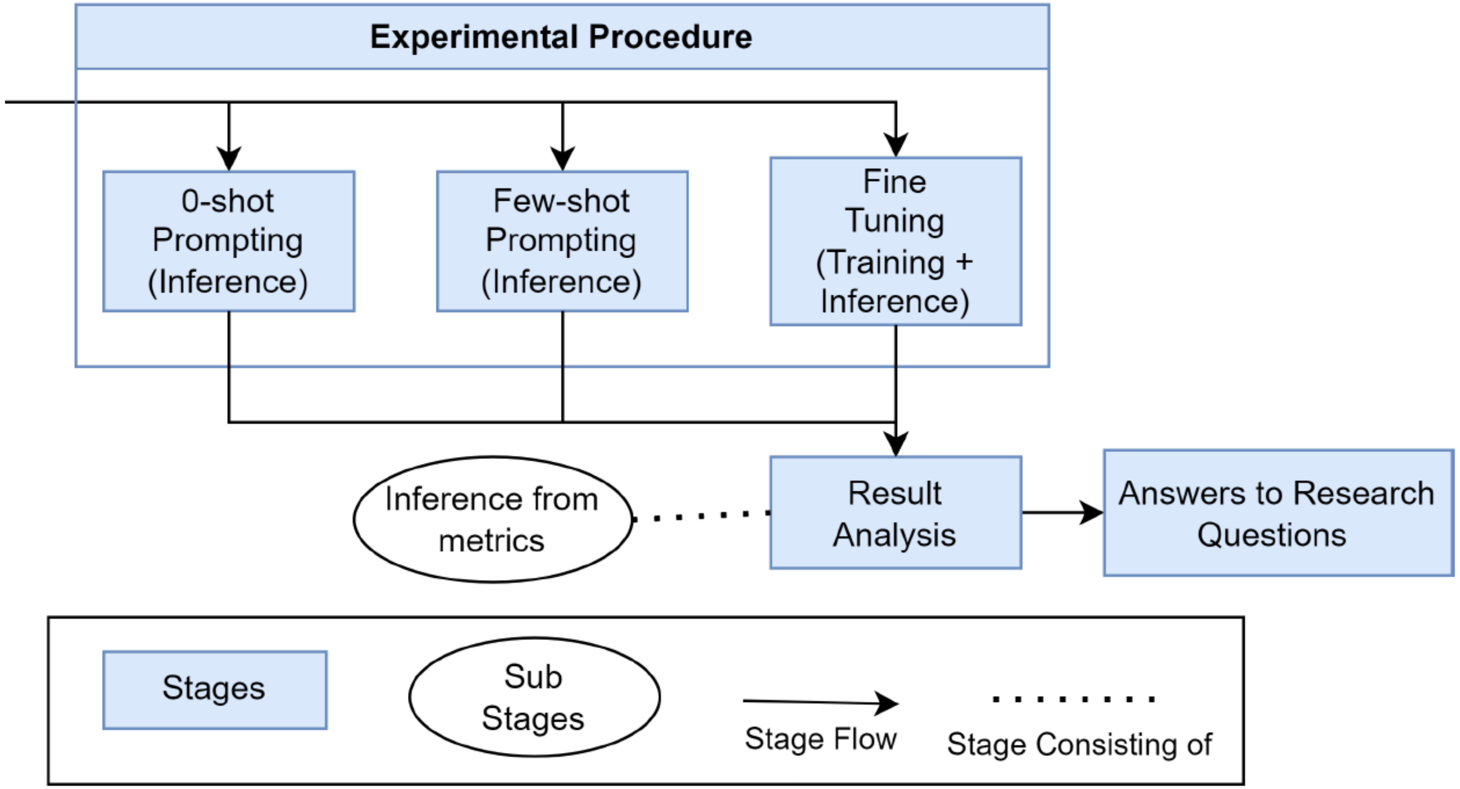
Context

We need to decide on whether to use Python as a programming language for our project. Our project involves data analysis, machine learning, and web development.

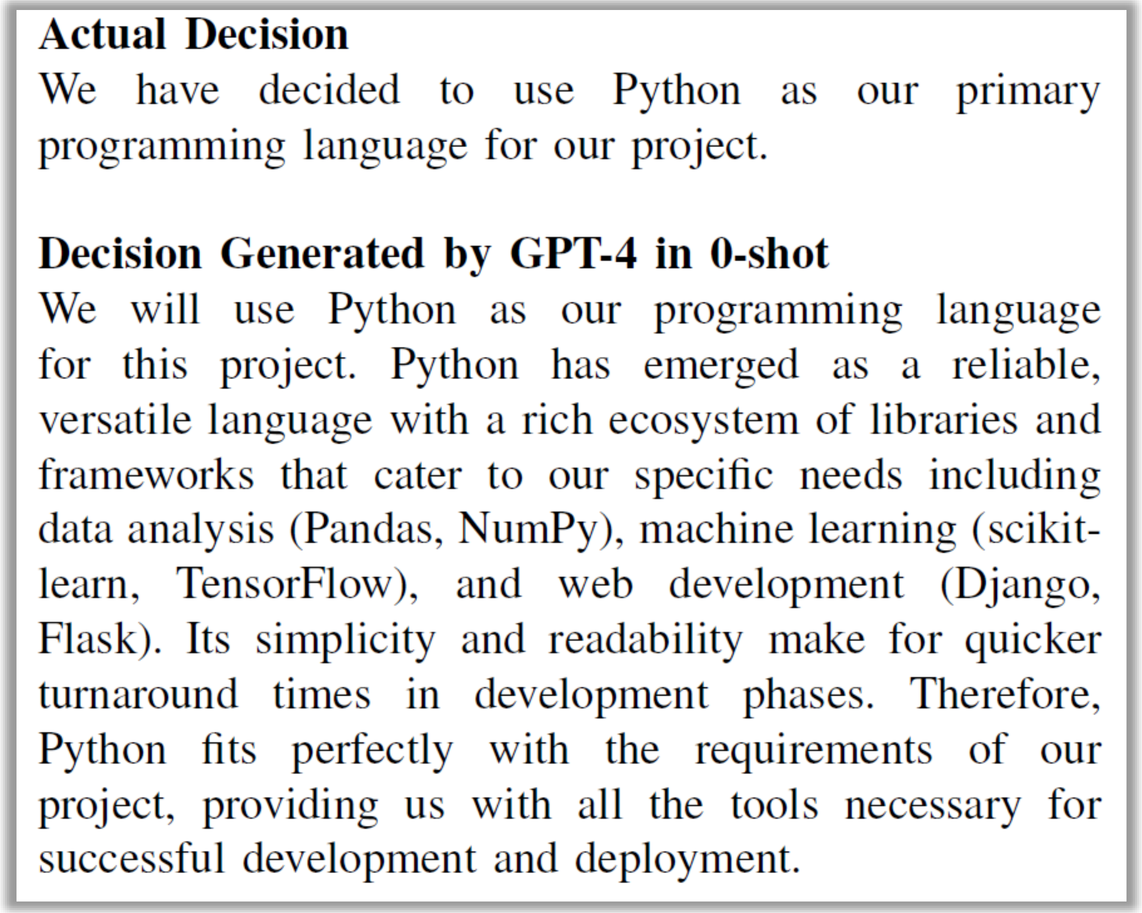
Decision

We have decided to use Python as our primary programming language for our project.

Architectural Decision Record (ADR)


Experimental Procedure
0-shot Prompting (Inference)
Few-shot Prompting (Inference)
Fine Tuning (Training + Inference)
Inference from metrics
Result Analysis
Answers to Research Questions
Stages
Sub Stages
Stage Flow
Stage Consisting of

Existing Work III – Completion of ADRs

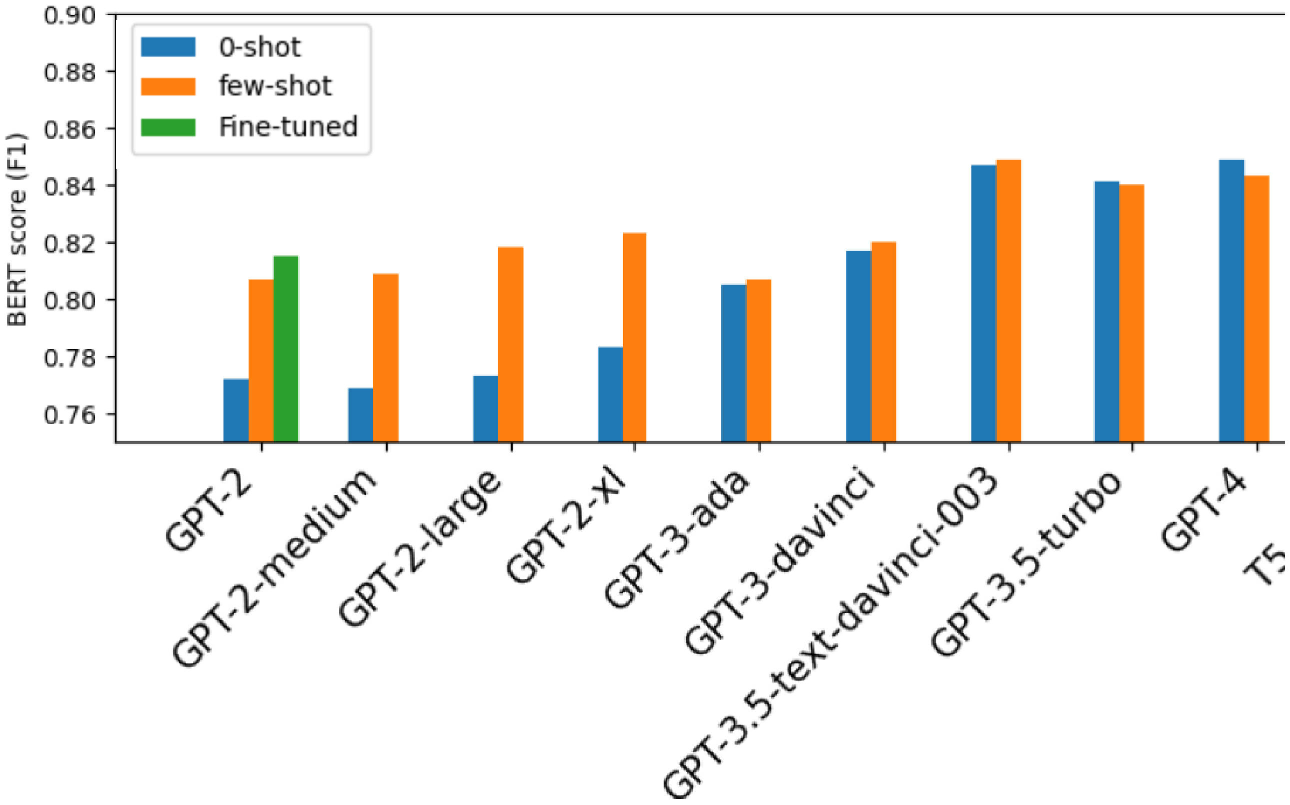


Actual Decision

We have decided to use Python as our primary programming language for our project.

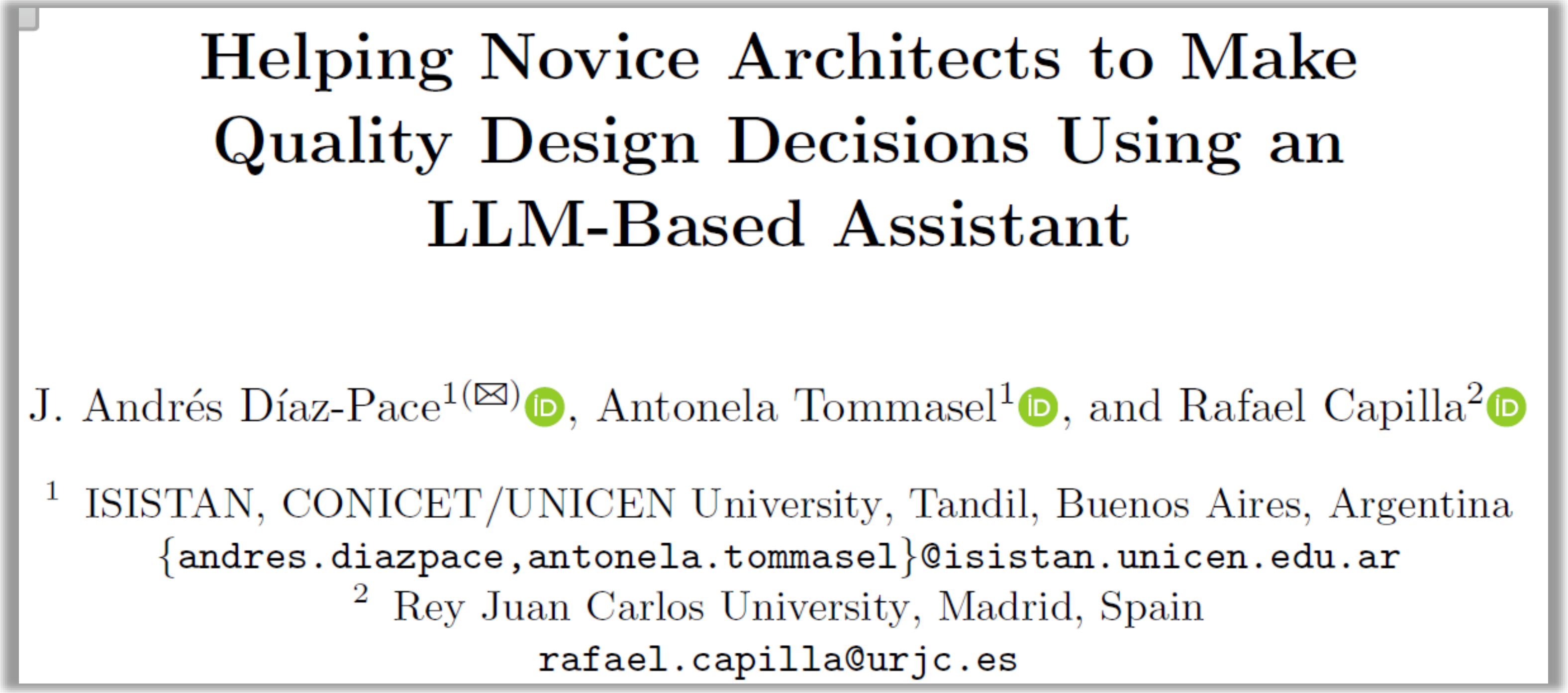
Decision Generated by GPT-4 in 0-shot

We will use Python as our programming language for this project. Python has emerged as a reliable, versatile language with a rich ecosystem of libraries and frameworks that cater to our specific needs including data analysis (Pandas, NumPy), machine learning (scikit-learn, TensorFlow), and web development (Django, Flask). Its simplicity and readability make for quicker turnaround times in development phases. Therefore, Python fits perfectly with the requirements of our project, providing us with all the tools necessary for successful development and deployment.



BERT Score Results
(0.76-0.86 out of 1)

Existing Work IV – Critiquing Architectural Decisions



Helping Novice Architects to Make Quality Design Decisions Using an LLM-Based Assistant

J. Andrés Díaz-Pace[1](✉), Antonela Tommasel[1], and Rafael Capilla[2]

[1] ISISTAN, CONICET/UNICEN University, Tandil, Buenos Aires, Argentina
{andres.diazpace,antonela.tommasel}@isistan.unicen.edu.ar
[2] Rey Juan Carlos University, Madrid, Spain
rafael.capilla@urjc.es

ECSA 2024
(Short Paper)

Existing Work IV – Critiquing Architectural Decisions

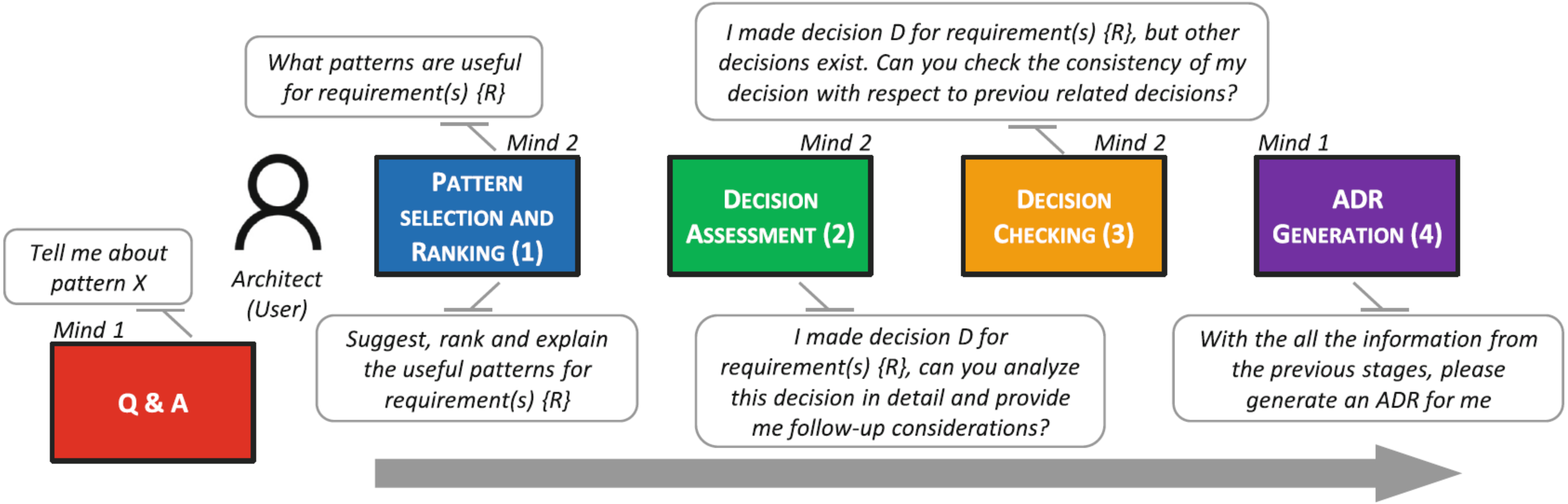
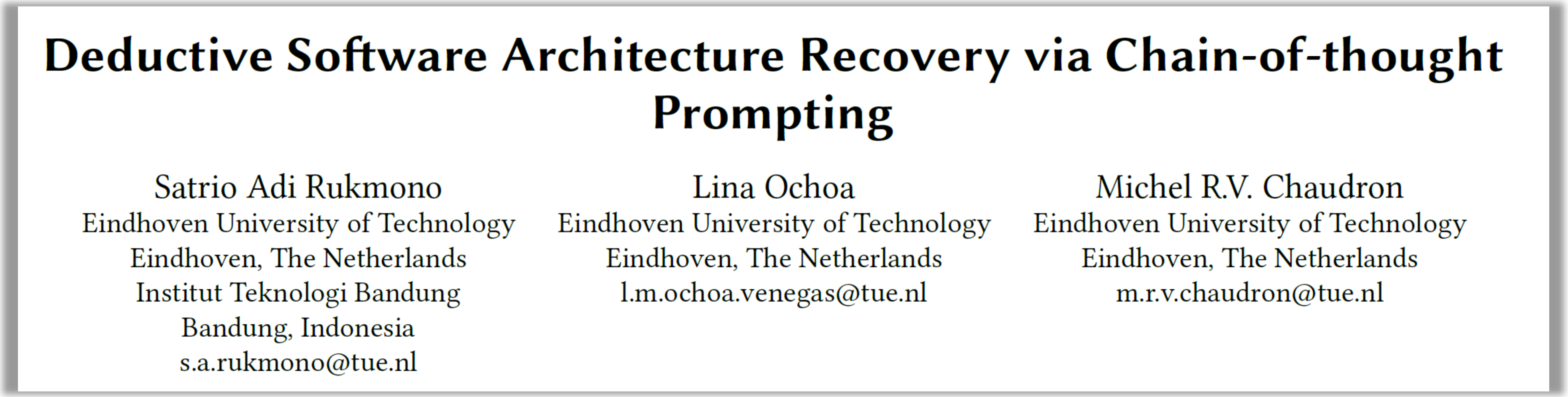

What patterns are useful for requirement(s) {R}
I made decision D for requirement(s) {R}, but other decisions exist. Can you check the consistency of my decision with respect to previou related decisions?
Mind 2
Mind 2
Mind 2
Mind 1
PATTERN SELECTION AND RANKING (1)
DECISION ASSESSMENT (2)
DECISION CHECKING (3)
ADR GENERATION (4)
Tell me about pattern X
Architect (User)
Mind 1
Q & A
Suggest, rank and explain the useful patterns for requirement(s) {R}
I made decision D for requirement(s) {R}, can you analyze this decision in detail and provide me follow-up considerations?
With the all the information from the previous stages, please generate an ADR for me

Fig. 2. Example of an architect's journey and questions supported by each copilot (numbered boxes). The copilots have a one-to-one mapping with the tasks.

