



# Generative AI for Software Engineering at FPT Software AI Center

Presenter: NghiBDQ, Ph.D.

Contact: [nghibdq@fpt.com](mailto:nghibdq@fpt.com)

# About



School of  
**Computing and  
Information Systems**



**Ph.D. in Computer Science**  
School of Computing & Information Systems,  
Singapore Management University

2016-2020

**Principal Research Scientist,**  
Trustworthy Software Innovation Lab –  
Huawei Research Center, Ireland

2019-2022

**Mentor/Head of AI Research  
Department**  
FPT Software AI Center

**Research Manager/Adjunct  
Professor**  
Intelligent Software Lab, SMU  
X Salesforce Research Asia



2022- Now

- Research Interests:** Software Engineering (ICSE, FSE, ASE), Artificial Intelligence (AAAI), Natural Language Processing (ACL, EMNLP), Information Retrieval (SIGIR).
- Website:** <https://bdqnghi.github.io/>
- Owner of open-source projects (~4000 star in total):** <https://github.com/bdqnghi>

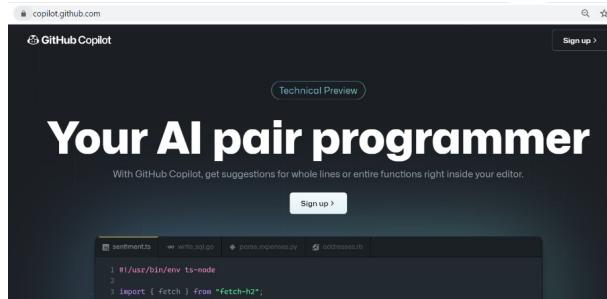
# What is AI4Software?

## AI4Software <==> Applications of AI into Software Engineering

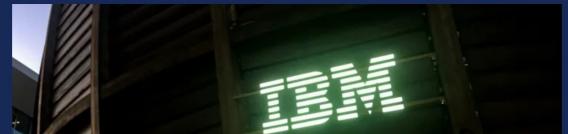
### Einstein for Salesforce Developers

RENÉ WINKELMEYER

AI-powered tools are changing the way we write and analyze code. Learn how Einstein will change the software development landscape with Salesforce.



**IBM taps watsonx generative AI to help modernize COBOL on mainframes**



### AI gives software development tools a boost

GitHub Copilot, DeepDev, IntelliCode, and other code-focused applications of machine learning can help us deliver better code, faster.



Visual Studio

Products

Downloads

Buy

Support

**Visual Studio IntelliCode**  
AI-assisted development

Sign up for news and updates >

© Copyright FPT So

**Intel Machine Programming Tool Detects Bugs in Code**

Today, Intel unveiled ControlFlag – a machine programming research system that can autonomously detect errors in code.



#### News

- December 3, 2020
- Contact Intel PR

More New Technologies  
News →

» Watch video: "Intel Labs Day 2020: Justin Gottschlich Second Session from Intel PR on Vimeo"

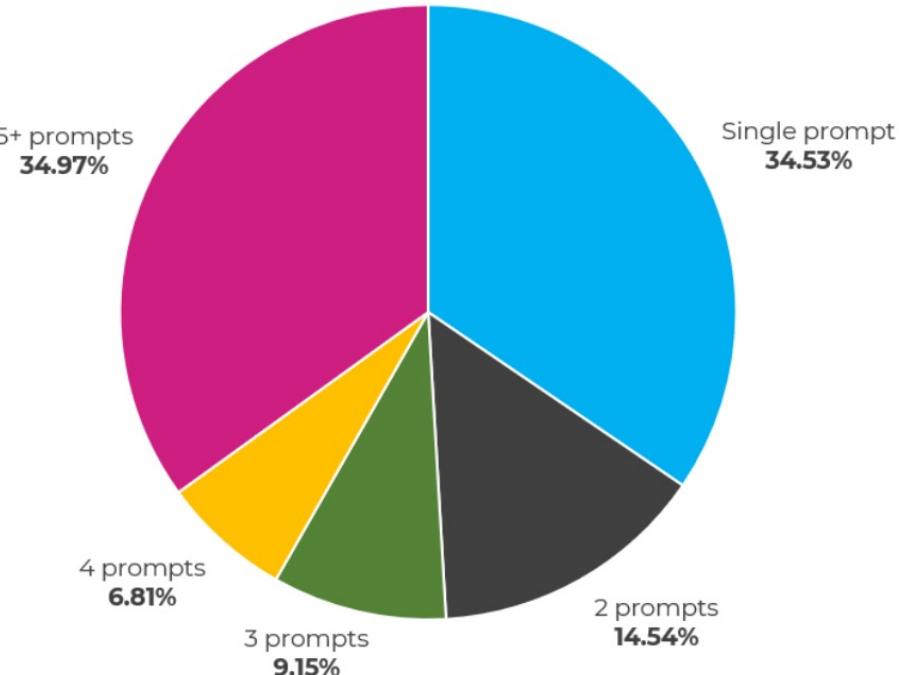
**What's New:** Today, Intel unveiled ControlFlag – a machine programming research system that can autonomously detect errors in code. Even in its infancy, this novel, self-supervised system shows promise as a powerful productivity tool to assist software developers with the labor-intensive task of debugging. In preliminary tests, ControlFlag trained and learned novel defects on over 1 billion unlabeled lines of production-quality code.

