Requirements Engineering for Trustworthy Human-AI Synergy in Software Engineering 2.0

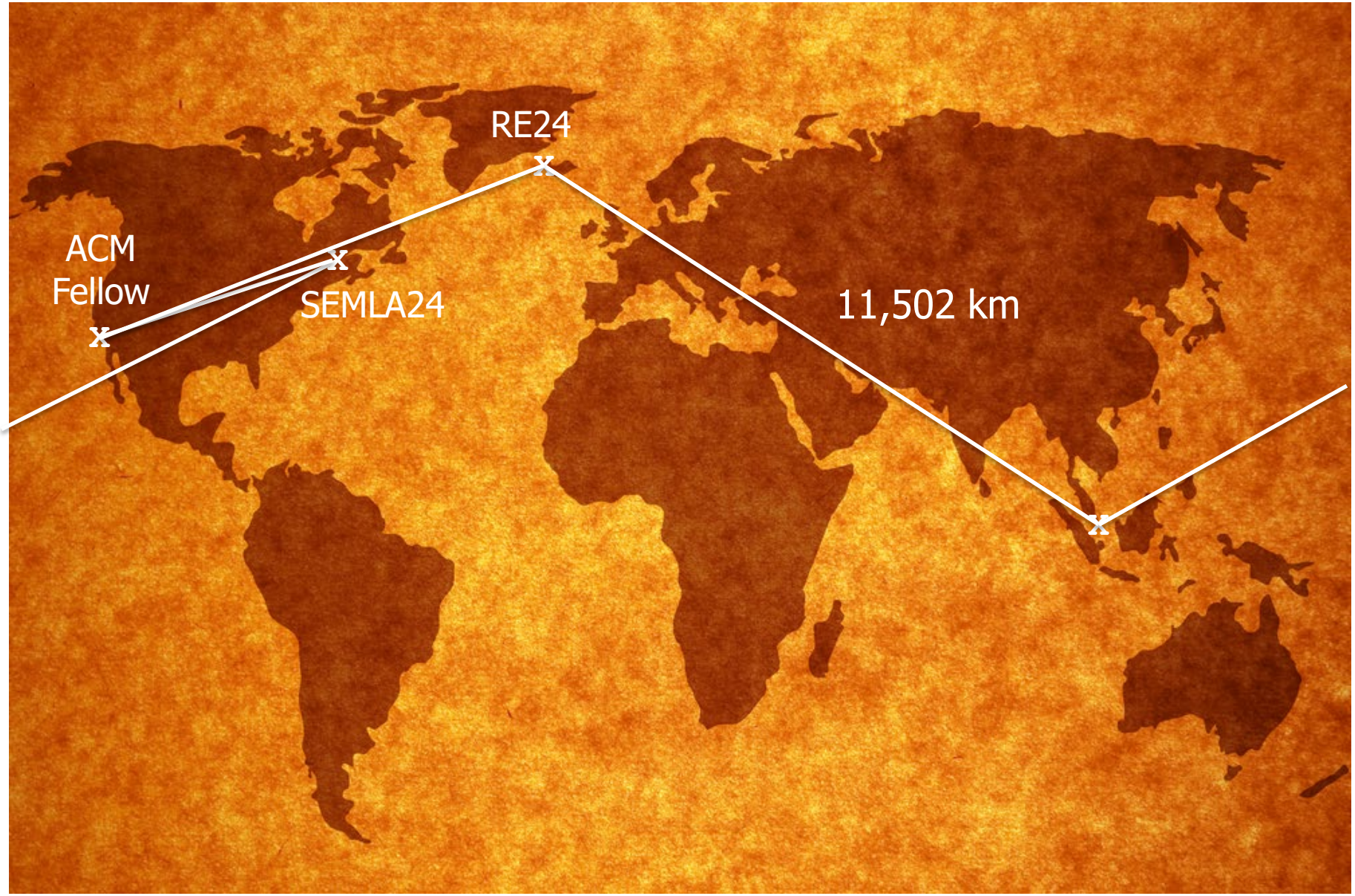
David Lo

OUB Chair Professor of Computer Science
Director, Center for Research on Intelligent SE (RISE)

SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and
Information Systems

Self-Introduction



Self-Introduction

Self-Introduction

Self-Introduction

Singapore Management University

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

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, June 2024

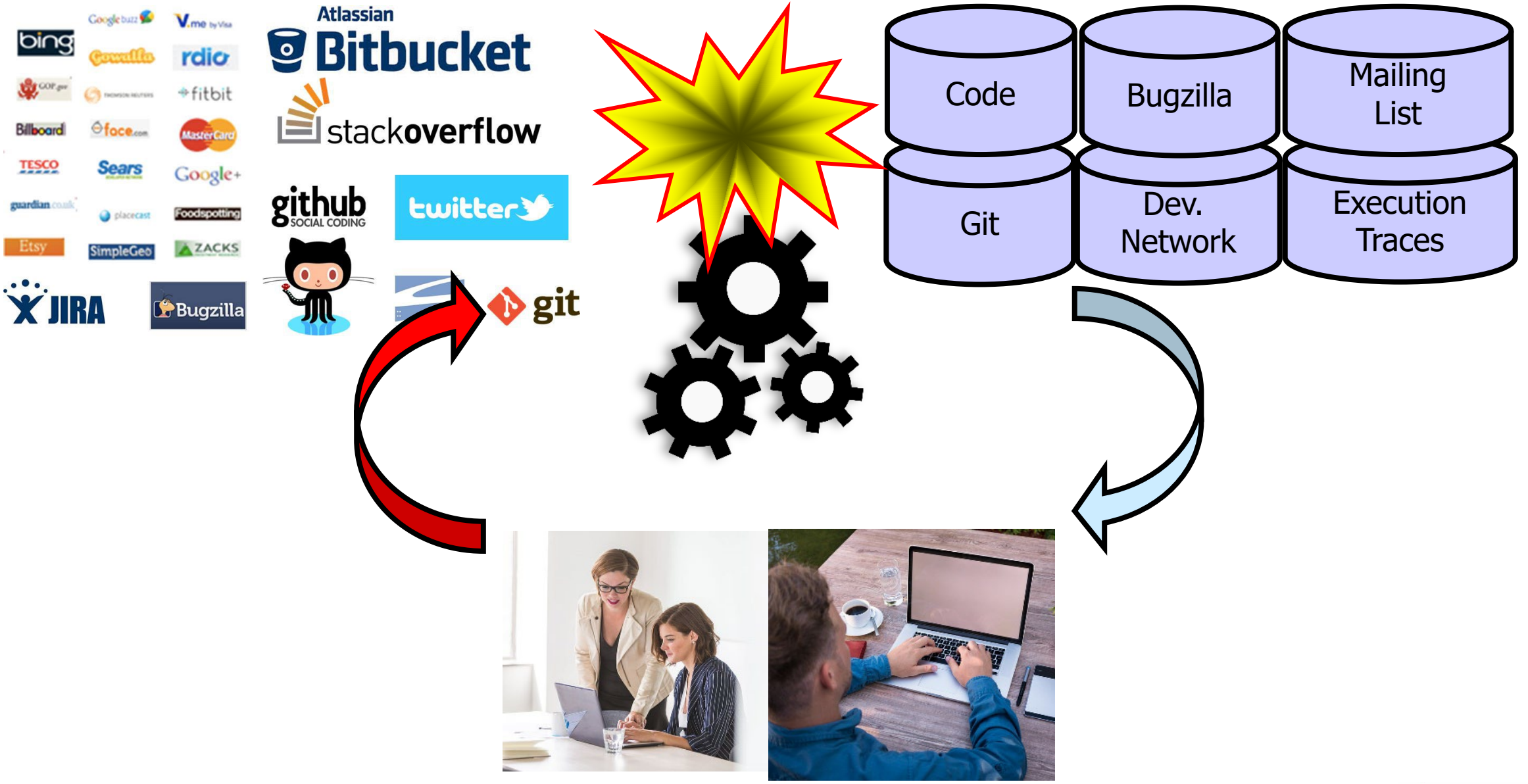
#	Institution	Count	Faculty
1	Nanjing University	39.0	38
2	Carnegie Mellon University	31.6	17
3	Peking University	28.5	21
4	Singapore Management University	22.7	8

SMU SINGAPORE MANAGEMENT UNIVERSITY

School of Computing and Information Systems

Centre for Research on Intelligent Software Engineering

AI for Software Engineering



Experience with AI4SE

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

FSE'06

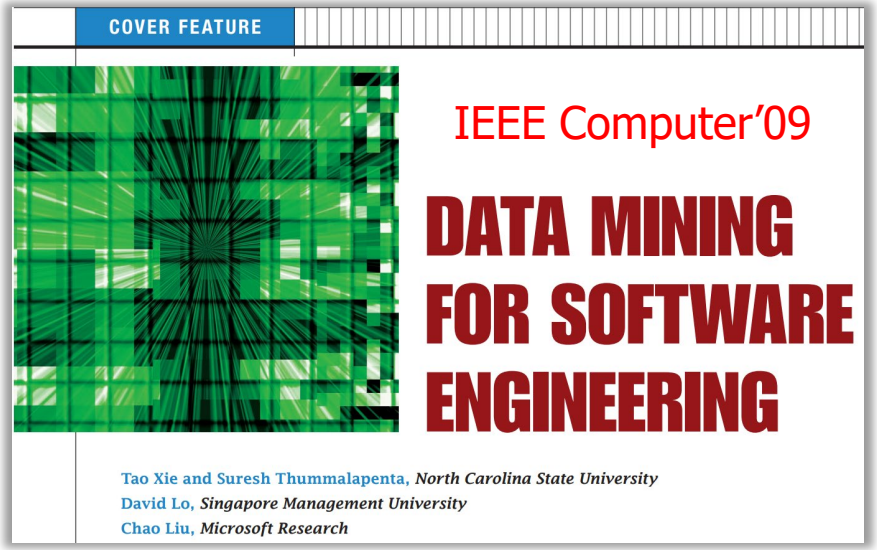
David Lo and Siau-Cheng Khoo
Department of Computer Science,National University of Singapore
{dlo,khoosc}@comp.nus.edu.sg

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



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

Requirements Engineering for Trustworthy Human-AI Synergy in Software Engineering 2.0

David Lo

OUB Chair Professor of Computer Science
Director, Center for Research on Intelligent SE (RISE)

SMU
SINGAPORE MANAGEMENT UNIVERSITY

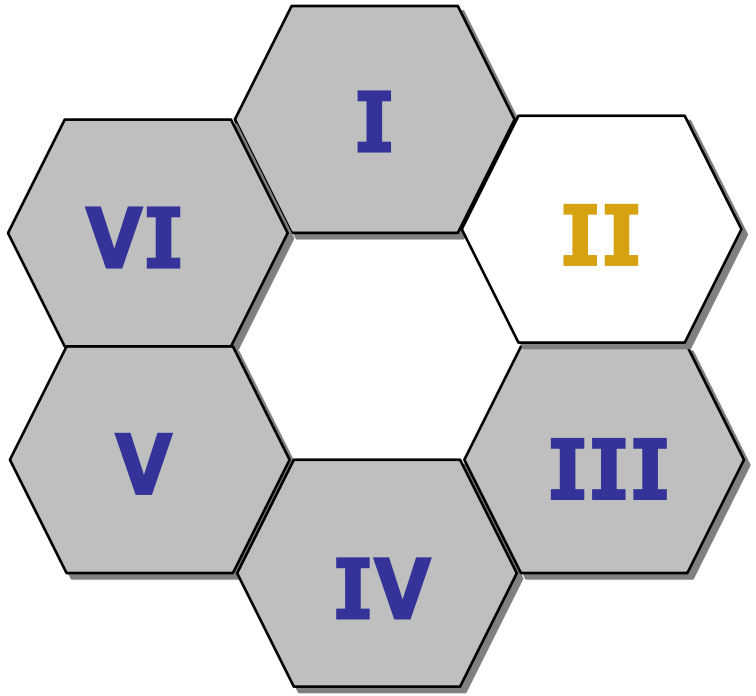
School of
Computing and
Information Systems

Talk Structure

Self-Intro.

RE4(SE2.0)

Req.-Guided AI4SE

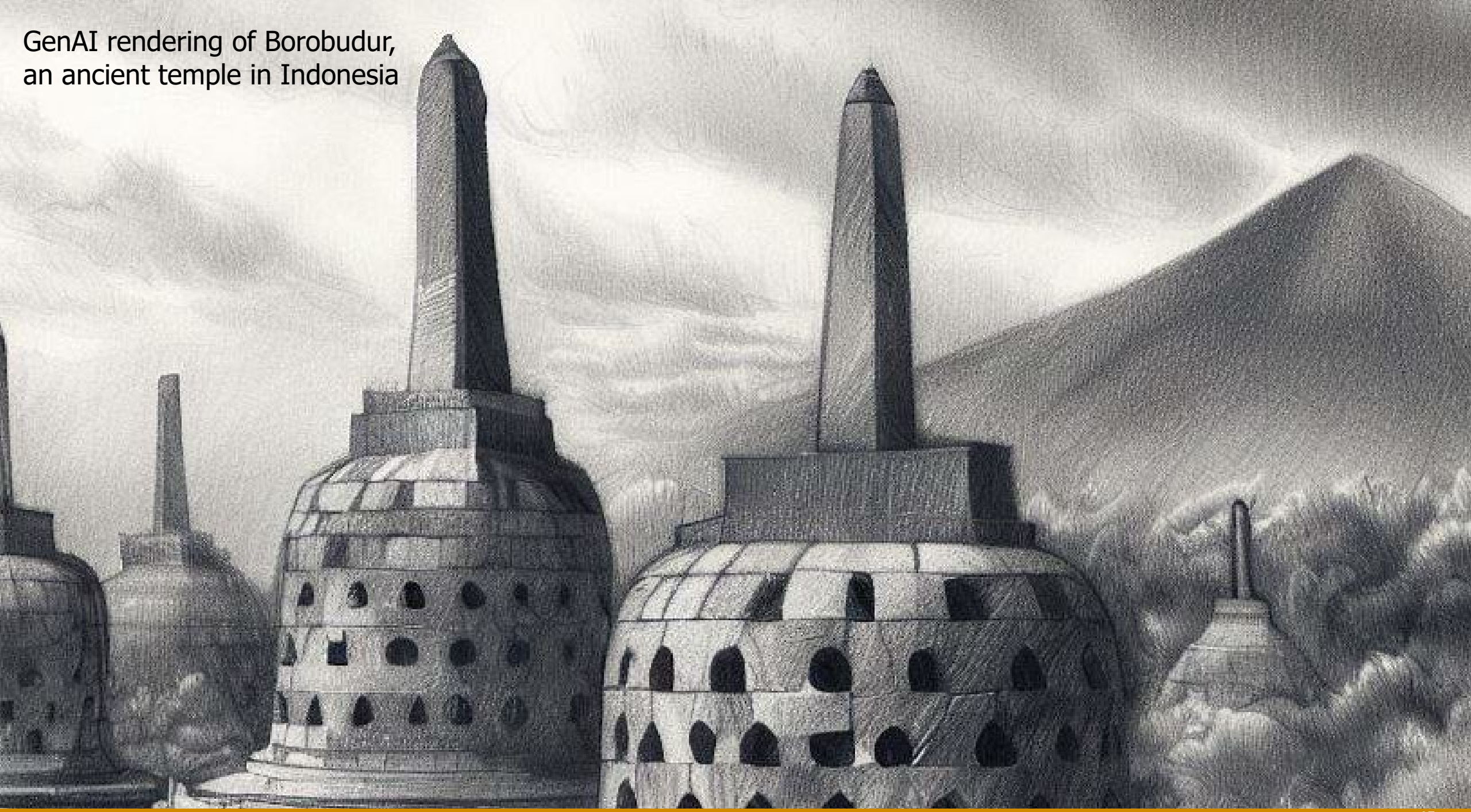


AI4SE

SE2.0

RE4(AI4SE)

GenAI rendering of Borobudur,
an ancient temple in Indonesia



AI for Software Engineering (AI4SE)

MSR 2004: International Workshop on Mining Software Repositories
2004.msrconf.org

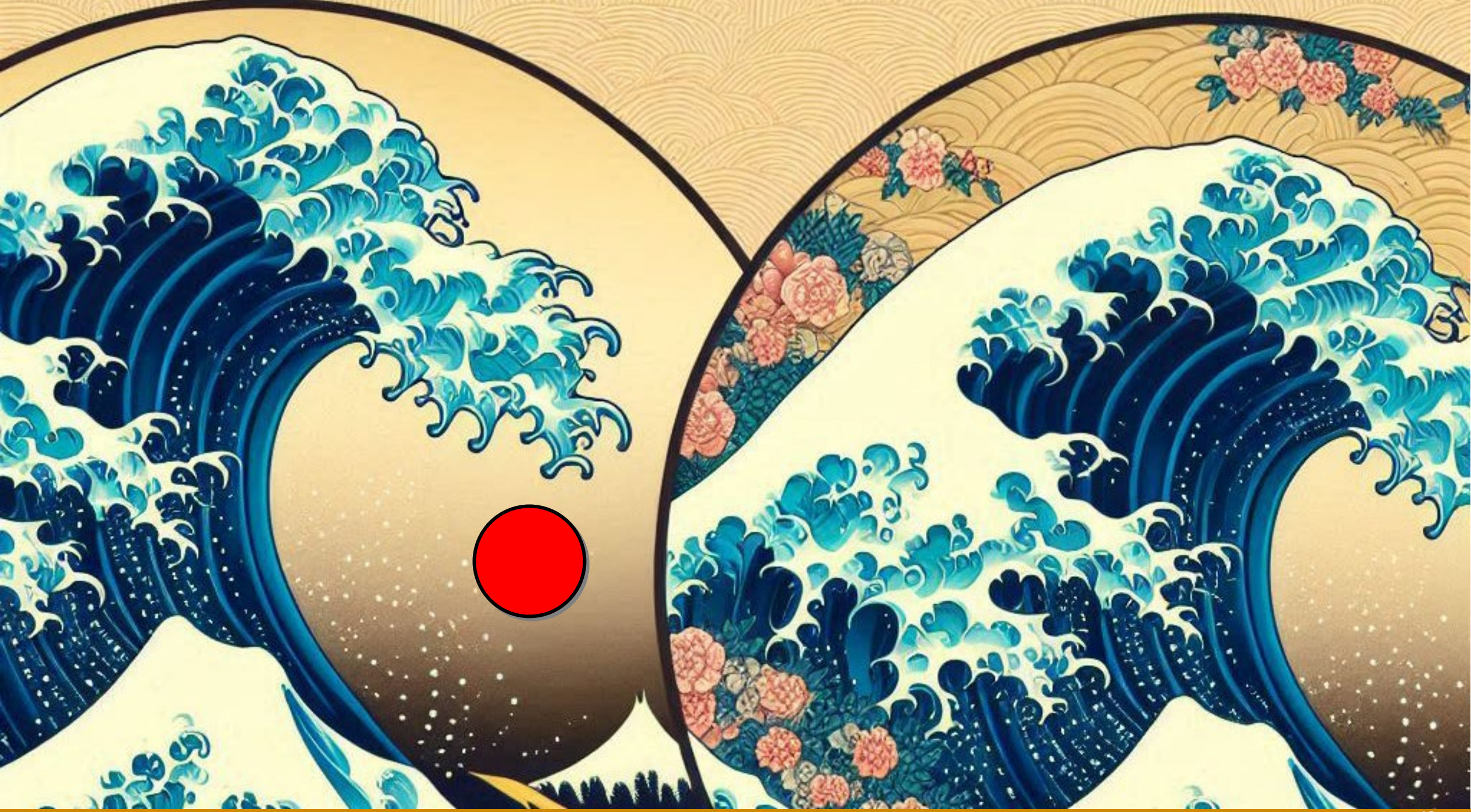
AI for Software Engineering (AI4SE)

Andreas Zeller @AndreasZeller · Mar 31, 2022

Machine learning rules! Machine Learning with and for Software Engineering is now the #1 topic in terms of submitted papers, before all-time favorites such as software testing and analysis.

Top 10 Topics – Submitted

Topics	# Submitted Papers	# Accepted Papers	Acceptance Rate
Machine Learning with and for SE	237	74	31,22%
Software Testing	181	47	25,97%
Program Analysis	117	35	29,91%
Evolution and maintenance	105	31	29,52%
Mining Software Repositories	105	23	21,90%
Software Security	85	25	29,41%
Human Aspects of SE	68	20	29,41%
Validation and Verification	53	15	28,30%
Tools and Environments	49	12	24,49%
Reliability and Safety	46	15	32,61%



Wave 1: Deep Learning

Deep Learning for Just-In-Time Defect Prediction

Xinli Yang*, David Lo†, Xin Xia*‡, Yun Zhang*, and Jianling Sun*
*College of Computer Science and Technology, Zhejiang University, Hangzhou, China
†School of Information Systems, Singapore Management University, Singapore
zdyxl@zju.edu.cn, davidlo@smu.edu.sg, {xxia, yunzhang28, sunjl}@zju.edu.cn

Toward Deep Learning Software Repositories

Martin White, Christopher Vendome, Mario Linares-Vásquez, and Denys Poshyvanyk
Department of Computer Science
College of William and Mary
Williamsburg, Virginia 23187–8795
Email: {mgwhite, cvendome, mlinarev, denys}@cs.wm.edu

First few papers of deep learning for software engineering (2015, 300+ citations each)

Wave 1: Deep Learning

Key Question: How Can We Learn Better Representations of Code and Other Software Artifacts?

Deep Belief Network (DBN)

Convolutional Neural Network (CNN)

Recurrent Neural Network (RNN)

Long Short-Term Memory Network (LSTM)

Transformer

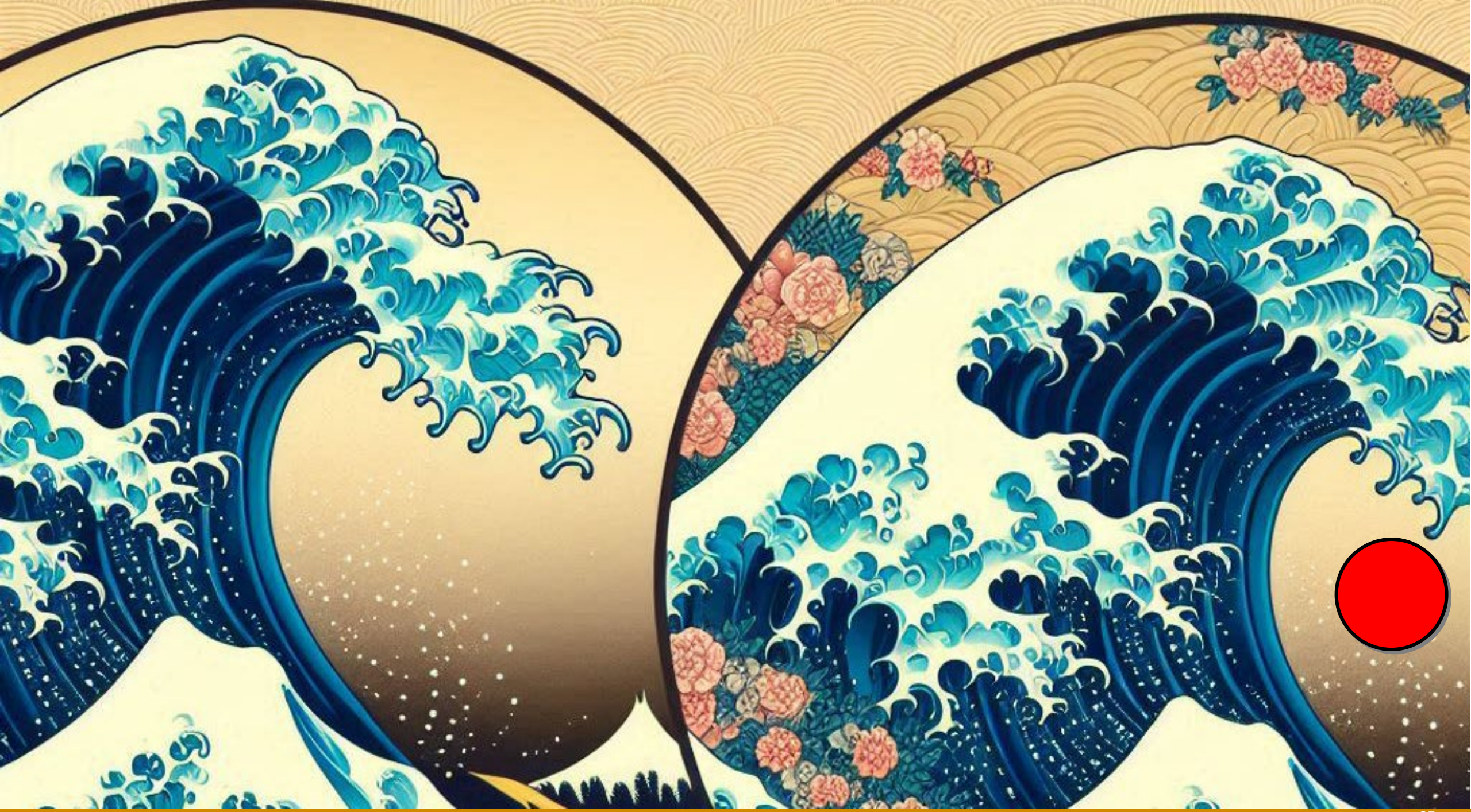
...