Existing Work IV – Critiquing Architectural Decisions

- Initial experiments were conducted for a tool chain on top of GPT3.5
- Experiment data & setting
  - An undergraduate project from an architecture course
  - Output of LLM-powered toolchain vs. 5 ADRs produced by students
- Results
  - Most ADRs as correct, with only a few misleading decisions
  - Issues:
    - Did not capture necessary context
    - Provide no answer (RAG) or generic comments (zero-shot)

Existing Work V – Inference of Architecture Traceability Links



Deductive Software Architecture Recovery via Chain-of-thought Prompting

Satrio Adi Rukmono
Eindhoven University of Technology
Eindhoven, The Netherlands
Institut Teknologi Bandung
Bandung, Indonesia
s.a.rukmono@tue.nl

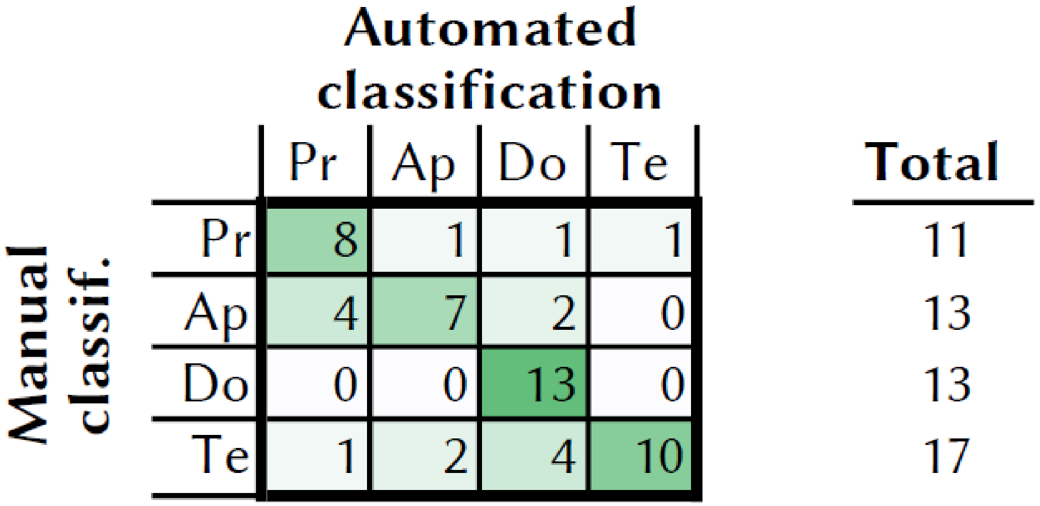
Lina Ochoa
Eindhoven University of Technology
Eindhoven, The Netherlands
l.m.ochoa.venegas@tue.nl

Michel R.V. Chaudron
Eindhoven University of Technology
Eindhoven, The Netherlands
m.r.v.chaudron@tue.nl

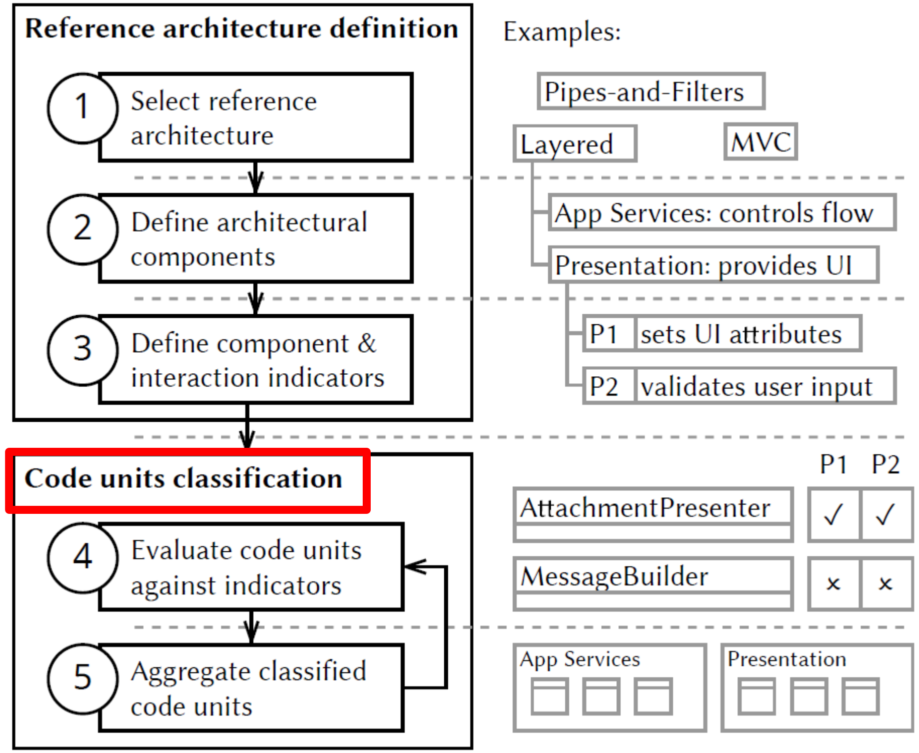
ICSE 2024
(NIER Track)

Existing Work V – Inference of Architecture Traceability Links

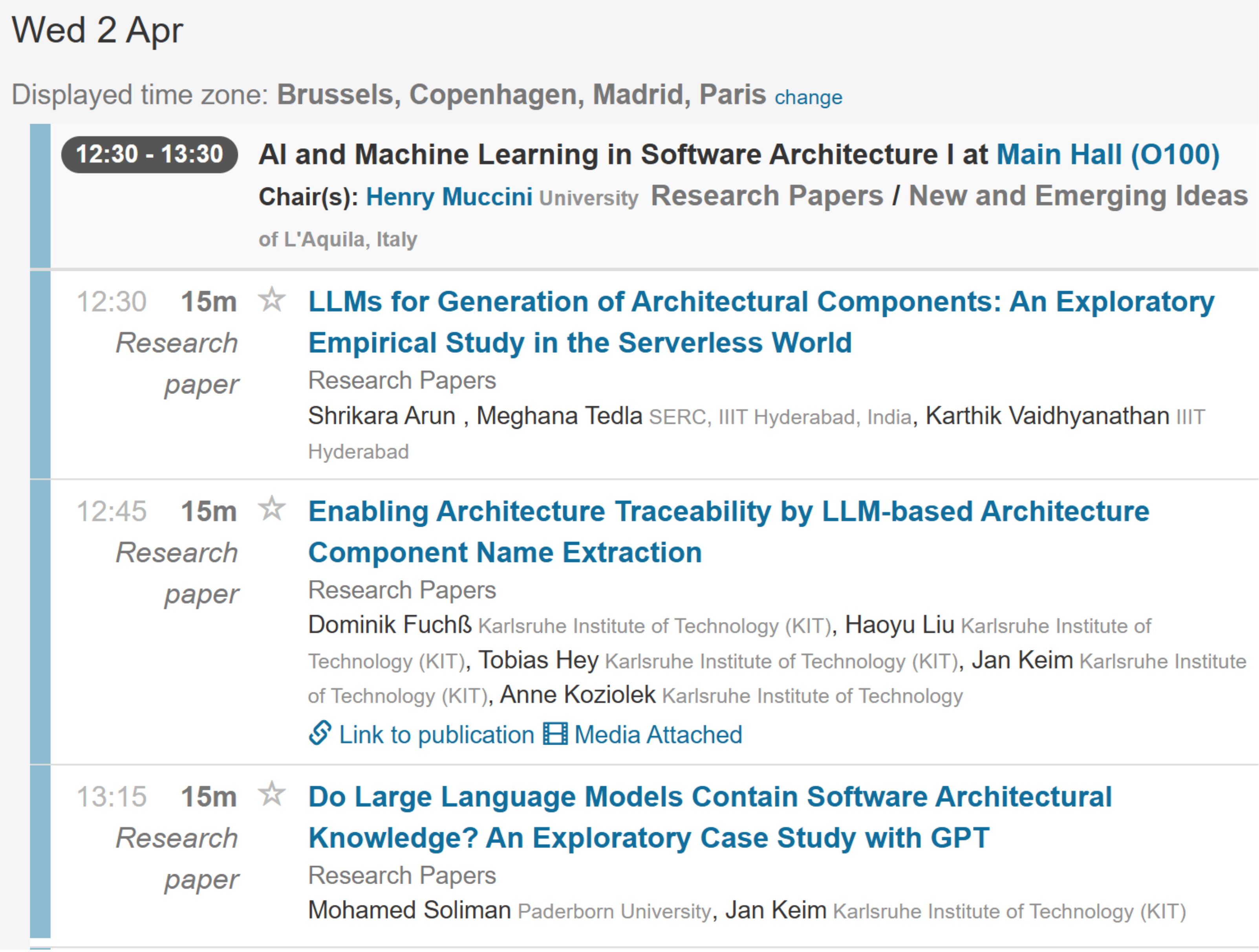
- Mapping between methods to 4 architectural component labels
  - Presentation [Pr], Application services [Ap], Domain services [Do] and Technical services [Te]
- Experiment data
  - A subset of methods of K-9 Mail



Manual classif.	Automated classification: Pr	Ap	Do	Te	Total
Pr	8	1	1	1	11
Ap	4	7	2	0	13
Do	0	0	13	0	13
Te	1	2	4	10	17


Reference architecture definition
1 Select reference architecture
2 Define architectural components
3 Define component & interaction indicators
Code units classification
4 Evaluate code units against indicators
5 Aggregate classified code units
Examples:
Pipes-and-Filters
Layered
MVC
App Services: controls flow
Presentation: provides UI
P1 sets UI attributes
P2 validates user input
P1
P2
AttachmentPresenter
✓
✓
MessageBuilder
×
×
App Services
Presentation

Existing Works on LLM4SA @ ICSA 2025



Wed 2 Apr

Displayed time zone: **Brussels, Copenhagen, Madrid, Paris** change

12:30 - 13:30 **AI and Machine Learning in Software Architecture I at Main Hall (O100)**
Chair(s): Henry Muccini University of L'Aquila, Italy **Research Papers / New and Emerging Ideas**

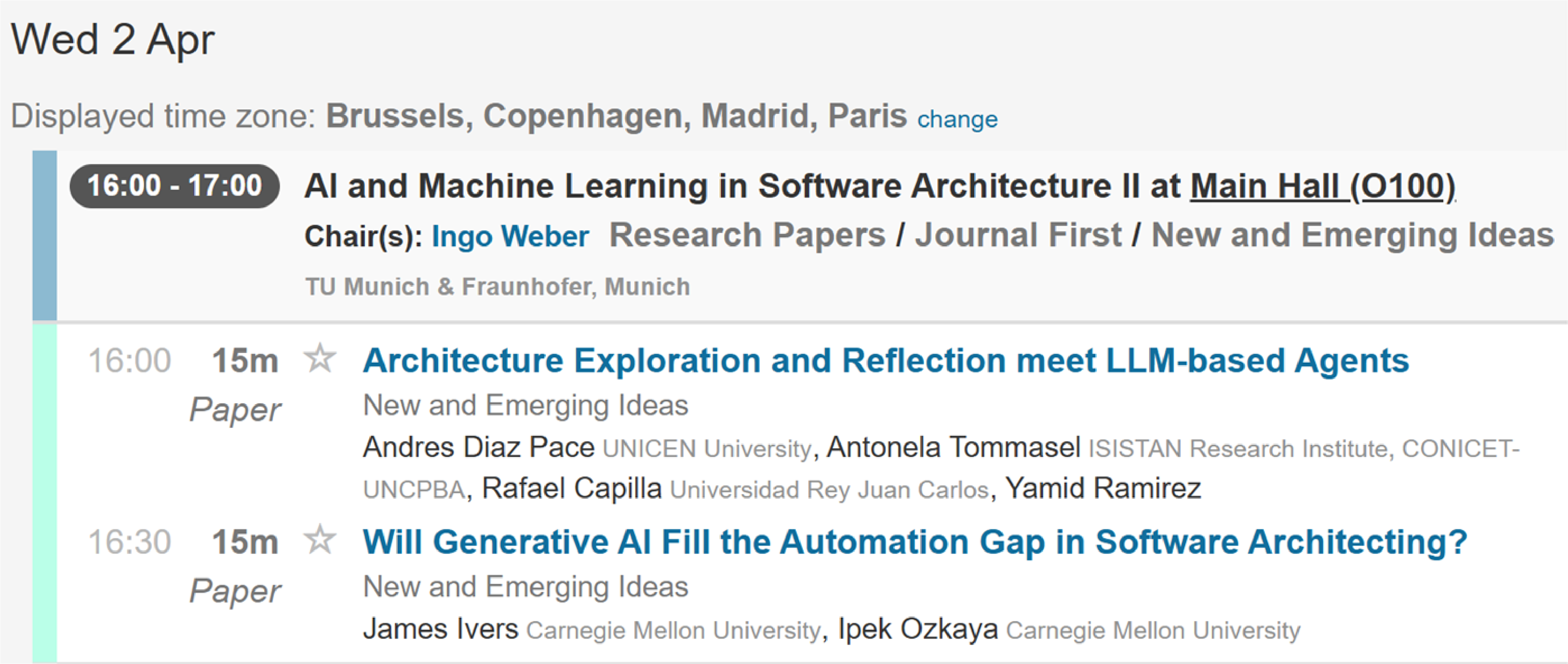
12:30 **15m** ☆ **LLMs for Generation of Architectural Components: An Exploratory Empirical Study in the Serverless World**
Research paper
Research Papers
Shrikara Arun , Meghana Tedla SERC, IIIT Hyderabad, India, Karthik Vaidhyanathan IIIT Hyderabad

12:45 **15m** ☆ **Enabling Architecture Traceability by LLM-based Architecture Component Name Extraction**
Research paper
Research Papers
Dominik Fuchß Karlsruhe Institute of Technology (KIT), Haoyu Liu Karlsruhe Institute of Technology (KIT), Tobias Hey Karlsruhe Institute of Technology (KIT), Jan Keim Karlsruhe Institute of Technology (KIT), Anne Koziolek Karlsruhe Institute of Technology
Link to publication Media Attached

13:15 **15m** ☆ **Do Large Language Models Contain Software Architectural Knowledge? An Exploratory Case Study with GPT**
Research paper
Research Papers
Mohamed Soliman Paderborn University, Jan Keim Karlsruhe Institute of Technology (KIT)

School of
Computing and
Information Systems

Existing Works on LLM4SA @ ICSA 2025



Wed 2 Apr

Displayed time zone: **Brussels, Copenhagen, Madrid, Paris** change

16:00 - 17:00 **AI and Machine Learning in Software Architecture II at Main Hall (O100)**

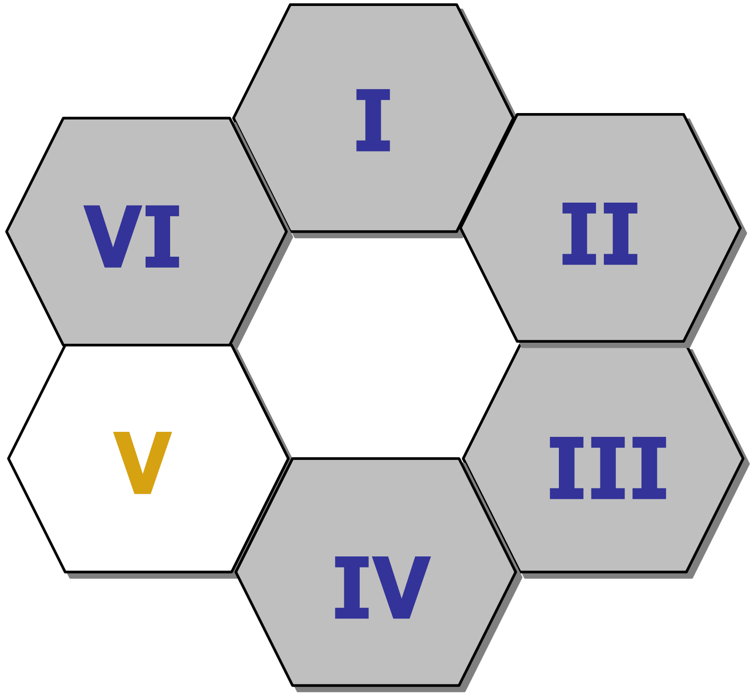
Chair(s): Ingo Weber TU Munich & Fraunhofer, Munich **Research Papers / Journal First / New and Emerging Ideas**

16:00 **15m** *Paper* ☆ **Architecture Exploration and Reflection meet LLM-based Agents**
New and Emerging Ideas
Andres Diaz Pace UNICEN University, Antonela Tommasel ISISTAN Research Institute, CONICET-UNCPBA, Rafael Capilla Universidad Rey Juan Carlos, Yamid Ramirez

16:30 **15m** *Paper* ☆ **Will Generative AI Fill the Automation Gap in Software Architecting?**
New and Emerging Ideas
James Ivers Carnegie Mellon University, Ipek Ozkaya Carnegie Mellon University

What open challenges are still to be solved?