# Programming Is The Most Popular Use Case on ChatGPT

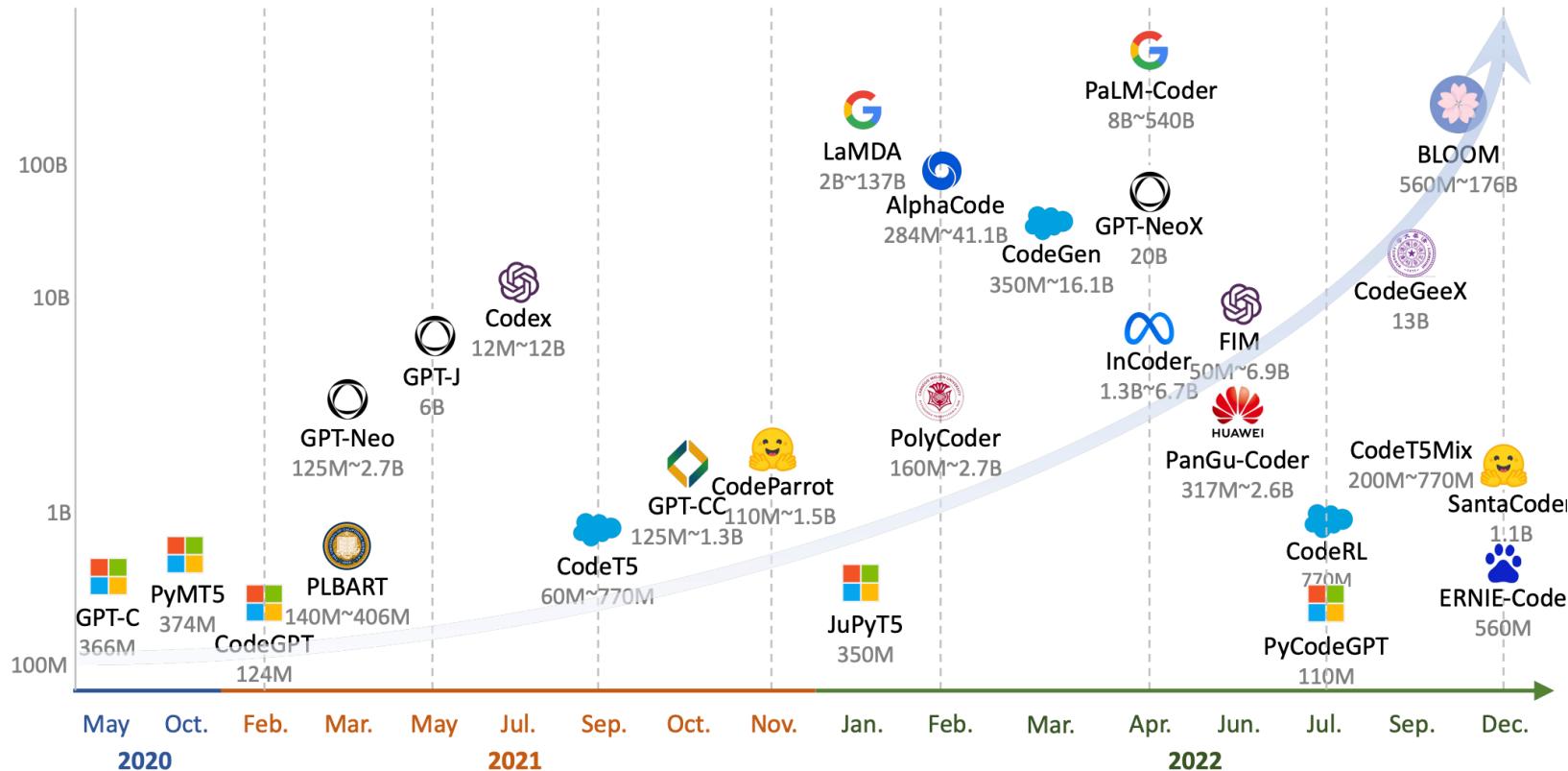
## # of Prompts per ChatGPT Session

(analysis of 4,098 unique users' sessions, May & June 2023)