Wave 1: Deep Learning

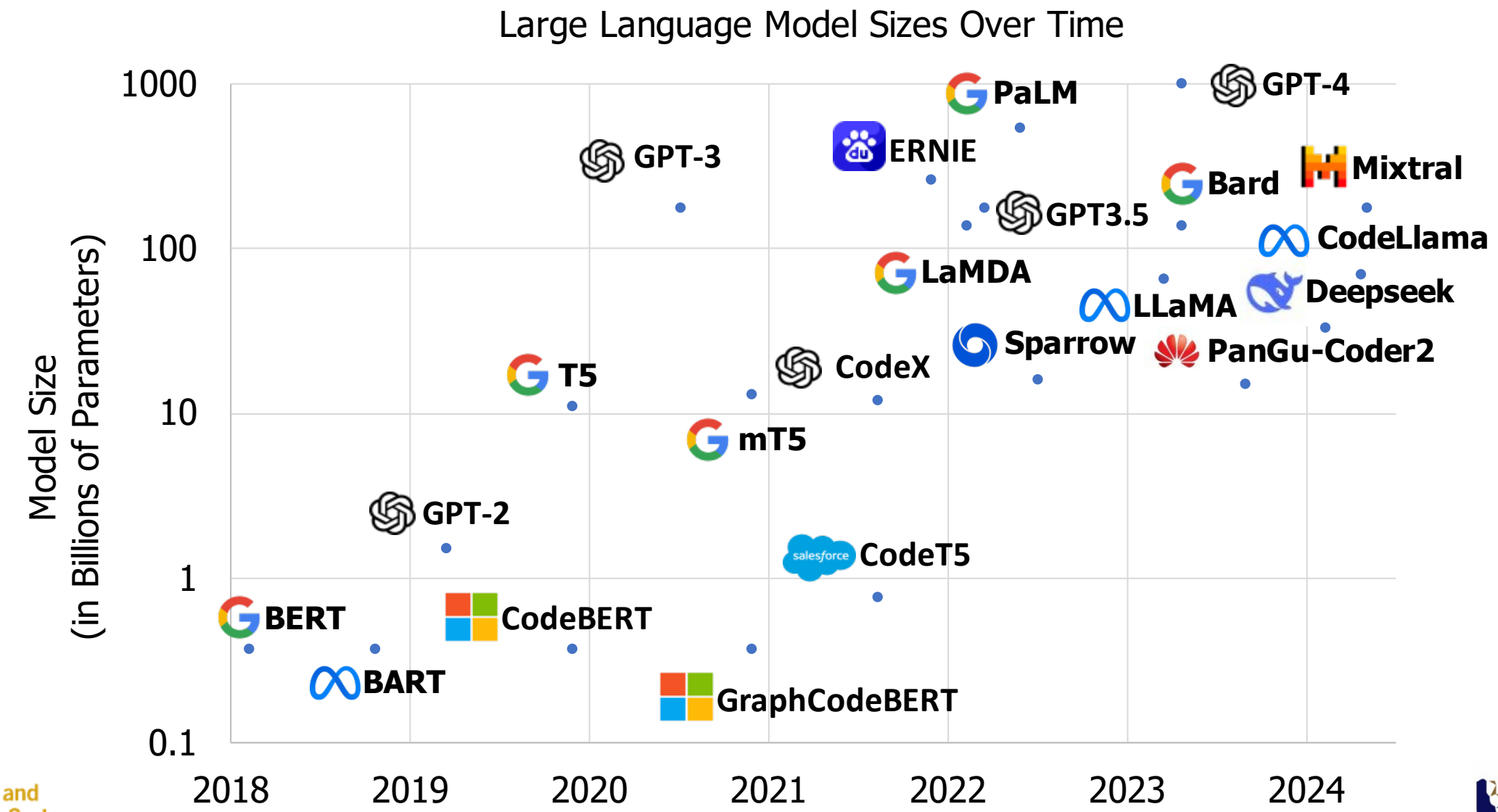
A Survey on Deep Learning for Software Engineering

YANMING YANG, Faculty of Information Technology, Monash University, Australia
XIN XIA, Faculty of Information Technology, Monash University, Australia
DAVID LO, School of Information Systems, Singapore Management University, Singapore
JOHN GRUNDY, Faculty of Information Technology, Monash University, Australia

ACM Computing Survey 2022



Wave 2: Large Language Models (LLMs)



LLMs Seem to Win for Many SE Tasks

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, most cited paper of ICSME 2020

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

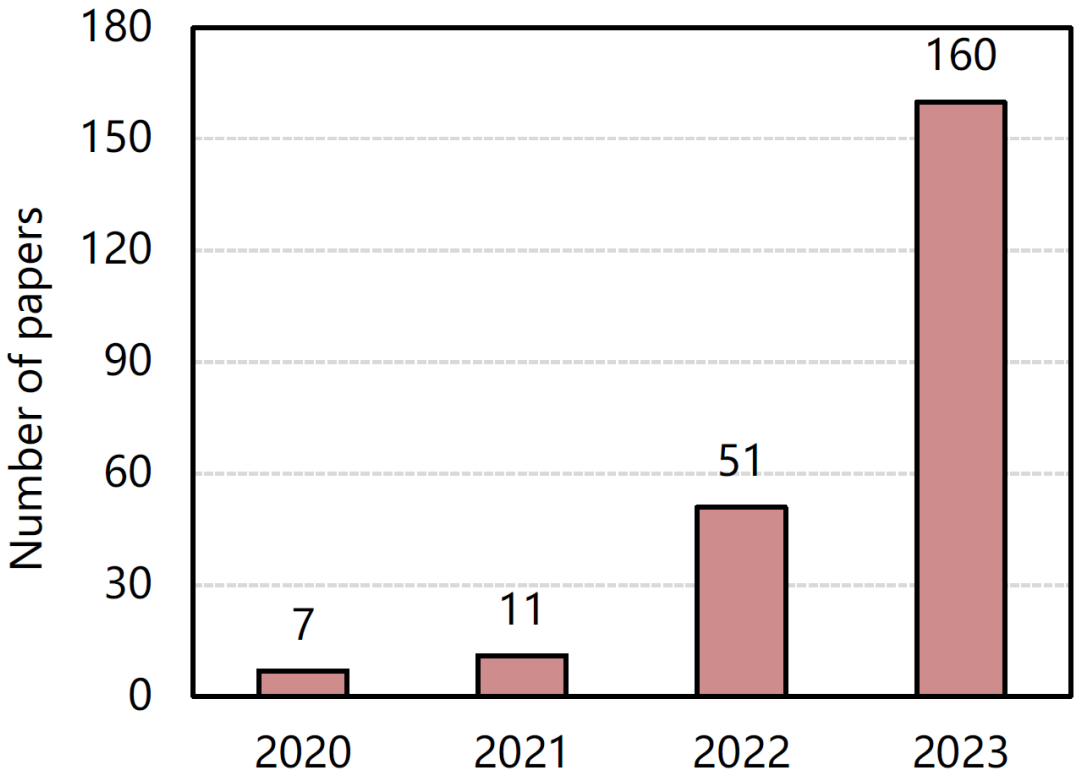
*Multi-LLM collaboration + data-centric innovation = **2x** efficacy*

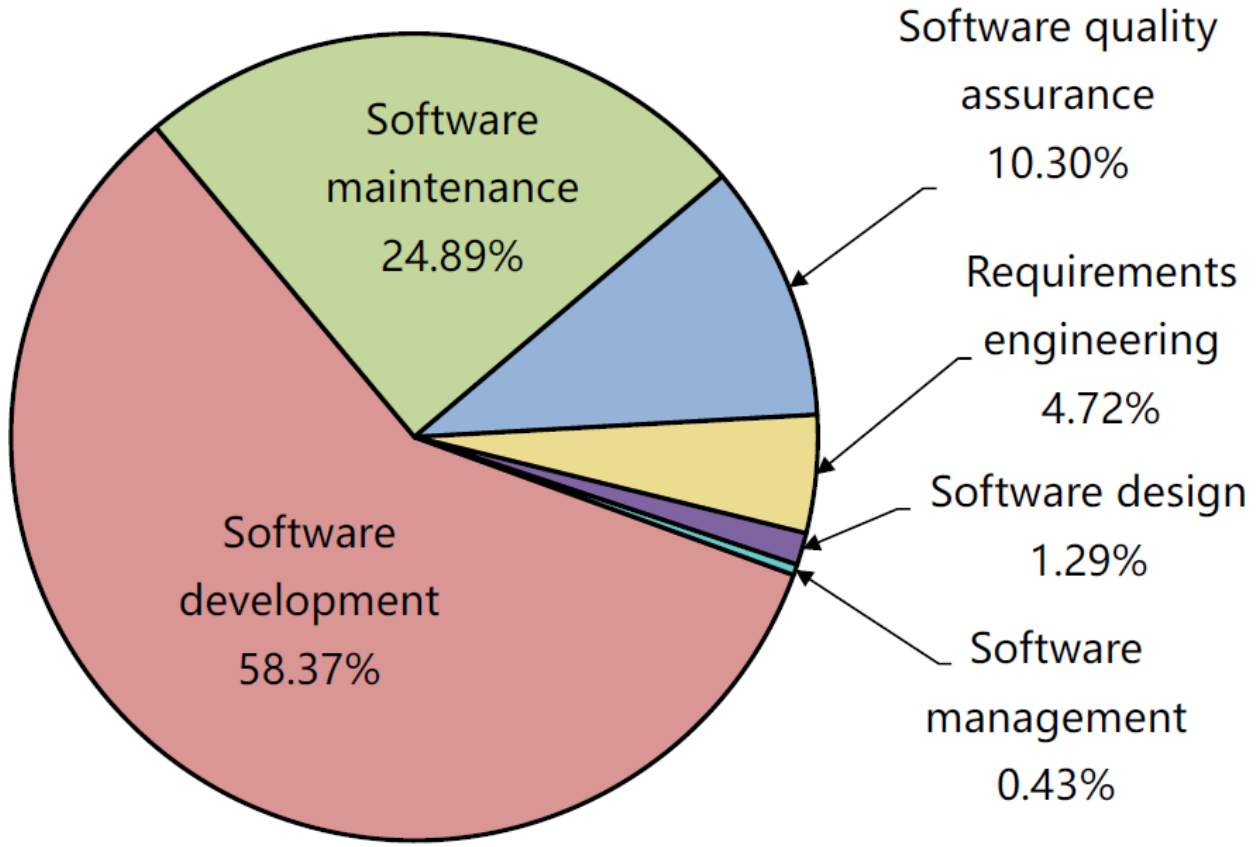
LLMs Seem to Win for Many SE Tasks

Large Language Models for Software Engineering: A Systematic Literature Review

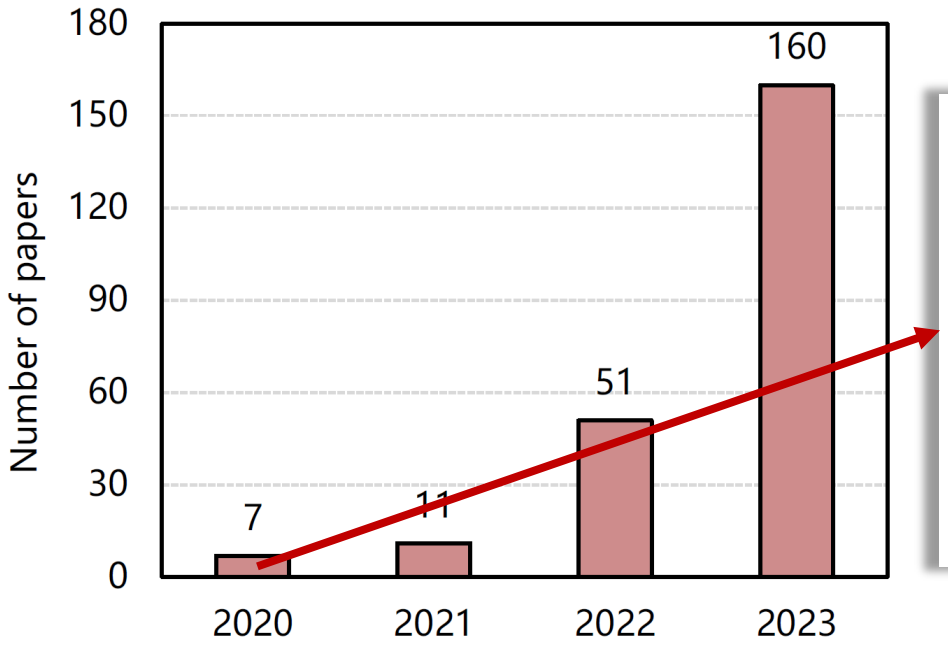
XINYI HOU*, Huazhong University of Science and Technology, China
YANJIE ZHAO*, Monash University, Australia
YUE LIU, Monash University, Australia
ZHOU YANG, Singapore Management University, Singapore
KAILONG WANG, Huazhong University of Science and Technology, China
LI LI, Beihang University, China
XIAPU 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 Tasks





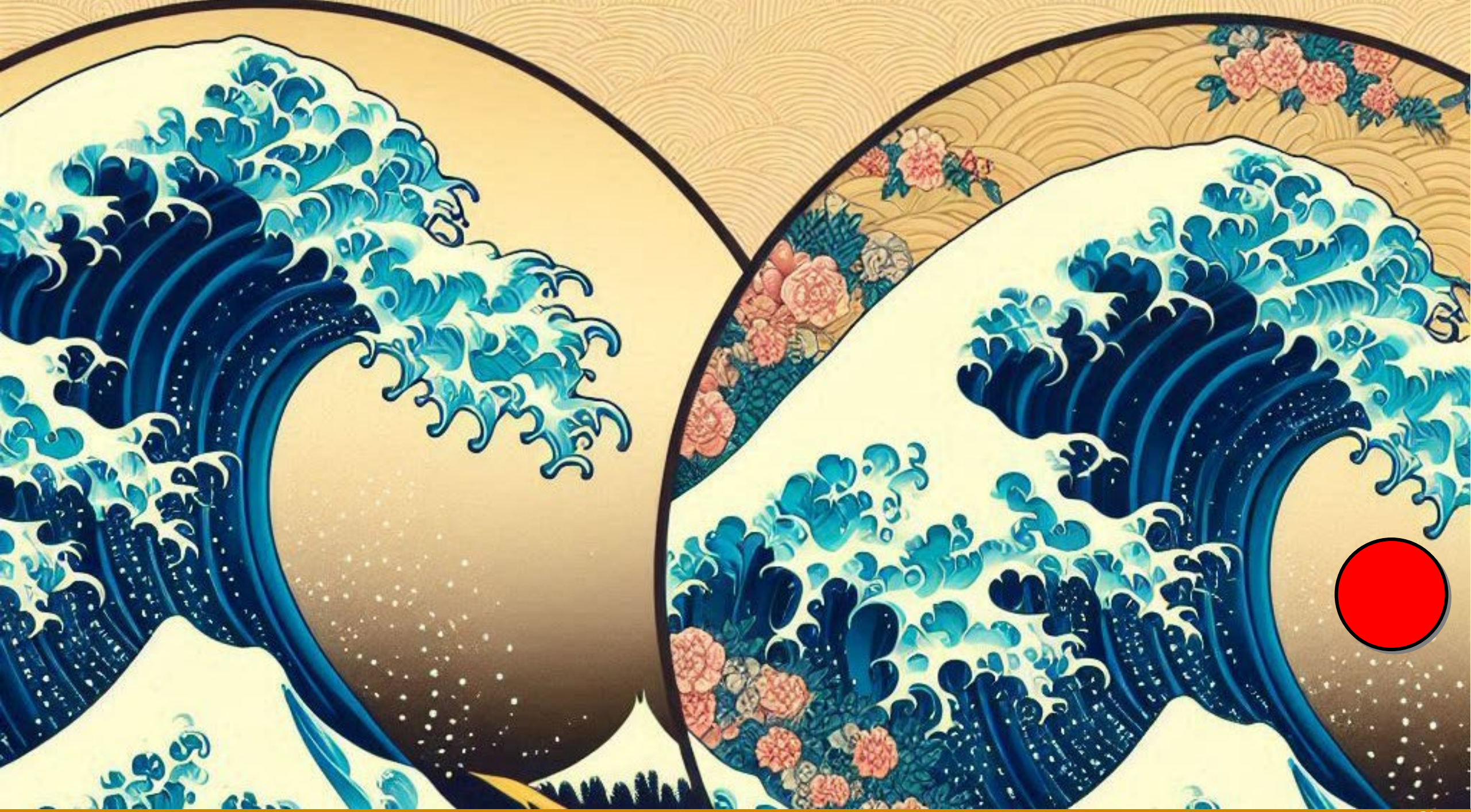
RE is Among Early Adopters of LLM in SE



NoRBERT: Transfer Learning for Requirements Classification

Tobias Hey, Jan Keim, Anne Koziolek, Walter F. Tichy
Karlsruhe Institute of Technology (KIT)
Institute for Program Structures and Data Organization
Karlsruhe, Germany
hey@kit.edu, jan.keim@kit.edu, koziolek@kit.edu, tichy@kit.edu

Early work on LLM4SE, most cited paper of RE 2020

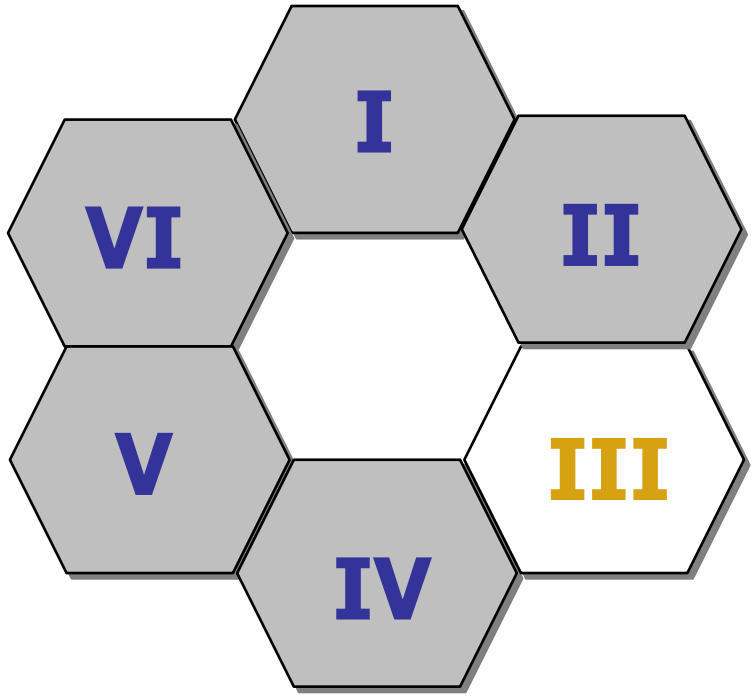


Talk Structure

Self-Intro.

Call to Arms

Req.-Guided AI4SE



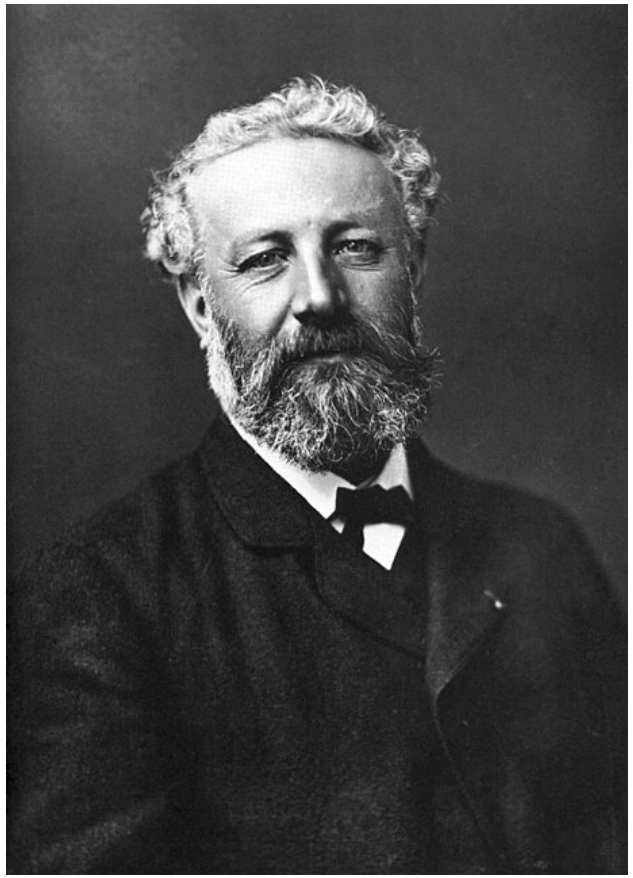
AI4SE

SE2.0

RE4(AI4SE)

Vision/Dream

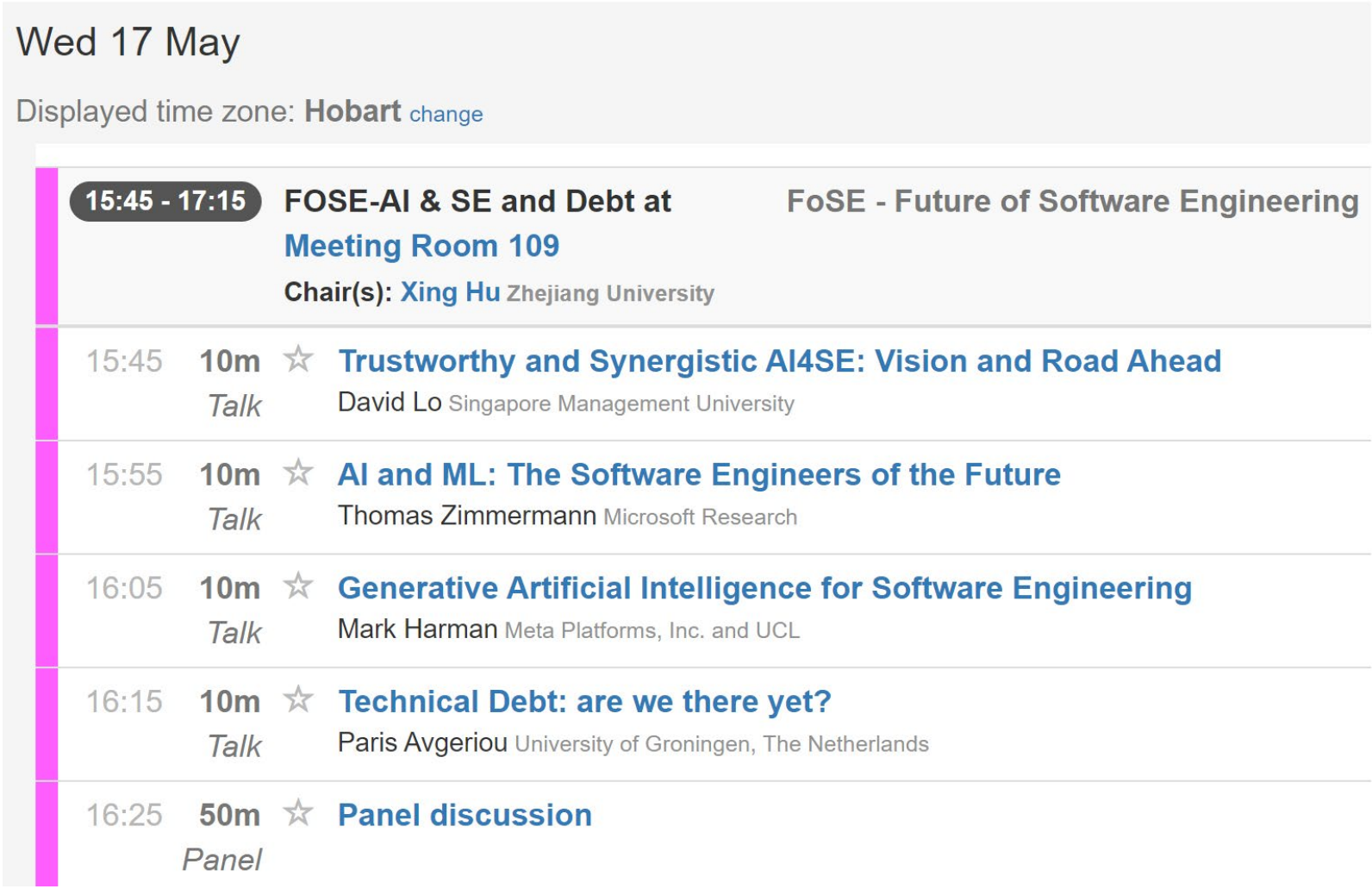
Future of Software Engineering Track, ICSE 2023



"Anything one person can imagine, other people can make real."

- Jules Verne

Vision/Dream



Wed 17 May

Displayed time zone: **Hobart** change

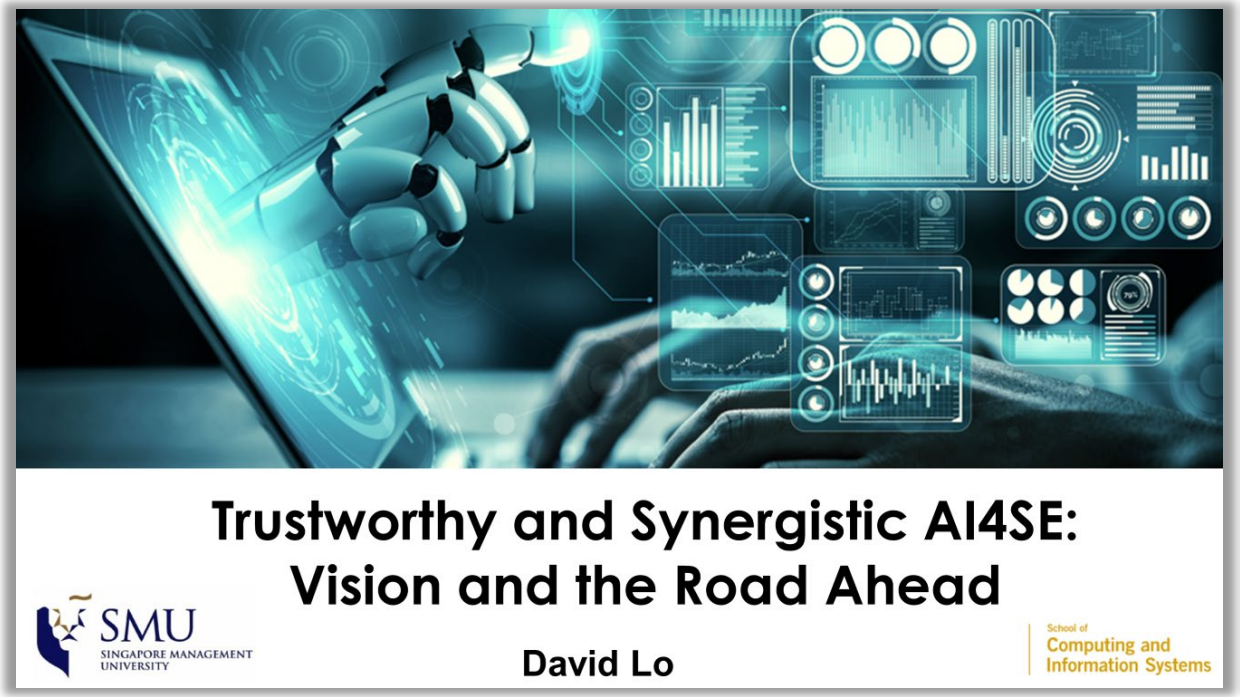
15:45 - 17:15 **FOSE-AI & SE and Debt at Meeting Room 109** — FoSE - Future of Software Engineering

Chair(s): Xing 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**

Future of Software Engineering, ICSE 2023

Vision/Dream: Software Engineering 2.0



What can we realize with trustworthy and synergistic AI4SE?