Talk Structure



Self-Intro.

SE 2.0

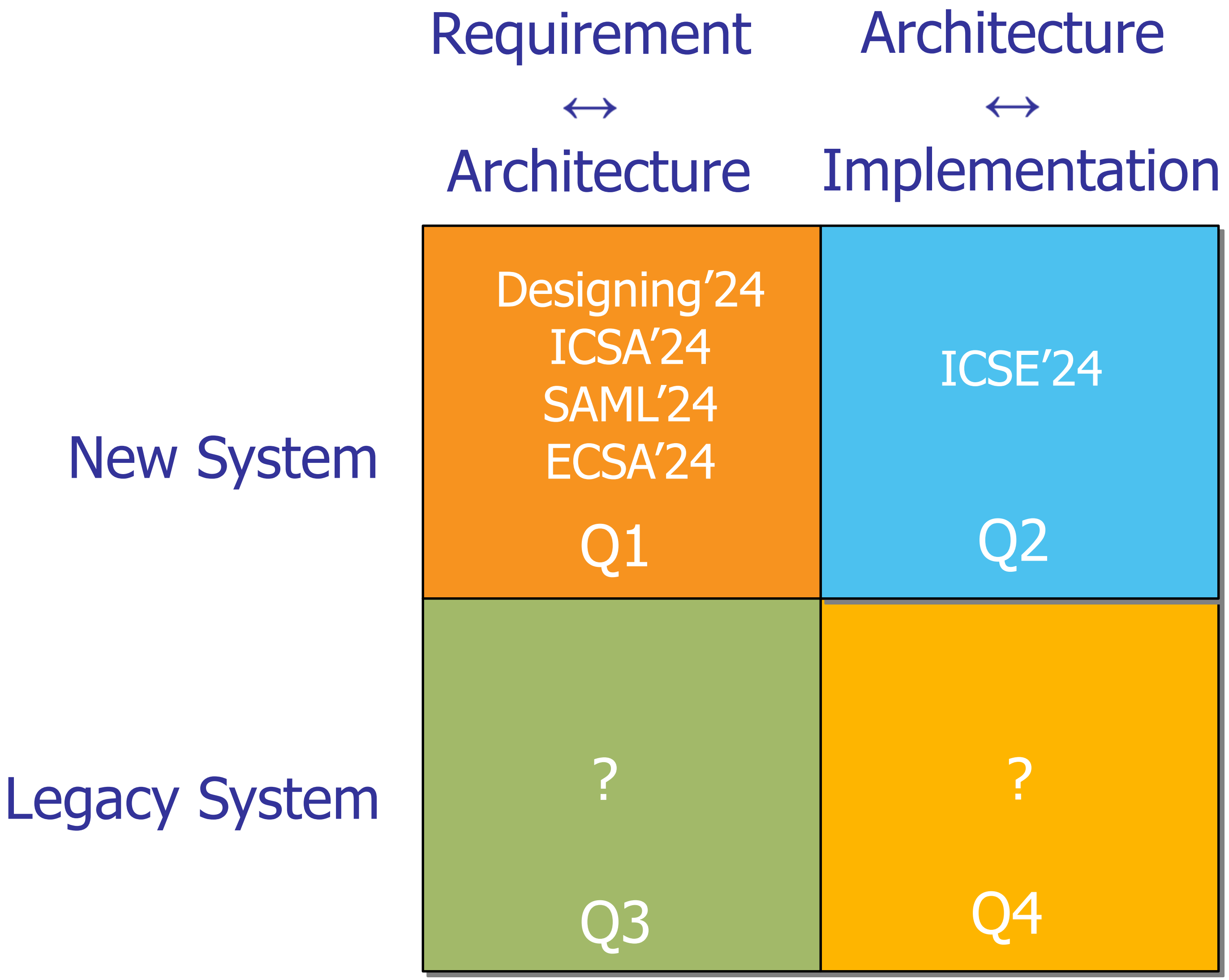
Software Architecture (SA) Challenges

LLM4SA Existing Work

LLM4SA Open Challenges

Realizing SE 2.0 & Implications to LLM4SA

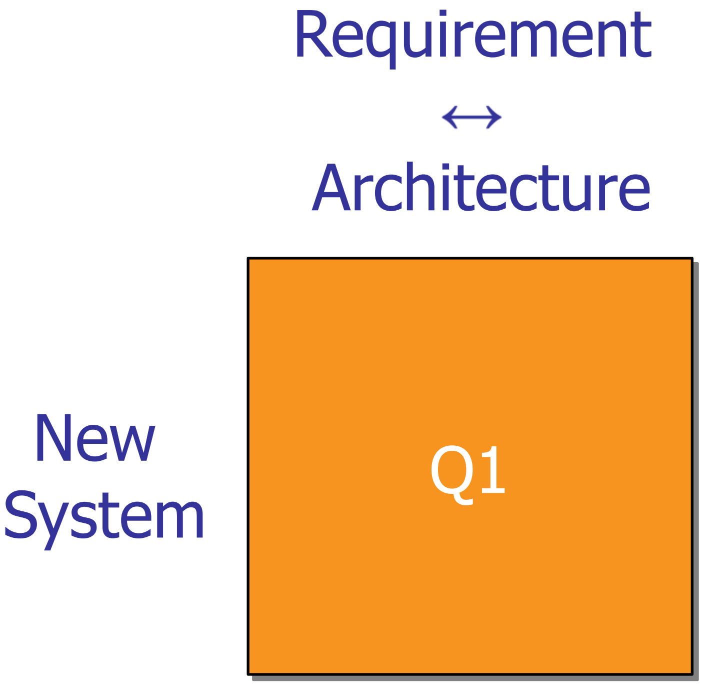
Open Challenges



	Requirement ↔ Architecture	Architecture ↔ Implementation
New System	Designing'24 ICSA'24 SAML'24 ECSA'24 Q1	ICSE'24 Q2
Legacy System	? Q3	? Q4

Not including new ICSA'25 papers — wouldn't want to steal their thunder 🙂

Quadrant 1: Existing Work (Summary)

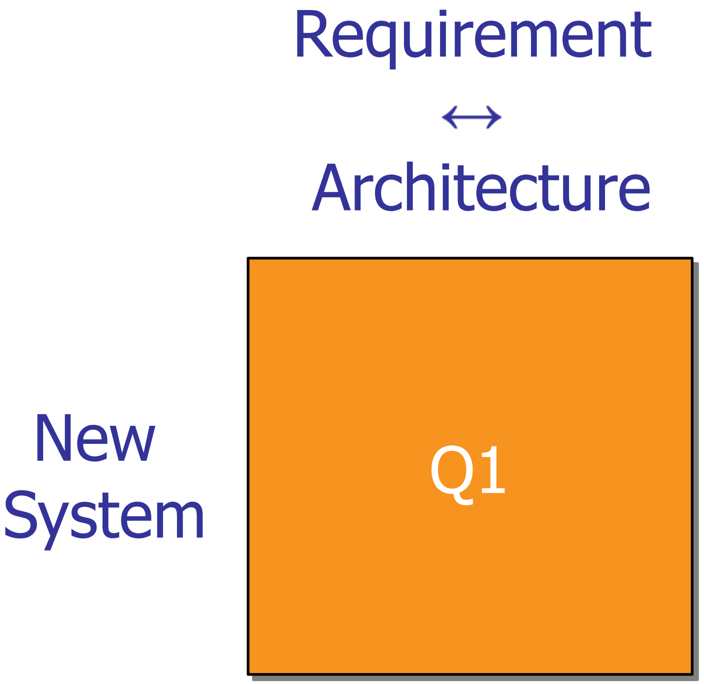


[C1] Completing architecture documentation

[Designing’24, SAML’24, ICSA’24]

- Architecture documentations:
  - Domain models [Designing’24]
  - C4 container solution diagrams [SAML’24]
  - Decisions in ADRs [ICSA’24]
- Evaluation: Examples, 95 ADRs [ICSA’24]

Quadrant 1: Existing Work (Summary)



[C2] Anticipating future requirements and tradeoffs

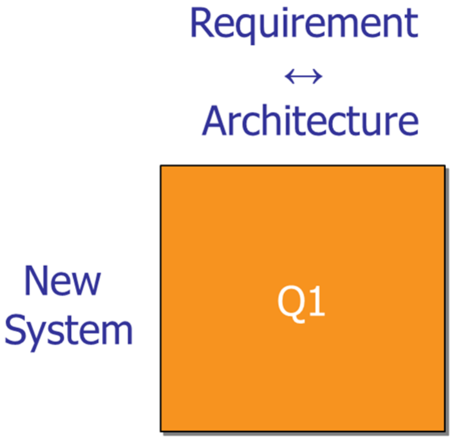
[ECSA'24]

- Focuses on patterns (design, microservice)
- Evaluation: 5 ADRs, Classroom

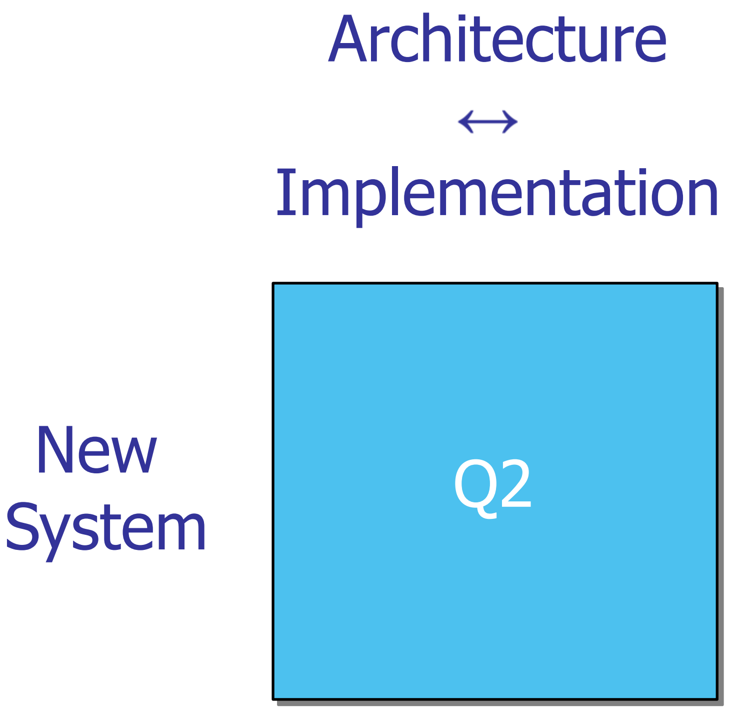
Quadrant 1: Future Possibility & Open Challenge

[FP1] A **robust** and intelligent LLM-powered agent that engages in an **active dialogue** to:

1. elicit and document *diverse and aligned* architectural decisions and diagrams, and
2. critique architectural decisions with *insightful* and sound reasoning about alternatives and *trends*



Quadrant 2: Existing Work (Summary)



[C3] Documenting traceability links between architecture and code

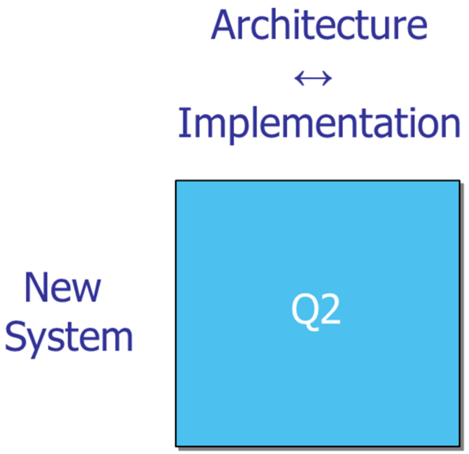
[ICSE’24]

- Link type: Methods ↔ Layer Type
- Evaluation: 184 methods from K9-Mail

Quadrant 2: Future Possibility & Open Challenge

[FP2] A robust & **integrated** LLM-powered agent that:

- *elicits*, infers, and documents linkages between source code and *various* architectural concepts and documents,
- as software engineers build a new system — in a *seamless* and *non-intrusive* manner that integrates well into their workflow



School of
Computing and
Information Systems

Quadrant 2: Existing Work (Summary)

Architecture
↔
Implementation

New
System

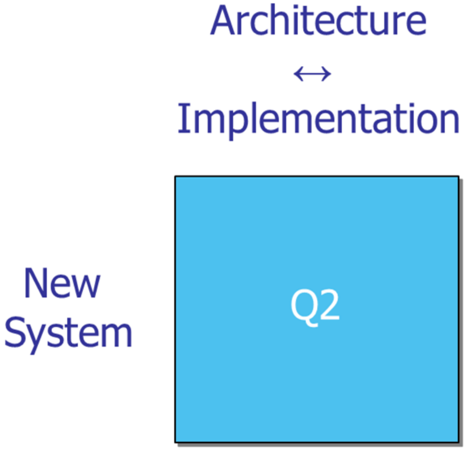
Q2

[C4] Ensuring architecture documentation aligns with implementation

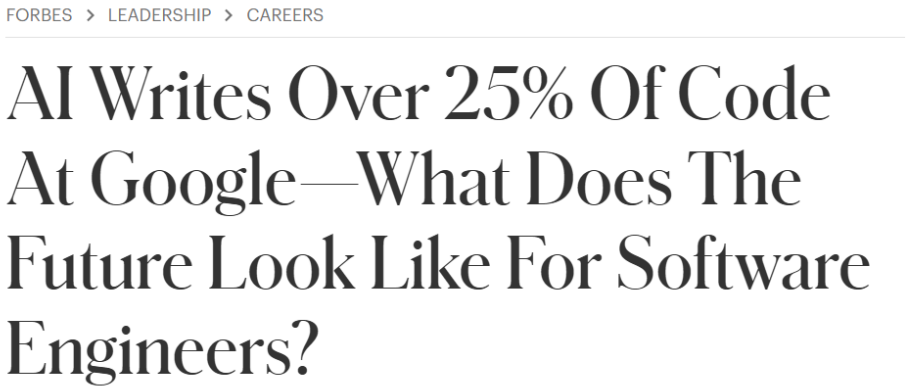
School of
Computing and
Information Systems

Quadrant 2: Future Possibility & Open Challenge

[FP3] An **explainable** LLM-powered agent that identify *divergences* between implementations and architectural documentations.



Quadrant 2: Future Possibility & Open Challenge



FORBES > LEADERSHIP > CAREERS

AI Writes Over 25% Of Code At Google—What Does The Future Look Like For Software Engineers?

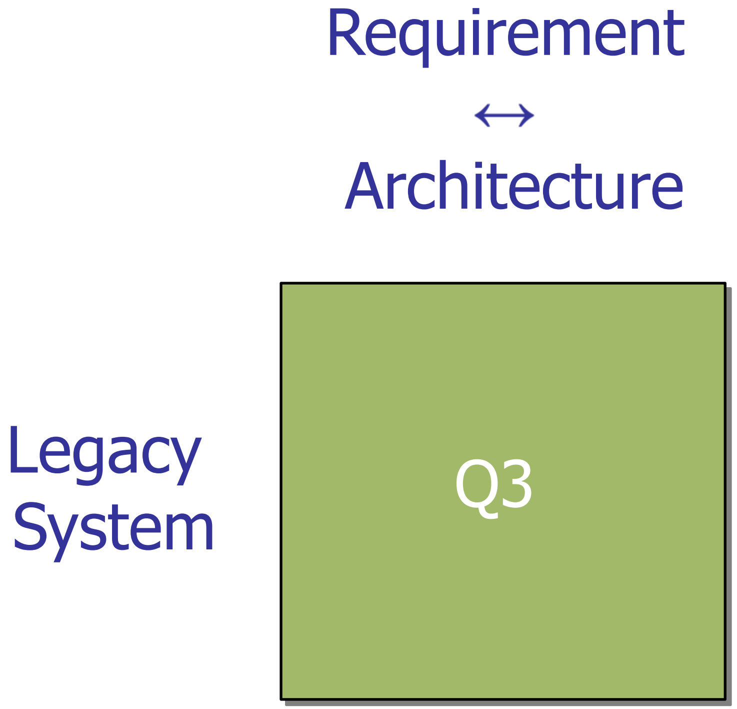
[FP4] A **code generation engine** that takes into consideration software architecture documentation and ensure **compliance by design**.