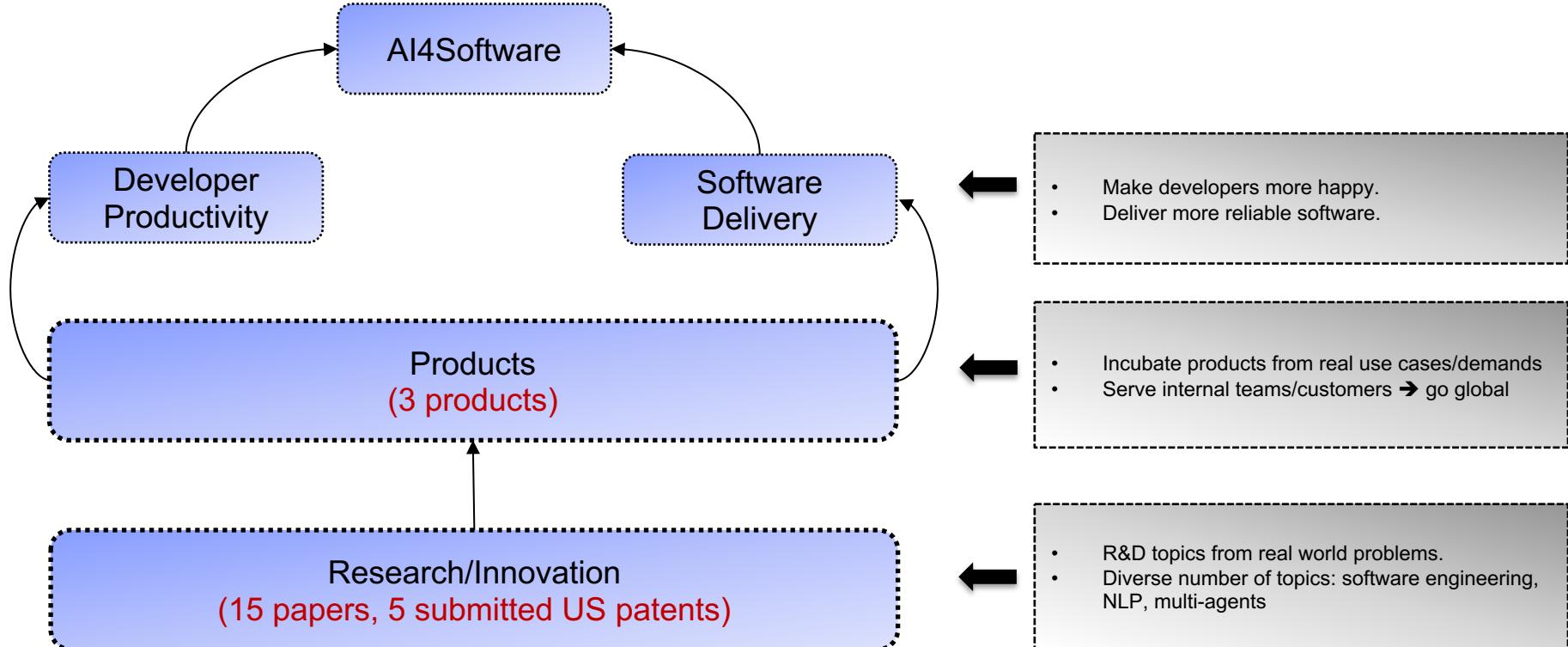
We Analyzed Millions of ChatGPT User Sessions: Visits are Down 29% since May, Programming Assistance is 30% of Use



# Investment to Build Large Language Models for Code from Industry



# AI4Software at FPT Software AI Center: What do we do?



# Why GenAI for Software Development at FSoft?



+30.000 Developers

+400 Outsourcing Projects

Large-Scale Internal Dataset

Technical Debts

**Productivity Boost**

# Notable Research Achievements – Research Papers

- ❖ We consistently publish scientific papers at top-tier conferences, including ICSE, FSE, ASE, ACL, EMNLP, NeuRIPS, AAAI.
- ❖ Selected Publications:
  - **The Vault: A Comprehensive Multilingual Dataset for Advancing Code Understanding and Generation, by Dung Nguyen Manh, Nam Le Hai, Anh T. V. Dau, Anh Minh Nguyen, Khanh Nghiêm, Jin Guo, Nghi D. Q. Bui,** in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (**EMNLP 2023**), **in collaboration with Mila**
  - **Better Language Models of Code through Self-Improvement, by Hung Quoc To, Nghi D. Q. BUI, Jin Guo, Tien N. Nguyen,** in Proceedings of The 61st Annual Meeting of the Association for Computational Linguistics (**ACL 2023**), **in collaboration with Mila .**
  - **HierarchyNet: Learning to Summarize Source Code with Heterogeneous Representations,** by Minh H. Nguyen, Nghi D. Q. Bui, Truong Son Hy, Long Tran-Thanh, Tien N. Nguyen., The 17th Conference of the European Chapter of the Association for Computational Linguistics (**EACL 2024**), **in collaboration with UT Dallas, UC San Diego.**
  - ..... **and 12 more**

# Notable Research Achievements – Global Competition

- We ranked high on global leaderboards.