Vision 2033:

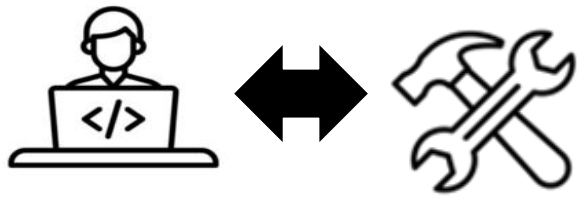
Symbiotic workforce of *autonomous*, *responsible*, *intelligent* bots and software engineers, creating a new way for us to engineer software (**Software Engineering 2.0**)

10

Future of Software Engineering Track, ICSE 2023

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

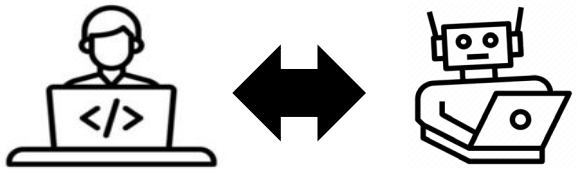


GitHub Copilot

aws Amazon CodeWhisperer

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

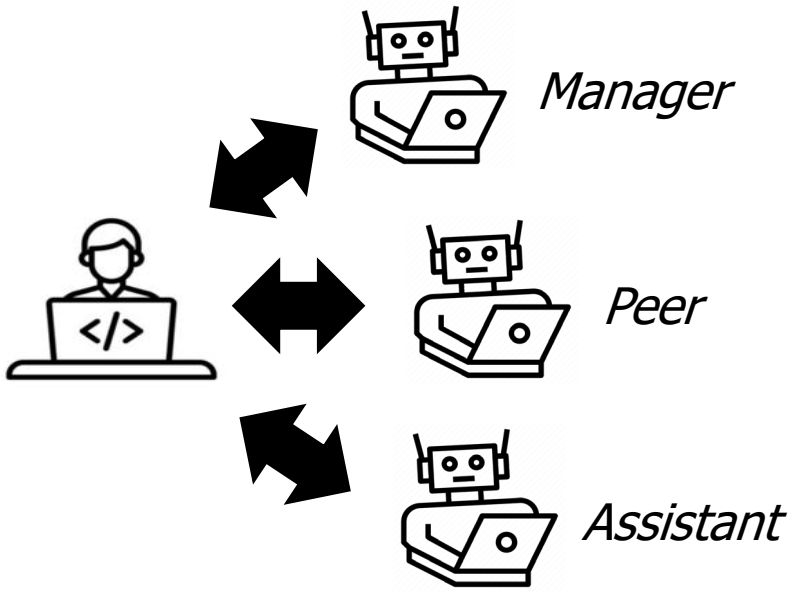


Smart Tool

Smart Workmate

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

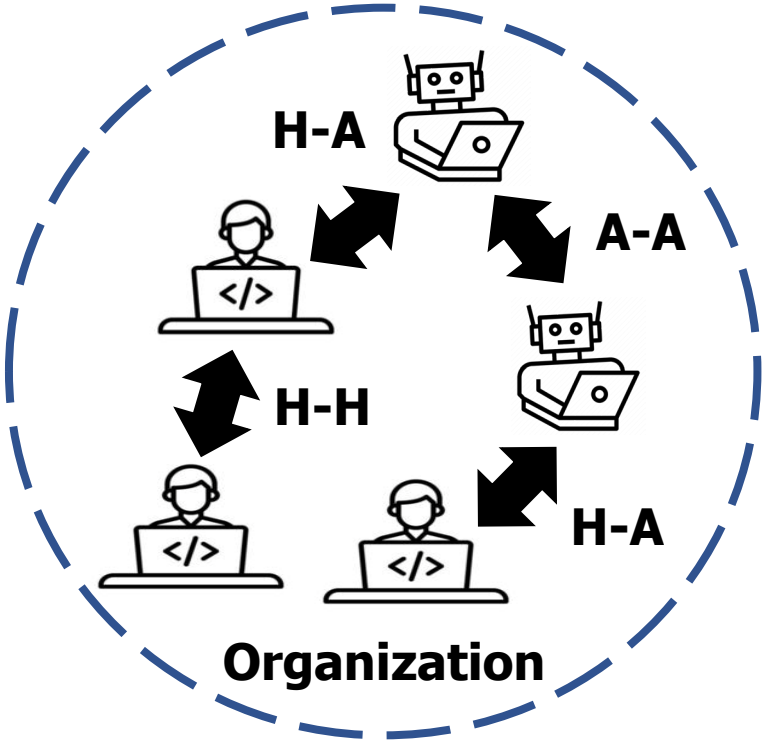


Smart Tool

Smart Workmate

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

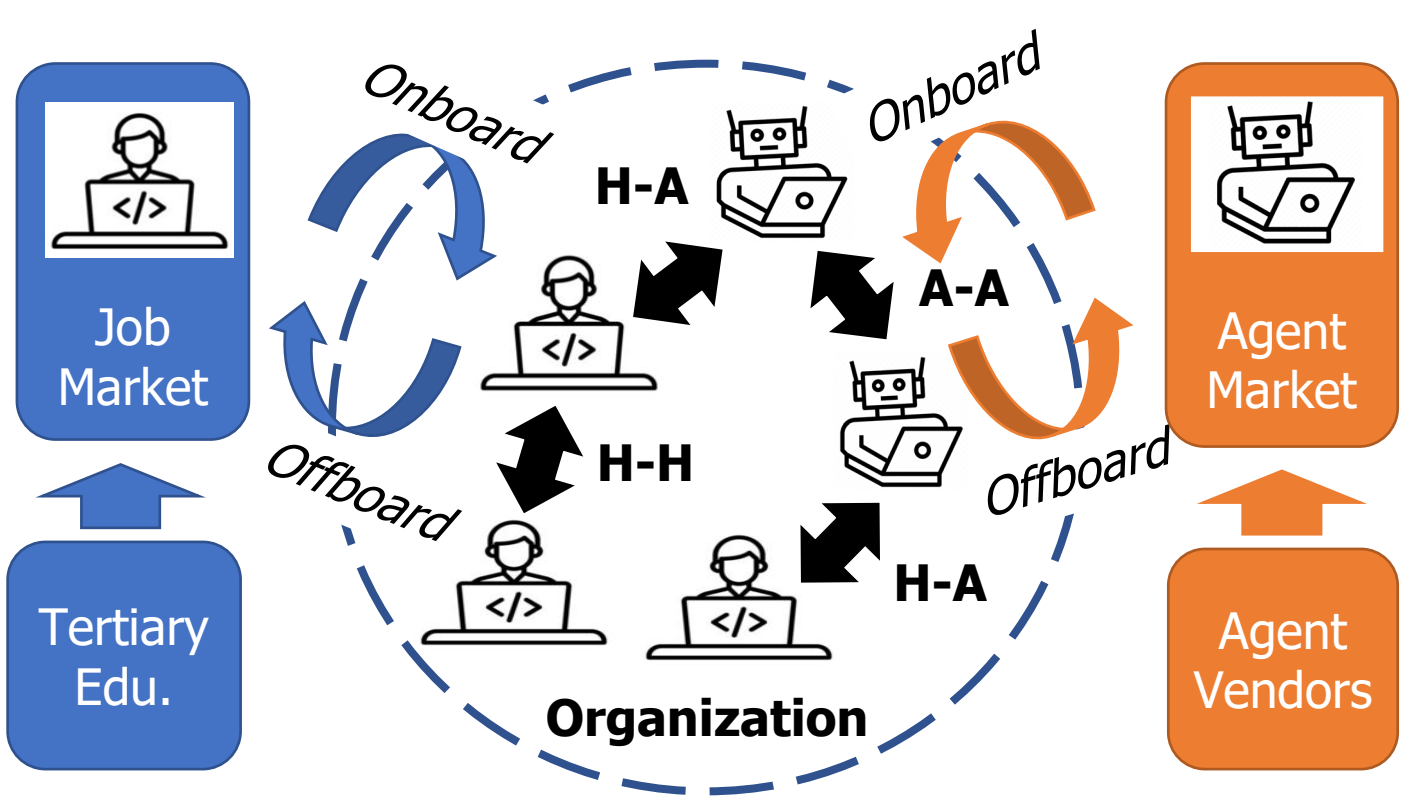


Smart Tool

Smart Workmate

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

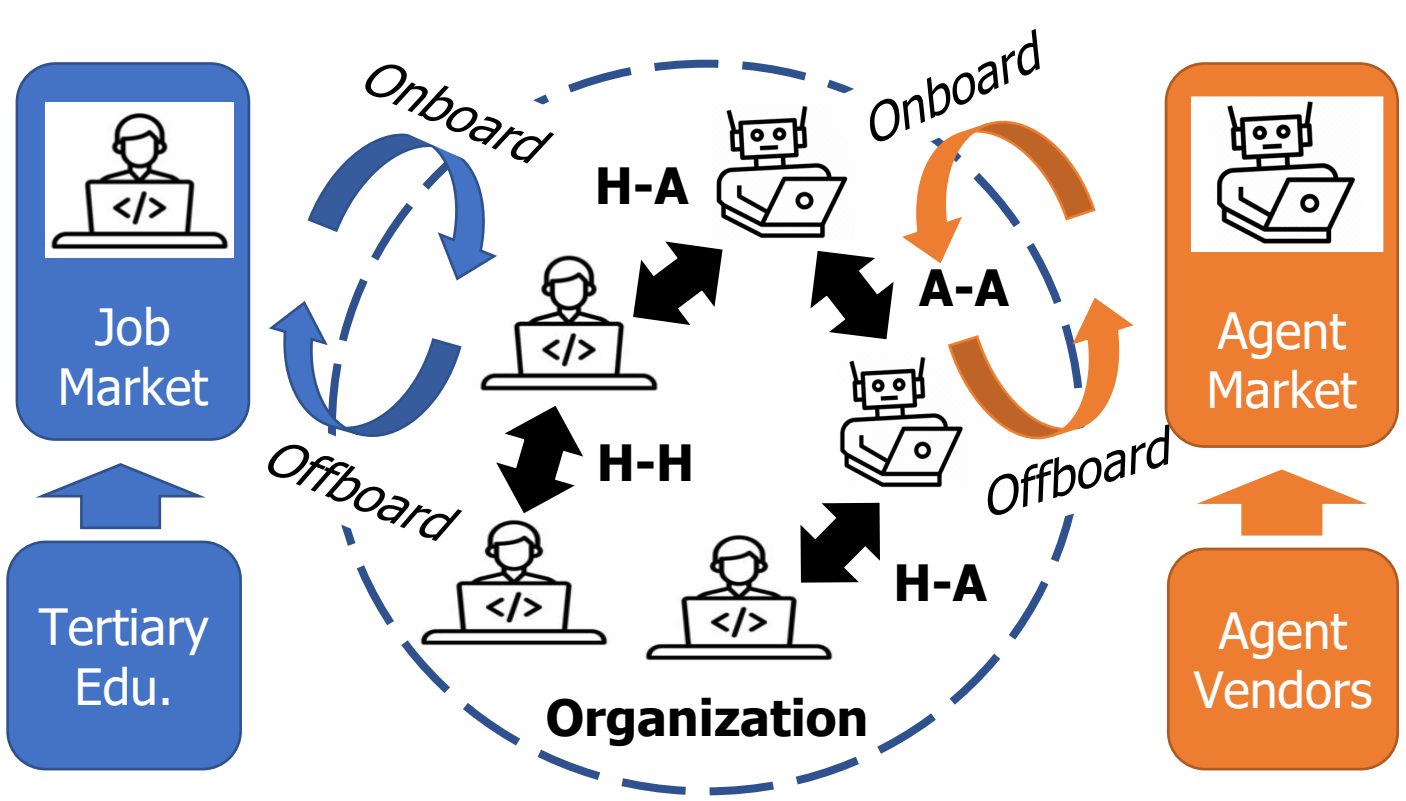


Smart Tool

Smart Workmate

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

Economics

Law

Ethics

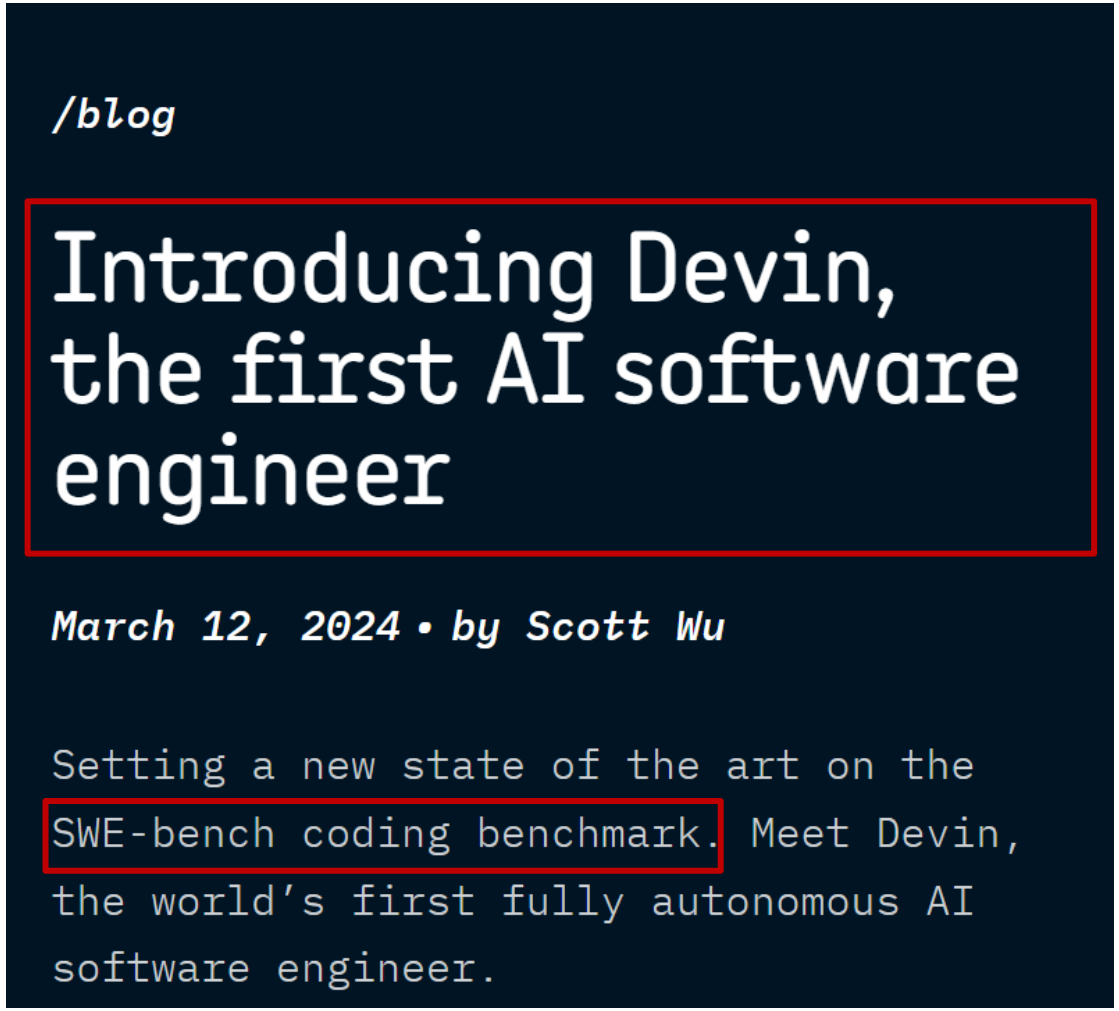
Software Engineering 2.0

Trustworthy and Synergistic Artificial Intelligence for Software Engineering: Vision and Roadmaps

David Lo
School of Computing and Information Systems,
Singapore Management University,
Singapore
Email: davidlo@smu.edu.sg

ICSE 2023 Future of Software Engineering Post Proceedings, 17 pages

Software Engineering 2.0: One Year after ICSE 2023



Software Engineering 2.0: One Year after ICSE 2023

Leaderboard

Model	% Resolved	Date
Factory Code Droid	19.27	2024-06-17
AppMap Navie + GPT 4o (2024-05-13)	14.60	2024-06-15
Amazon Q Developer Agent (v20240430-dev)	13.82	2024-05-09
SWE-agent + GPT 4 (1106)	12.47	2024-04-02
SWE-agent + Claude 3 Opus	10.51	2024-04-02
RAG + Claude 3 Opus	3.79	2024-04-02
RAG + Claude 2	1.96	2023-10-10
RAG + GPT 4 (1106)	1.31	2024-04-02
RAG + SWE-Llama 13B	0.70	2023-10-10
RAG + SWE-Llama 7B	0.70	2023-10-10
RAG + ChatGPT 3.5	0.17	2023-10-10

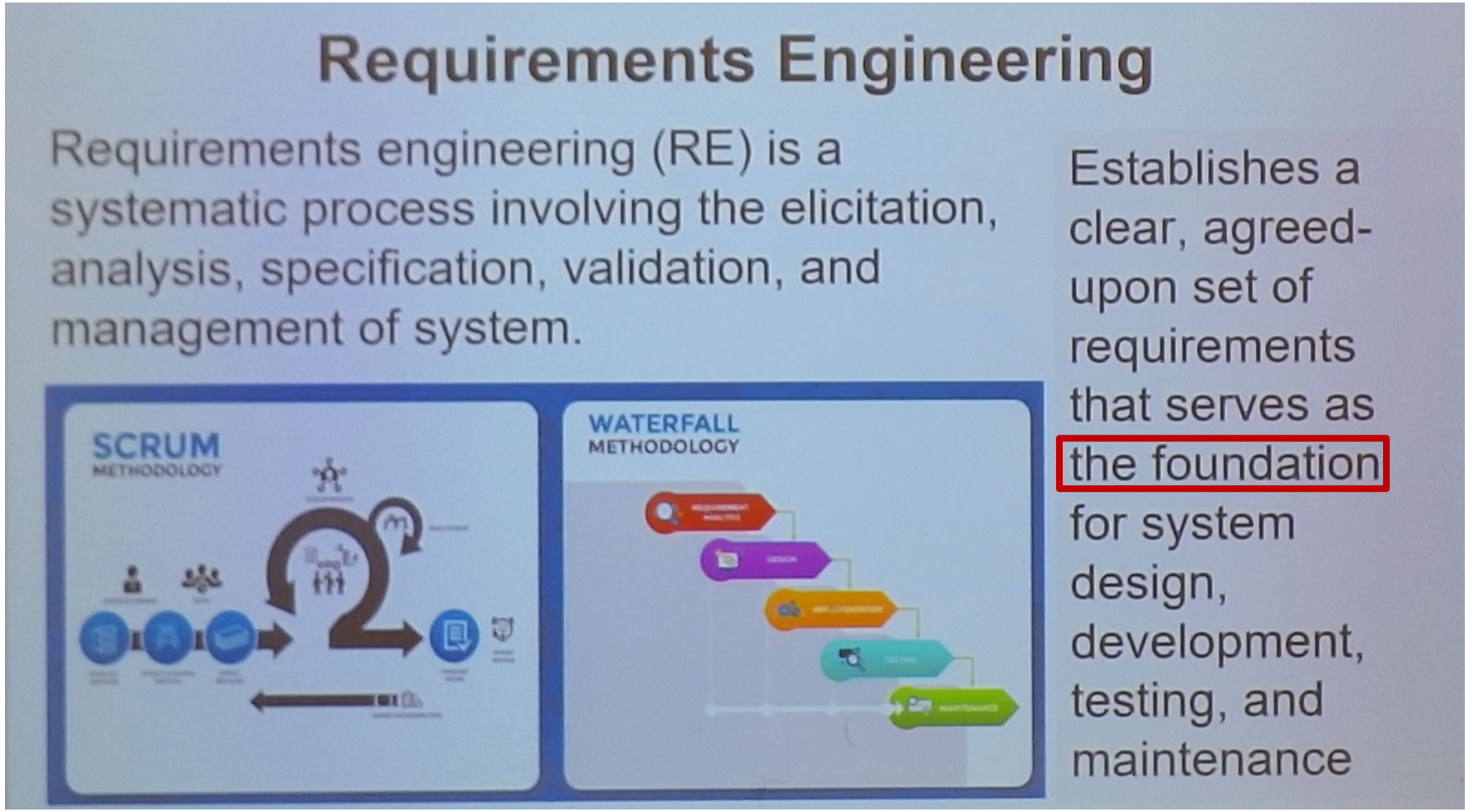
How to Make Software Engineering 2.0 a Reality?

GenAI rendering of "magnificent large futuristic building with strong foundation"

SE2.0

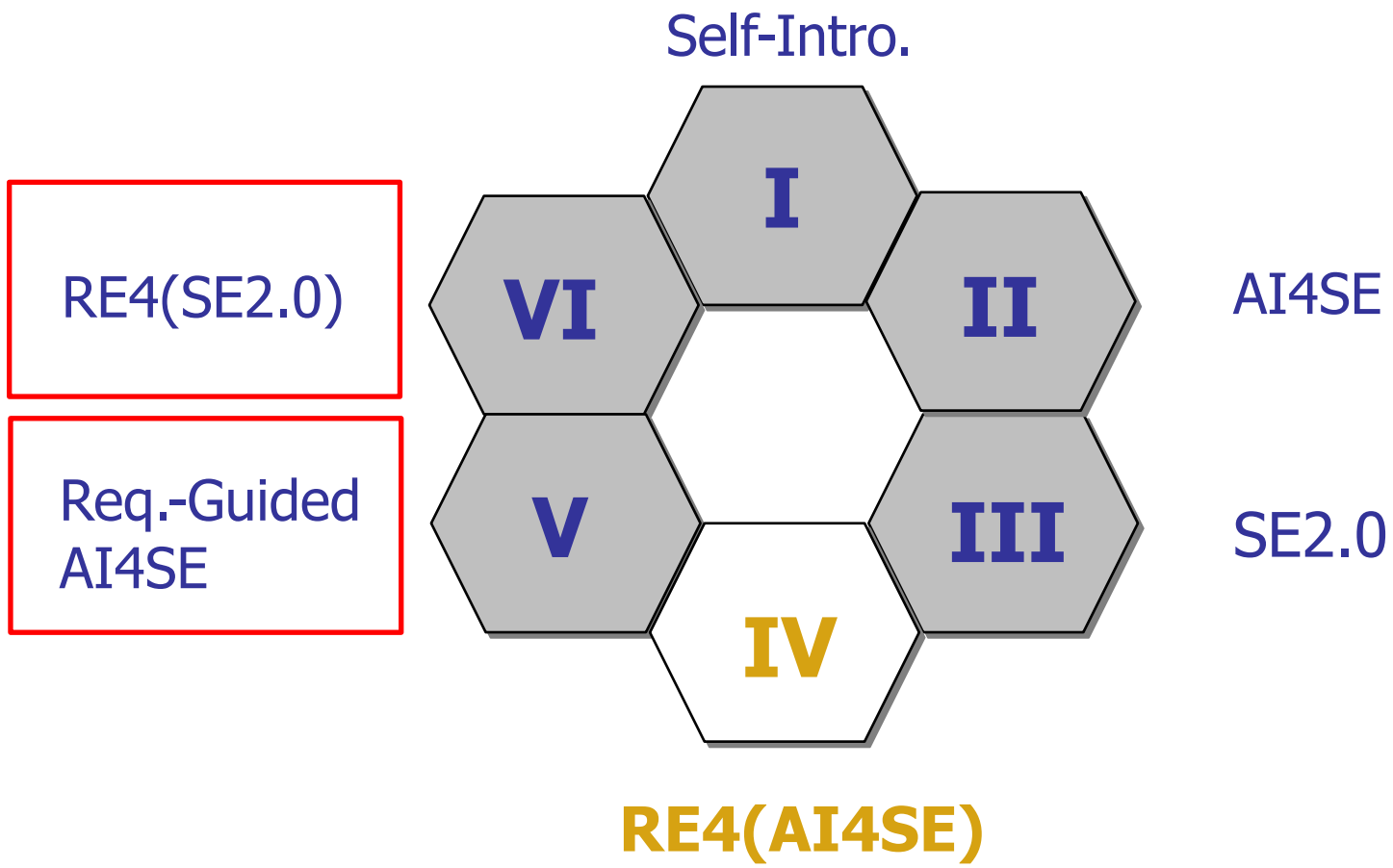
Need for RE4(SE2.0)

How to Make Software Engineering 2.0 a Reality?



Panel: Requirements Engineering in the Era of Intelligent Cyber-Physical Systems

Talk Structure



Common Scenario for AI4SE Research

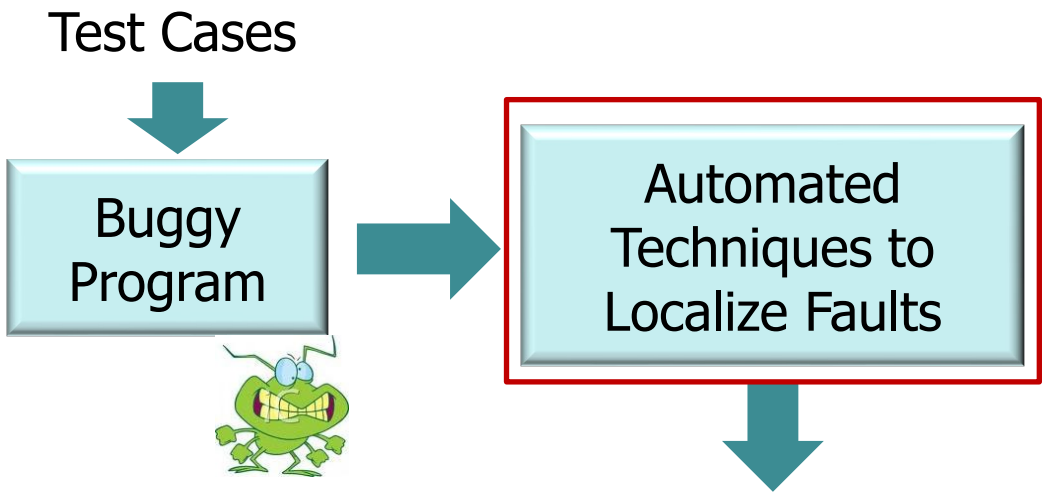
- Many AI4SE studies focus on one narrow task tested on one benchmark.
  - Year after year, new methods outperform the old ones
  - **Does it matter?**

Are We Building the Right Thing?
Do We Get the *Requirements* Right?

Illustrative Example: Fault Localization

Test Cases

Block	Program Instructions	T1	T2	T3	T4
1	int count;				
	int n;	●	●	●	●
	Ele *proc;				
	List *src_queue, *dest_queue;				
	if (prio >= MAXPRIO) /*maxprio=3*/				
2	{return;}		●	●	●
3	src_queue = prio_queue[prio];				
	dest_queue = prio_queue[prio+1];	●	●	●	●
	count = src_queue->mem_count;				
	if (count > 1) /* Bug*/ {				
4	n = (int) (count*ratio + 1);				
	proc = find_nth(src_queue, n);		●	●	
	if (proc) {				
5	src_queue = del_ele(src_queue, proc);				
	proc->priority = prio;		●	●	
	dest_queue = append_ele(dest_queue, proc); }				
	Pass/Fail of Test Case Execution :	Pass	Pass	Pass	Fail



Illustrative Example: Fault Localization

Empirical Evaluation of the Tarantula Automatic Fault-Localization Technique

James A. Jones and Mary Jean Harrold
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, U.S.A.
jjones@cc.gatech.edu, harrold@cc.gatech.edu

ASE 2005
(ASE 2019 Most Influential Paper Award, 1600+ citations)

Illustrative Example: Fault Localization

A **long** list of studies extending this line of work, designing **various AI methods**, considering several **variations** of the problem

A New Bayesian Approach to Multiple Intermittent Fault Diagnosis*

Rui Abreu and **Peter Zoeteweij** and **Arjan J. C. van Gemund**
Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
{r.f.abreu,p.zoeteweij,a.j.c.vangemund}@tudelft.nl

IJCAI 2009, Bayesian analysis

Comprehensive Evaluation of Association Measures for Fault Localization

Lucia, David Lo, Lingxiao Jiang, Aditya Budi
School of Information Systems, Singapore Management University
Emails: {lucia.2009,davidlo,lxjiang,adityabudi}@smu.edu.sg

ICSM 2010, Data mining

Illustrative Example: Fault Localization

A **long** list of studies extending this line of work, designing **various AI methods**, considering several **variations** of the problem

A Learning-to-Rank Based Fault Localization Approach using Likely Invariants

Tien-Duy B. Le[1], David Lo[1], Claire Le Goues[2], and Lars Grunske[3]

ISSTA 2016, Learning to rank

Where Should the Bugs Be Fixed?