Architecture
↔
Implementation

Q2

New

Quadrant 3: Existing Work (Summary)


Requirement
↔
Architecture

Legacy
System

Q3

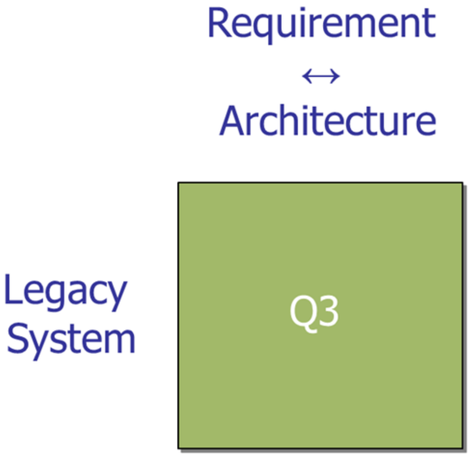
[C5] Consolidating architecture knowledge from scattered sources

School of
Computing and
Information Systems

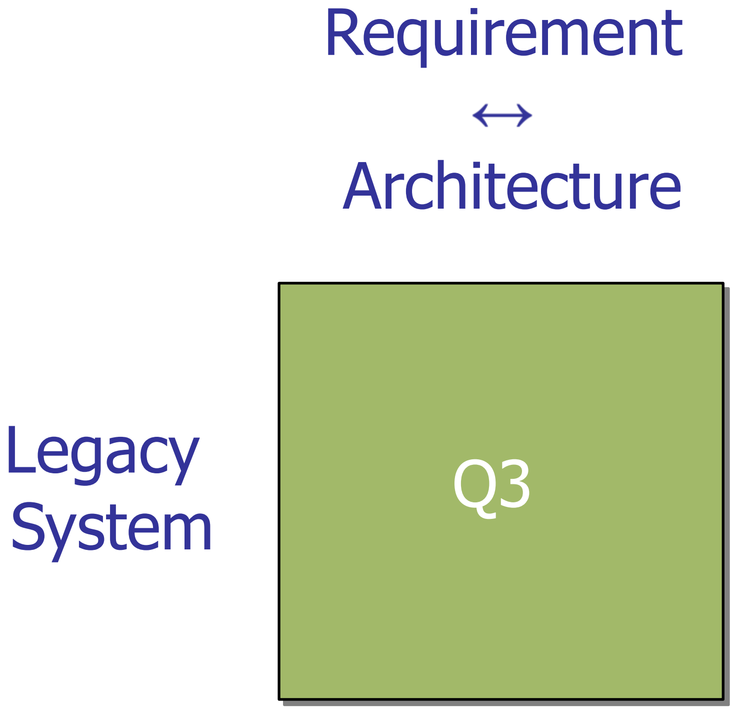
Quadrant 3: Future Possibility & Open Challenge

[FP5] An LLM-powered agent that:

- takes *diverse* architecture documents from a legacy systems, and
- generates well-formatted and **coherent** one,
- while identifies **inconsistencies**.



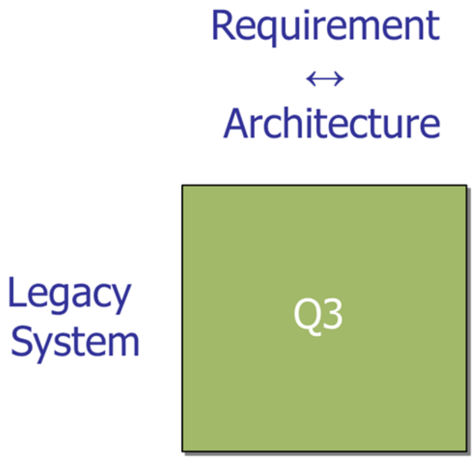
Quadrant 3: Existing Work (Summary)



[C6] Mapping and propagating business requirement changes to architectural changes

Quadrant 3: Future Possibility & Open Challenge

[FP6] An LLM-powered agent that takes **changes in business requirements** and suggests update to software architecture with *humans-in-the-loop*.



Quadrant 4: Existing Work (Summary)

[C7] Maintaining up-to-date architecture documentation

Architecture
↔
Implementation

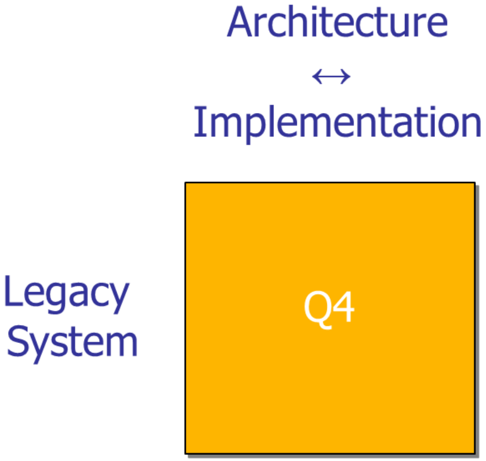
Legacy System

Q4

School of
Computing and
Information Systems

Quadrant 4: Future Possibility & Open Challenge

[FP7] An LLM-powered agent that takes **changes in source code** and suggests update to software architecture documentation.

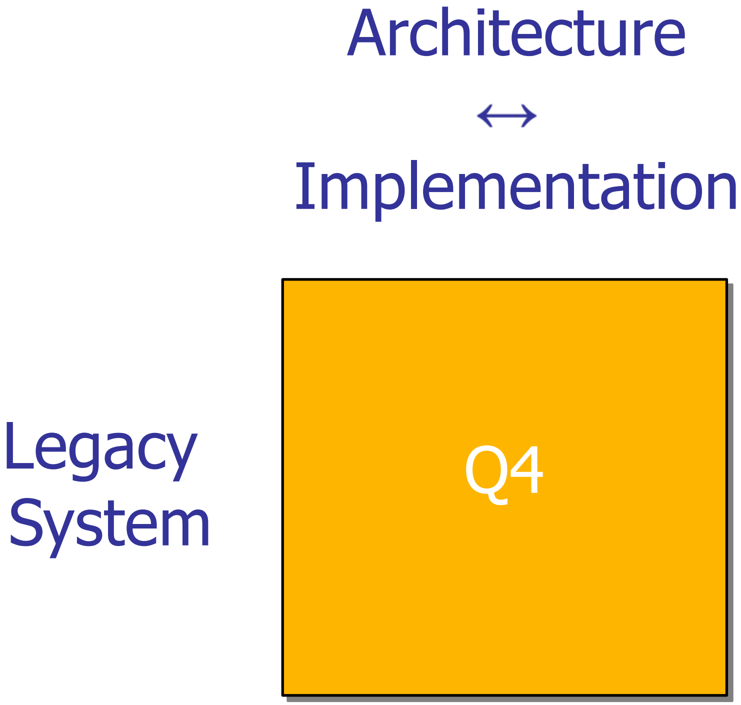

Architecture
↔
Implementation

Legacy System

Q4

School of
Computing and
Information Systems

Quadrant 4: Existing Work (Summary)



Architecture

↔

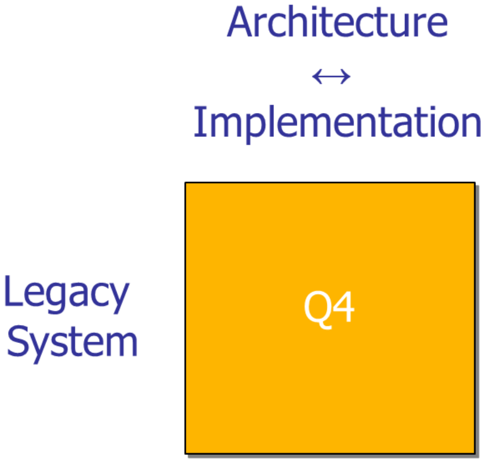
Implementation

Legacy System

[C8] Detecting and addressing architecture erosion

Quadrant 4: Future Possibility & Open Challenge

[FP8] An LLM-powered agent that **abstracts** low level code smells to *architectural erosion* and suggest *mitigation or fixing* actions.

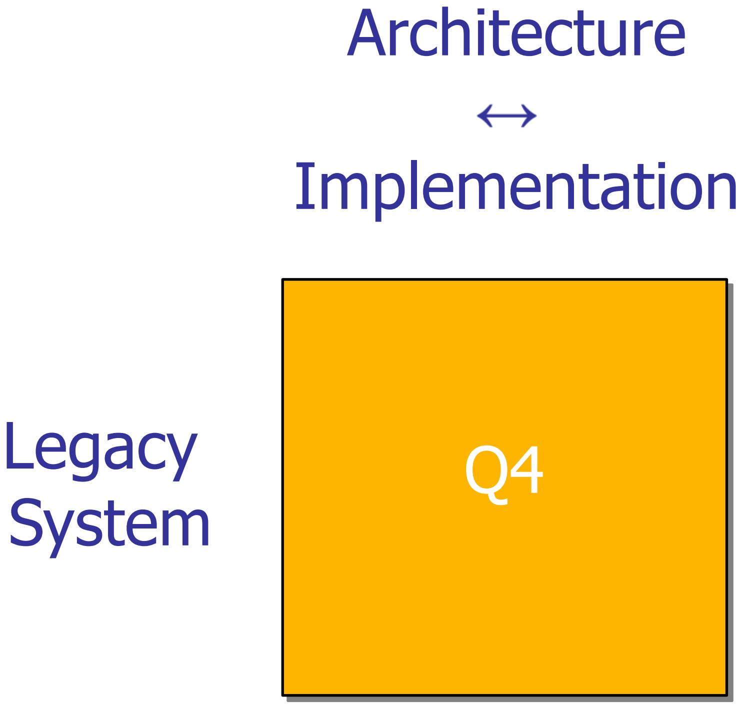

Architecture
↔
Implementation

Legacy System

Q4

School of
Computing and
Information Systems

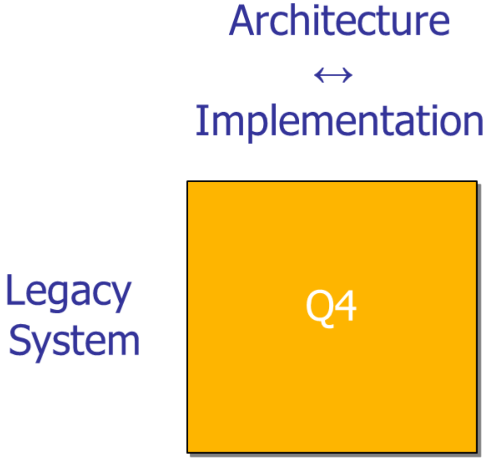
Quadrant 4: Existing Work (Summary)



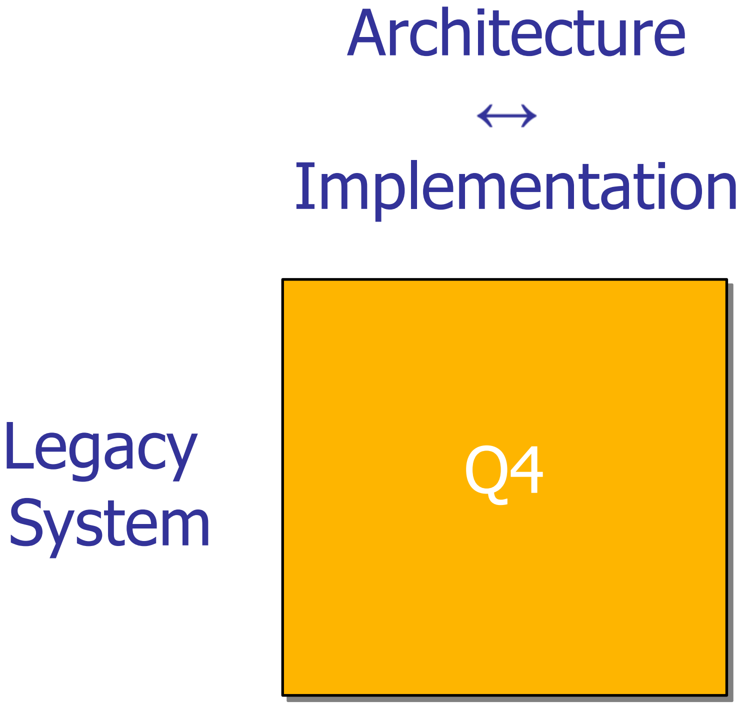
[C9] Analyzing the impact of architectural changes

Quadrant 4: Future Possibility & Open Challenge

[FP9] An LLM-powered agent that **assesses impacts** of architectural changes, and measures as well as **explains** *risks, costs and benefits*.



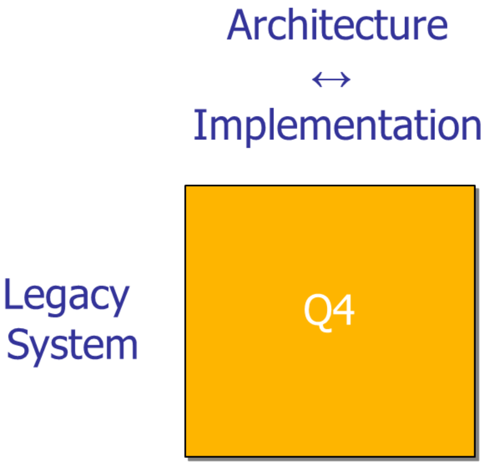
Quadrant 4: Existing Work (Summary)



[C10] Refactoring architecture in a cost-effective manner

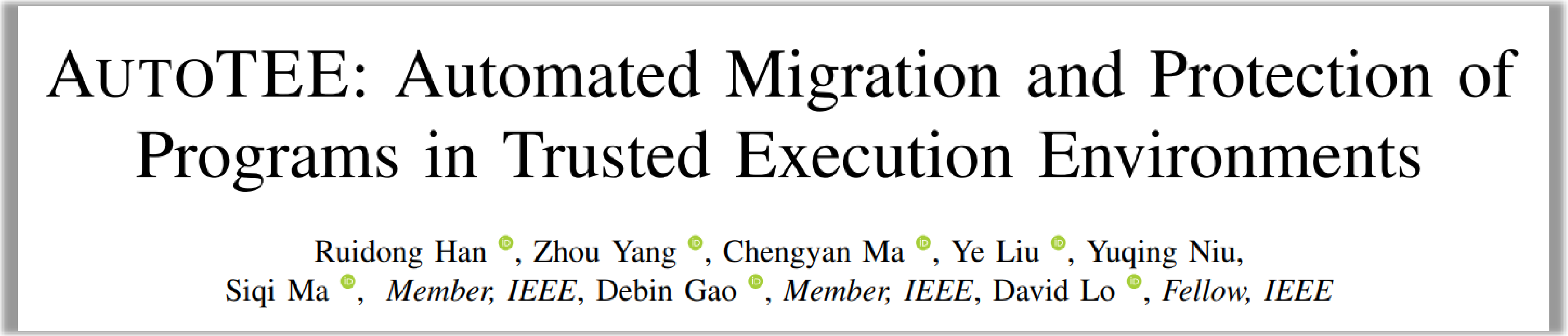
Quadrant 4: Future Possibility & Open Challenge

[FP10] An LLM-powered agent that *trustworthily* **refactors** a software project to upgrade a legacy architecture.



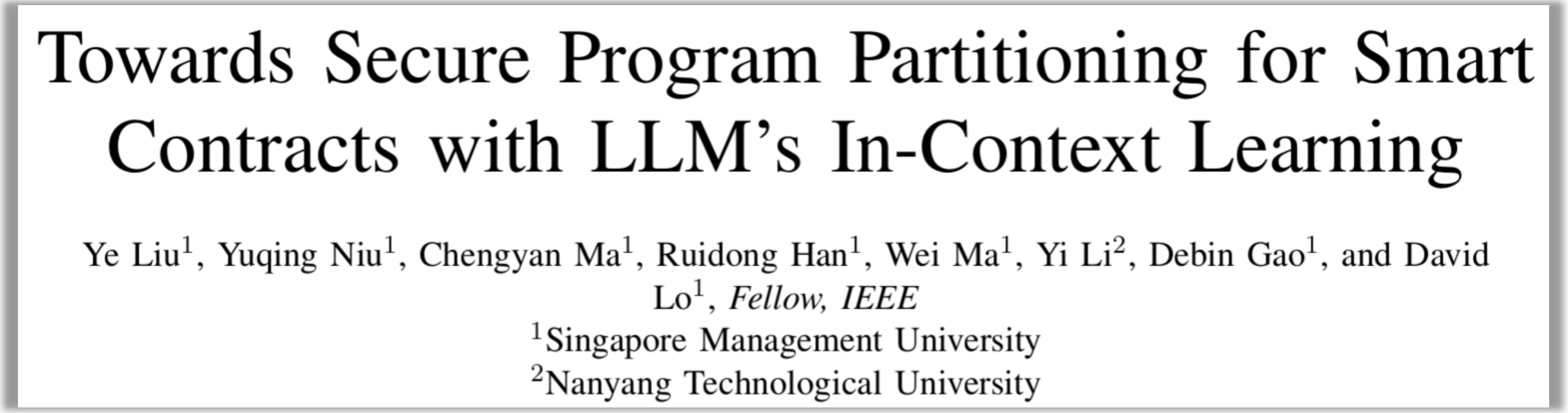
School of
Computing and
Information Systems

Quadrant 4: Future Possibility & Open Challenge



AUTOTEE: Automated Migration and Protection of Programs in Trusted Execution Environments

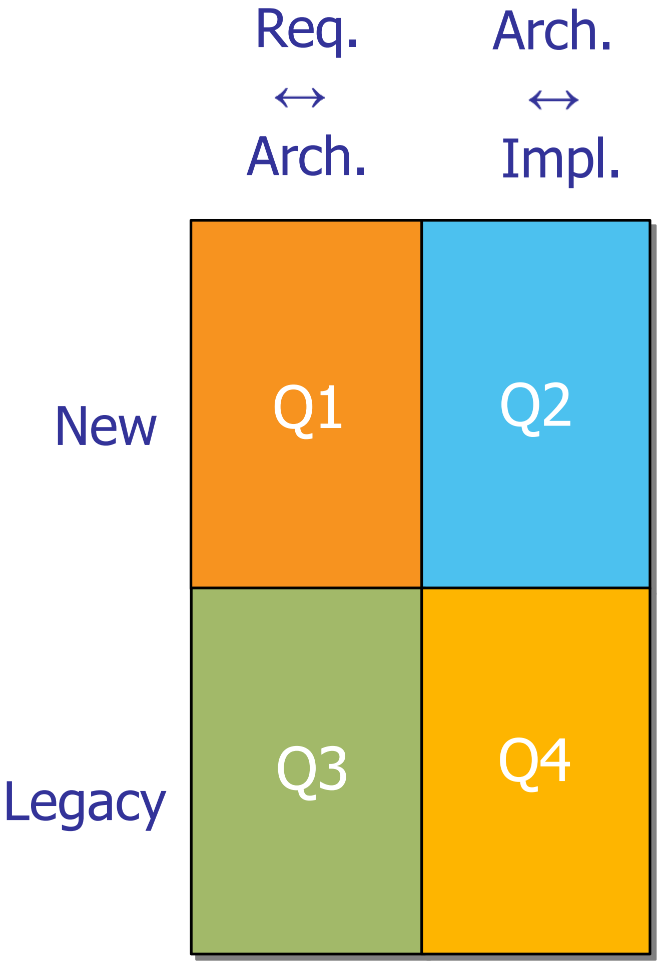
Ruidong Han, Zhou Yang, Chengyan Ma, Ye Liu, Yuqing Niu,
Siqi Ma, *Member, IEEE*, Debin Gao, *Member, IEEE*, David Lo, *Fellow, IEEE*