microsoft.github.io/CodeXGLUE/

## CodeXGLUE

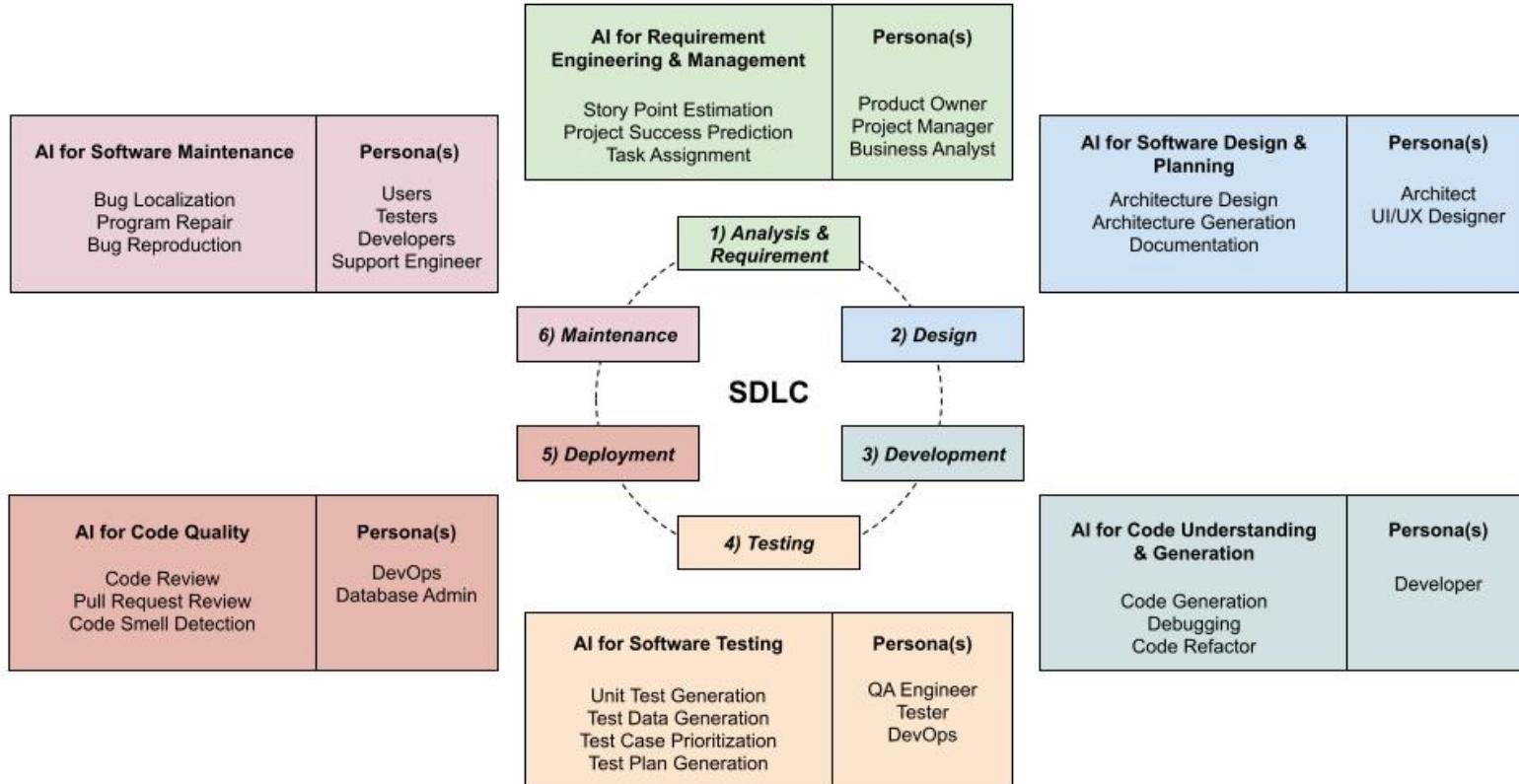
### Code Summarization (Code-Text)

Rank	Model	Organization	Date
1	DistillCodeT5	FSOFT AI Lab	2022-04-01
2	PolyglotCodeBERT	UC Davis	2021-07-30
3	CoTexT	Case Western R...	2021-04-23
4	ProphetNet-X	USTC & MSRA	2021-05-07
5	PLBART	UCLA & Columbi...	2021-04-02
6	CodeBERT	CodeXGLUE Team	2020-08-30
7	RnRFTa	CodeXGI LIF Team	2020-08-30

paperswithcode.com/sota/code-generation-on-humaneval

1	Language Agent Tree Search (GPT-4)	94.4	✓	Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models	2023
2	Reflexion (GPT-4)	91.0	✓		2023
3	AgileCoder (GPT-4)	87.80	✗		2024
4	OctorCoder (GPT-4)	86.6	✗	OctoPack: Instruction Tuning Code Large Language Models	2023
5	ANPL (GPT-4)	86.6	✗	ANPL: Towards Natural Programming with Interactive Decomposition	2023
6	MetaGPT (GPT-4)	85.9	✗	MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework	2023
7	Parsel (GPT-4 + CodeT)	85.1	✗	Parsel: Algorithmic Reasoning with Language Models by Composing Decompositions	2022
8	Language Agent Tree Search (GPT-3.5)	83.8	✗	Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models	2023
9	ANPL (GPT-3.5)	76.2	✗	ANPL: Towards Natural Programming with Interactive Decomposition	2023
10	INTERVENOR	75.6	✗	INTERVENOR: Prompting the Coding Ability of Large Language Models with the Interactive Chain of Repair	2023
11	SRank (WizardCoder)	75.31	✗	Neural Rankers for Code Generation via Inter-Cluster Modeling	2023
12	Gemini Ultra (zero-shot)	74.4	✗	Gemini: A Family of Highly Capable Multimodal Models	2023
13	Gemini Pro	73.7	✗	Gemini: A Family of Highly Capable Multimodal Models	2023

# How AI Can Be Applied Into Software Development Life Cycle?





# State-Of-The-Arts CodeLLMs

# CodeT5 and CodeT5+

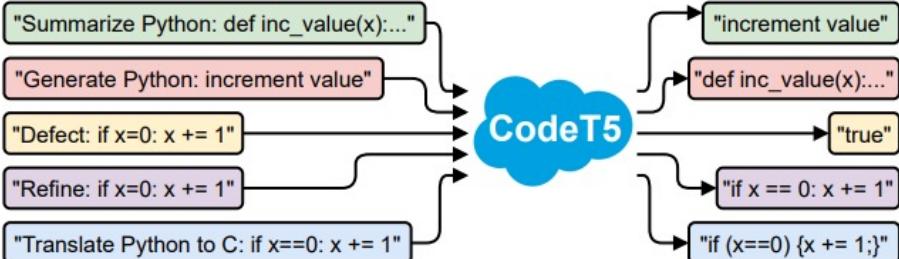


Figure 1: Illustration of our CodeT5 for code-related understanding and generation tasks.