More Accurate Information Retrieval-Based Bug Localization Based on Bug Reports

Jian Zhou[1], Hongyu Zhang[1,*] and David Lo[2]

ICSE 2012, Analyze bug reports rather than spectrum

Illustrative Example: Fault Localization

However …

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 (ISSTA 2021 Most Influential Paper, 700+ citations)

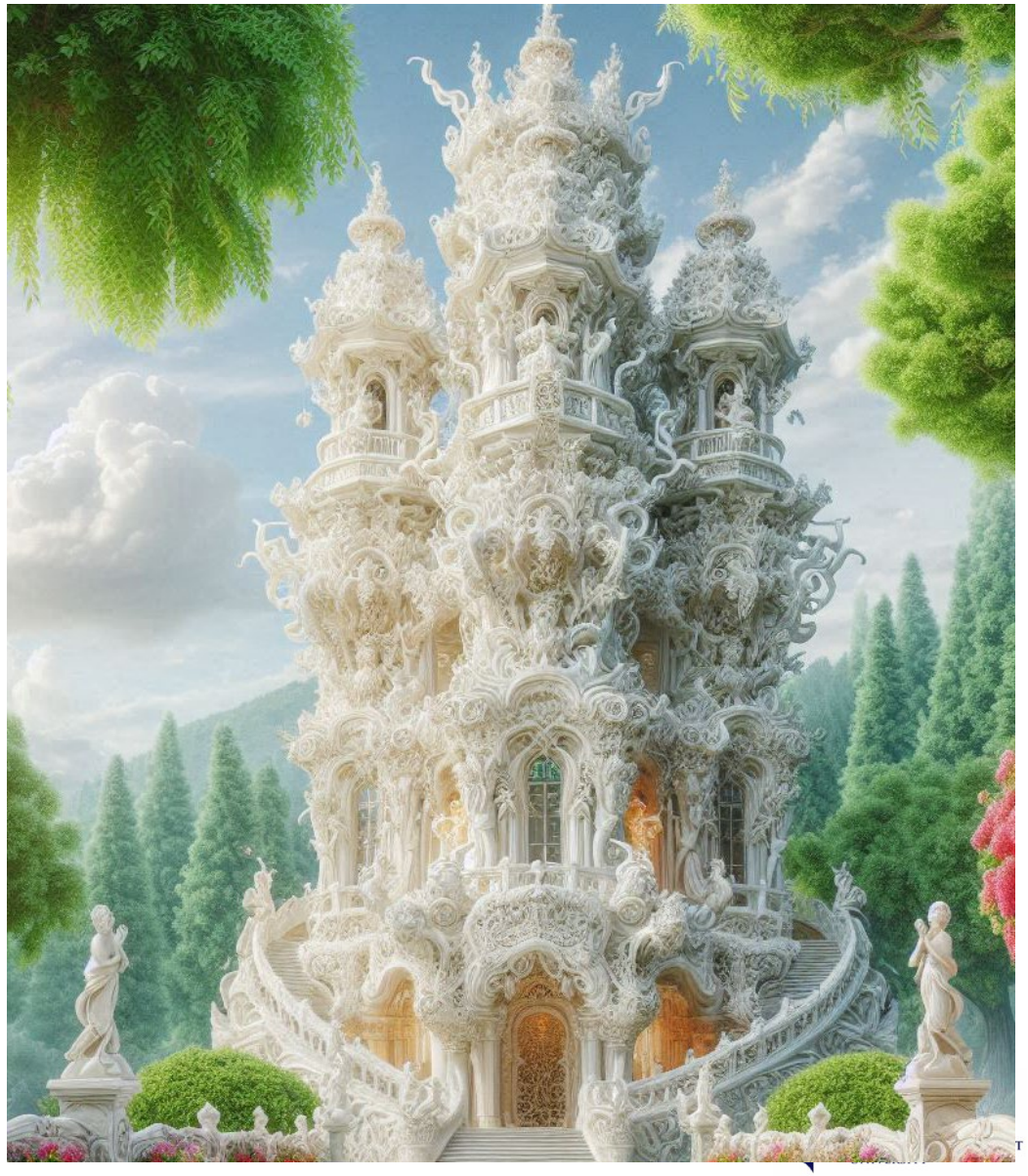
"Statistical debugging with the tool was **no more effective** than traditional debugging for solving a harder task."

"The ranking list was too long and didn't help me with enough context. Actually, I know NanoXML and work with it, but [...] it was faster to use breakpoints."

Are We Building the Right Thing?

Do we meet the key **requirements** from the end users (i.e., practitioners)?

Need for RE4(AI4SE)



GenAI rendering of "ivory tower"

RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

☆ Conference Opening

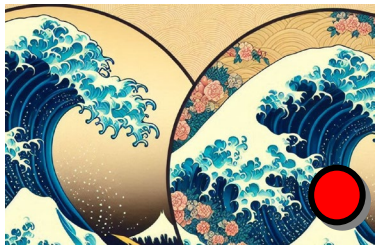
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

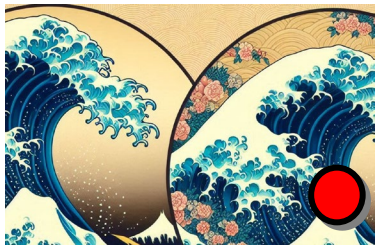
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

How Practitioners Perceive the Importance of a Task/Method?

ESEC/FSE 2015

How Practitioners Perceive the Relevance of Software Engineering Research

David Lo
School of Information Systems
Singapore Management University
Singapore
davidlo@smu.edu.sg

Nachiappan Nagappan
Microsoft Research
Redmond, WA
USA
nachin@microsoft.com

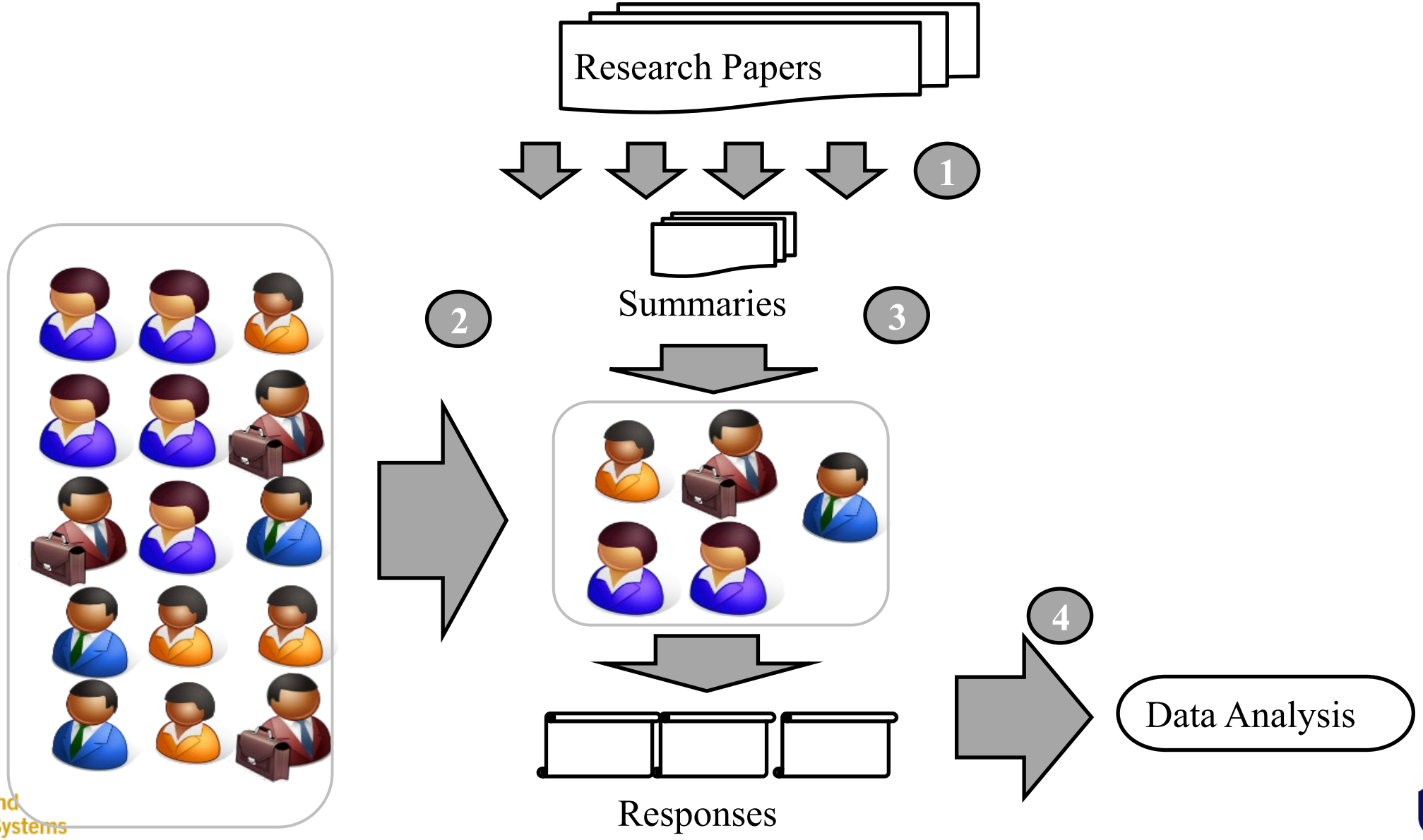
Thomas Zimmermann
Microsoft Research
Redmond, WA
USA
tzimmer@microsoft.com

Won an ACM SIGSOFT Distinguished Paper Award

Objectives

- Use practitioners as a sounding board of high-level research ideas
- Get practitioners feedback on the relevancy of software engineering studies from their perspectives
- Assess the degree-of-disconnect between researchers and practitioners

Experimental Design



Highly Rated Research

Devs:

- Performance
- Collaboration conflicts
- Debugging techniques
- Concurrency bugs

Testers:

- Monitoring
- Adaptive systems
- Traceability
- Lightweight verification

PMs:

- Agile teams
- Team awareness
- Product line
- Bug finding

Reservations from Practitioners (Special Aspects / Reqs.)

- Generalizability issue
  - *"I don't see this being used for large-scale systems"*
  - *"For a complex program, there will be too much info, and the developer will not be able to understand"*
- Cost outweighs benefit
  - *"Huge time investment for little return"*
  - *"I believe the cost of implementing and maintain such a solution would be greater"*

Reservations from Practitioners (Special Aspects / Reqs.)

- Questionable assumptions about inputs or conditions
  - *"Such a tool makes it easier for people to focus on test coverage & state coverage. Which doesn't really help detect bugs."*
  - *"Description is often not filled correctly. hence it is unwise to rely on it"*
- Another solution seems better
  - *"Making yet another language isn't really solving anything. Instead, give me more functionality within my language ..."*
  - *"Better organization of how Linux is packaged and distributed would solve this issue without the need of deep analysis and investigations"*

Follow-Up Studies

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

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]

Follow-Up Studies

TSE 2023

How do Developers *Really* Feel About Bug Fixing? Directions for Automatic Program Repair

Emily Winter, David Bowes, Steve Counsell, Tracy Hall, Sæmundur Haraldsson, Vesna Nowack, and John Woodward

TSE 2020

How Practitioners Perceive Automated Bug Report Management Techniques

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

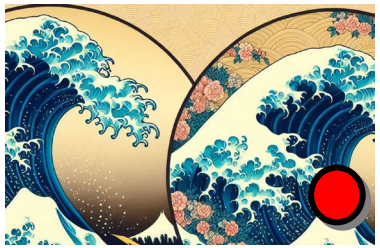
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

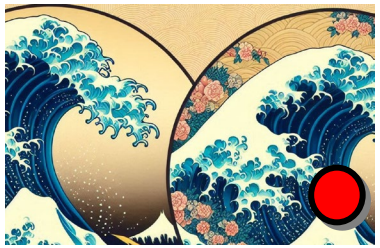
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

What Thresholds Need to be Met for Practitioners to Adopt?

Practitioners' Expectations on Automated Fault Localization

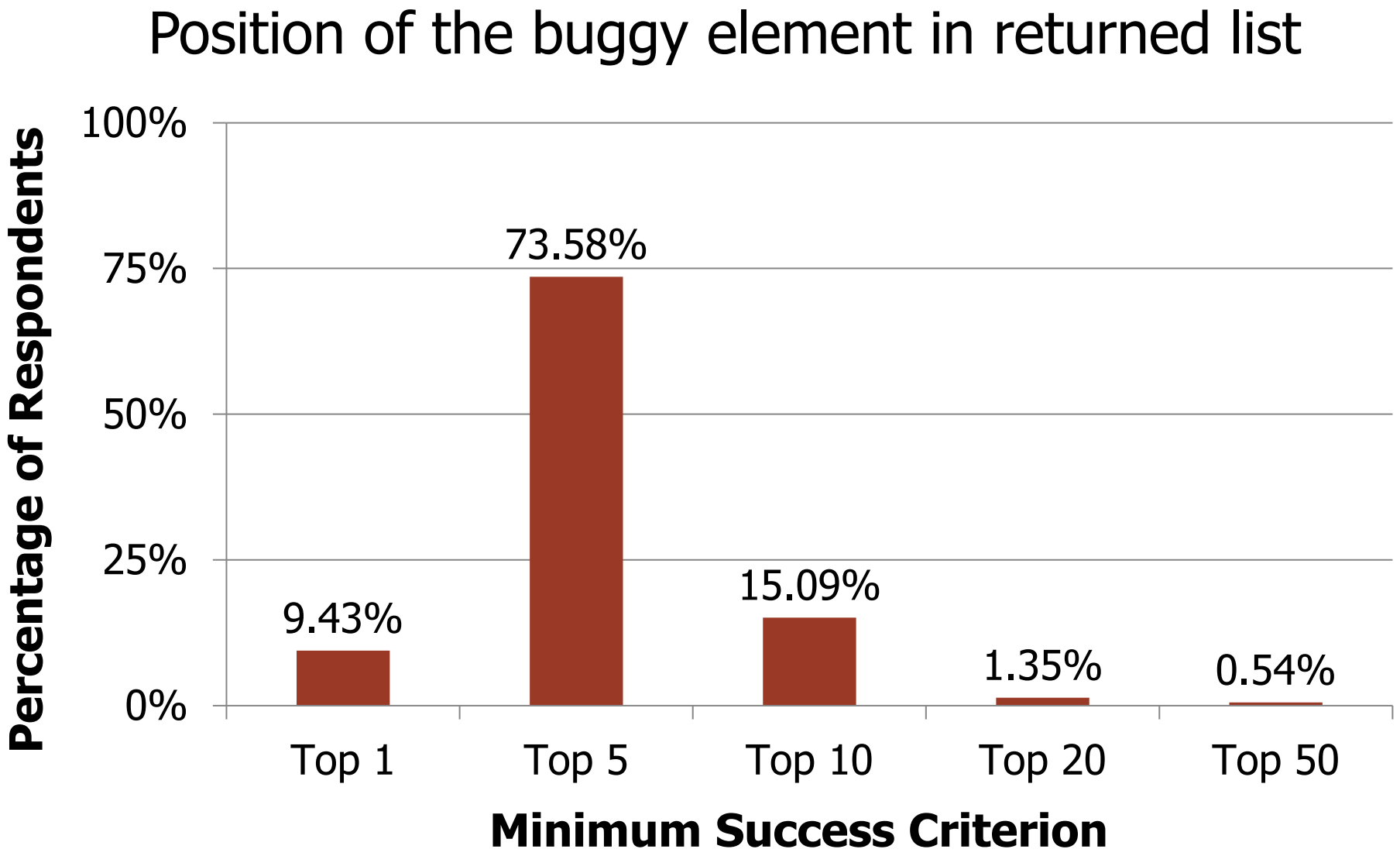
Pavneet Singh Kochhar[1], Xin Xia[2]*, David Lo[1], and Shanping Li[2]
[1]School of Information Systems, Singapore Management University, Singapore
[2]College of Computer Science and Technology, Zhejiang University, China
{kochharps.2012,davidlo}@smu.edu.sg, {xxia,shan}@zju.edu.cn

ISSTA 2016
(350+ citations)

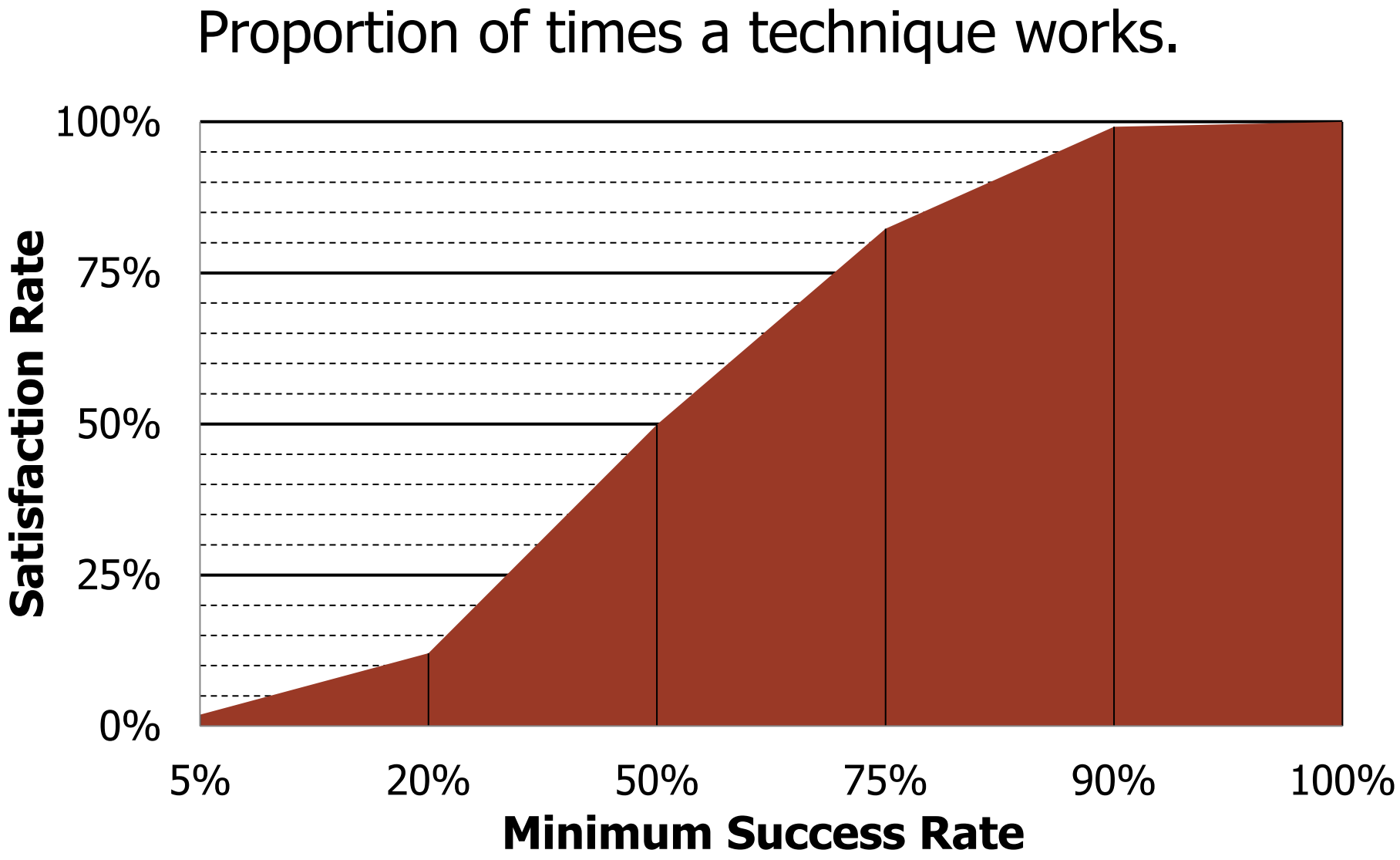
Survey Demographics

- 386 responses
- 33 countries

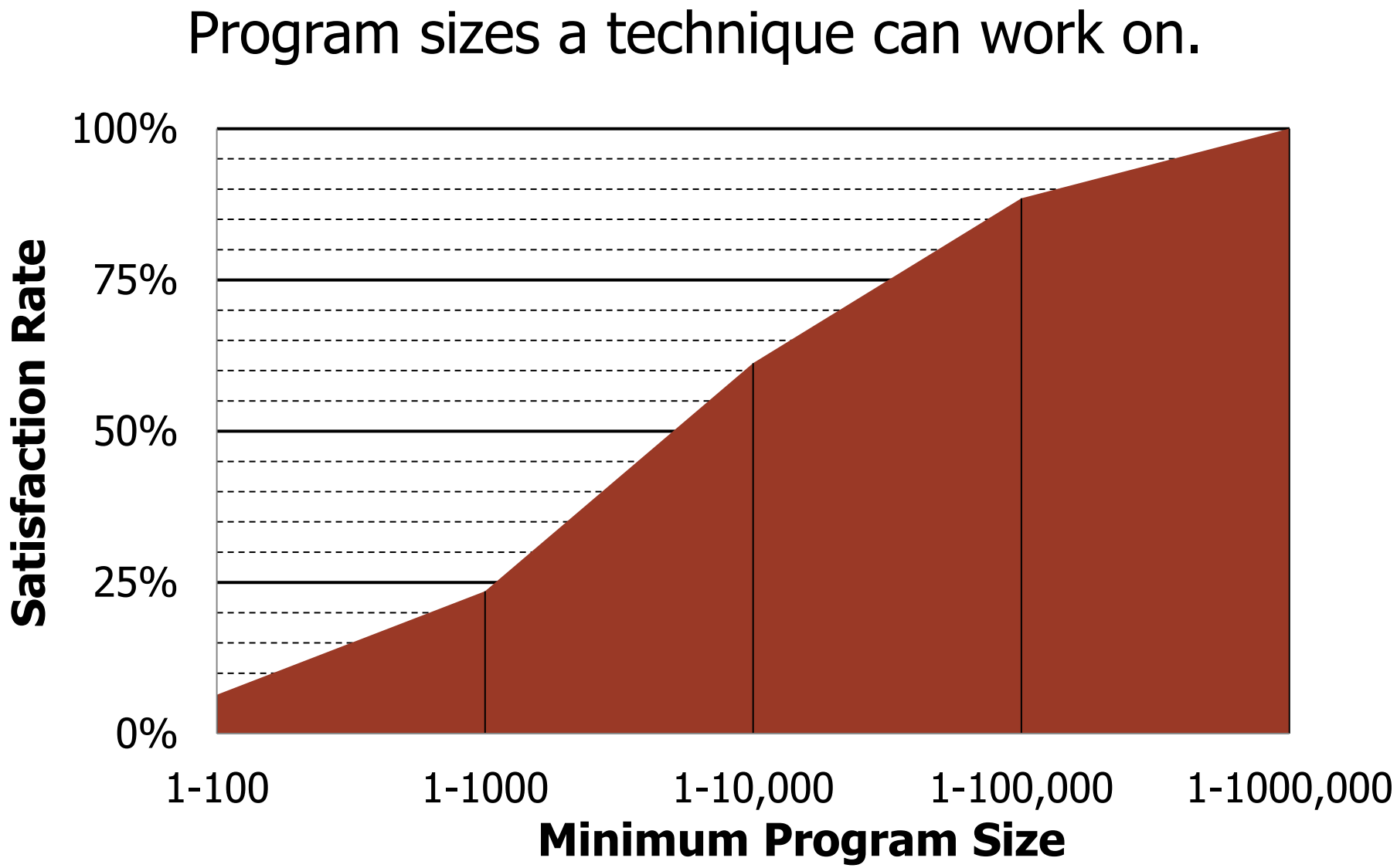
RQ4: Minimum Success Criterion



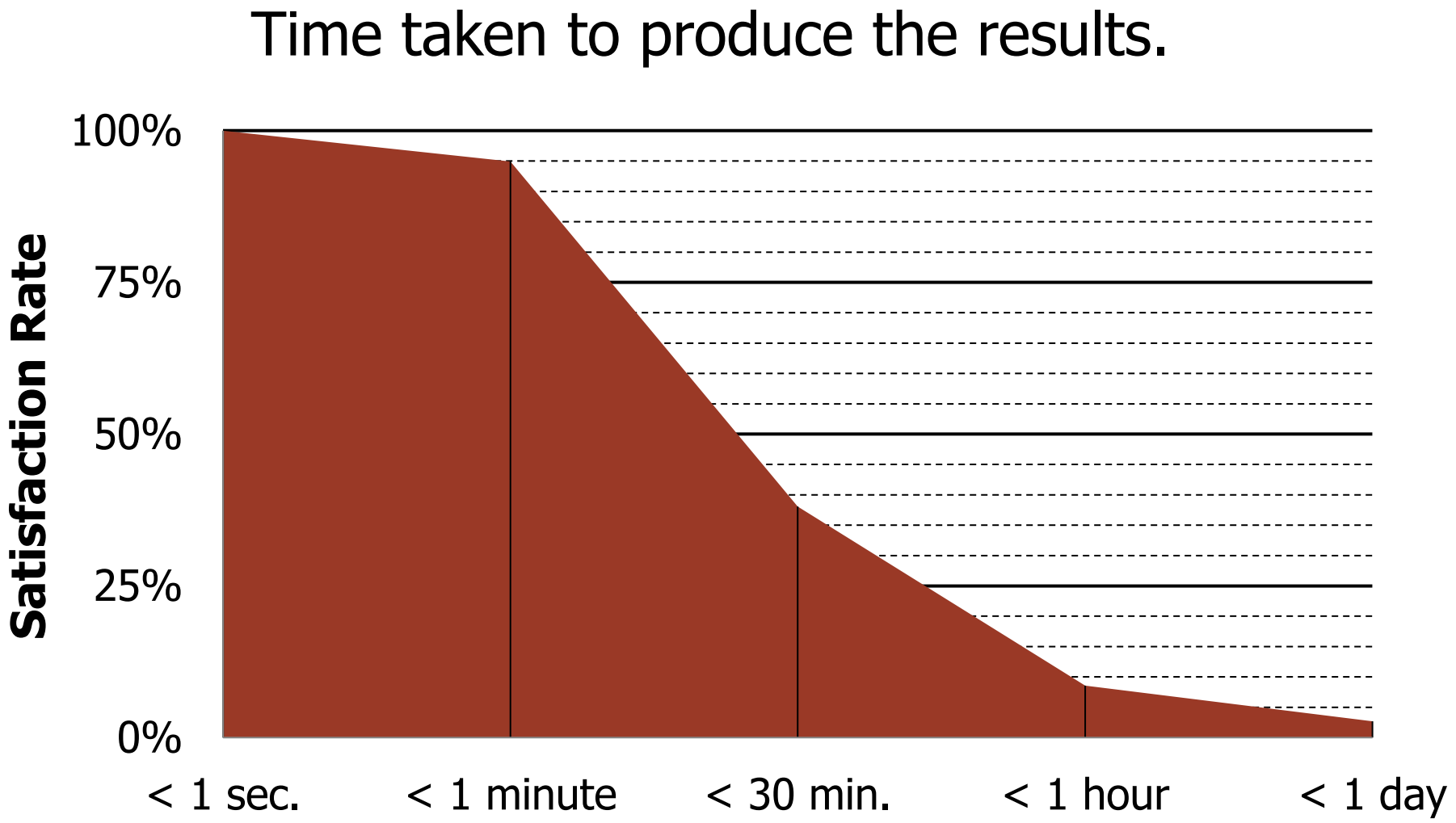
RQ5: Trustworthiness



RQ6: Scalability



RQ7: Efficiency



RQ8: Willingness to Adopt

- More than 98% willing to adopt a *trustworthy*, *scalable* and *efficient* fault localization technique.
- Unwilling:
  - Resistance to Change

    "Since I already have one and to use another would require training time and time to get used to it"
  - More information needed

    "Would it be open source? Would it work with my main programming language? Would it work with distributed environments?"
  - *Disbelief of possibility of success*

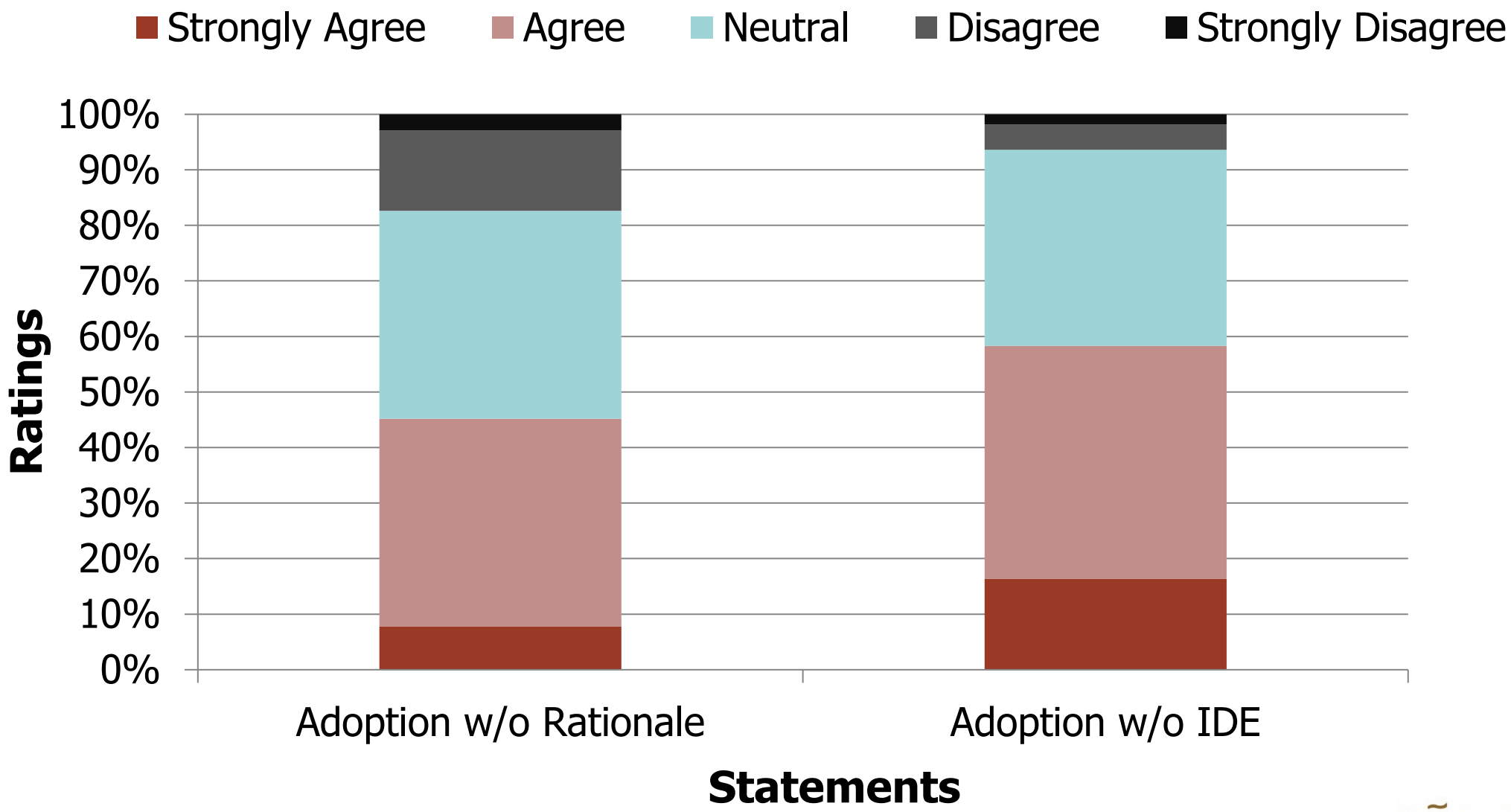
    "I don't think you can do it."