Towards Secure Program Partitioning for Smart Contracts with LLM's In-Context Learning

Ye Liu[1], Yuqing Niu[1], Chengyan Ma[1], Ruidong Han[1], Wei Ma[1], Yi Li[2], Debin Gao[1], and David Lo[1], *Fellow, IEEE*
[1]Singapore Management University
[2]Nanyang Technological University

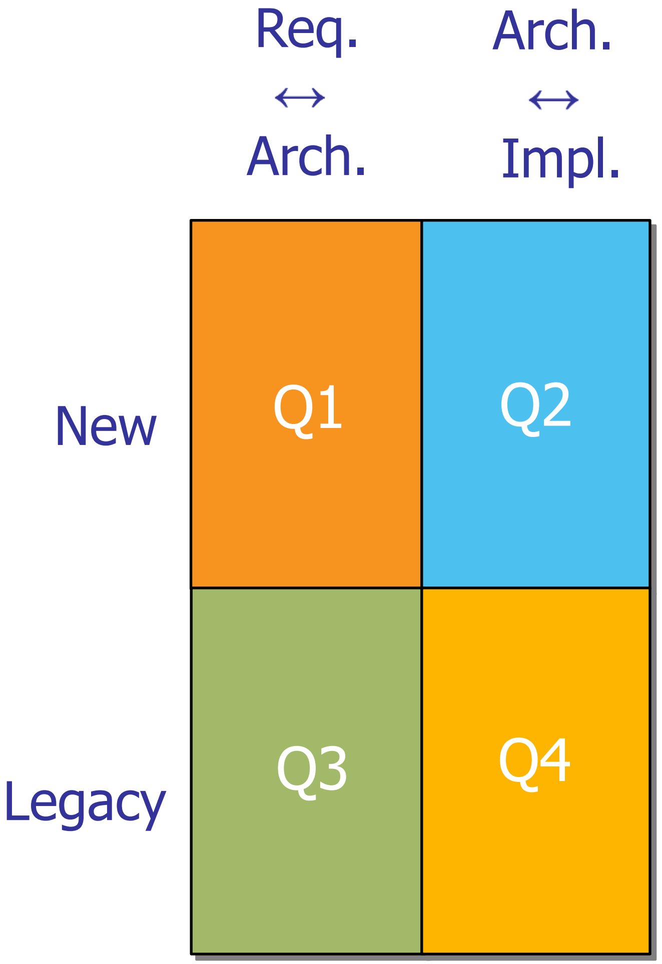
Ten Future Possibilities and Open Challenges



	Req. ↔ Arch.	Arch. ↔ Impl.
New	Q1	Q2
Legacy	Q3	Q4

An LLM-powered agent that …
[FP1] engages in an active dialogue to elicit and document diverse and aligned architectural decisions and diagrams, and critique architectural decisions with insightful and sound reasoning about alternatives and trends
[FP2] elicits, infers, and documents linkages between source code and various architectural concepts and documents, as software engineers build a new system — in a seamless and non-intrusive manner that integrates well into their workflow
[FP3] explains divergences (if any) between implementations and architectural documentations
[FP4] generates code considering software architecture documentation and ensure compliance by design

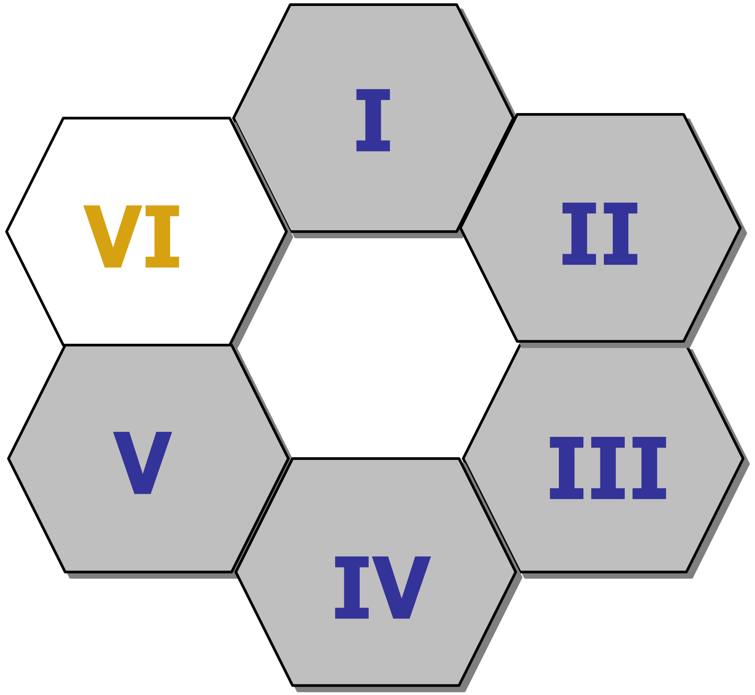
Ten Future Possibilities and Open Challenges



An LLM-powered agent that …
[FP5] takes diverse architecture documents from a legacy systems, and generates well-formatted and coherent one, while identifies inconsistencies
[FP6] takes changes in business requirements and suggests update to software architecture with humans-in-the-loop
[FP7] takes changes in source code and suggests update to software architecture documentation
[FP8] abstracts low level code smells to architectural erosion and suggest mitigation or fixing actions
[FP9] assesses impacts of architectural changes, and measures as well as explains risks, costs and benefits
[FP10] refactors a software project to upgrade a legacy architecture

How to solve these open challenges?

Talk Structure



Self-Intro.

SE 2.0

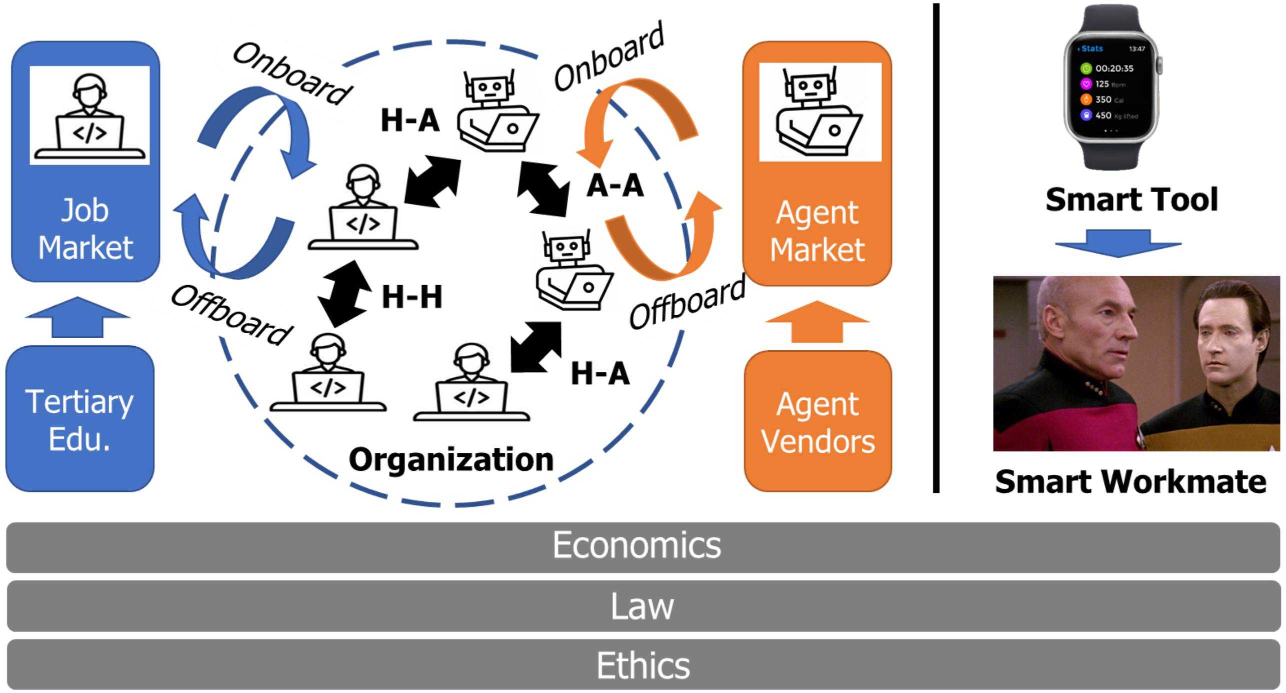
Software Architecture
(SA) Challenges

LLM4SA
Existing Work

LLM4SA
Open Challenges

Realizing SE 2.0 & Implications to LLM4SA

Challenges for AI4SE towards SE 2.0



SE 2.0
Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

Practitioners' Needs

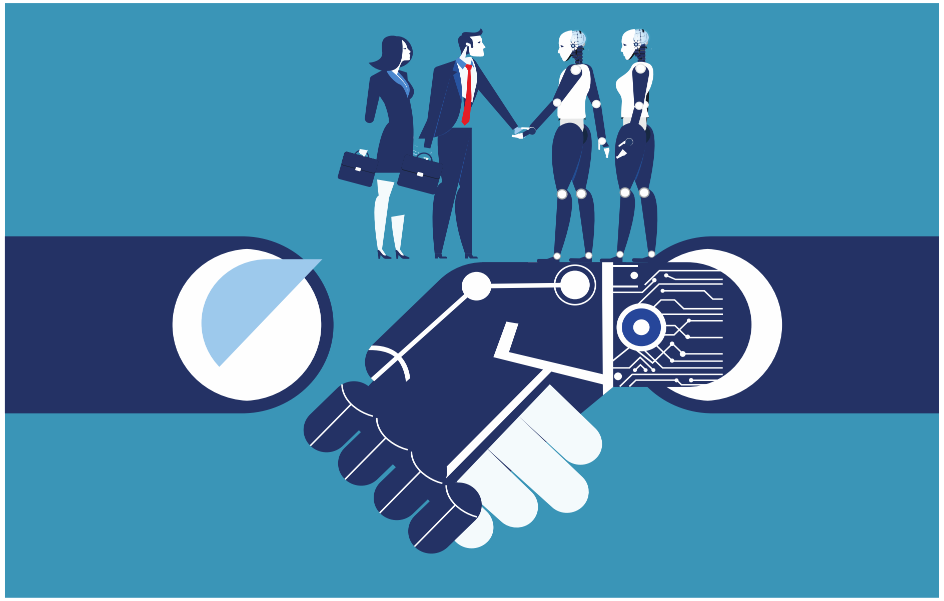
Gap

AI4SE Capabilities

Realizing SE 2.0: Human-Centered AI4SE

- Putting humans (software practitioners) **central** in the design of AI4SE tools
    - to augment and amplify their capabilities

- Without considering humans, it is easy for AI4SE to:
    - Solve issues that practitioners do not or no longer care about
    - Hinder practitioners in their work
    - Introduce unacceptable side-effects



Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

Synergize

Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

Synergize

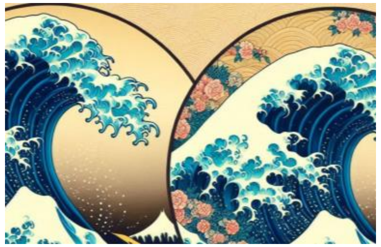
Listen: Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters?
- Are there special aspects to consider?

What **thresholds** need to be met for adoption?

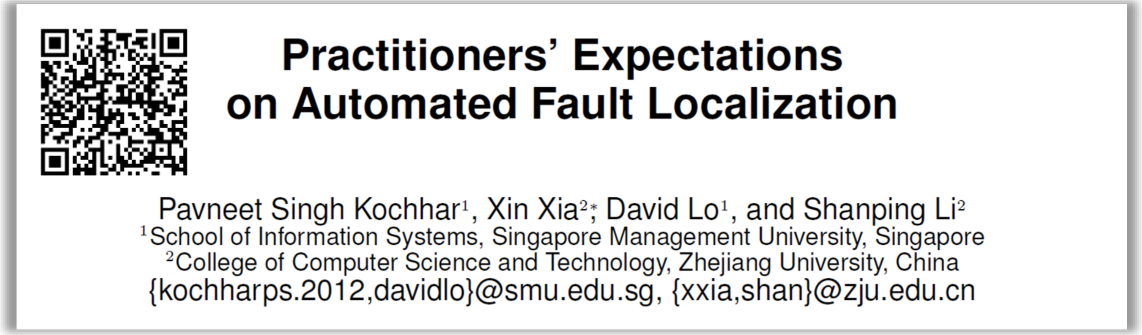
- What efficacy requirements must a tool meet before it can be considered helpful?



Are there **new** *problems* when tech. changes?

Practitioners’ Perceptions to SE Research

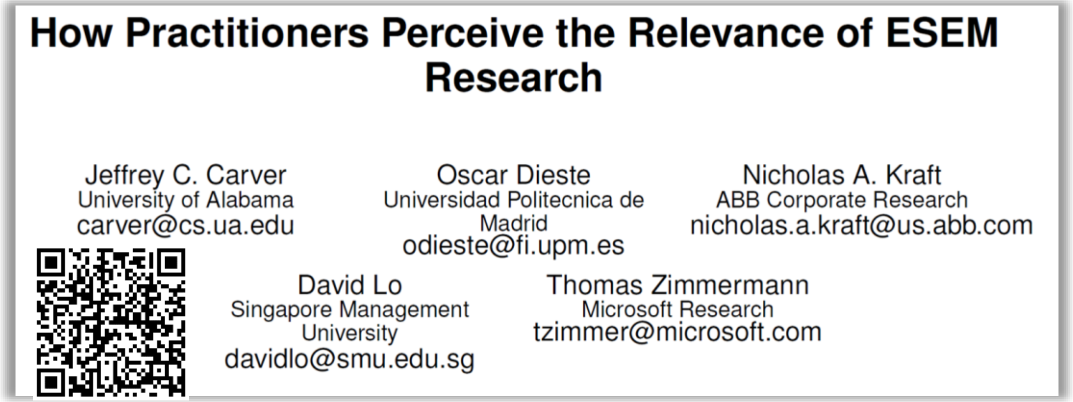
ISSTA 2016



Practitioners’ Expectations on Automated Fault Localization

Pavneet Singh Kochhar[1], Xin Xia[2]*, David Lo[1], and Shanping Li[2]
[1]School of Information Systems, Singapore Management University, Singapore
[2]College of Computer Science and Technology, Zhejiang University, China
{kochharps.2012,davidlo}@smu.edu.sg, {xxia,shan}@zju.edu.cn

ESEM 2016



How Practitioners Perceive the Relevance of ESEM Research

Jeffrey C. Carver
University of Alabama
carver@cs.ua.edu

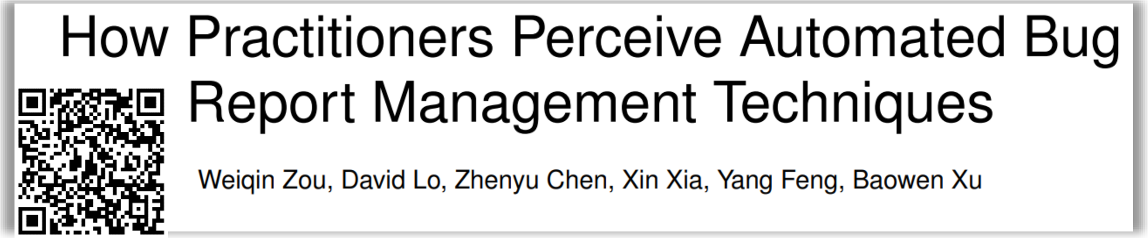
Oscar Dieste
Universidad Politecnica de Madrid
odieste@fi.upm.es

Nicholas A. Kraft
ABB Corporate Research
nicholas.a.kraft@us.abb.com

David Lo
Singapore Management University
davidlo@smu.edu.sg

Thomas Zimmermann
Microsoft Research
tzimmer@microsoft.com

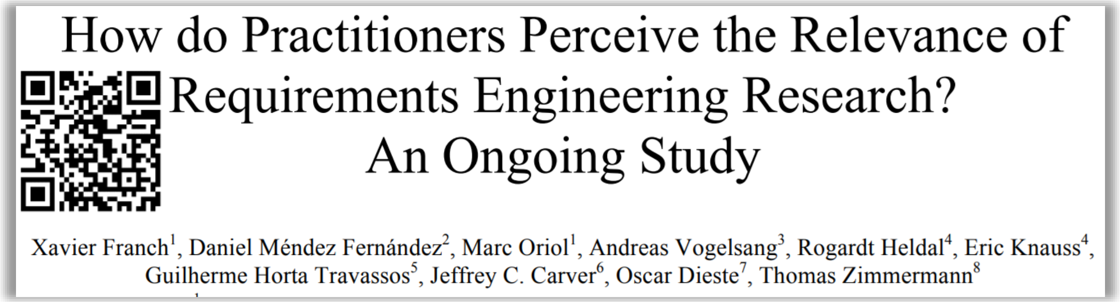
TSE 2020



How Practitioners Perceive Automated Bug Report Management Techniques

Weiqin Zou, David Lo, Zhenyu Chen, Xin Xia, Yang Feng, Baowen Xu

RE 2017

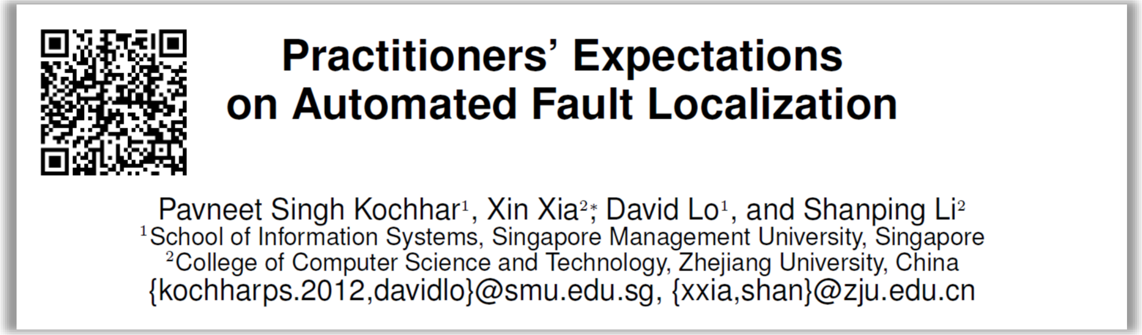


How do Practitioners Perceive the Relevance of Requirements Engineering Research? An Ongoing Study