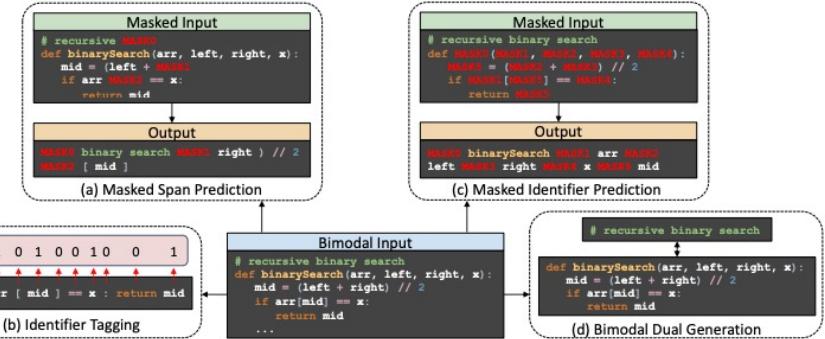


Figure 2: Pre-training tasks of CodeT5. We first alternately train span prediction, identifier prediction, and identifier tagging on both unimodal and bimodal data, and then leverage the bimodal data for dual generation training.

## CodeT5+: Open Code Large Language Models for Code Understanding and Generation

Yue Wang\*, Hung Le\*, Akhilesh Deepak Gotmare, Nghi D.Q. Bui, Junnan Li, Steven C.H. Hoi  
 Salesforce AI Research  
<https://github.com/salesforce/CodeT5/tree/main/CodeT5+>

## CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation

Yue Wang<sup>1</sup>, Weishi Wang<sup>12</sup>, Shafiq Joty<sup>12</sup>, and Steven C.H. Hoi<sup>1</sup>

<sup>1</sup> Salesforce Research Asia

<sup>2</sup> Nanyang Technological University, Singapore

{wang.y, weishi.wang, sjoty, shoi}@salesforce.com

# CodeGen, CodeGen2 and CodeGen2.5

Published as a conference paper at ICLR 2023

## CODEGEN: AN OPEN LARGE LANGUAGE MODEL FOR CODE WITH MULTI-TURN PROGRAM SYNTHESIS

Erik Nijkamp\*, Bo Pang\*, Hiroaki Hayashi\*

Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, Caiming Xiong

Salesforce Research

## CODEGEN2: LESSONS FOR TRAINING LLMs ON PROGRAMMING AND NATURAL LANGUAGES

Erik Nijkamp\*, Hiroaki Hayashi\*, Caiming Xiong, Silvio Savarese, Yingbo Zhou

Salesforce Research

## CodeGen2.5: Small, but mighty

8 min read

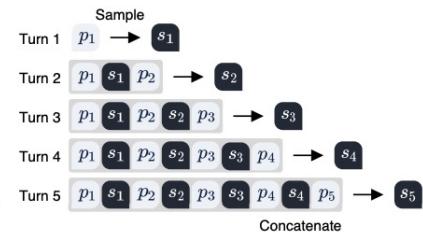
Erik Nijkamp Hiroaki Hayashi Yingbo Zhou Caiming Xiong

### ① Discourse

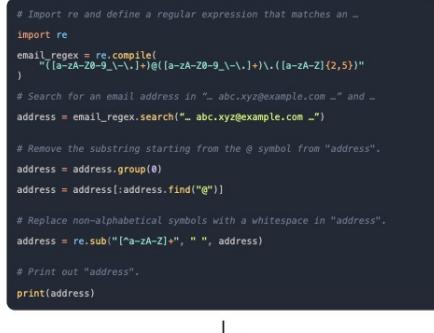


Expected Output "abc xyz"

### ② Generation



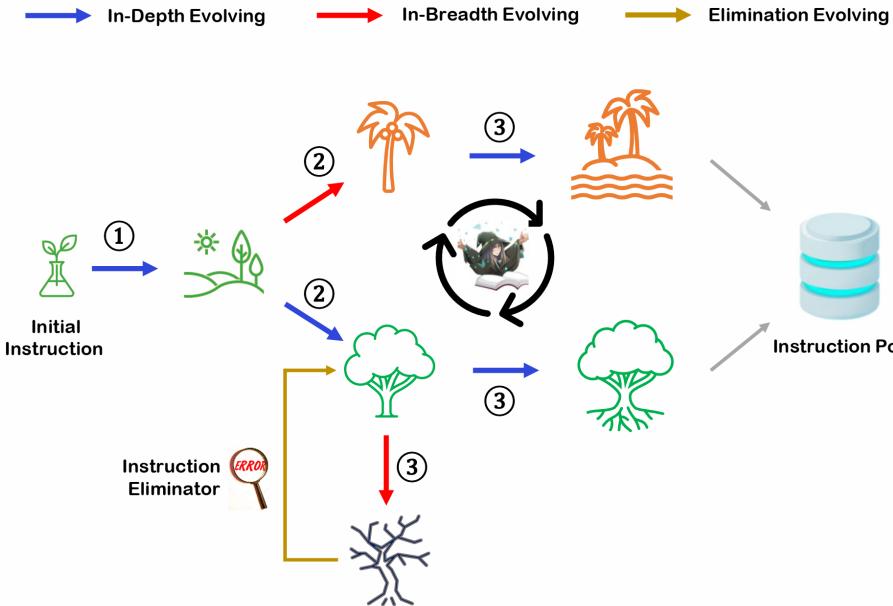
### ③ Evaluation



✓

"abc xyz" Actual Output

# WizardCoder



## WizardCoder: Empowering Large Language Models with Evol-Instruct

Ziyang Luo<sup>2\*</sup> Can Xu<sup>1\*</sup> Pu Zhao<sup>1</sup> Qingfeng Sun<sup>1</sup> Xiubo Geng<sup>1</sup>  
 Wenxiang Hu<sup>1</sup> Chongyang Tao<sup>1</sup> Jing Ma<sup>2</sup> Qingwei Lin<sup>1</sup> Dixin Jiang<sup>1†</sup>