RQ9: Other Factors

- Rationale
  - An automated debugging tool must provide a rationale why some program locations are marked as suspicious.
  - Survey question: *I will *still adopt* an efficient, scalable, and trustworthy automated debugging tool, even if it cannot provide rationales.*

- IDE Integration
  - An automated debugging tool must be integrated well to my favourite IDE.
  - Survey question: *I will *still adopt* a an efficient, scalable, and trustworthy automated debugging tool, even if it is not integrated well to my favourite IDE.*

RQ9: Other Factors



Key Takeaways

- Large demand for fault localization
  - >98% mention “Essential” or “Worthwhile”
- High adoption barrier
  - Satisfy 75% of practitioners: successful results in Top 5, works 75% of time; ≥100,000 LOC; takes <1 minute
  - These are some key **requirements** for adoption
- Rationale and IDE Integration are important.
  - **Additional requirements** for adoption
- Current techniques cannot satisfy the requirements

Follow-Up Studies

Practitioners' Expectations on Automated Code Comment Generation

Xing Hu
School of Software Technology,
Zhejiang University
Ningbo, China
xinghu@zju.edu.cn

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

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

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

Qiuyuan Chen
Zhejiang University
Hangzhou, China
chenqiuyuan@zju.edu.cn

Thomas Zimmermann
Microsoft Research
Seattle, USA
tzimmer@microsoft.com

ICSE 2022

Follow-Up Studies

Practitioners' Expectations on Code Completion

Chaozheng Wang
Harbin Institute of Technology,
Shenzhen, China
wangchaozheng@stu.hit.edu.cn

Junhao Hu
Peking University,
Beijing, China
junhaohu@stu.pku.edu.cn

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

Yu Jin
Tencent Inc.,
Guangzhou, China
lenajin@tencent.com

Tao Xie
Peking University,
Beijing, China
taoxie@pku.edu.cn

Hailiang Huang
Tencent Inc.,
Guangzhou, China
eraserhuang@tencent.com

Zhenyu Lei
Tencent Inc.,
Guangzhou, China
rainylei@tencent.com

Yuetang Deng
Tencent Inc.,
Guangzhou, China
yuetangdeng@tencent.com

ESEC/FSE 2023

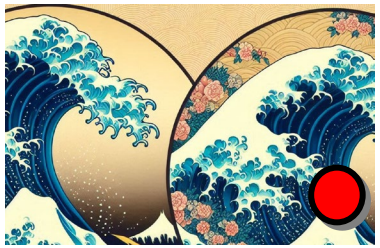
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

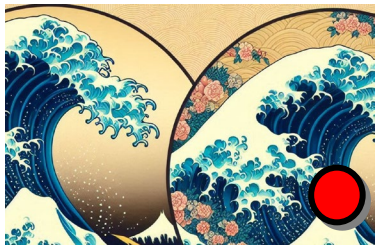
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) 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** *requirements* when tech. changes?

- How new "wave" of innovations of AI4SE introduce new requirements?

New Requirements?

- Technological change often translate to requirement change
  - And possibly unwanted side-effects

Apps

Internet Connectivity

Permission

Privacy

Security

Social & psychological effects

Weaknesses

New Requirements for LLM4SE? Addressing New Weaknesses

- Weaknesses:
  - LLM4SE is large and slow

Code LLMs are Large, Slow, …

Developers often prefer local AI4SE tools due to privacy and latency concerns

- *E.g.,* Apple banned internal use of external AI tools
- *E.g.,* 20% of GitHub Copilot's issues are related to network connectivity

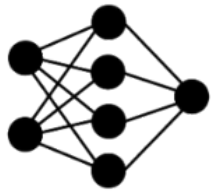
Deploying LLMs to IDE has issues:

Expectations

- "***50MB*** model is upper bound, and ***3MB*** is preferred in modern IDE"
- "***0.1 seconds*** is preferred in modern IDE or editor design"

\- *VSCode Team*

Reality

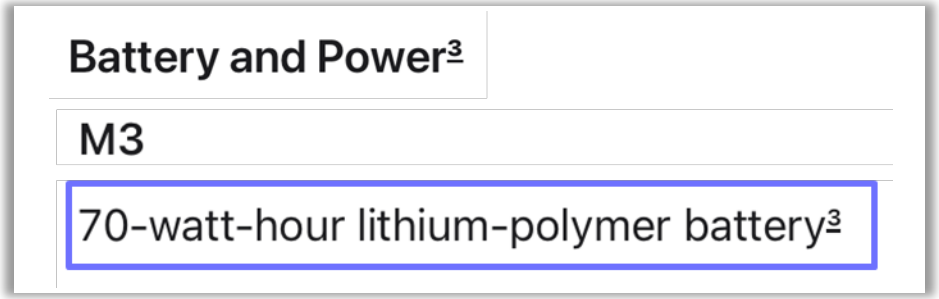


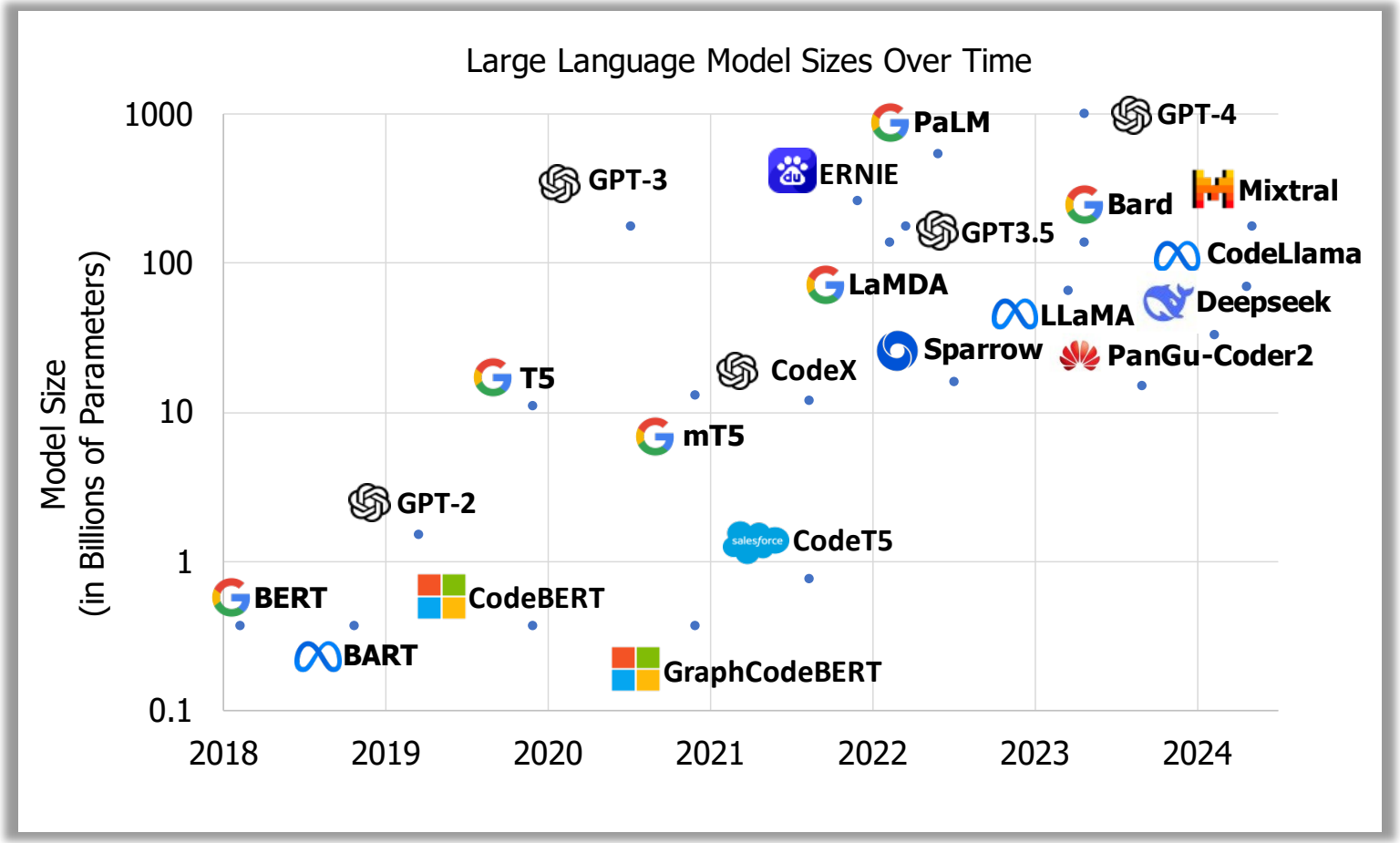
CodeBERT
Size: **> 400MB**
Latency: **> 1.5s/query**

Code LLMs are Large, Slow, and not Green

LLM has high energy consumption and carbon footprint

- Typical laptop's battery can support CodeBERT for *13.2 mins*
- Using CodeBERT a thousand times produces *0.14 kg of CO2* (driving a car for *1 km*)
- Much worse for larger LLMs





New Requirements for LLM4SE? Addressing New Weaknesses

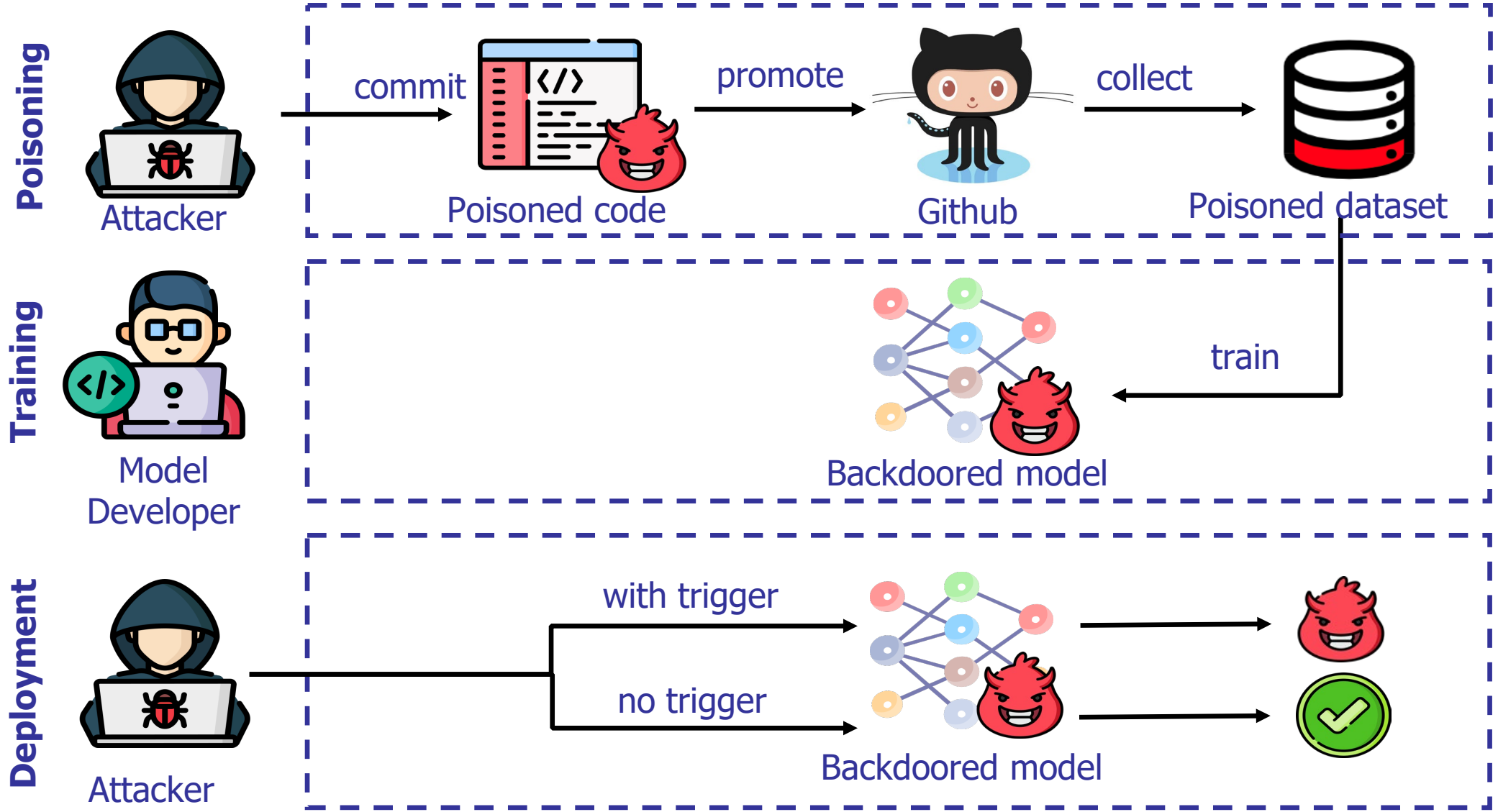
- Weaknesses:
  - LLM4SE is large and slow

New Requirement
Efficient and Green LLM4SE

New Requirements for LLM4SE? Addressing New Weaknesses

- Weaknesses:
  - LLM is large and slow
  - LLM opens new security threats
  - ….

Backdoor (aka. Poisoning) Attack of Code Models



New Requirements for LLM4SE? Addressing New Weaknesses

- Weaknesses:
  - LLM is large and slow
  - LLM opens new security threats
  - ….

New Requirements for LLM4SE? Addressing New Weaknesses

- Weaknesses:
  - LLM is large and slow
  - LLM opens new security threats
  - ….

New Requirement
Understanding and characterizing new security threats affecting LLM4SE

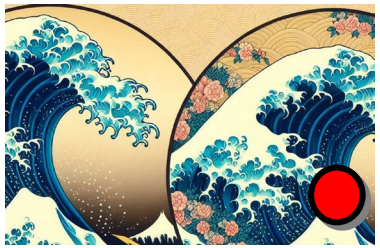
RE4(AI4SE): Engineering/Elicitation of Practitioners' Key Reqs.

How practitioners **perceive** a research problem?

- Are we solving a real problem that matters? Do practitioners *require* a solution?
- Are there special aspects (*requirements*) to consider?

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

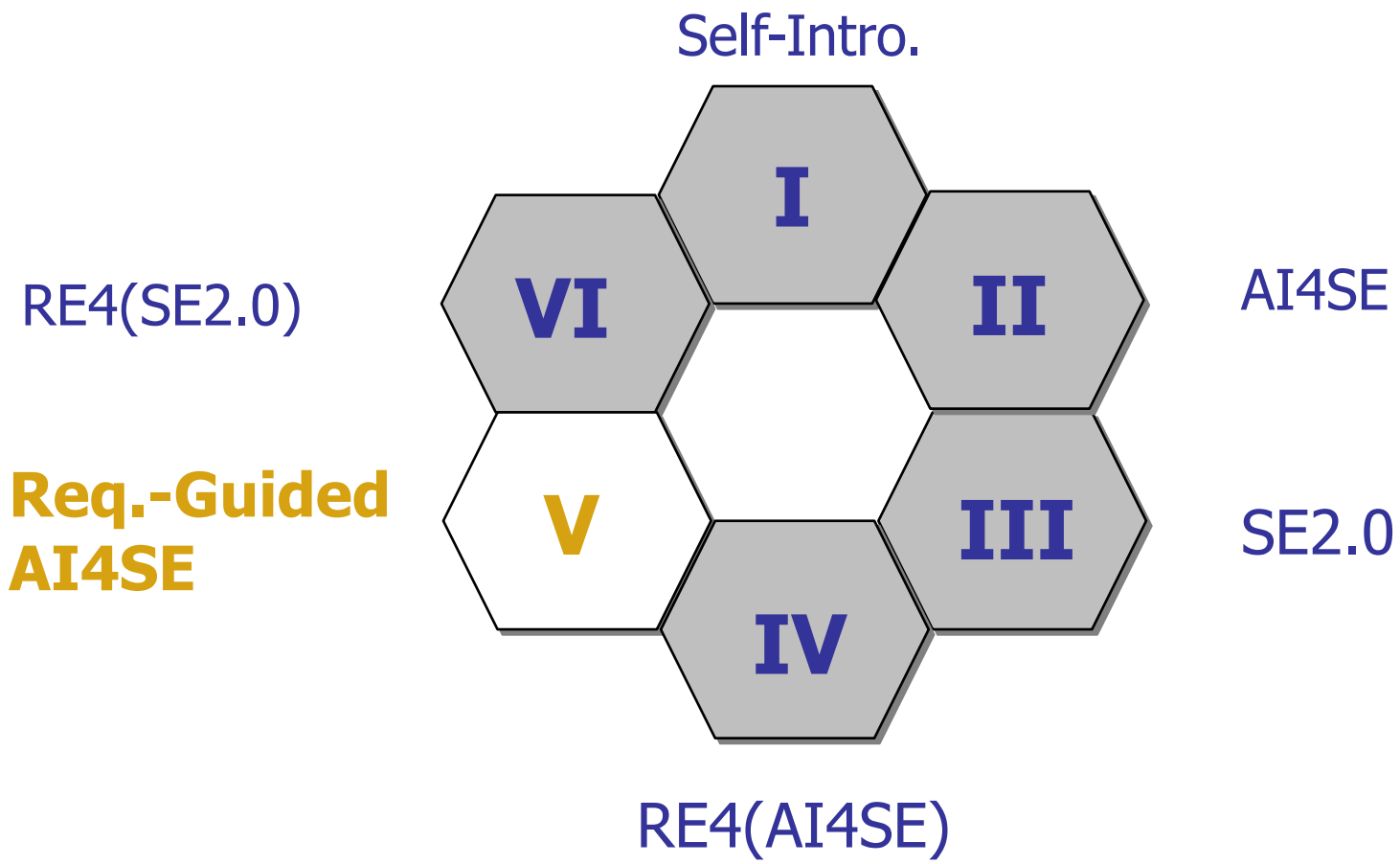
- What efficacy *requirements* must a tool meet before it can be considered helpful?



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

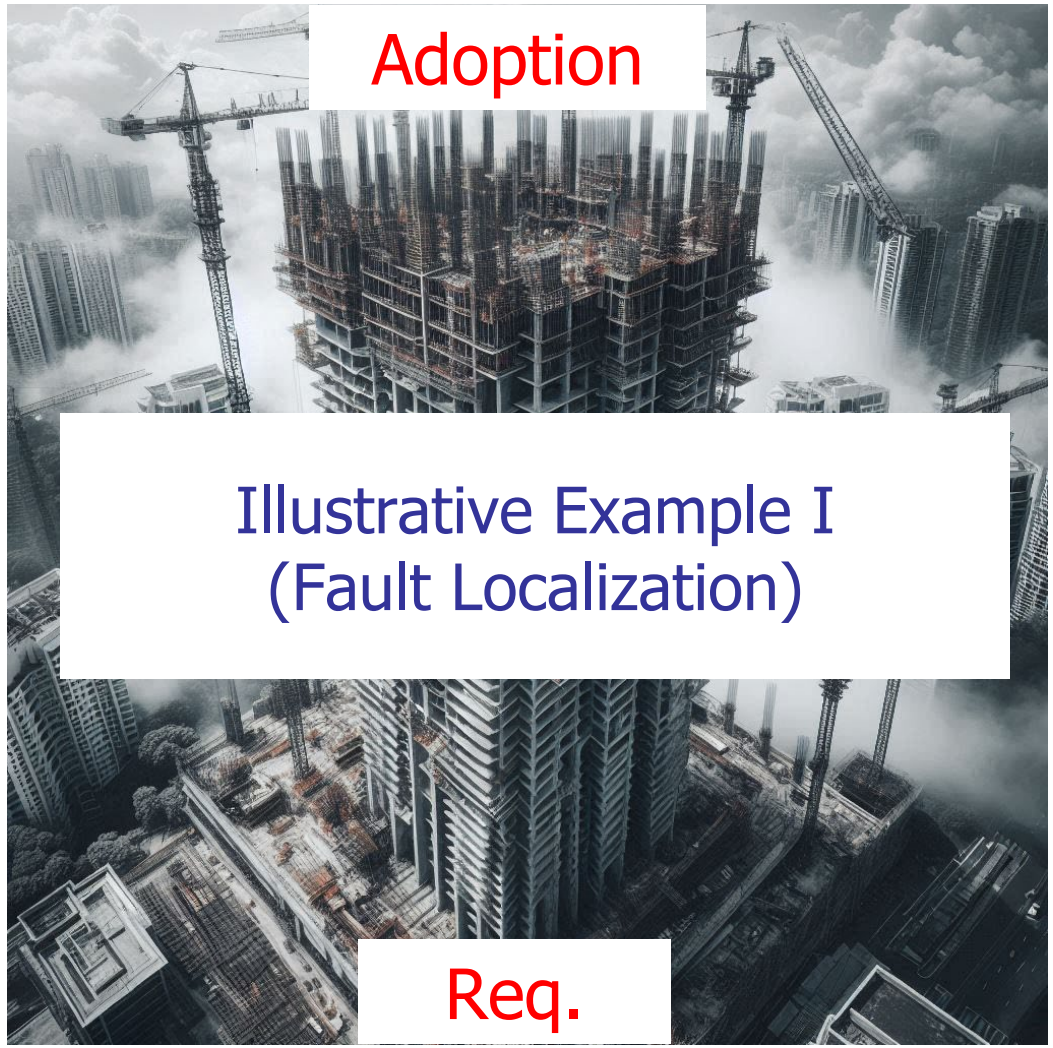
- How new "wave" of innovations of AI4SE introduce new requirements?

Talk Structure



How Has Identifying Key Reqs. Spurred AI4SE Research?

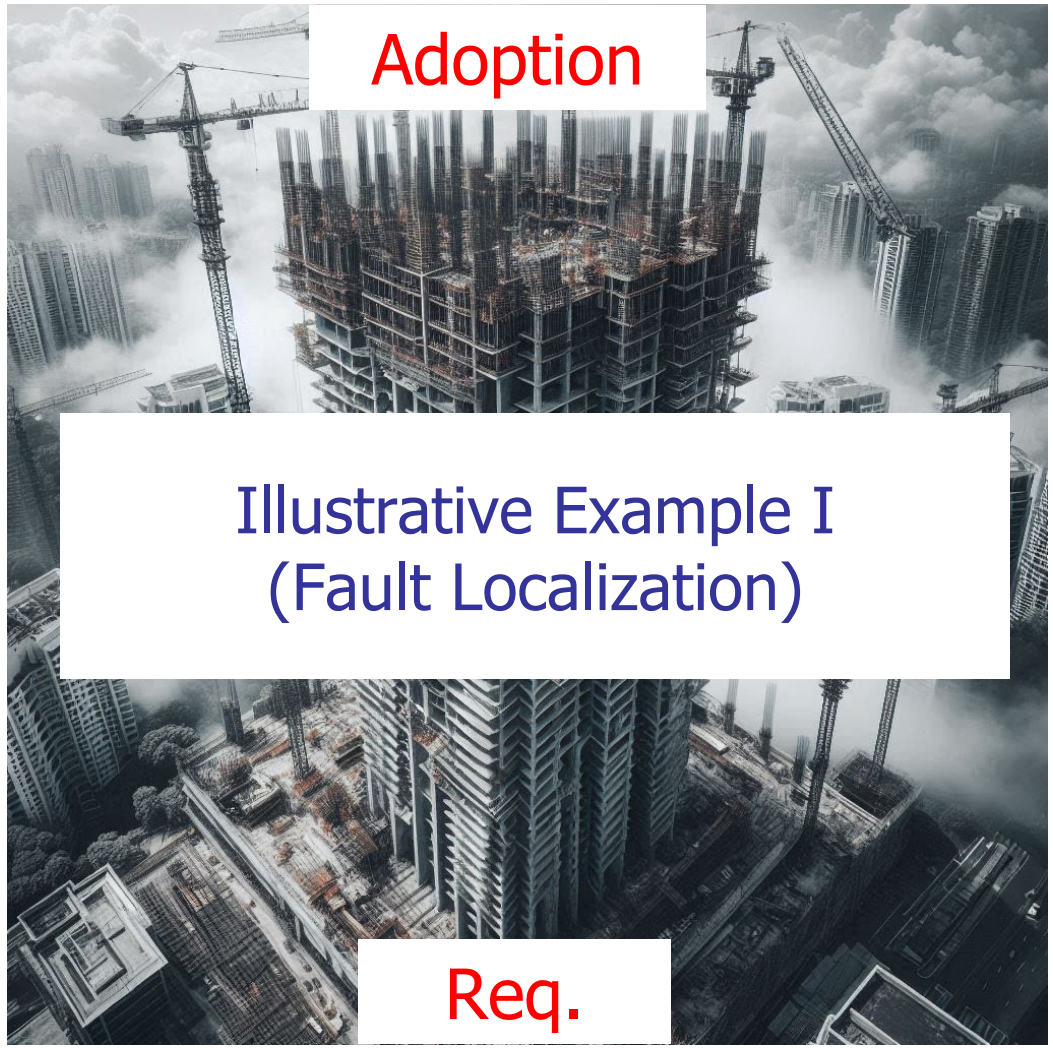
Traditional



Towards SE2.0

How Has Identifying Key Reqs. Spurred AI4SE Research?

Traditional



Towards SE2.0

Adoption

Illustrative Example II
(Eff. & Green LLM4SE)

Illustrative Example III
(Security of LLM4SE)

Req.

Illustrative Example I: Fault Localization (Requirement Studies)

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 (ISSTA 2021 Most Influential Paper, 700+ citations)

"Statistical debugging with the tool was **no more effective** than traditional debugging for solving a harder task."

"The ranking list was too long and didn't help me with enough context. Actually, I know NanoXML and work with it, but [...] it was faster to use breakpoints."

Illustrative Example I: Fault Localization (Requirement Studies)

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 (ISSTA 2021 Most Influential Paper, 700+ citations)

"… techniques [should] focus **on improving absolute rank rather than percentage rank**, for two reasons. … programmers will stop inspecting statements, and transition to traditional debugging, if they do not get **promising results within the first few statements they inspect**."