Xavier Franch[1], Daniel Méndez Fernández[2], Marc Oriol[1], Andreas Vogelsang[3], Rogardt Heldal[4], Eric Knauss[4], Guilherme Horta Travassos[5], Jeffrey C. Carver[6], Oscar Dieste[7], Thomas Zimmermann[8]

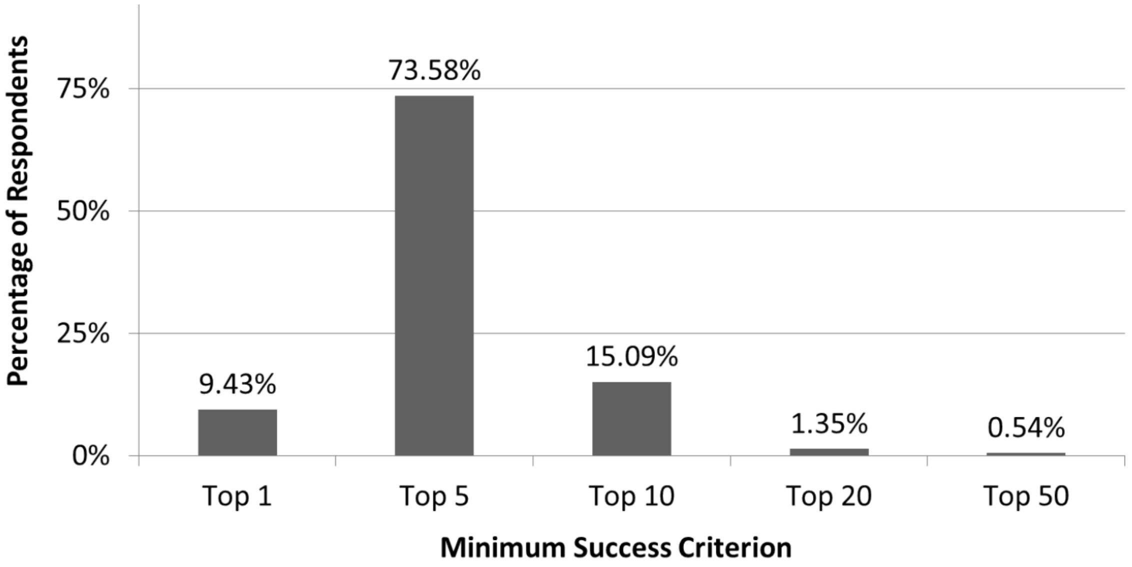
Practitioners’ Thresholds for Adoption

ISSTA 2016

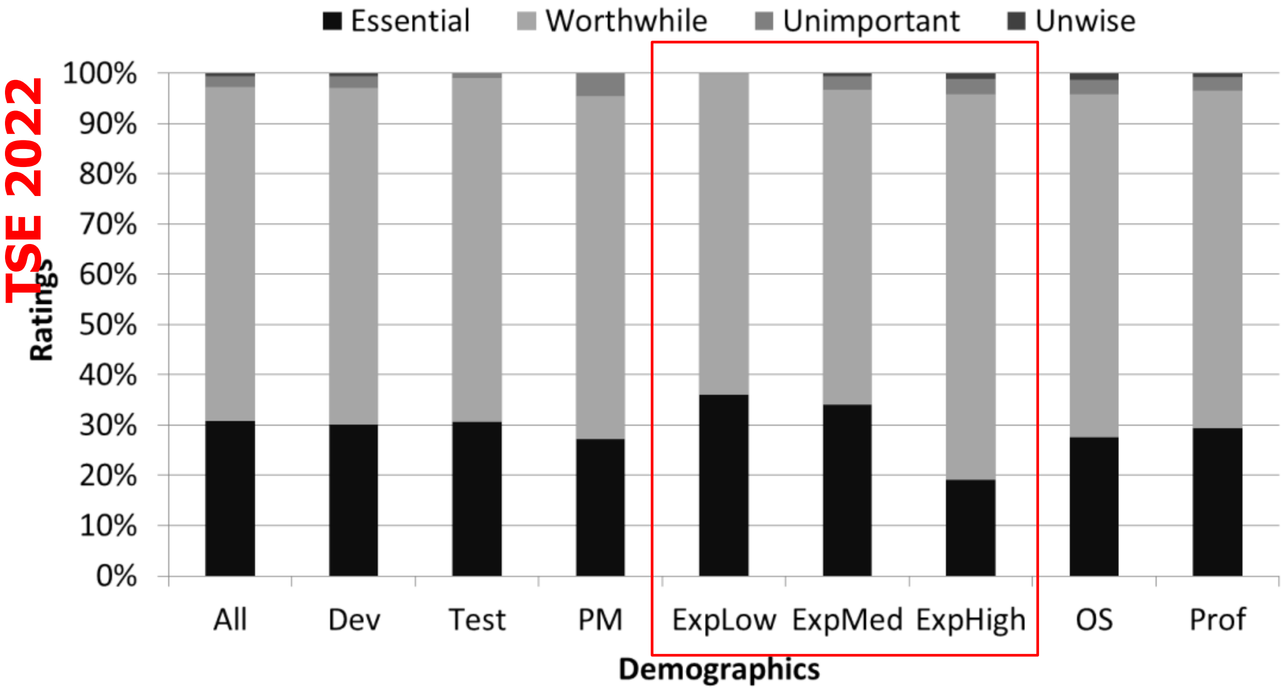


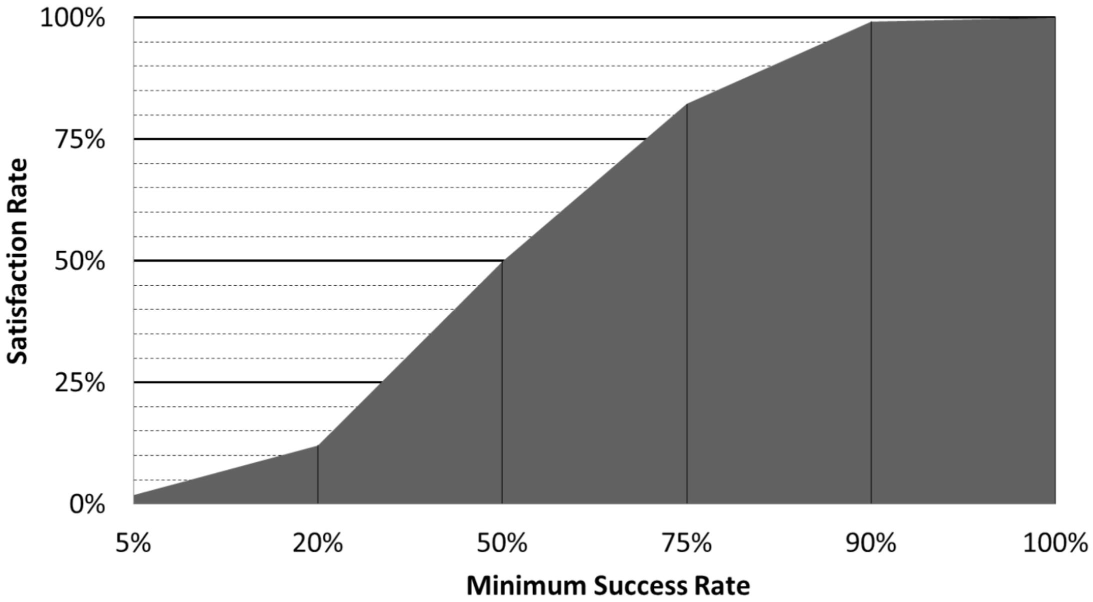
Practitioners’ Expectations on Automated Fault Localization

Pavneet Singh Kochhar[1], Xin Xia[2]*, David Lo[1], and Shanping Li[2]
[1]School of Information Systems, Singapore Management University, Singapore
[2]College of Computer Science and Technology, Zhejiang University, China
{kochharps.2012,davidlo}@smu.edu.sg, {xxia,shan}@zju.edu.cn

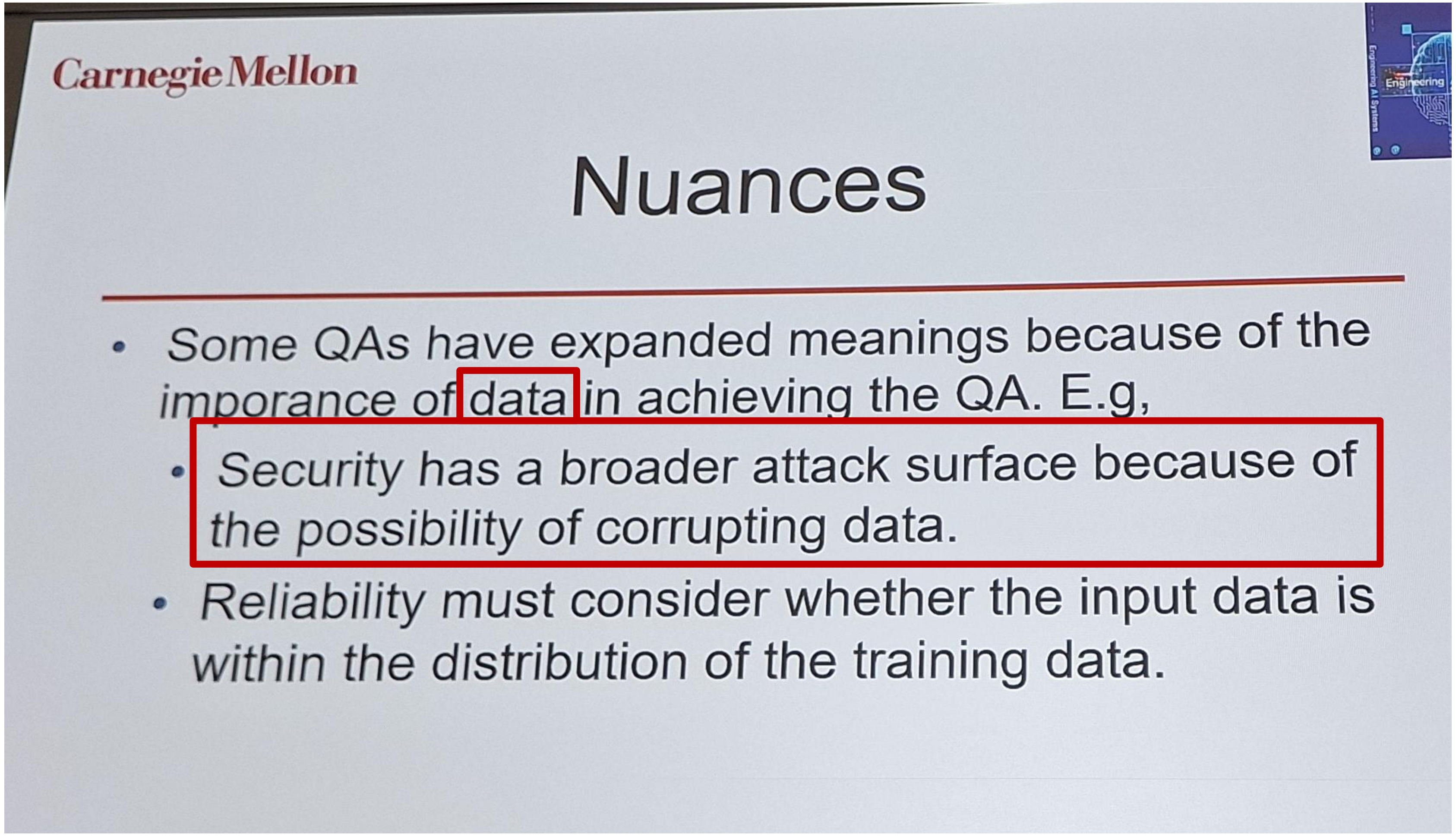


TSE 2022





New Problems / New Requirements

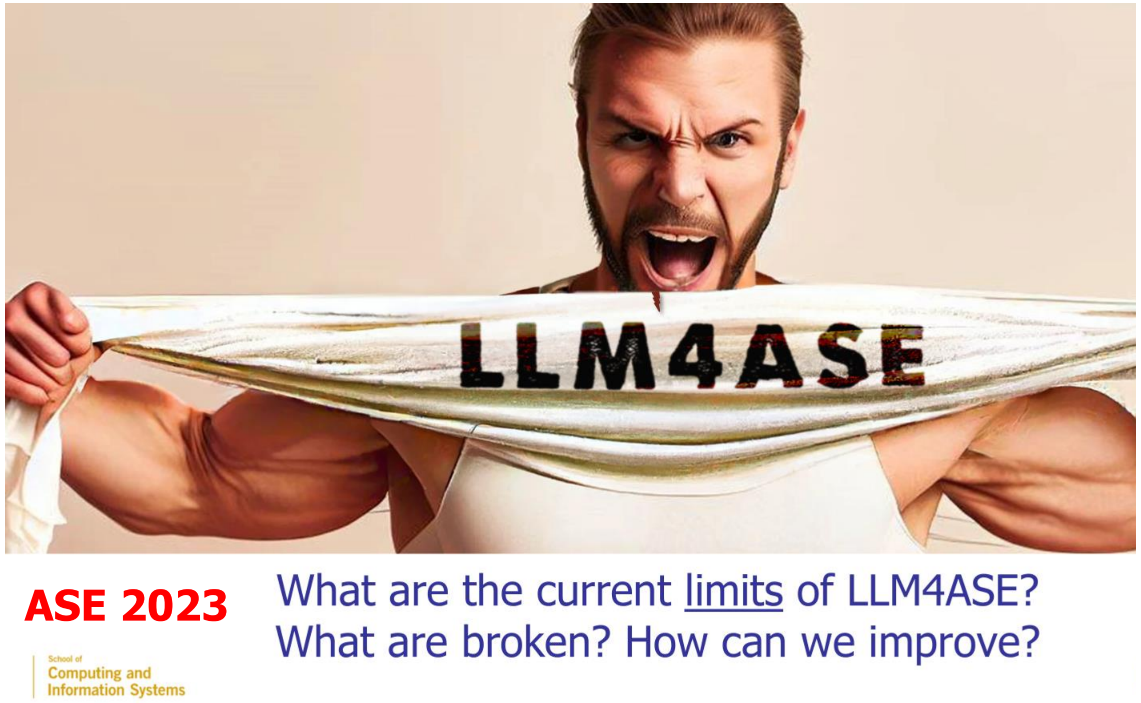


Carnegie Mellon

Nuances

- Some QAs have expanded meanings because of the imporance of data in achieving the QA. E.g,
  - Security has a broader attack surface because of the possibility of corrupting data.
  - Reliability must consider whether the input data is within the distribution of the training data.

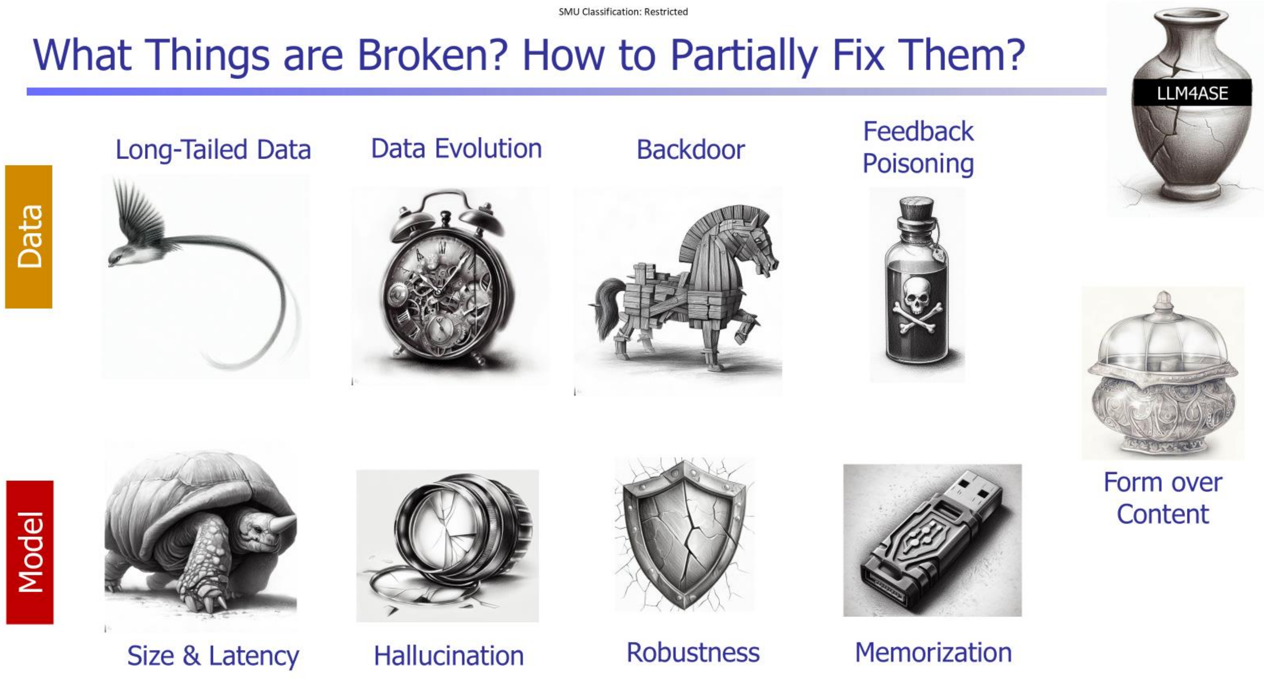
New Problems / New Requirements



ASE 2023

What are the current limits of LLM4ASE?
What are broken? How can we improve?