<sup>1</sup>Microsoft

<sup>2</sup>Hong Kong Baptist University

{caxu, puzhao, qins, xigeng, wenxh, chongyang.tao, qlin, djiang}@microsoft.com  
 {czsyluo, majing}@comp.hkbu.edu.hk

The process of plant photosynthesis is commonly written as:  
 $6CO_2 + 6H_2O \rightarrow C_6H_{12}O_6 + 6O_2$   
 Please explain the main role of chlorophyll in above formula.

In-Breadth Evolving

```
import math
import random

# choose a random integer between 1 and 10
x = random.randint(1, 10)
1/(math.sqrt(x) + x^2) = ?
```

Complicate Input (Code)

How many times faster is light than sound in a vacuum?

How is the speed of light in a vacuum measured and defined?

Increase Reasoning

What is the speed of light in a vacuum?

If you have one apple and someone gives you another banana, how many fruits do you have?

Deepening

Concretizing

Add Constraints

Initial Instruction

1 + 1 = ?

In what situation does 1+1 not equal to 2?

What is the value of x, if  $x^3 + 2x + 3 = 7$ ?

What is the value of x, if  $x^3 + 2x + 3 = 7$ ?

Please fill in the table below with the approximate values of the speed of light in each medium.

Medium	Speed of light (km/s)
Air	
Water	
Glass	

# Other Notable CodeLLMs

Textbooks Are All You Need

Suriya Gunasekar	Yi Zhang	Jyoti Aneja	Cao César Teodoro Mendes
Allie Del Girono	Sivakanth Gopi	Mojan Javaheripi	Piero Kauffmann
Gustavo de Rosa	Olli Saarikivi	Adil Salim	Shital Shah Harkirat Singh Behl
Xin Wang	Sébastien Bubeck	Ronen Eldan	Adam Tauman Kalai Yin Tat Lee
			Yuanzhi Li

Microsoft Research

Published as a conference paper at ICLR 2023



**STARCODER:**

MAY THE SOURCE BE WITH YOU!

Raymond Li<sup>2</sup> Loubna Ben Allal<sup>1</sup> Yangtian Zi<sup>4</sup> Niklas Muennighoff<sup>1</sup> Denis Kocetkov<sup>2</sup>  
 Chenghao Mou<sup>1</sup> Marc Marone<sup>3</sup> Christopher Akiki<sup>9,10</sup> Jia Li<sup>5</sup> Jenny Chim<sup>11</sup>  
 Qian Liu<sup>13</sup> Evgenii Zheltonozhskii<sup>14</sup> Terry Yu Zhuo<sup>15,16</sup> Thomas Wang<sup>1</sup>  
 Olivier Dehaene<sup>1</sup> Mishig Davaadorj<sup>1</sup> Joel Lamy-Poirier<sup>2</sup> João Monteiro<sup>2</sup>  
 Oleh Shlizhko<sup>2</sup> Nicolas Gontier<sup>2</sup> Nicholas Meade<sup>6,17</sup> Armel Zebaze<sup>1</sup> Ming-Ho Yee<sup>4</sup>  
 Logesh Kumar Umapathi<sup>18</sup> Jian Zhu<sup>19</sup> Benjamin Lipkin<sup>20</sup> Muhtasham Oblokulov<sup>21</sup>  
 Zhiruo Wang<sup>7</sup> Rudra Murthy<sup>22</sup> Jason Stillerman<sup>23</sup> Siva Sankalp Patel<sup>22</sup>  
 Dmitry Abulkhanov<sup>5</sup> Marco Zocca<sup>24</sup> Manan Dey<sup>25</sup> Zhihan Zhang<sup>26</sup> Nour Fahmy<sup>27</sup>  
 Urvasi Bhattacharyya<sup>28</sup> Wenhao Yu<sup>26</sup> Swapayam Singh<sup>30</sup>  
 Sasha Luccioni<sup>1</sup> Paulo Villegas<sup>31</sup> Maxim Kunakov<sup>2</sup> Fedor Zhdanov<sup>32</sup>  
 Manuel Romero<sup>5</sup> Tony Lee<sup>33</sup> Nadav Timor<sup>34</sup> Jennifer Ding<sup>35</sup> Claire Schlesinger<sup>4</sup>  
 Hailey Schoelkopf<sup>37</sup> Jan Ebert<sup>38</sup> Tri Dao<sup>33</sup> Mayank Mishra<sup>32</sup> Alex Gu<sup>20</sup>  
 Jennifer Robinson<sup>3</sup> Carolyn Jane Anderson<sup>36</sup> Brendan Dolan-Gavitt<sup>29</sup>  
 Danish Contractor<sup>3</sup> Siva Reddy<sup>2,6</sup> Daniel Fried<sup>2</sup> Dzmitry Bahdanau<sup>2</sup> Yacine Jernite<sup>1</sup>  
 Carlos Muñoz Ferrandis<sup>1</sup> Sean Hughes<sup>3</sup> Thomas Wolf<sup>1</sup> Arjun Guha<sup>4,12</sup>  
 Leandro von Werra<sup>1,\*</sup> Harm de Vries<sup>2,\*</sup>