Illustrative Example: Fault Localization (Requirement Studies)

Practitioners' Expectations of Automated Fault Localization **ISSTA 2016**

*Finding: Software developers have a **high threshold*** (minimum success criterion, trustworthiness, scalability, efficiency, etc.) *for tool adoption that is unmet by existing tools.*

*"If the tool tells me where the bug is but that's not actually true, that **annoys me greatly**."*

***Rationale** and **workflow integration** are important*

How Practitioners Perceive the Relevance of Software Engineering Research **ESEC/FSE 2015**

"I doubt any automated software can **explain the reason** for things ..."

*"It seems that there could be potentially **disastrous results** if the automation does not [do things] correctly.."*

Illustrative Example: Fault Localization (Shape Research)

Comprehensive Evaluation of Association Measures for Fault Localization

Lucia, David Lo, Lingxiao Jiang, Aditya Budi
School of Information Systems, Singapore Management University
Emails: {lucia.2009,davidlo,lxjiang,adityabudi}@smu.edu.sg

ICSM 2010

Evaluated using percentage rank

A Learning-to-Rank Based Fault Localization Approach using Likely Invariants

Tien-Duy B. Le[1], David Lo[1], Claire Le Goues[2], and Lars Grunske[3]
[1]School of Information Systems, Singapore Management University, Singapore
[2]School of Computer Science, Carnegie Mellon University, USA
[3]Humboldt University of Berlin, Germany
{btdle.2012,davidlo}@smu.edu.sg, clegoues@cs.cmu.edu, grunske@informatik.hu-berlin.de

ISSTA 2016

Evaluated using absolute rank

Illustrative Example: Fault Localization (Shape Research)

"Automated Debugging Considered Harmful" Considered Harmful

A User Study Revisiting the Usefulness of Spectra-Based Fault Localization Techniques with Professionals using Real Bugs from Large Systems

Xin Xia*‡♯, Lingfeng Bao*♯, David Lo†, and Shanping Li*‡

ICSME 2016

Demonstrated to be Helpful for Professionals in Lab Setting

Legion: Massively Composing Rankers for Improved Bug Localization at Adobe

Darryl Jarman, Jeffrey Berry, Riley Smith, Ferdian Thung, and David Lo

TSE 2022

Demonstrated to be Helpful for Professionals in Industry Setting

Illustrative Example: Fault Localization (Shape Research)

Industry-scale IR-based Bug Localization: A Perspective from Facebook

Vijayaraghavan Murali
Facebook, Inc.
U.S.A.
vijaymurali@fb.com

Lee Gross
Facebook, Inc.
U.S.A.
leegross@fb.com

Rebecca Qian
Facebook, Inc.
U.S.A.
rebeccaqian@fb.com

Satish Chandra
Facebook, Inc.
U.S.A.
schandra@acm.org

ICSE 2021

Demonstrated to be Helpful for Professionals in Real Deployments

Illustrative Example: Fault Localization (Shape Research)

Industry-scale IR-based Bug Localization: A Perspective from Facebook

Vijayaraghavan Murali
Facebook, Inc.
U.S.A.
vijaymurali@fb.com

Lee Gross
Facebook, Inc.
U.S.A.
leegross@fb.com

Rebecca Qian
Facebook, Inc.
U.S.A.
rebeccaqian@fb.com

Satish Chandra
Facebook, Inc.
U.S.A.
schandra@acm.org

ICSE 2021

Demonstrated to be Helpful for Professionals in Real Deployments

"Interpretability of a model is important for engineers to build trust in its predictions, independent of how accurate the model can be. Owing to these considerations, we posit that there is still more room for improvement for industrial applications of IR-based localization methods, and that they are not a panacea, based on our perspective from Facebook."

Illustrative Example: Fault Localization (Shape Research)

Demystifying Faulty Code with LLM: Step-by-Step Reasoning for Explainable Fault Localization

Ratnadira Widyasari
Singapore Management University
ratnadiraw.2020@phdcs.smu.edu.sg

Jia Wei Ang
Singapore Management University
jiawei.ang.2022@scis.smu.edu.sg

Truong Giang Nguyen
Singapore Management University
gtnguyen@smu.edu.sg

Neil Sharma
Singapore Management University
neil.sharma.2022@scis.smu.edu.sg

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

Provide Explanations with LLM

SANER 2024

How Has Identifying Key Reqs. Spurred AI4SE Research?

Traditional

Adoption

Illustrative Example I
(Fault Localization)

Req.

Towards SE2.0

Adoption

Illustrative Example II
(Eff. & Green LLM4SE)

Illustrative Example III
(Security of LLM4SE)

Req.

Illustrative Example II: Efficient & Green LLM4SE

Compressing Pre-trained Models of Code into 3 MB

ASE 2022
Compressor

Jieke Shi, Zhou Yang, Bowen Xu*, Hong Jin Kang and David Lo
School of Computing and Information Systems
Singapore Management University
{jiekeshi,zyang,bowenxu.2017,hjkang.2018,davidlo}@smu.edu.sg

First work to optimize code LLMs: **160× smaller** and **4.23× faster**

Nominated for ACM SIGSOFT Distinguished Paper Award

Greening Large Language Models of Code

ICSE 2024
Avatar

Jieke Shi[◇], Zhou Yang[◇], Hong Jin Kang[♠], Bowen Xu[♣], Junda He[◇], and David Lo[◇]
[◇]School of Computing and Information Systems, Singapore Management University, Singapore
[♠]Department of Computer Science, University of California, Los Angeles, USA
[♣]Department of Computer Science, North Carolina State University, Raleigh, USA
{jiekeshi, zyang, jundahe, davidlo}@smu.edu.sg, hjkang@cs.ucla.edu, bxu22@ncsu.edu

Optimize code LLMs: **160x smaller, 76× faster, 184× more energy-saving,** and **157× less in carbon footprint**

School of Computing and Information Systems

SMU SINGAPORE MANAGEMENT UNIVERSITY

One Door Opens, Many Doors to Be Opened

- More experimentation and adaptation:
  - Compressing more and larger models
  - Consideration of various SE tasks
- More LLM inference acceleration methods *in combination with* compression:
  - Quantization, dynamic model inference, static program optimization, etc.
- LLM training acceleration, e.g., training data reduction

Efficient and Green Large Language Models for Software Engineering: Vision and the Road Ahead

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

TOSEM SE Vision 2030 @ FSE 2024

Illustrative Example III: Security of LLM4SE

FSE 2022

You See What I Want You to See: Poisoning Vulnerabilities in Neural Code Search

Yao Wan*
School of Computer Science and Technology, Huazhong University of Science and Technology, China
wanyao@hust.edu.cn

Shijie Zhang*
School of Computer Science and Technology, Huazhong University of Science and Technology, China
shijie_zhang@hust.edu.cn

Hongyu Zhang
University of Newcastle
Australia
hongyu.zhang@newcastle.edu.au

ICPR 2022

Backdoors in Neural Models of Source Code

Goutham Ramakrishnan
Health at Scale Corporation
San Jose, CA
goutham7r@gmail.com

Aws Albarghouthi
University of Wisconsin–Madison
Madison, WI
aws@cs.wisc.edu

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*

Abstract—Code models, such as CodeBERT and CodeT5, offer general-purpose representations of code and play a vital role in supporting downstream automated software engineering tasks. Most recently, code models were revealed to be vulnerable to backdoor attacks. A code model that is backdoor-attacked can behave normally on clean examples but will produce pre-defined malicious outputs on examples injected with *triggers* that activate the backdoors. Existing backdoor attacks on code models use unstealthy and easy-to-detect triggers. This paper aims to investigate the vulnerability of code models with *stealthy* backdoor attacks. To this end, we propose AFRAIDOOR (*A*dversarial *F*eatur*e* as *A*dapt*i*ve Back*door*). AFRAIDOOR achieves stealthiness by leveraging adversarial perturbations to inject adaptive triggers into different inputs. We apply AFRAIDOOR to three widely adopted code models (CodeBERT, PLBART, and CodeT5) and two downstream tasks (code summarization and method name prediction). We evaluate three widely used defense methods and find that AFRAIDOOR is more unlikely to be detected by the defense methods than by baseline methods. More specifically, when using spectral signature as defense, around 85% of adaptive triggers in AFRAIDOOR bypass the detection in the defense process. By contrast, only less than 12% of the triggers from previous work bypass the defense. When the defense method is not applied, both AFRAIDOOR and baselines have almost perfect attack success rates. However, once a defense is applied, the attack success rates of baselines decrease dramatically, while the success rate of AFRAIDOOR remains high. Our finding exposes security weaknesses in code models under stealthy backdoor attacks and shows that state-of-the-art defense methods cannot provide sufficient protection. We call for more research efforts in understanding security threats to code models and developing more effective countermeasures.

Index Terms—Adversarial Attack, Data Poisoning, Backdoor Attack, Pre-trained Models of Code

IEEE Transactions on Software Engineering (TSE 2024)

One Door Opens, Many Doors to Be Opened

- Need to investigate novel attack vectors
  - beyond adversarial attack, data poisoning attack, etc.
- Need more effective data auditing tools
  - identify and sanitize poisoned data examples
- Need for trustworthy LLM4Code ecosystem
  - trusted datasets and models that developers can reuse and build upon

Ecosystem of Large Language Models for Code

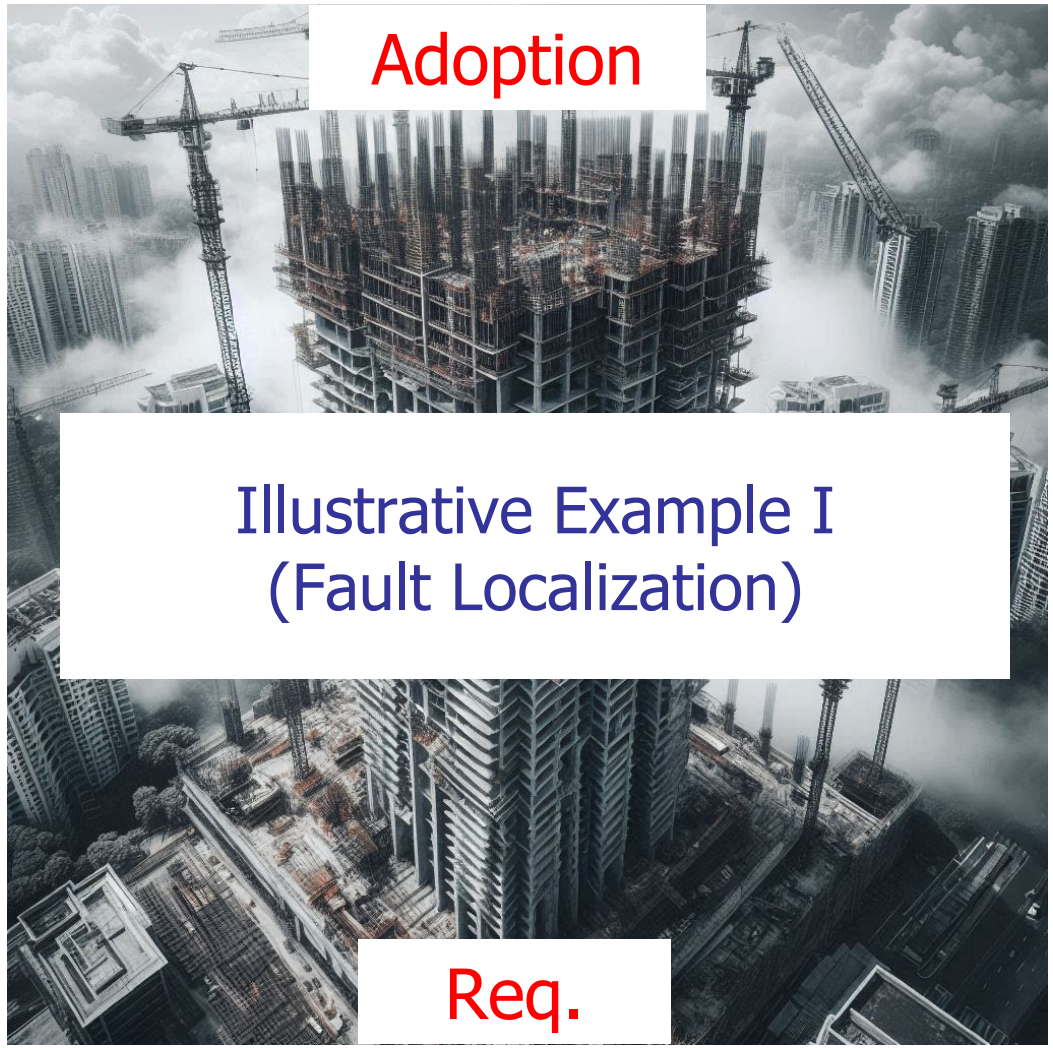
Zhou Yang, Jieke Shi, and David Lo *Fellow, IEEE*

Abstract—The availability of vast amounts of publicly accessible data of source code and the advances in modern language models, coupled with increasing computational resources, have led to a remarkable surge in the development of large language models for code (LLM4Code, for short). The interaction between code datasets and models gives rise to a complex ecosystem characterized by intricate dependencies that are worth studying. This paper introduces a pioneering analysis of *code model ecosystem*. Utilizing Hugging Face 🤗—the premier hub for transformer-based models—as our primary source, we curate a list of datasets and models that are manually confirmed to be relevant to software engineering. By analyzing the ecosystem, we first identify the popular and influential datasets, models, and contributors. The popularity is quantified by various metrics, including the number of downloads, the number of likes, the number of reuses, etc. The ecosystem follows a power-law distribution, indicating that users prefer widely recognized models and datasets. Then, we manually categorize how models in the ecosystem are reused into nine categories, analyzing prevalent model reuse practices. The top-3 most popular reuse types are *fine-tuning*, *architecture sharing*, and *quantization*. We also explore the practices surrounding the publication of LLM4Code, specifically focusing on documentation practice and license selection. We find that the documentation in the ecosystem contains less information than that in general artificial intelligence (AI)-related repositories hosted on GitHub. Additionally, the license usage is also different from other software repositories. Models in the ecosystem adopt some AI-specific licenses, e.g., RAIL (Responsible AI Licenses) and AI model license agreement.

Index Terms—Pre-trained Models for Code, Software Ecosystem, Mining Software Repository

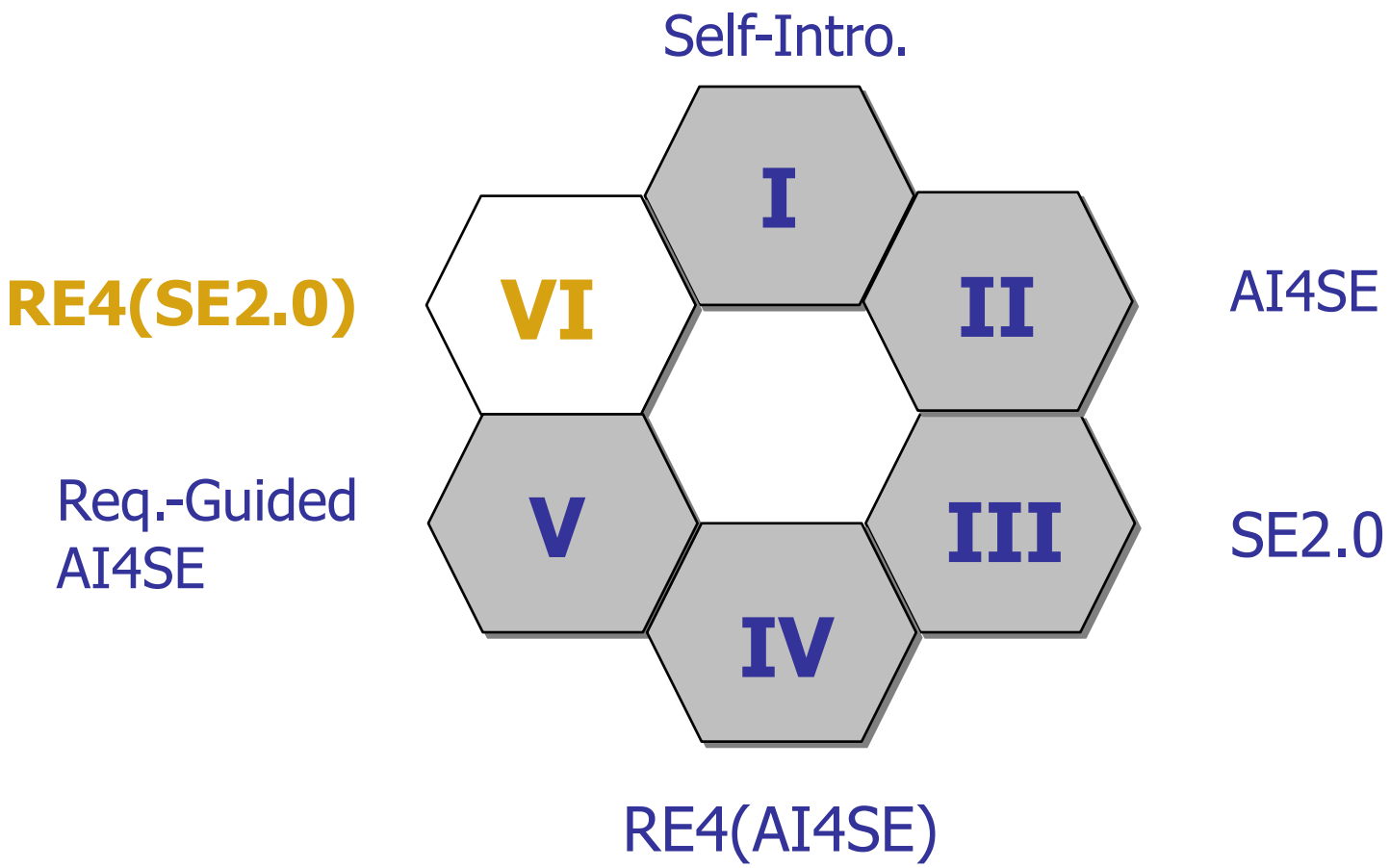
How Has Identifying Key Reqs. Spurred AI4SE Research?

Traditional



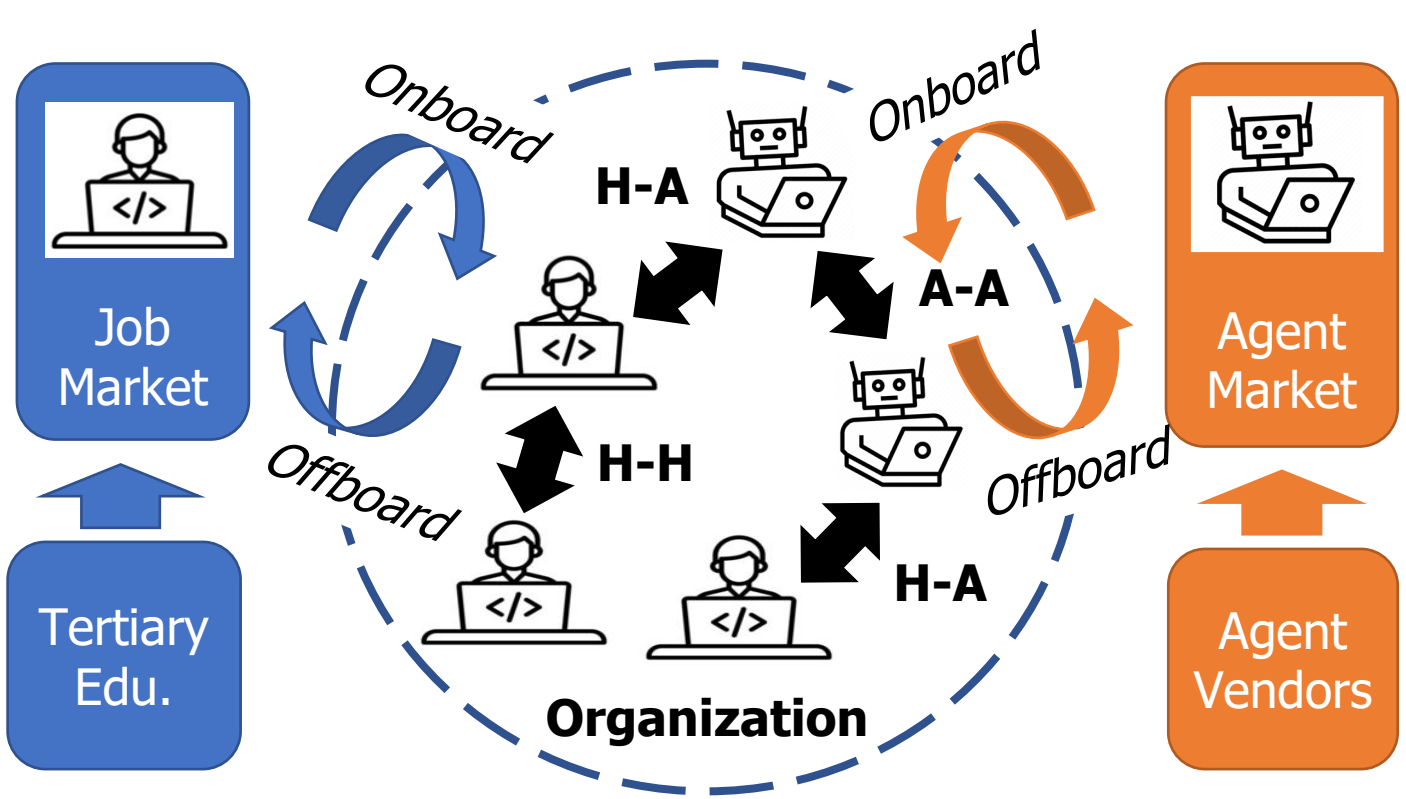
Towards SE2.0

Talk Structure



Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

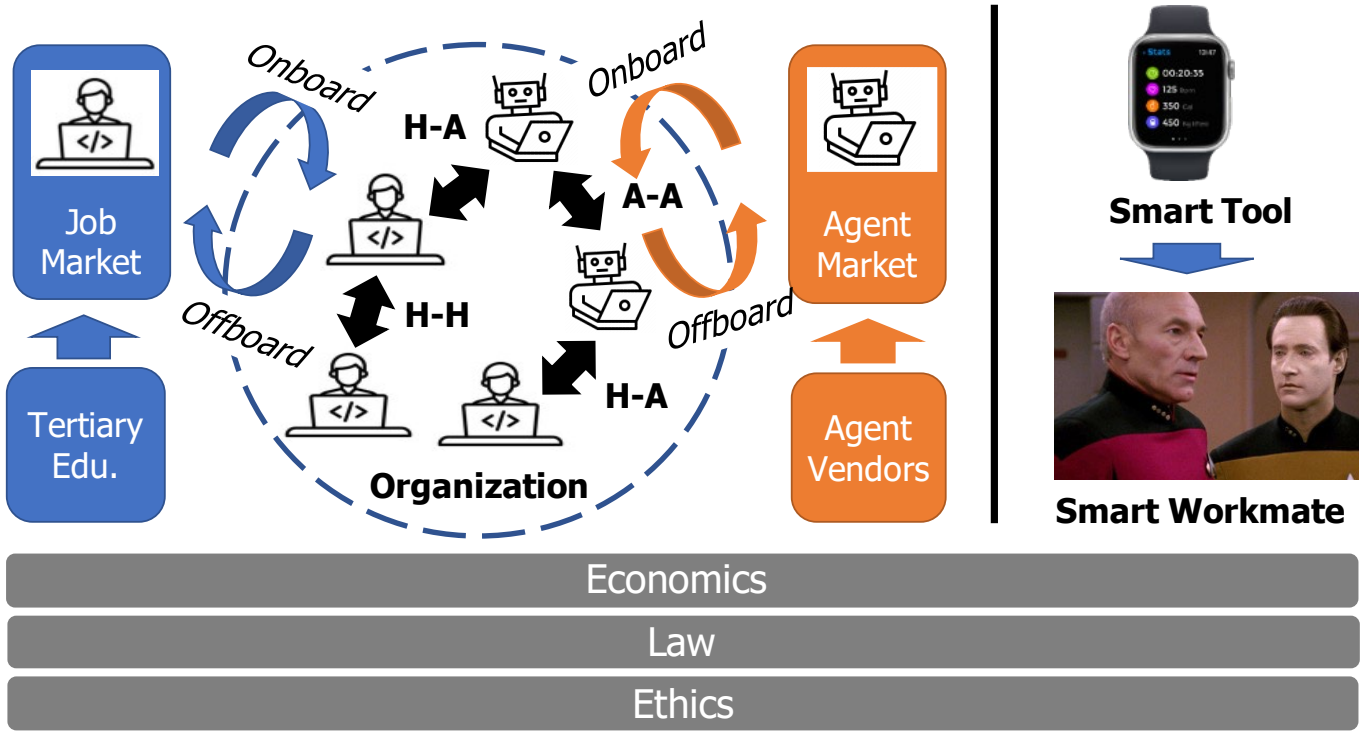
Economics

Law

Ethics

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



What requirements on AI agents' intelligence should be met before we can find them useful for diverse tasks?

Many Unknown Requirements: Much Research to Be Done

What requirements on AI agents' intelligence should be met before we can find them useful for diverse tasks?

ISSTA 2016

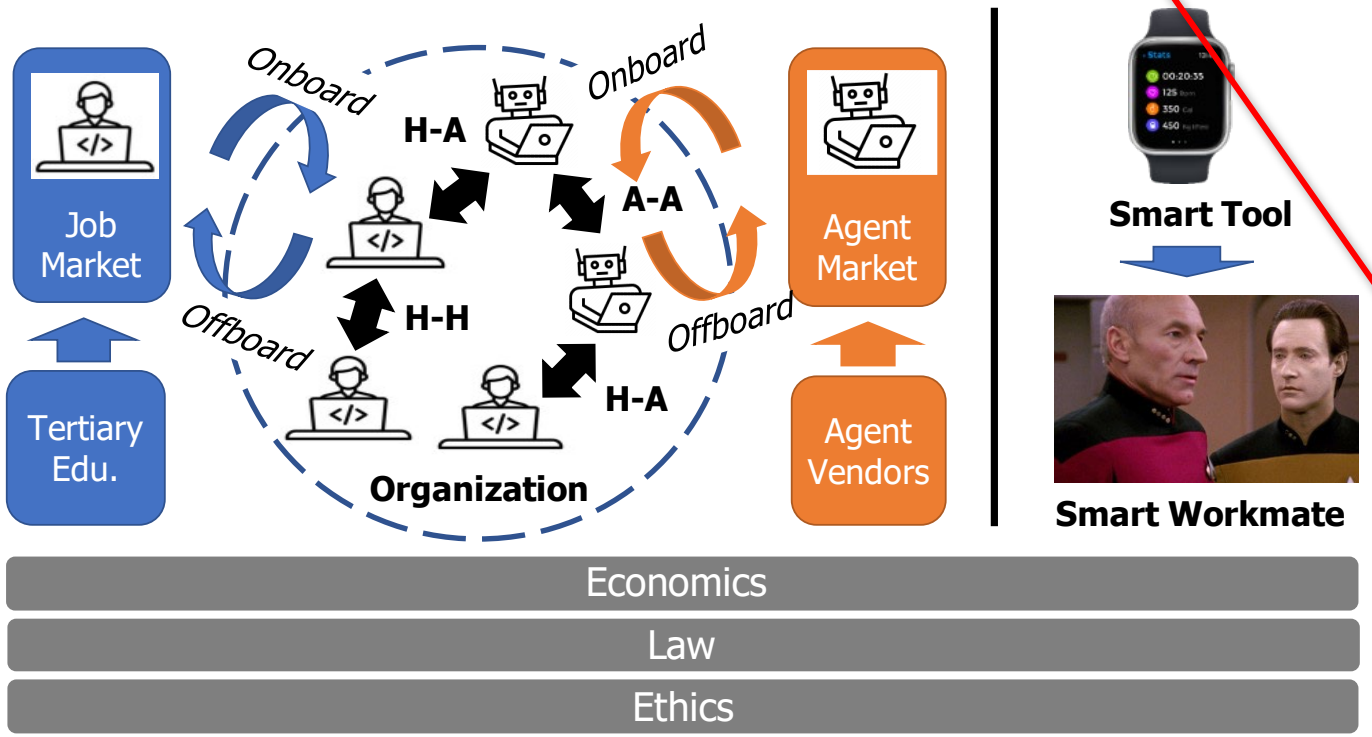
Practitioners' Expectations on Automated Fault Localization

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

- High adoption threshold
  - Satisfy 75% of practitioners: successful results in Top 5, works 75% of time; ≥100,000 LOC; takes <1 minute.
- Rationale and IDE integration are important.