School of
Computing and
Information Systems



SMU Classification: Restricted

What Things are Broken? How to Partially Fix Them?

Data
- Long-Tailed Data
- Data Evolution
- Backdoor
- Feedback Poisoning

Model
- Size & Latency
- Hallucination
- Robustness
- Memorization

Form over Content

7

TSE 2024

Stealthy Backdoor Attack for Code Models

Zhou Yang, Bowen Xu, Jie M. Zhang, Hong Jin Kang, Jieke Shi, Junda He, and David Lo *Fellow, IEEE*

ICSE 2022

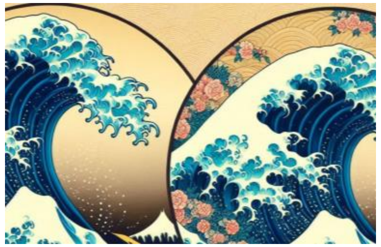
Natural Attack for Pre-trained Models of Code

Zhou Yang, Jieke Shi, Junda He and David Lo
School of Computing and Information Systems
Singapore Management University
{zyang,jiekeshi,jundahe,davidlo}@smu.edu.sg

School of
Computing and
Information Systems

Perception

Thresholds



New Probs.

LLM for Software Architecture?

Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

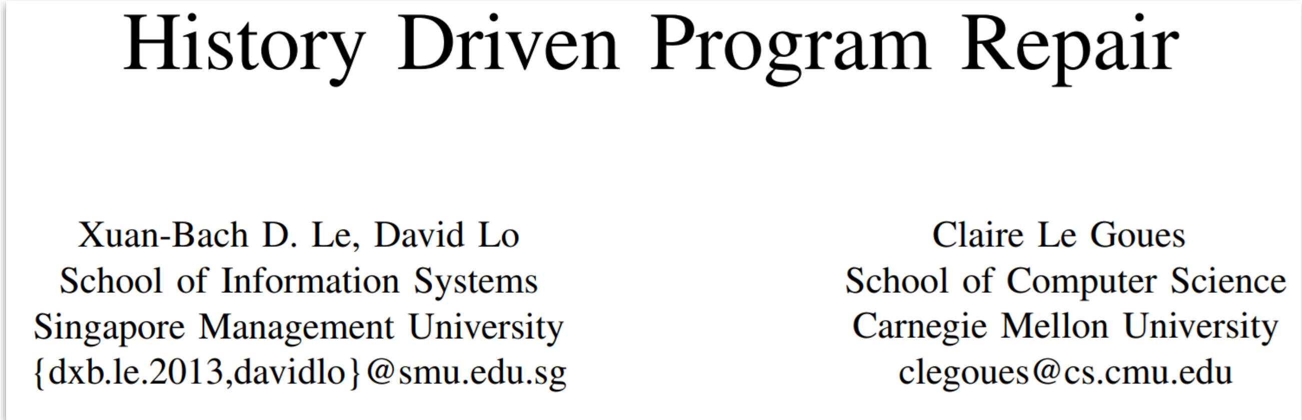
Synergize

Goals

- Able to learn from humans well to build more intelligent AI4SE
- AI4SE can:
  - Learn from (past) human decisions
  - Learn from human thinking process
    - i.e., learn like humans

Example: Learning from Past Human Decisions

SANER 2016



History Driven Program Repair

Xuan-Bach D. Le, David Lo
School of Information Systems
Singapore Management University
{dxb.le.2013,davidlo}@smu.edu.sg

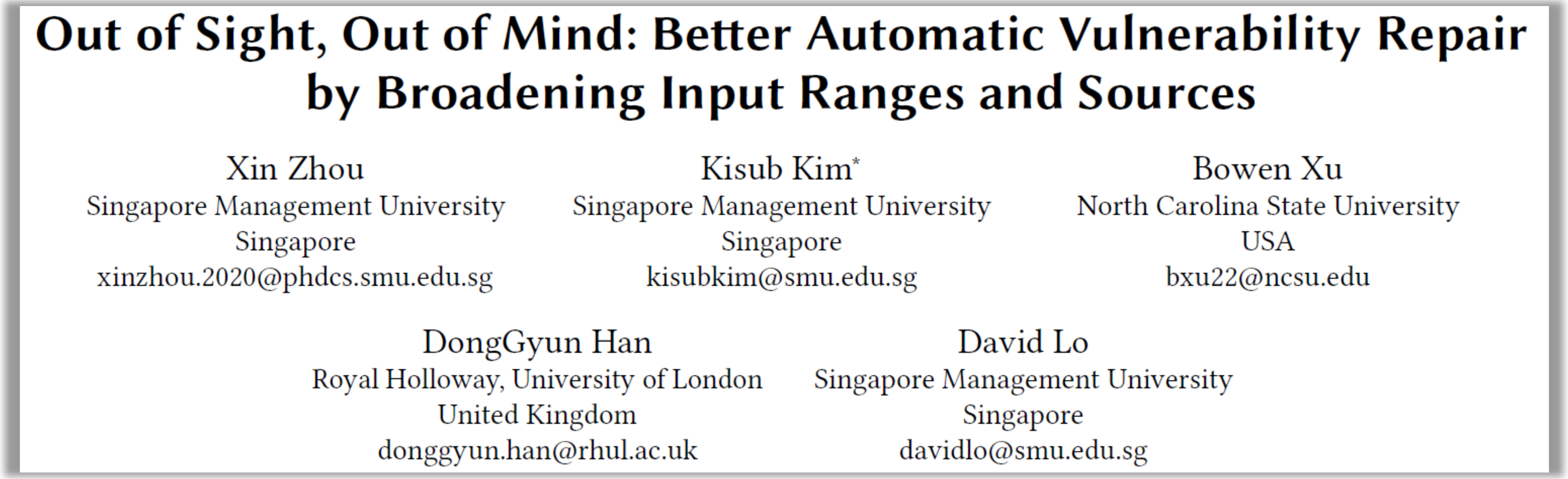
Claire Le Goues
School of Computer Science
Carnegie Mellon University
clegoues@cs.cmu.edu

First automated program repair solution that learns
from large number of fixed mistakes in large number of repos;
Most Cited Research Paper of SANER 2016

"***History driven program repair** as investigated by Le et al. **influenced** our work, the overall diffing/mining pipeline is similar."* – Facebook Engineer

Example: Learning from Human Learning Process

ICSE 2024



Out of Sight, Out of Mind: Better Automatic Vulnerability Repair by Broadening Input Ranges and Sources

Xin Zhou
Singapore Management University
Singapore
xinzhou.2020@phdcs.smu.edu.sg

Kisub Kim*
Singapore Management University
Singapore
kisubkim@smu.edu.sg

Bowen Xu
North Carolina State University
USA
bxu22@ncsu.edu

DongGyun Han
Royal Holloway, University of London
United Kingdom
donggyun.han@rhul.ac.uk

David Lo
Singapore Management University
Singapore
davidlo@smu.edu.sg

*First multi-LLM collaboration mimicking human learning for vulnerability repair; lightweight model, achieving **2x** efficacy*

Learn
(from & like)
Humans

LLM for Software Architecture?

Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

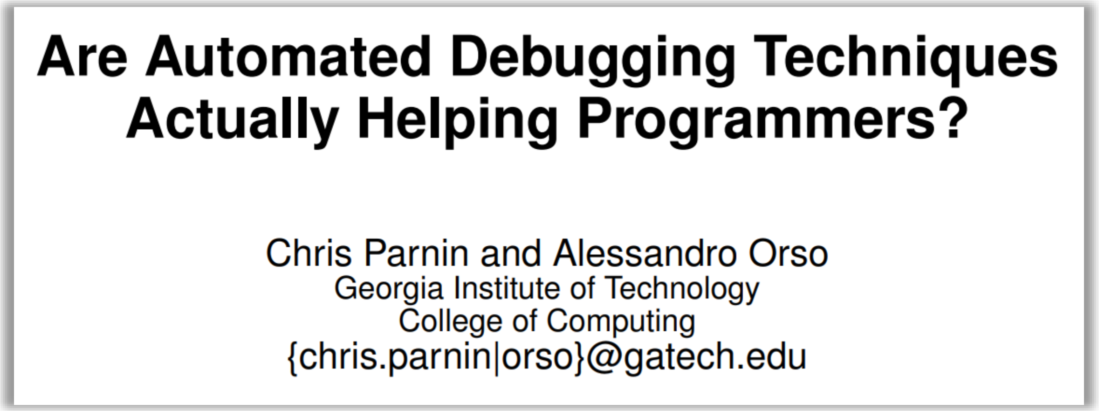
Synergize

Need for Synergy

*“the interaction or **cooperation** of two or more … agents to produce a **combined effect greater than the sum of their separate effects**”* - Oxford

Human & intelligent agent
may not work well together

1:1



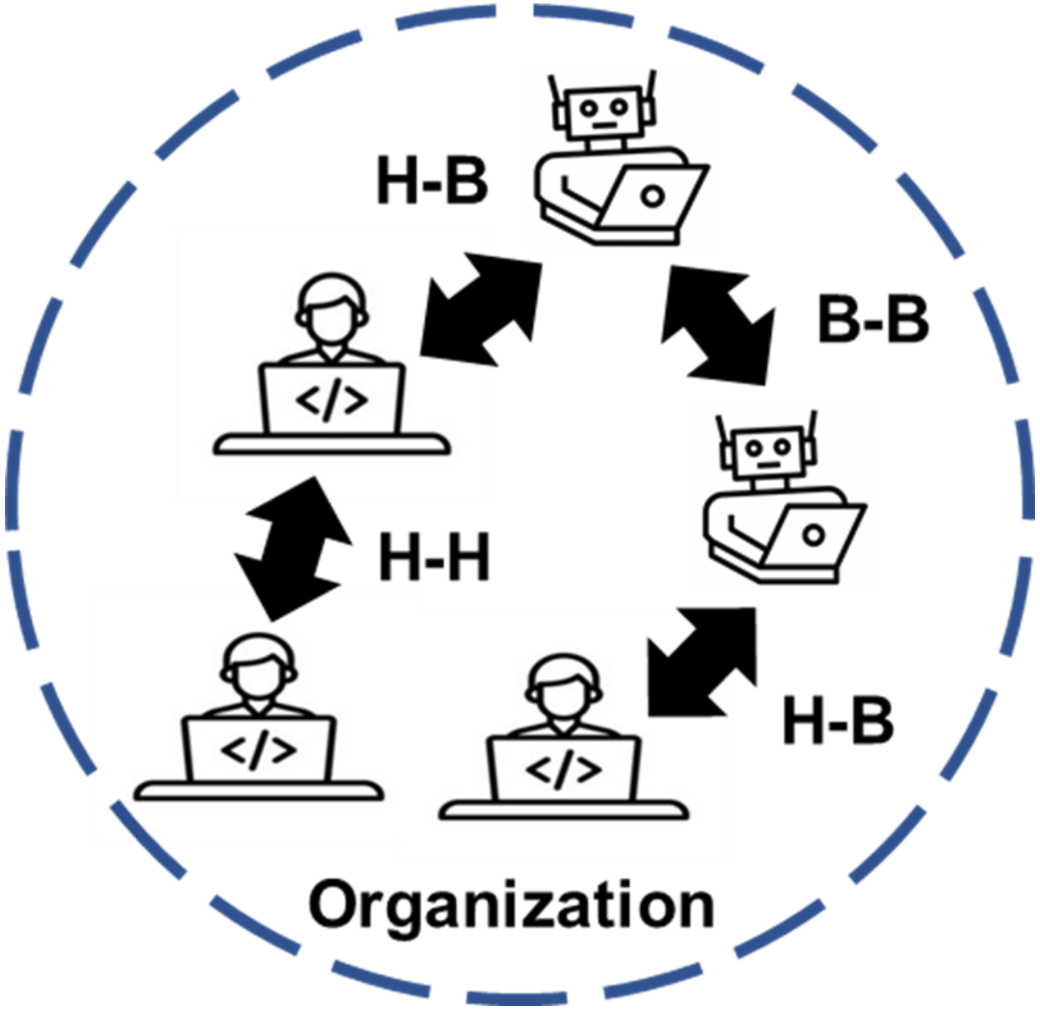
Are Automated Debugging Techniques Actually Helping Programmers?

Chris Parnin and Alessandro Orso
Georgia Institute of Technology
College of Computing
{chris.parnin|orso}@gatech.edu

ISSTA 2011
Won MIP Award

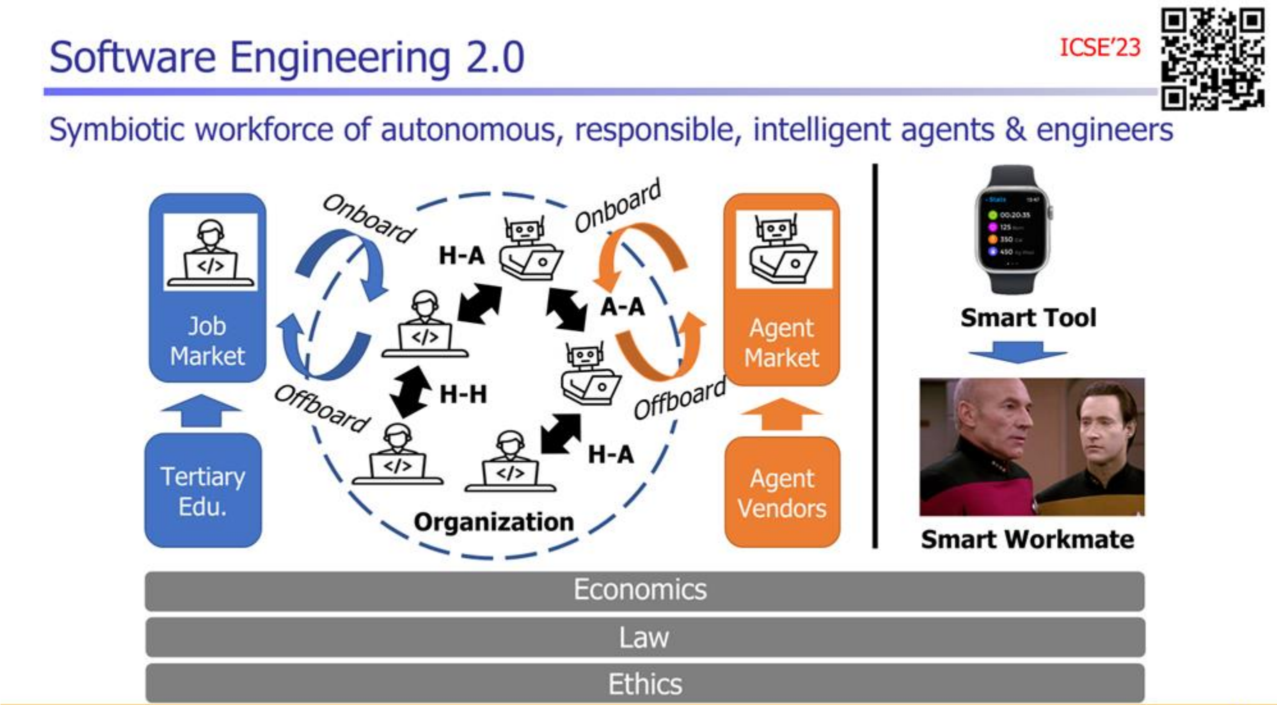
Need for Synergy

N:M??



Synergize
(AI and Humans)

LLM for Software Architecture?