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



What requirements on AI agents' intelligence should be met before we can find them useful for diverse tasks?

What are the requirements of a responsible AI agents?

Many Unknown Requirements: Much Research to Be Done

What are the requirements of a responsible AI agents?

Boy who cried wolf

Smart Aleck GenAI Rendering

Should I follow this fault localization tool's output?

Automated prediction of fault localization effectiveness

Tien-Duy B. Le · David Lo · Ferdian Thung

ICSM 2013 & EMSE 2015

BBC Sign in Home News Sport Reel Worklife

NEWS

Friend or foe: Can computer coders trust ChatGPT?

31 March

The Register®

SOFTWARE 59

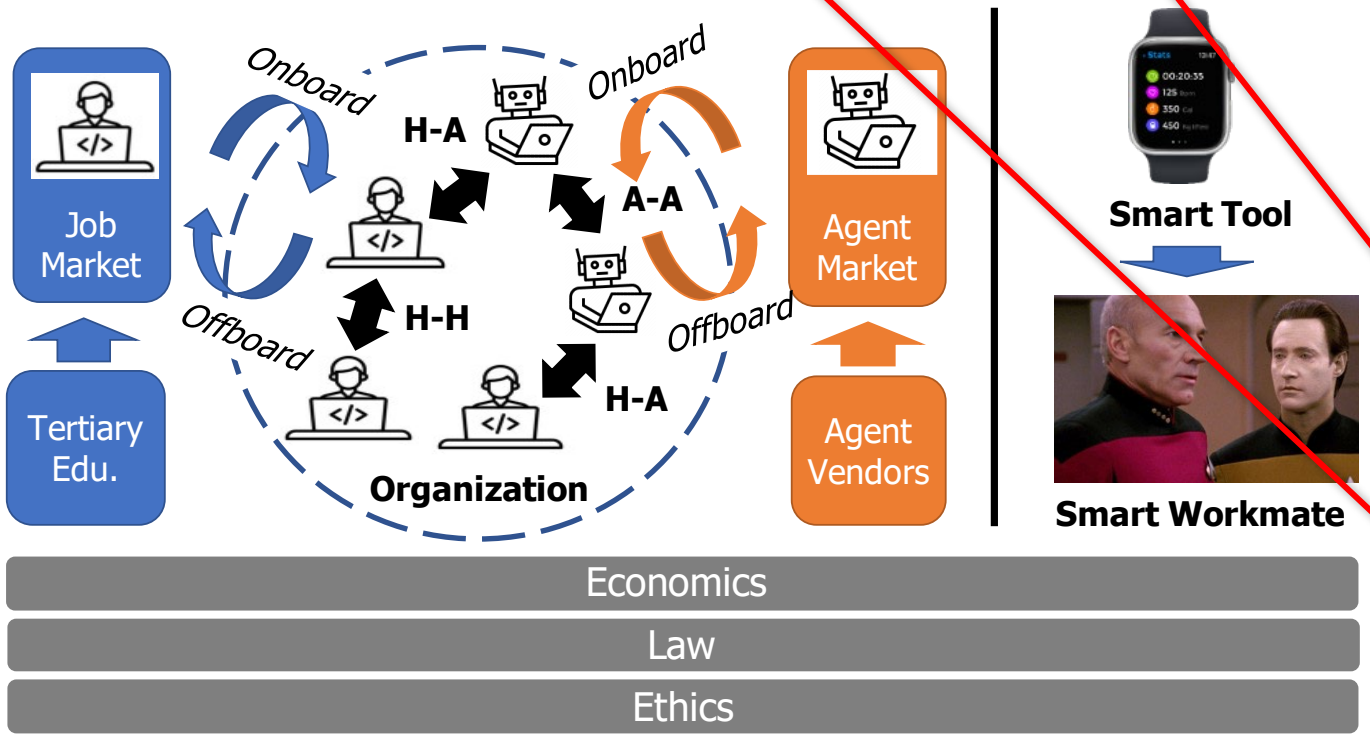
ChatGPT creates mostly insecure code, but won't tell you unless you ask

Boffins warn of risks from chatbot model that, Dunning–Kruger style, fails to catch its own bad advice

Thomas Claburn Fri 21 Apr 2023 // 01:28 UTC

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



What requirements on AI agents' intelligence should be met before we can find them useful for diverse tasks?

What are the requirements of a responsible AI agents?

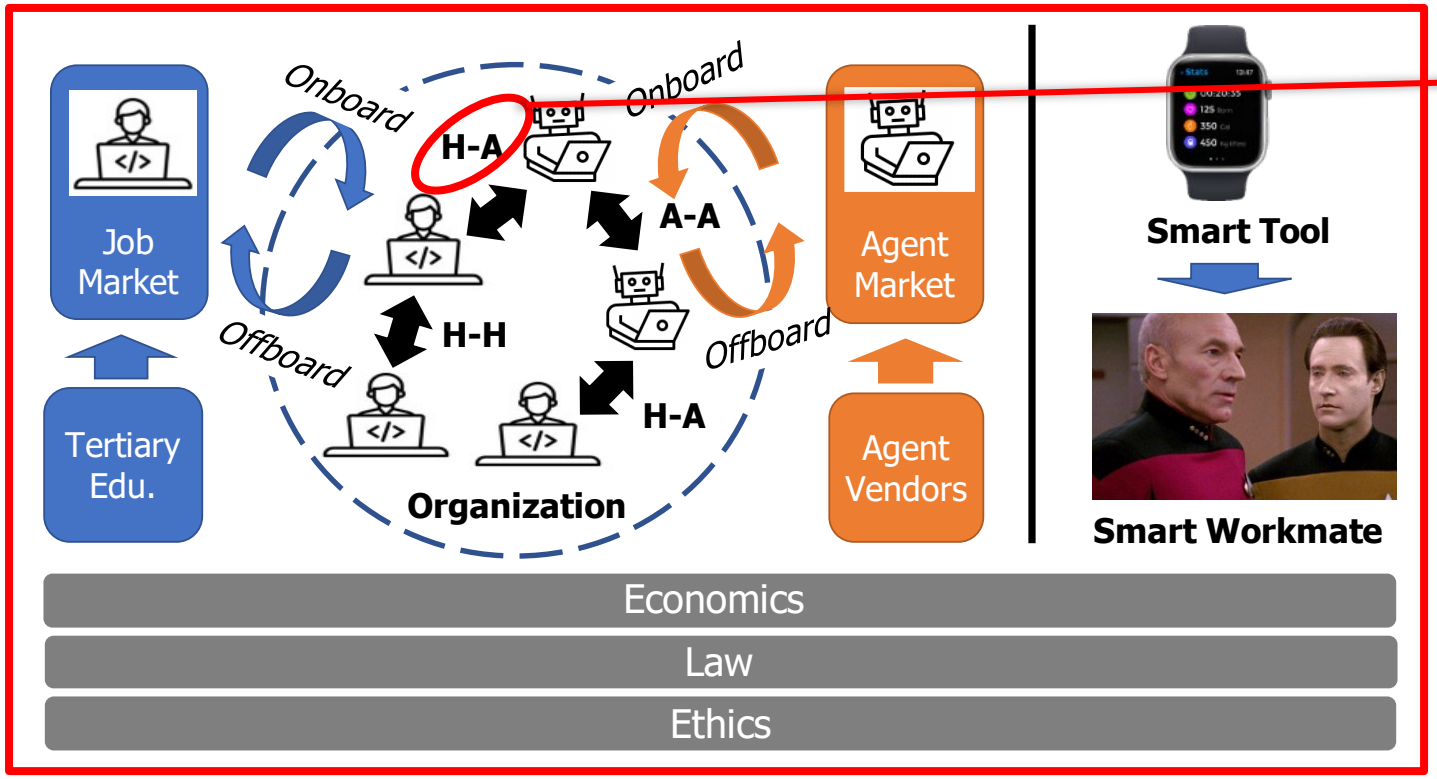
What requirements should be met before we can give higher degree of autonomy to AI agents?

How Practitioners Perceive the Relevance of Software Engineering Research **ESEC/FSE 2015**

*"It seems that there could be potentially **disastrous results** if the automation does not [do things] correctly.."*

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers

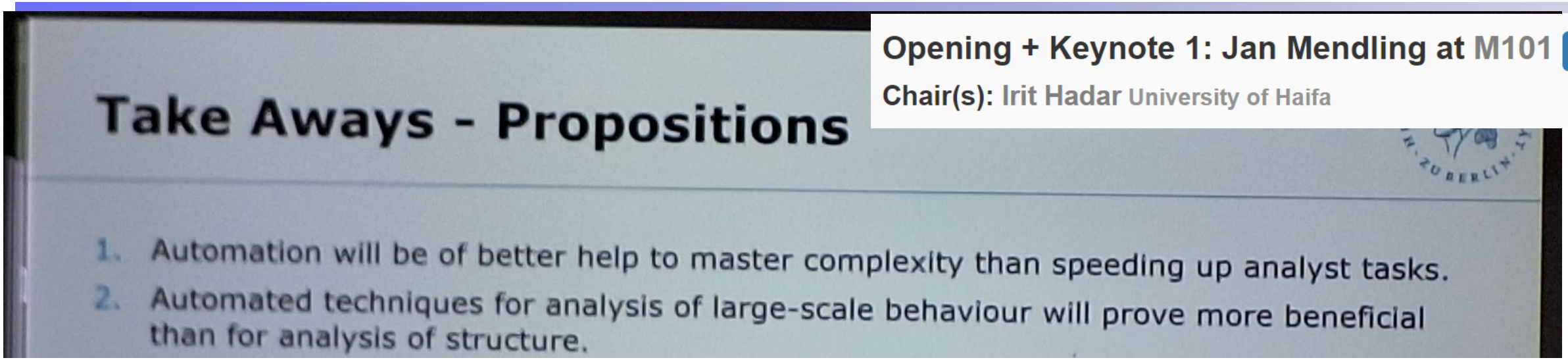


Smart Tool

Smart Workmate

What are the requirements for effective Human-Agent interactions?

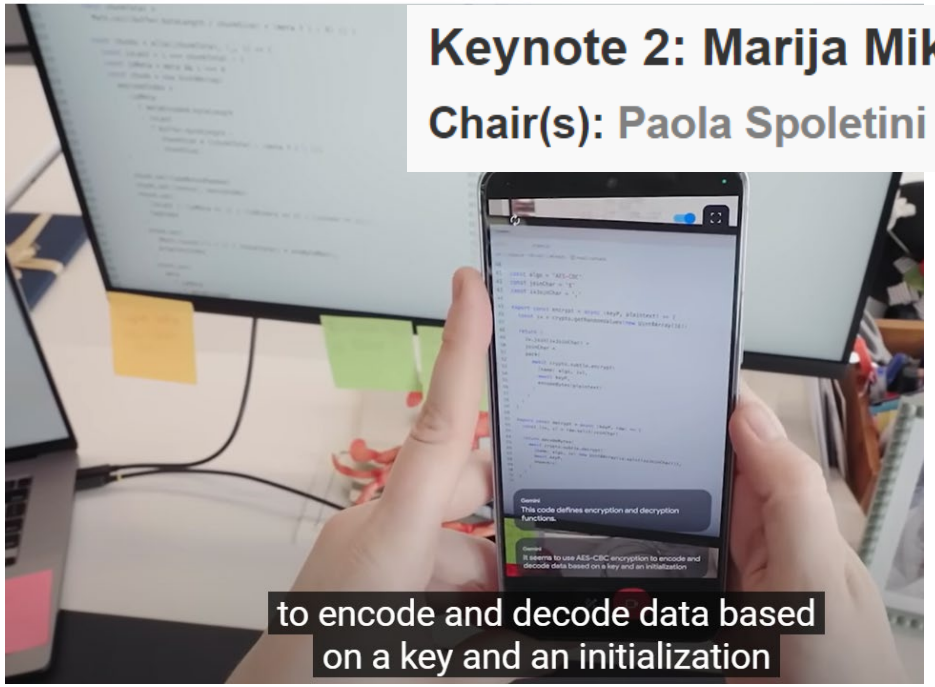
Many Unknown Requirements: Much Research to Be Done



Opening + Keynote 1: Jan Mendling at M101

Chair(s): Irit Hadar University of Haifa

What are the requirements for effective Human-Agent interactions?

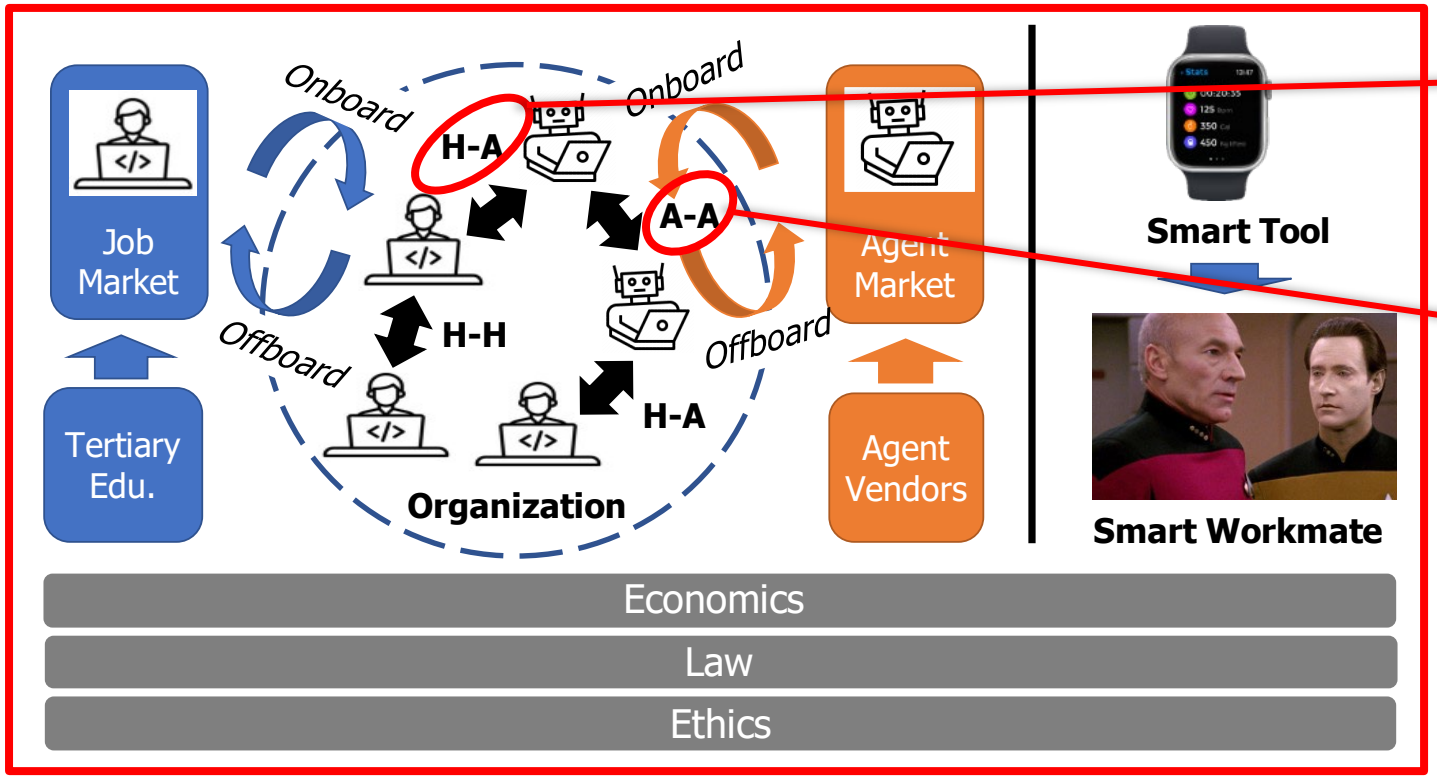


Keynote 2: Marija Mikic at M101

Chair(s): Paola Spoletini Kennesaw State University

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

What are the requirements for effective Human-Agent interactions?

What are the requirements for effective Agent-Agent interactions?

Many Unknown Requirements: Much Research to Be Done

What are the requirements for effective Agent-Agent interactions?

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

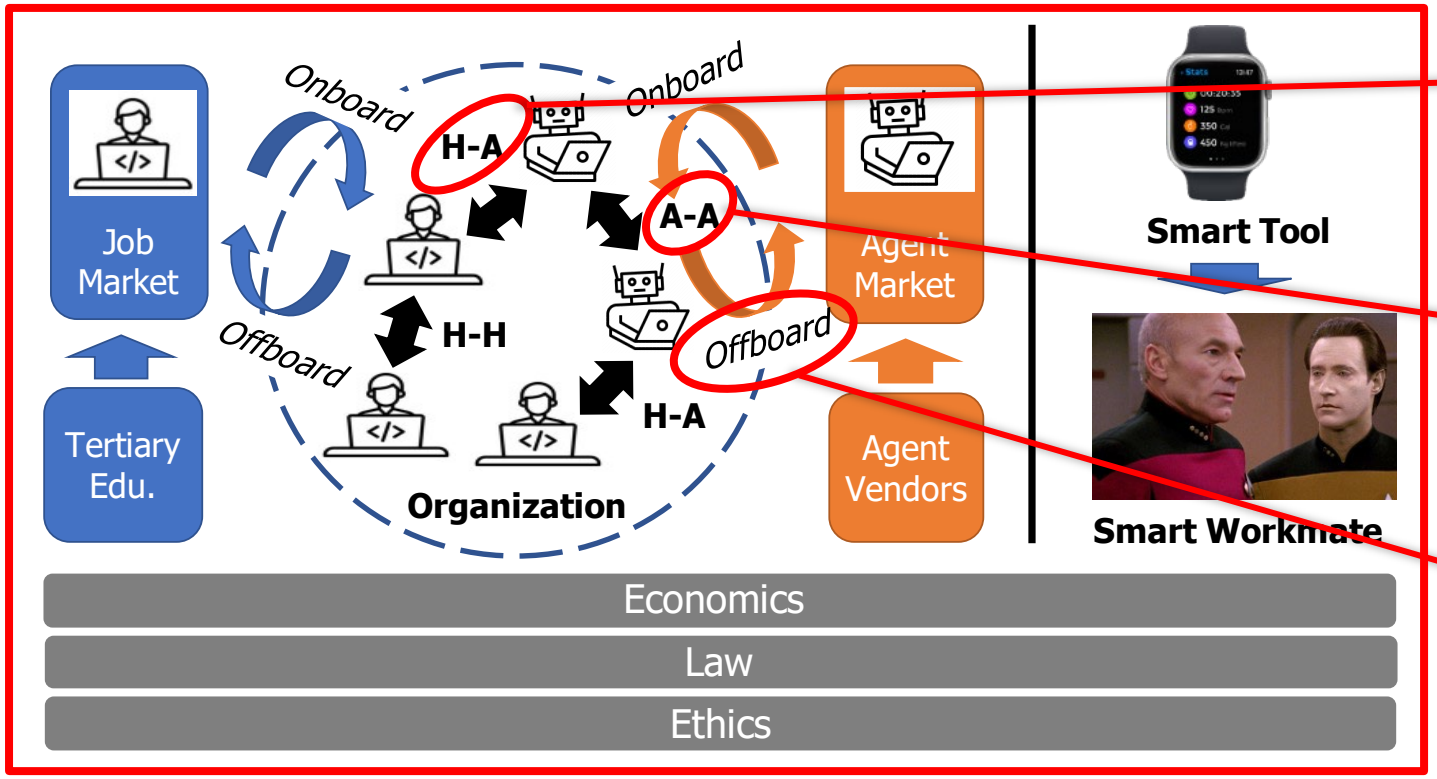
Zhou Yang
Singapore Management University
Singapore
zyang@smu.edu.sg

Li Li
Beihang University
China
lilicoding@ieee.org

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

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

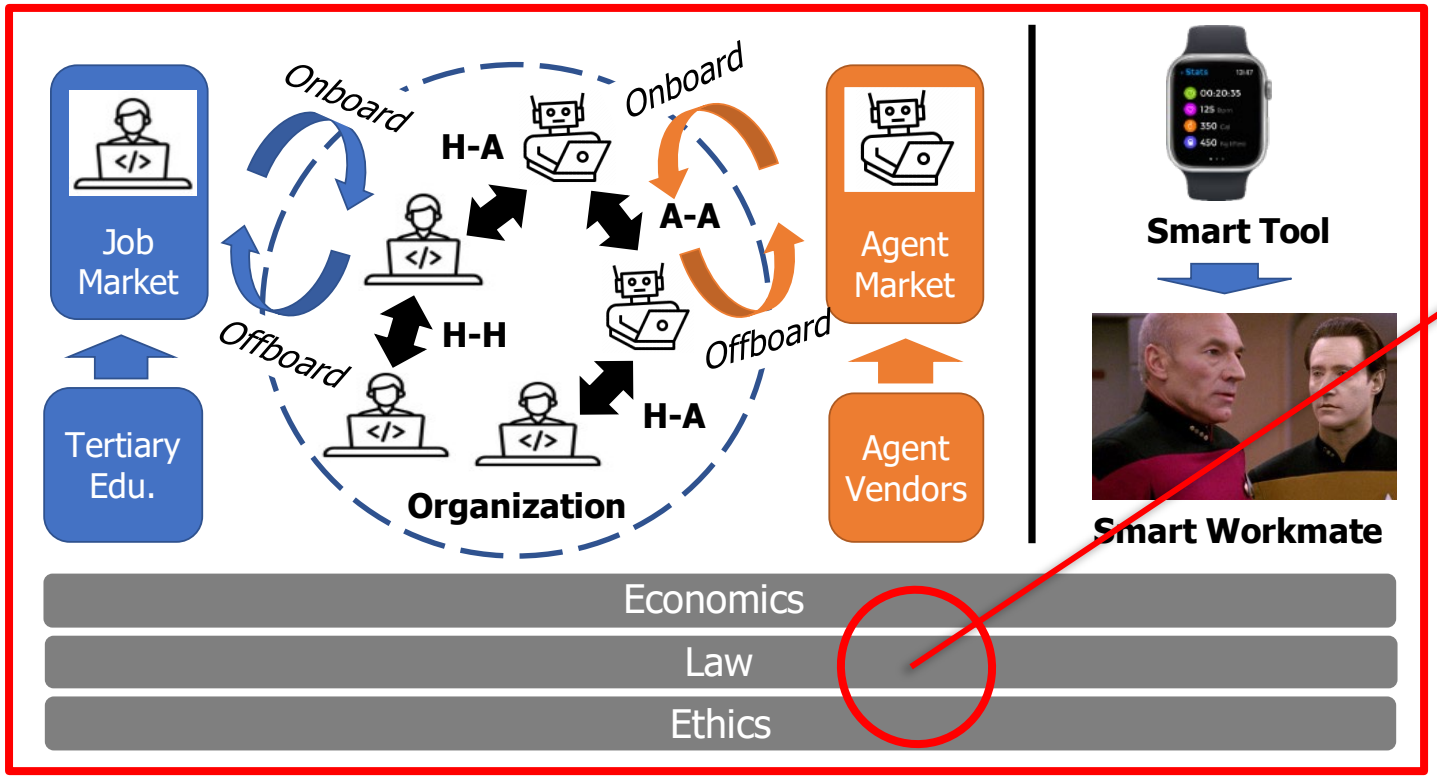
What are the requirements for effective Human-Agent interactions?

What are the requirements for effective Agent-Agent interactions?

What are the requirements for effective Agent onboarding and offboarding?

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



What are the economics, legal, and ethical requirements for different SE 2.0 stakeholders?

Fri 28 Jun

Displayed time zone: **(UTC) Coordinated Universal Time** change

10:45 - 12:15 Legal Compliance at M124

RE@Next! Papers / Research Papers

Chair(s): Chetan Arora Monash University

10:45 **30m** *Paper* ☆

AI-enabled Regulatory Change Analysis of Legal Requirements

Research Papers

Sallam Abualhaija University of Luxembourg, Marcello Ceci University of Luxembourg, Nicolas Sannier University of Luxembourg, SnT, Domenico Bianculli University of Luxembourg, Lionel Briand University of Ottawa, Canada; Lero centre, University of Limerick, Ireland, Dirk Zetzsche University of Luxembourg, Marco Bodellini University of Luxembourg

Pre-print

11:15 **30m** *Paper* ☆

Defining a Model for Content Requirements from the Law: an Experience Report

Research Papers

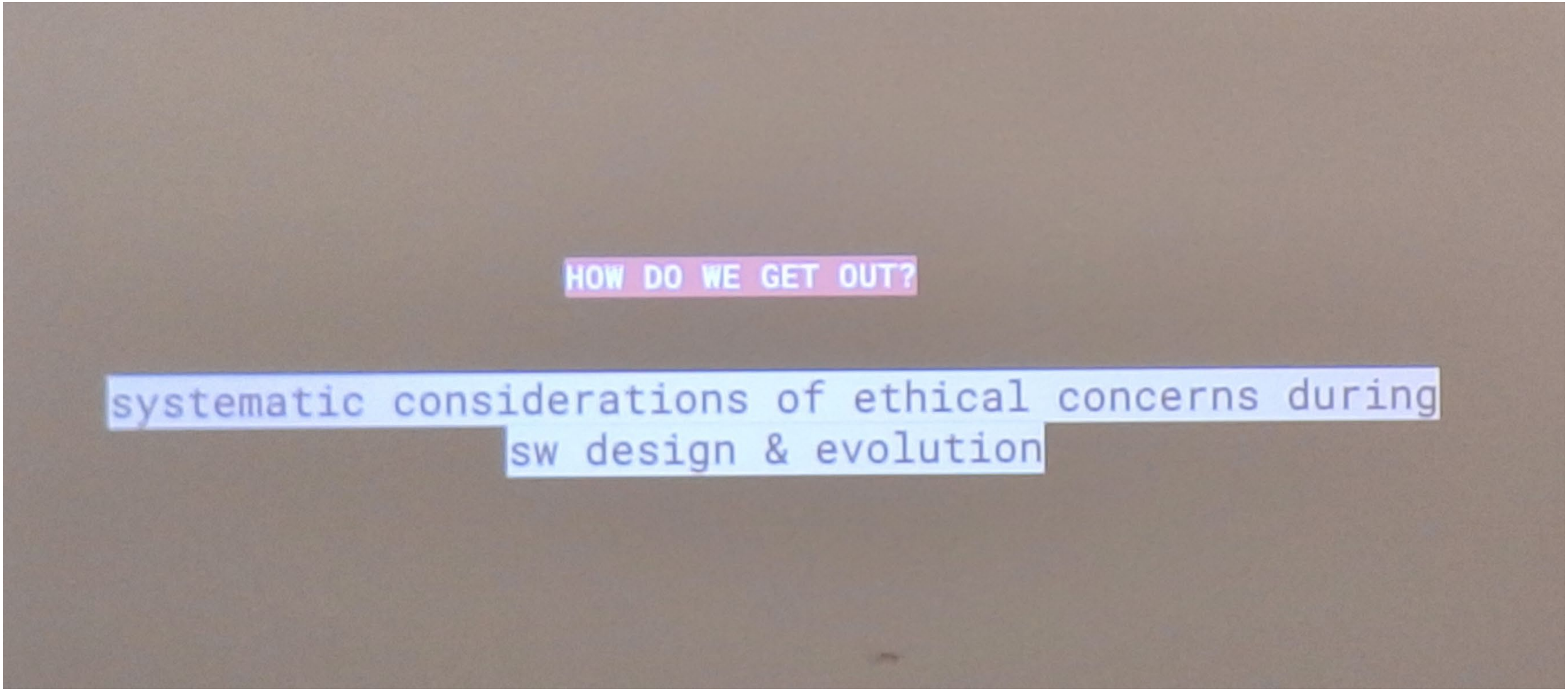
Marcello Ceci University of Luxembourg, Domenico Bianculli University of Luxembourg, Lionel Briand University of Ottawa, Canada; Lero centre, University of Limerick, Ireland

11:45 **30m** *Paper* ☆

Rethinking Legal Compliance Automation: Opportunities with Large Language Models

RE@Next! Papers

Shabnam Hassani University of Ottawa, Mehrdad Sabetzadeh University of Ottawa, Daniel Amyot University of Ottawa, Jian Liao



How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews

Most Influential Paper

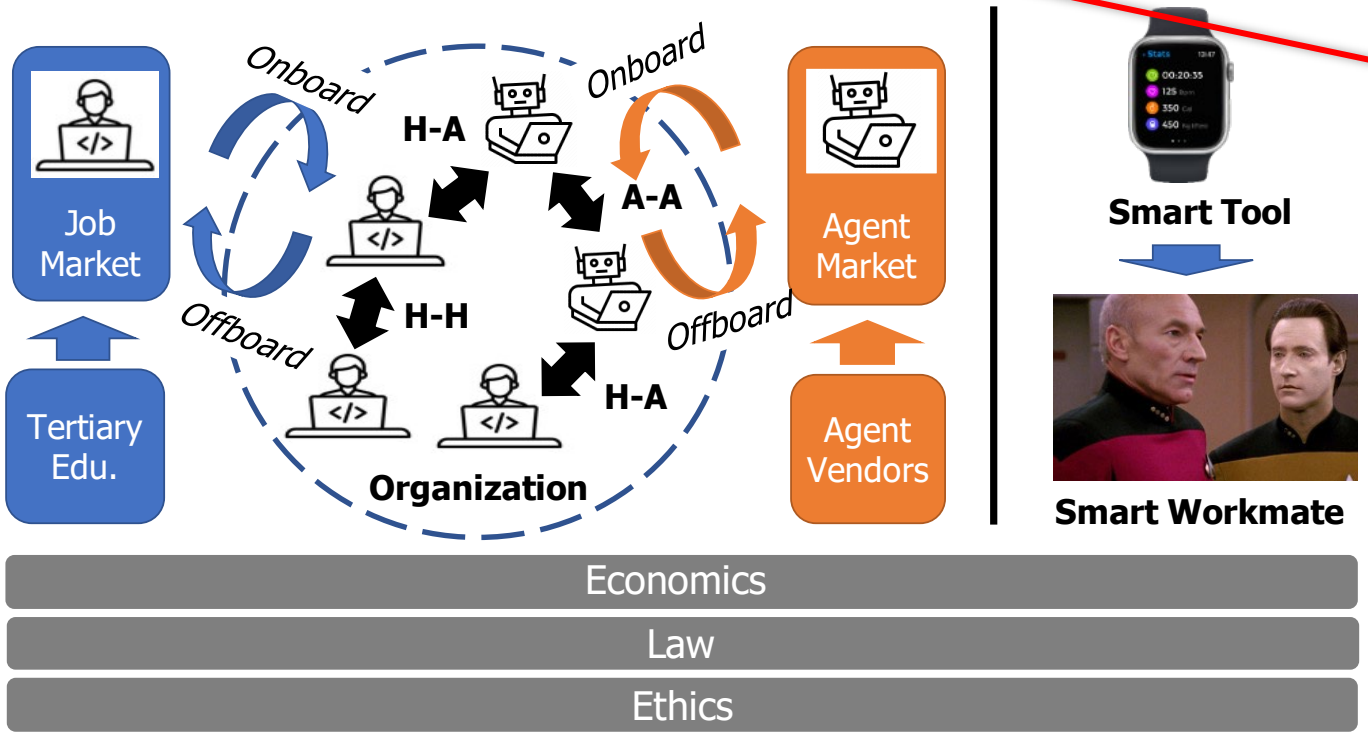
A: Emitzá Guzmán Vrije Universiteit Amsterdam, **A:** Walid Maalej University of Hamburg

School of Computing and Information Systems

SMU Singapore Management University

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

What are the requirements to achieve symbiotic workforce?

Which tasks should be delegated to AI agents, and which one should be kept by humans?

Should nature of jobs for engineers change?

One Perspective

TSE 2023

How do Developers *Really* Feel About Bug Fixing? Directions for Automatic Program Repair

Emily Winter, David Bowes, Steve Counsell, Tracy Hall, Sæmundur Haraldsson, Vesna Nowack, and John Woodward

- "We find that bug finding and fixing is not necessarily as onerous for developers as has often been suggested, being rated as more satisfying than developers' general work."
- "The fact that developers derive satisfaction and benefit from bug fixing indicates that APR adoption is not as simple as APR replacing an unwanted activity."
- "When it comes to potential APR approaches, we find a strong preference for developers being kept in the loop (for example, choosing between different fixes or validating fixes) as opposed to a fully automated process."

Trend: End-User Programming

Microsoft Powerapps

Zoho Creator

Appian

Airtable

Kissflow

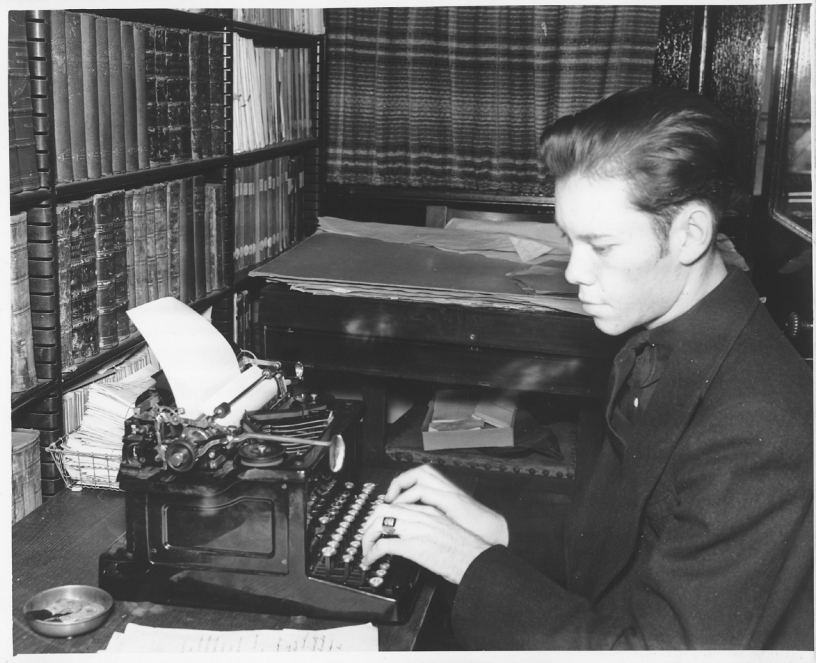
Low-Code Programming Frameworks

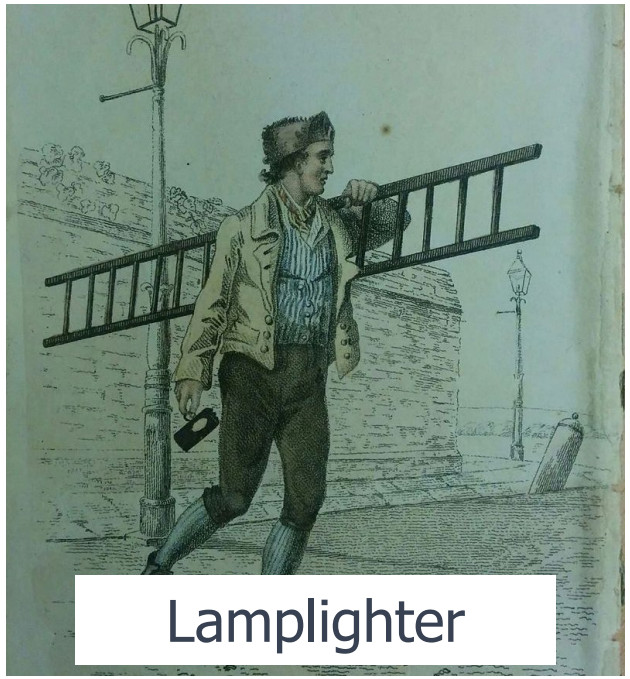
INFOCOMM MEDIA DEVELOPMENT AUTHORITY | CODE@SG

From 2020, all upper primary students will undergo a 10-hour computational thinking and coding enrichment class to expose them to the skills needed to embrace new opportunities in the digital economy.

Computational Thinking and Coding becomes More Pervasive

Another Perspective: Programmers? History Repeats?

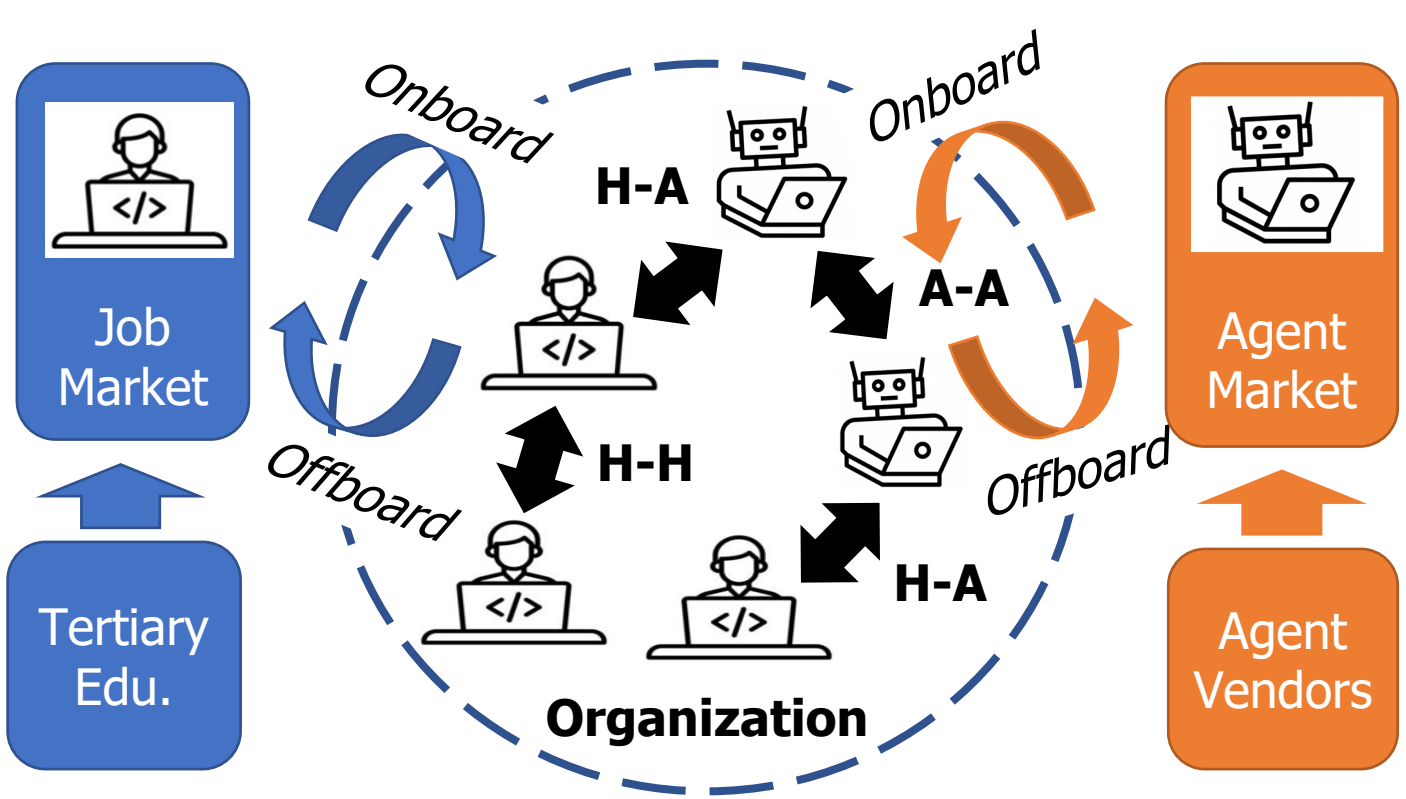




"One historical job that has become nearly obsolete because the necessary competencies are now widespread is that of the typist. With the advent of personal computers and the spread of typing skills, many people are now able to do their own typing rather than relying on a professional typist." – ChatGPT

Possible Future?

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

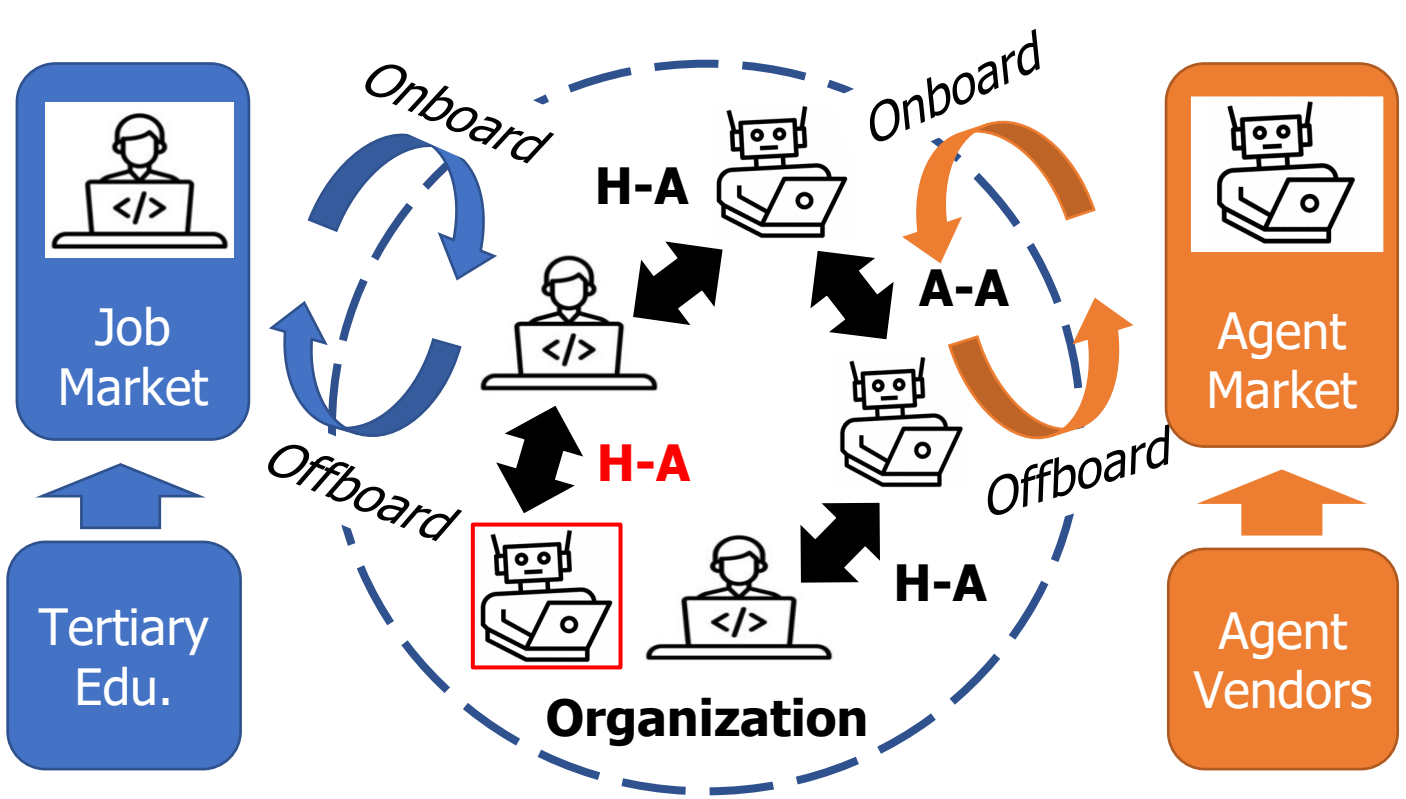
Economics

Law

Ethics

Possible Future?

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

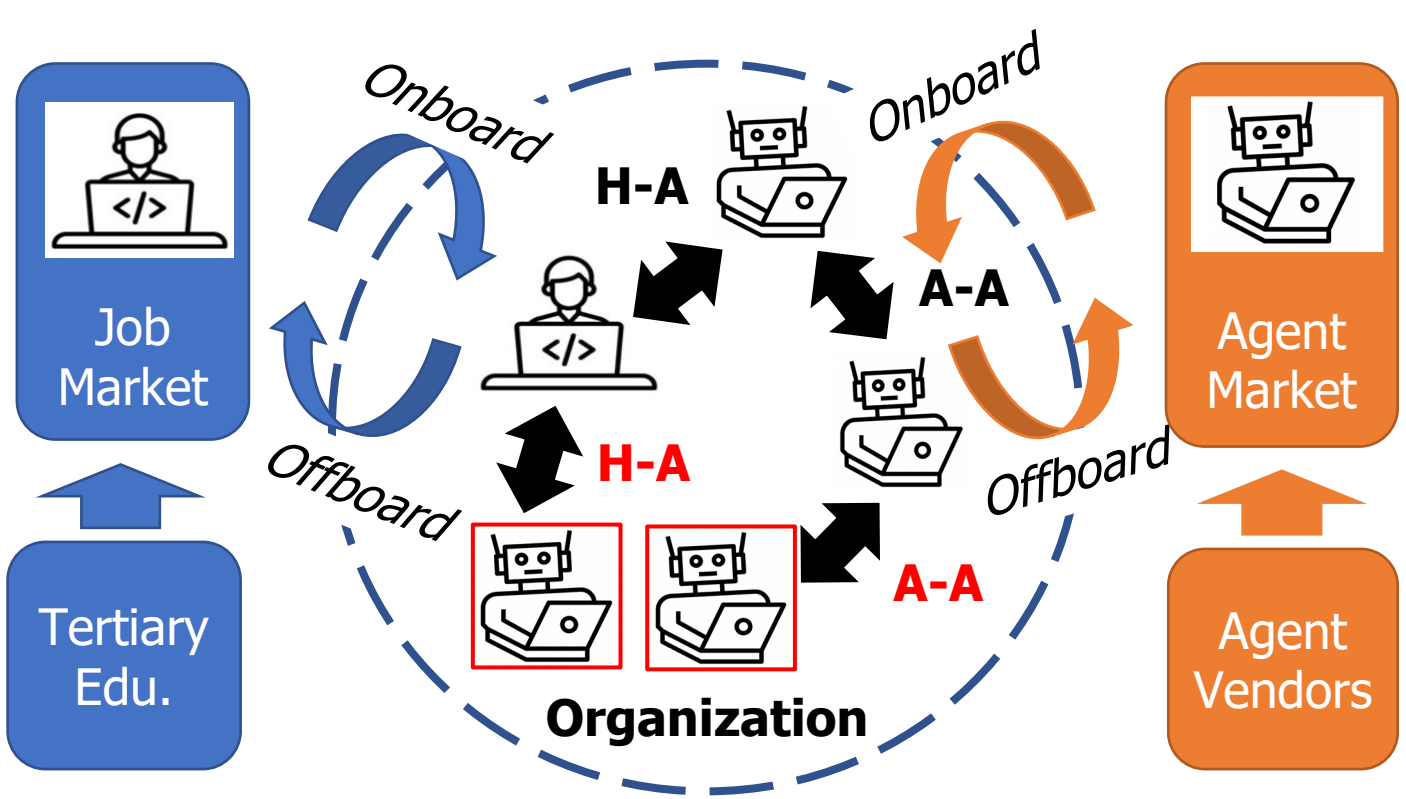
Economics

Law

Ethics

Possible Future?

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



Smart Tool

Smart Workmate

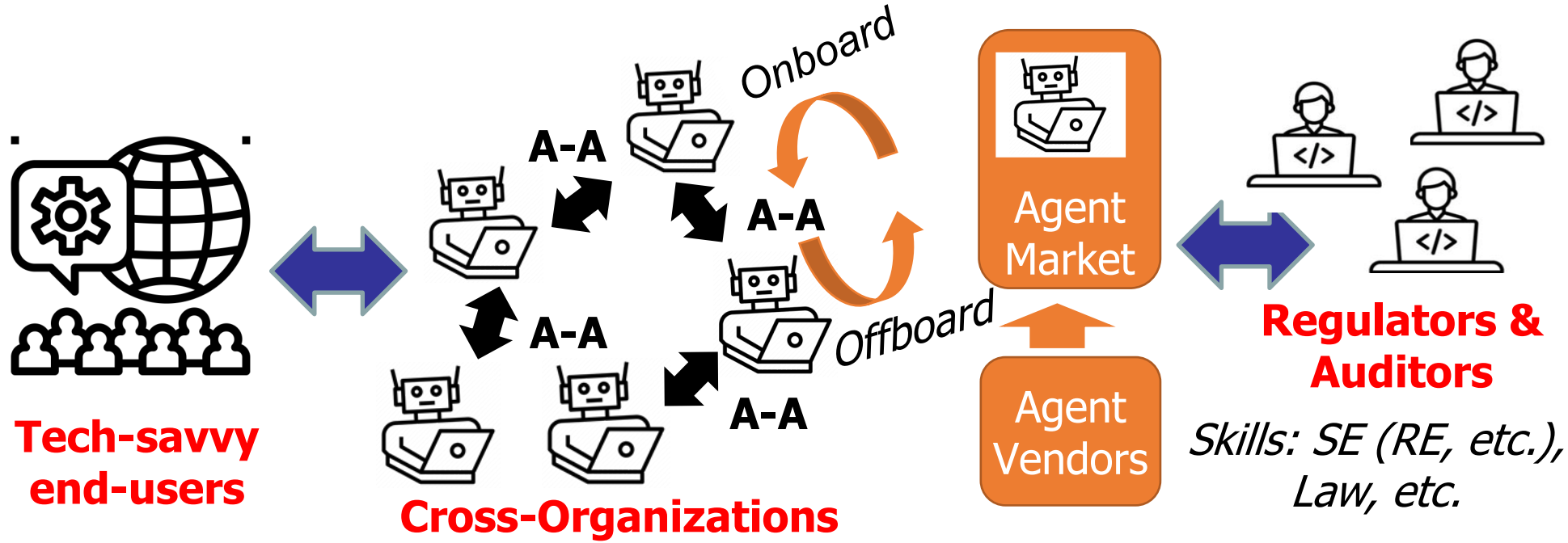
Economics

Law

Ethics

Possible Future?

Symbiotic workforce of autonomous, responsible, intelligent agents creating **myriads of software we need and want with tech-savvy end-users + regulators & auditors**



Economics

Law

Ethics

Possible Future?

Symbiotic workforce of autonomous, responsible, intelligent agents ing **myriads of software we need and want with tech-savvy end ors & auditors**

Requirements to bring net-benefit to society?
Requirements to make this work?
Requirements for skills of regulators and auditors?

子曰："人无远虑，必有近忧。"

"A person who does not plan long ahead will find trouble at his door." – Confucius

How to Make Software Engineering 2.0 a Reality?

GenAI rendering of "magnificent large futuristic building with strong foundation"

SE2.0

Need for RE4(SE2.0)

How to Make Software Engineering 2.0 a Reality?

"[We need to be] engineering technology with consequences that we want."

Panel: Requirements Engineering in the Era of Intelligent Cyber-Physical Systems

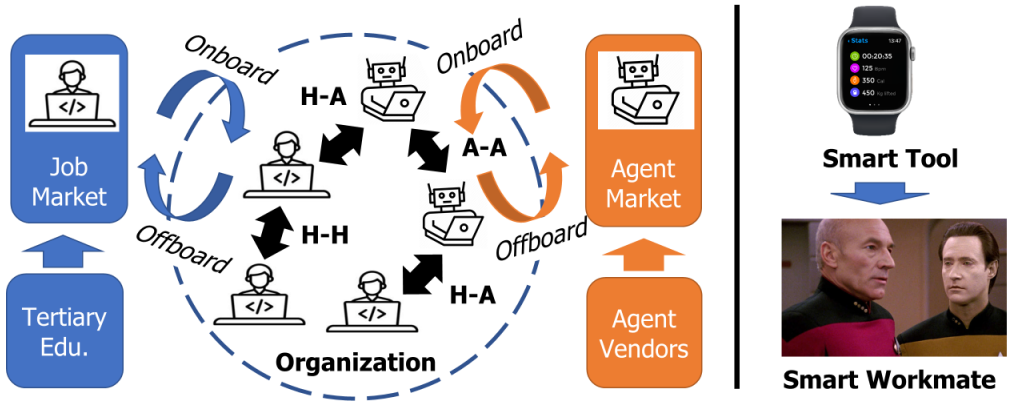
Panels

S: Jane Cleland-Huang University of Notre Dame, **P:** Bashar Nuseibeh The Open University, UK, **P:** Jan-Philipp Steghöfer XITASO GmbH IT & Software Solutions, **P:** Nelly Bencomo Durham University, **P:** Chattie McBotface

Need for RE4(SE2.0)

Software Engineering 2.0

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



How to Make Software Engineering 2.0 a Reality?

GenAI rendering of "magnificent large futuristic building with strong foundation"

SE2.0

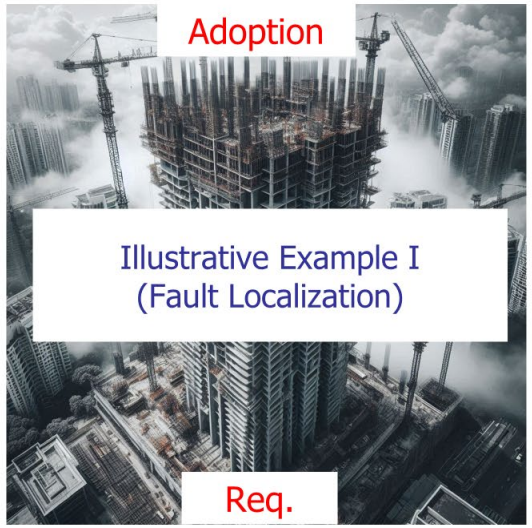
Need for RE4(SE2.0)

School of Computing and Information Systems

SMU Singapore Management University

How Has Identifying Key Reqs. Spurred AI4SE Research?

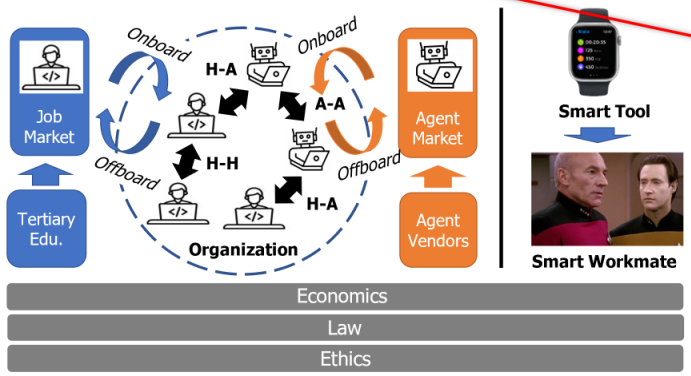
Traditional



Towards SE2.0

Many Unknown Requirements: Much Research to Be Done

Symbiotic workforce of autonomous, responsible, intelligent agents & engineers



- What are the requirements to achieve symbiotic workforce?
- Should nature of jobs for engineers change?
- Which tasks should be delegated to AI agents, and which one should be kept by humans?

School of Computing and Information Systems

SMU Singapore Management University

Icarus and Daedalus
Greek Mythology

starcoder

Acknowledgements

OUB Chair
Professorship Fund

Thank you!

Questions? Comments? Advice?

davidlo@smu.edu.sg