Software Engineering 2.0

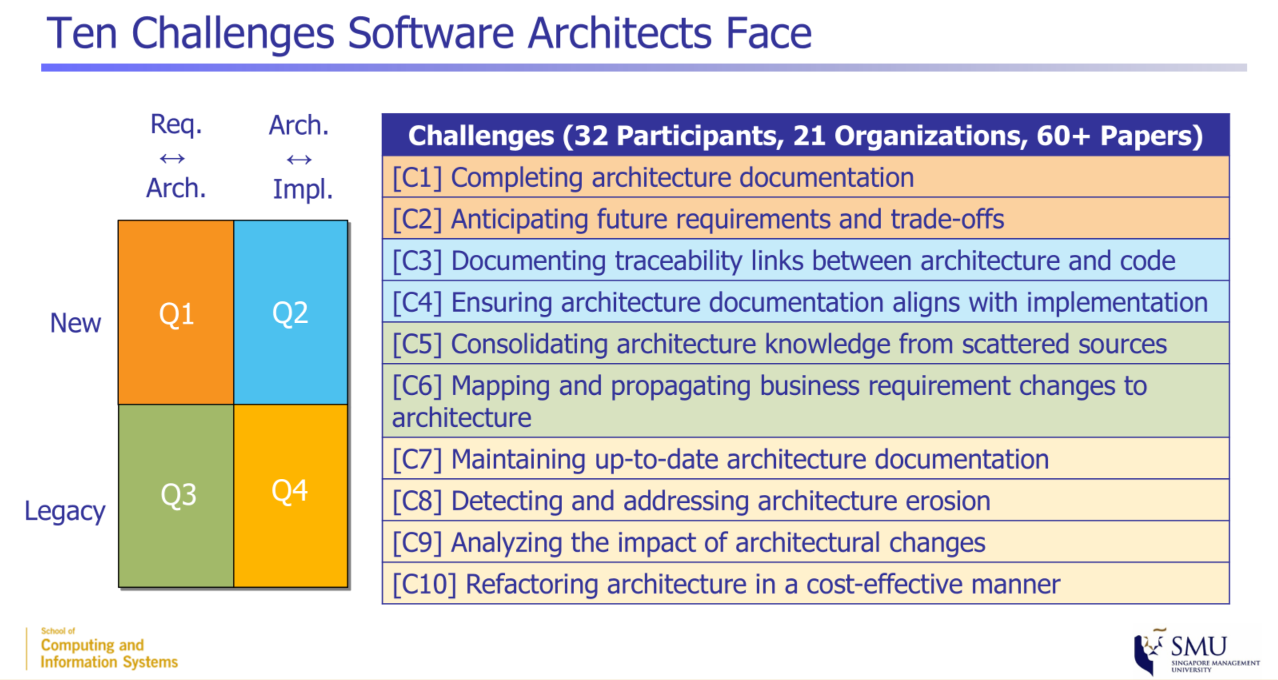
ICSE'23

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

Economics

Law

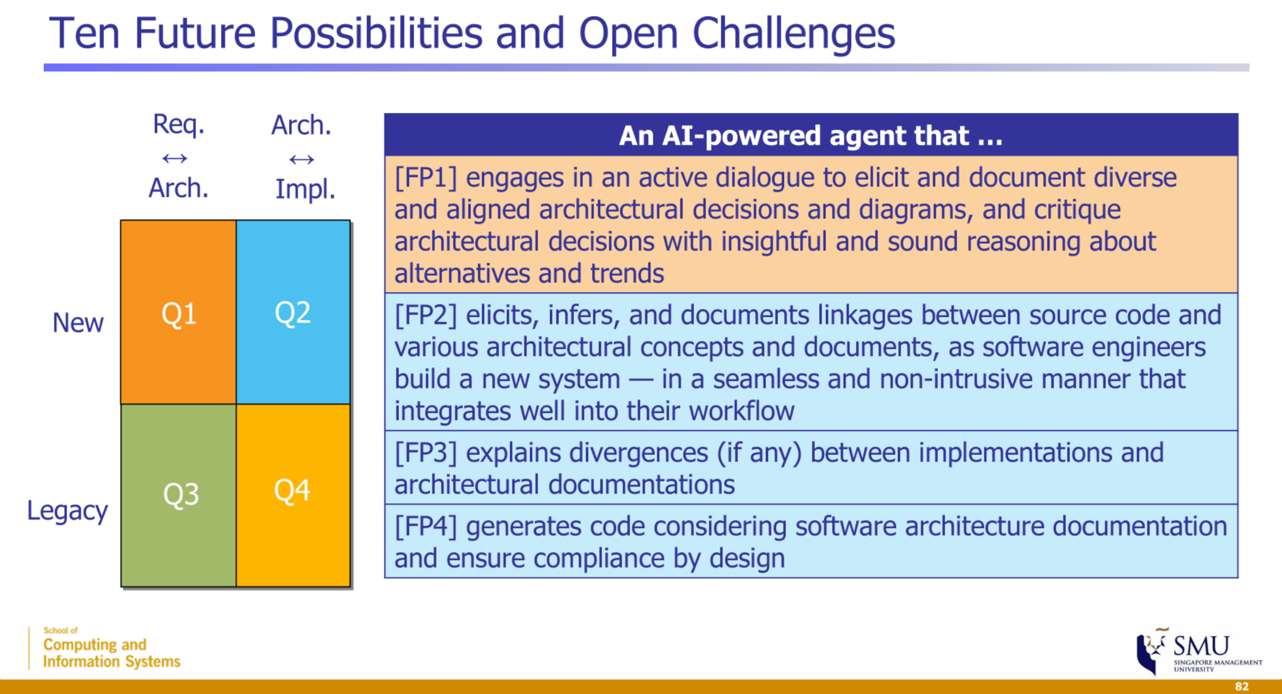
Ethics



Ten Challenges Software Architects Face

	Req. ↔ Arch.	Arch. ↔ Impl.
New	Q1	Q2
Legacy	Q3	Q4

Challenges (32 Participants, 21 Organizations, 60+ Papers)
[C1] Completing architecture documentation
[C2] Anticipating future requirements and trade-offs
[C3] Documenting traceability links between architecture and code
[C4] Ensuring architecture documentation aligns with implementation
[C5] Consolidating architecture knowledge from scattered sources
[C6] Mapping and propagating business requirement changes to architecture
[C7] Maintaining up-to-date architecture documentation
[C8] Detecting and addressing architecture erosion
[C9] Analyzing the impact of architectural changes
[C10] Refactoring architecture in a cost-effective manner



Ten Future Possibilities and Open Challenges

	Req. ↔ Arch.	Arch. ↔ Impl.
New	Q1	Q2
Legacy	Q3	Q4

An AI-powered agent that …
[FP1] engages in an active dialogue to elicit and document diverse and aligned architectural decisions and diagrams, and critique architectural decisions with insightful and sound reasoning about alternatives and trends
[FP2] elicits, infers, and documents linkages between source code and various architectural concepts and documents, as software engineers build a new system — in a seamless and non-intrusive manner that integrates well into their workflow
[FP3] explains divergences (if any) between implementations and architectural documentations
[FP4] generates code considering software architecture documentation and ensure compliance by design

---

Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

Synergize

aiXcoder

overflow AI

tabnine

GitHub Copilot

JET BRAINS

starcoder

salesforce

aws Amazon CodeWhisperer

Acknowledgements

ODENSE • DENMARK
ICSA 2025
31 MARCH - 4 APRIL

SMU
SINGAPORE MANAGEMENT UNIVERSITY

OUB Chair
Professorship Fund

Acknowledgements

Software Architecture in Practice: Challenges and Opportunities

31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023)

Zhiyuan Wan Yun Zhang Xin Xia Yi Jiang David Lo

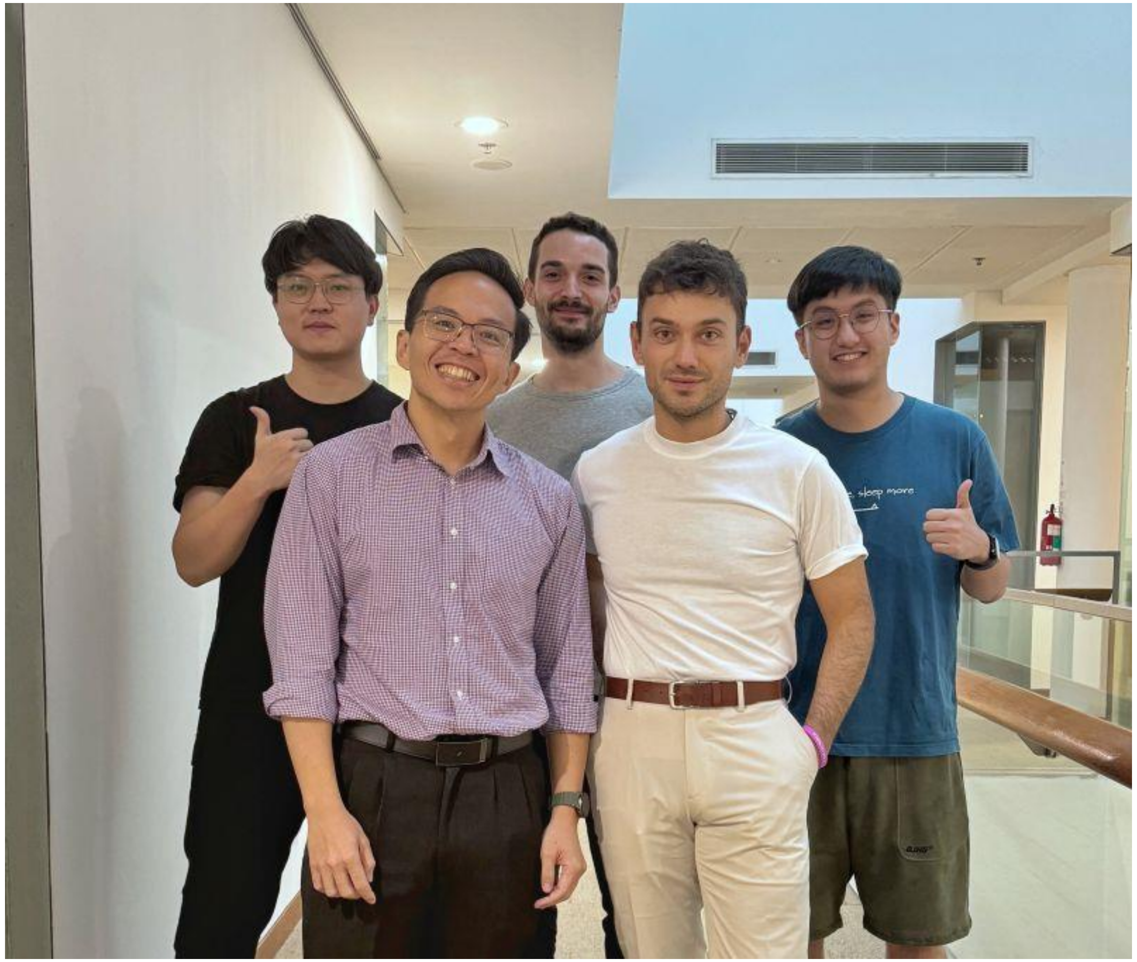
HUAWEI

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Acknowledgements

Artificial Intelligence for Software Architecture: Literature Review and the Road Ahead

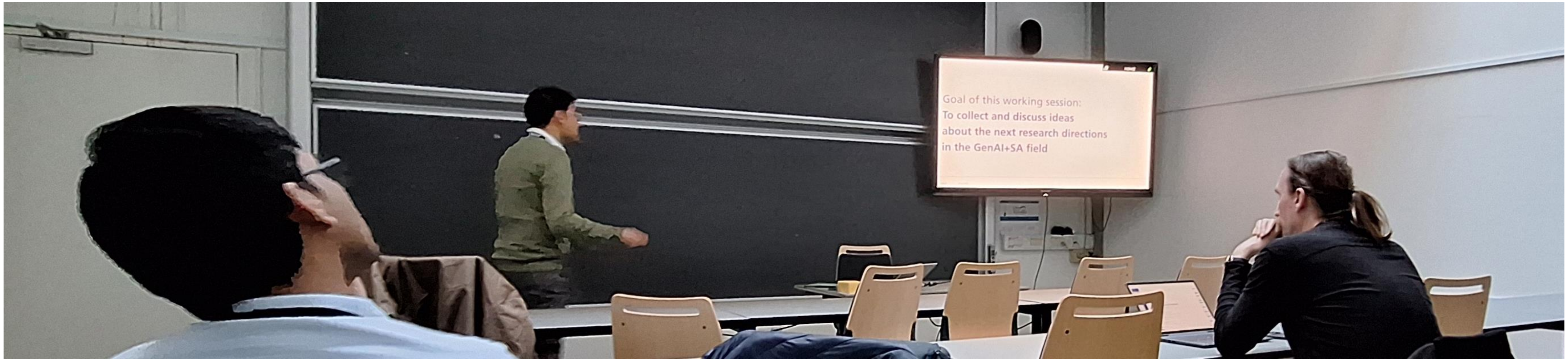
2030 Software Engineering Workshop @ FSE 2025



Acknowledgements

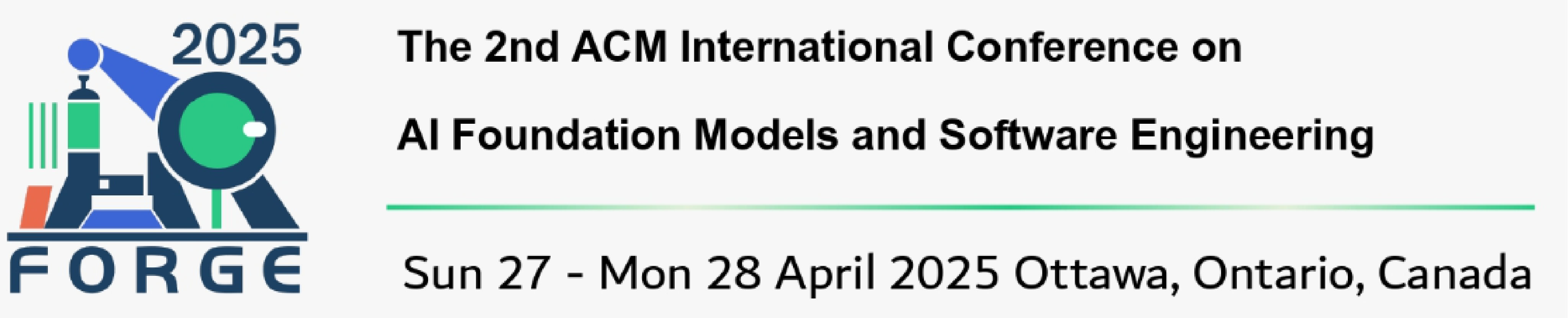
Acknowledgements

SAML 2025 – Panel Discussion


Goal of this working session:
To collect and discuss ideas
about the next research directions
in the GenAI+SA field

SAGAI 2025 – Discussion

Advertisements

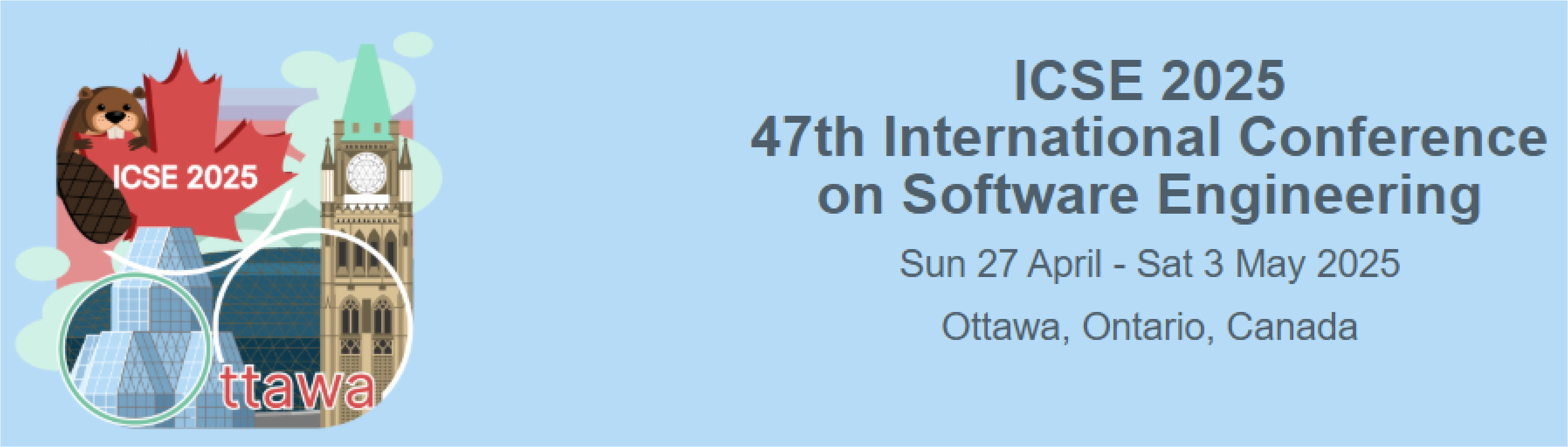


The 2nd ACM International Conference on

AI Foundation Models and Software Engineering

Sun 27 - Mon 28 April 2025 Ottawa, Ontario, Canada

Please submit to FORGE 2026, deadlines: Dec 2025


ICSE 2025
ttawa

ICSE 2025
47th International Conference
on Software Engineering

Sun 27 April - Sat 3 May 2025

Ottawa, Ontario, Canada

Interested to be Session Chairs? Design and Architecture, SE4AI

Thank you!

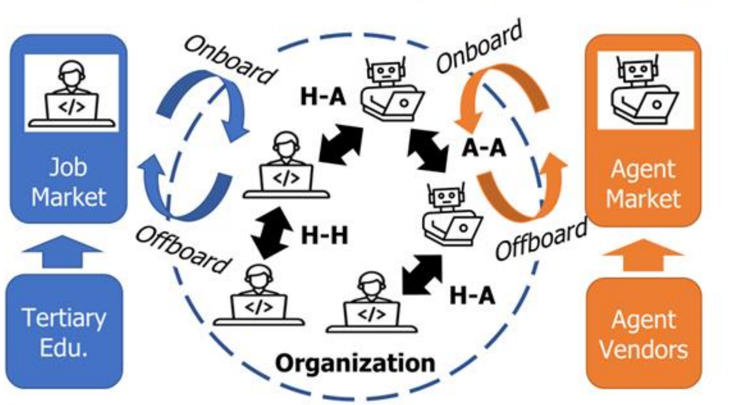
Questions? Comments? Advice?

davidlo@smu.edu.sg

Software Engineering 2.0

ICSE'23

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

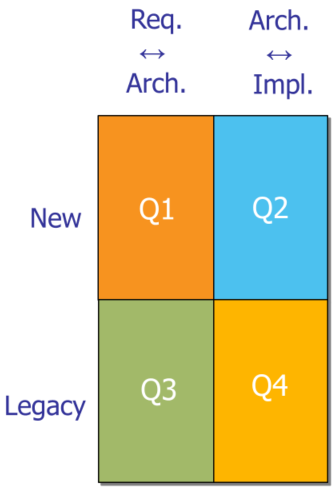


Economics

Law

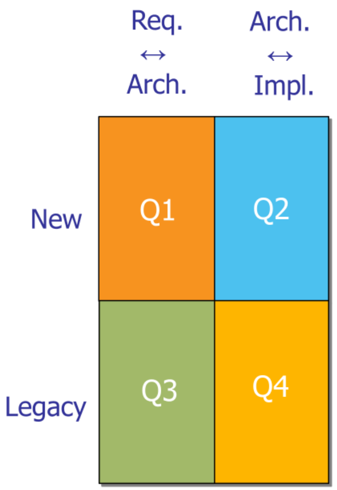
Ethics

Ten Challenges Software Architects Face



Challenges (32 Participants, 21 Organizations, 60+ Papers)
[C1] Completing architecture documentation
[C2] Anticipating future requirements and trade-offs
[C3] Documenting traceability links between architecture and code
[C4] Ensuring architecture documentation aligns with implementation
[C5] Consolidating architecture knowledge from scattered sources
[C6] Mapping and propagating business requirement changes to architecture
[C7] Maintaining up-to-date architecture documentation
[C8] Detecting and addressing architecture erosion
[C9] Analyzing the impact of architectural changes
[C10] Refactoring architecture in a cost-effective manner

Ten Future Possibilities and Open Challenges



An AI-powered agent that ...
[FP1] engages in an active dialogue to elicit and document diverse and aligned architectural decisions and diagrams, and critique architectural decisions with insightful and sound reasoning about alternatives and trends
[FP2] elicits, infers, and documents linkages between source code and various architectural concepts and documents, as software engineers build a new system — in a seamless and non-intrusive manner that integrates well into their workflow
[FP3] explains divergences (if any) between implementations and architectural documentations
[FP4] generates code considering software architecture documentation and ensure compliance by design

---

Realizing SE 2.0: Human-Centered AI4SE

Listen

Learn

Synergize