<sup>1</sup>Hugging Face <sup>2</sup>ServiceNow Research <sup>3</sup>ServiceNow <sup>4</sup>Northeastern University <sup>5</sup>Independent  
<sup>6</sup>Mila <sup>7</sup>Carnegie Mellon University <sup>8</sup>Johns Hopkins University <sup>9</sup>Leipzig University  
<sup>10</sup>ScADS.AI <sup>11</sup>Queen Mary University of London <sup>12</sup>Roblox <sup>13</sup>Sea AI Lab <sup>14</sup>Technion –  
 Israel Institute of Technology <sup>15</sup>Monash University <sup>16</sup>CSIRO's Data61 <sup>17</sup>McGill University  
<sup>18</sup>Saama AI Research Lab <sup>19</sup>University of British Columbia <sup>20</sup>MIT <sup>21</sup>Technical University of  
 Munich <sup>22</sup>IBM Research <sup>23</sup>University of Vermont <sup>24</sup>UnfoldML <sup>25</sup>SAP <sup>26</sup>University of  
 Notre Dame <sup>27</sup>Columbia University <sup>28</sup>Discover Dollar Pvt Ltd <sup>29</sup>NYU <sup>30</sup>University of  
 Allahabad <sup>31</sup>Telefonica I+D <sup>32</sup>Toloka <sup>33</sup>Stanford University <sup>34</sup>Weizmann Institute of Science  
<sup>35</sup>The Alan Turing Institute <sup>36</sup>Wellesley College <sup>37</sup>Eleuther AI <sup>38</sup>Forschungszentrum Jülich

Corresponding authors (\*) can be contacted at [contact@bigcode-project.org](mailto:contact@bigcode-project.org)

## PANGU-CODER: Program Synthesis with Function-Level Language Modeling

TECHNICAL REPORT

Fenia Christopoulou<sup>1\*</sup> Gerasimos Lampouras<sup>1\*</sup> Milan Grittia<sup>1\*</sup> Guchun Zhang<sup>1\*</sup>  
 Yingpeng Guo<sup>1\*</sup> Zhongqi Li<sup>2\*</sup> Qi Zhang<sup>2\*</sup> Meng Xiao<sup>1</sup> Bo Shen<sup>2</sup> Lin Li<sup>2</sup>  
 Hao Yu<sup>2</sup> Li Yan<sup>2</sup> Pingyi Zhou<sup>1</sup> Xin Wang<sup>1</sup> Yuchi Ma<sup>2†</sup> Ignacio Iacobacci<sup>1†</sup>  
 Yasheng Wang<sup>1†</sup> Guangtai Liang<sup>2</sup> Jiansheng Wei<sup>1</sup> Xin Jiang<sup>1</sup>  
 Qianxiang Wang<sup>2</sup> Qun Liu<sup>1</sup>

<sup>1</sup>Huawei Noah's Ark Lab

<sup>2</sup>Huawei Cloud



2022-3-16

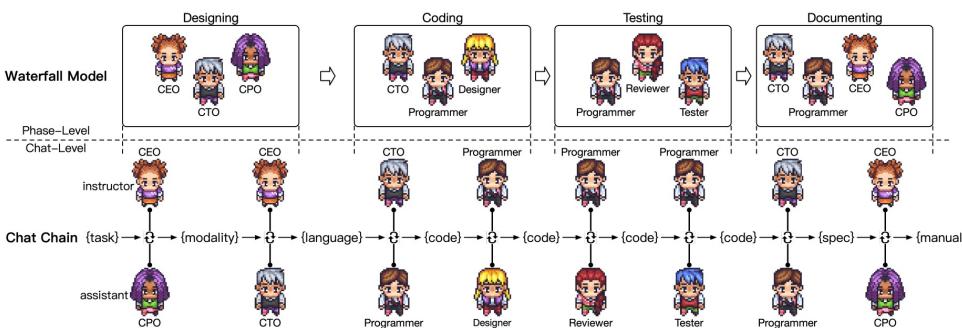
## Competition-Level Code Generation with AlphaCode

Yujia Li<sup>\*</sup>, David Choi<sup>\*</sup>, Junyoung Chung<sup>\*</sup>, Nate Kushman<sup>\*</sup>, Julian Schrittewieser<sup>\*</sup>, Rémi Leblond<sup>\*</sup>, Tom Eccles<sup>\*</sup>, James Keeling<sup>\*</sup>, Felix Gimeno<sup>\*</sup>, Agustin Dal Lago<sup>\*</sup>, Thomas Hubert<sup>\*</sup>, Peter Choy<sup>\*</sup>, Cyprien de Masson d'Autume<sup>\*</sup>, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu and Oriol Vinyals  
<sup>\*</sup>Joint first authors

**LLMs-Enabled Agents for Software Developments - 1**

# Communicative Agents for Software Development

Chen Qian<sup>\*</sup> Xin Cong<sup>\*</sup> Wei Liu<sup>\*</sup> Cheng Yang<sup>\*</sup> Weize Chen<sup>\*</sup> Yusheng Su<sup>\*</sup>  
Yufan Dang<sup>\*</sup> Jiahao Li<sup>\*</sup> Juyuan Xu<sup>\*</sup> Dahai Li<sup>†</sup> Zhiyuan Liu<sup>‡</sup> Maosong Sun<sup>‡</sup>  
<sup>\*</sup>Tsinghua University   <sup>†</sup>Beijing University of Posts and Telecommunications  
<sup>‡</sup>Dalian University of Technology   <sup>§</sup>Brown University   <sup>¶</sup>Modelbest Inc.  
qianc62@gmail.com liuyz@tsinghua.edu.cn sms@tsinghua.edu.cn



**Figure 2:** The proposed architecture of ChatDev consists of phase-level and chat-level components. At the phase level, the waterfall model is used to break down the software development process into four sequential phases. At the chat level, each phase is further divided into atomic chats. These atomic chats involve task-oriented role-playing between two agents, promoting collaborative communication. The communication follows an instruction-following style, where agents interact to accomplish a specific subtask within each chat.

# METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK

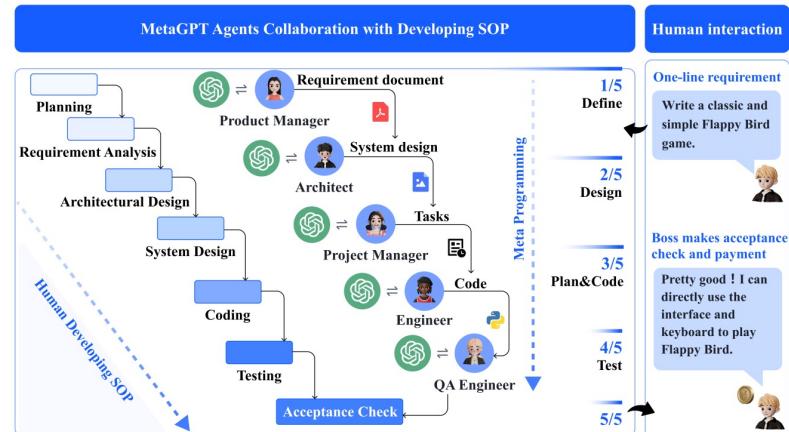
Sirui Hong<sup>1\*</sup>, Mingchen Zhu<sup>2\*</sup>, Jonathan Chen<sup>1</sup>, Xiawu Zheng<sup>3</sup>, Yuheng Cheng<sup>4</sup>, Ceyao Zhang<sup>4</sup>, Jinlin Wang<sup>1</sup>, Zili Wang<sup>1</sup>, Steven Ka Shing Yau<sup>5</sup>, Zijuan Lin<sup>4</sup>, Liyang Zhou<sup>6</sup>, Chenyu Ran<sup>1</sup>, Lingfeng Xiao<sup>1,7</sup>, Chenglin Wu<sup>1†</sup>, Jürgen Schmidhuber<sup>2,8</sup>

<sup>1</sup>DeepWisdom, <sup>2</sup>AI Initiative, King Abdullah University of Science and Technology,

<sup>3</sup>Xiamen University, <sup>4</sup>The Chinese University of Hong Kong, Shenzhen,

<sup>5</sup>Nanjing University, <sup>6</sup>University of Pennsylvania,

<sup>7</sup>University of California, Berkeley, <sup>8</sup>The Swiss AI Lab IDSIA/USI/SUPSI



**Figure 1: The software development SOPs between MetaGPT and real-world human teams.** In software engineering, SOPs promote collaboration among various roles. MetaGPT showcases its ability to decompose complex tasks into specific actionable procedures assigned to various roles (e.g., Product Manager, Architect, Engineer, etc.).

# LLMs-Enabled Agents for Software Developments - 2



## If LLM Is the Wizard, Then Code Is the Wand: A Survey on How Code Empowers Large Language Models to Serve as Intelligent Agents

Ke Yang\*, Jiateng Liu\*, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li,  
Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, Heng Ji, Chengxiang Zhai  
University of Illinois Urbana-Champaign

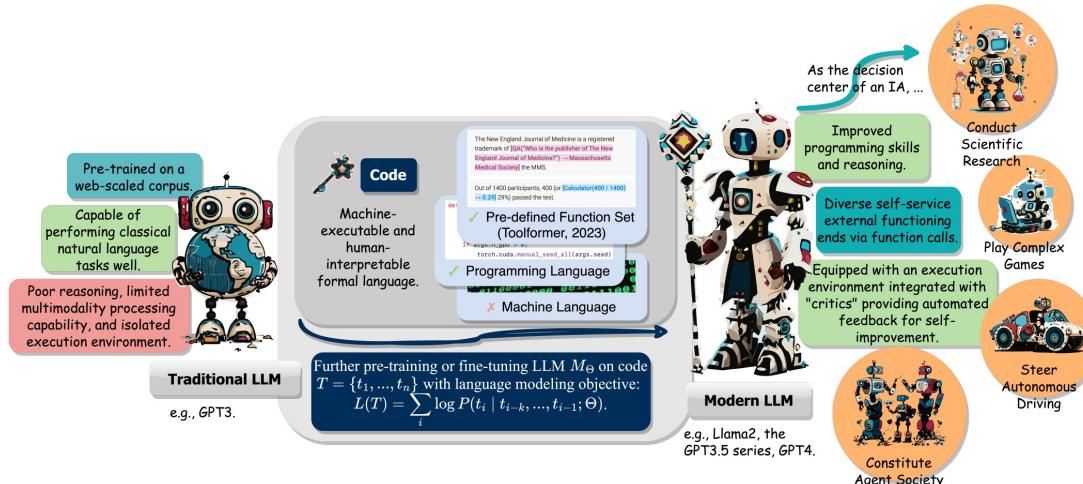
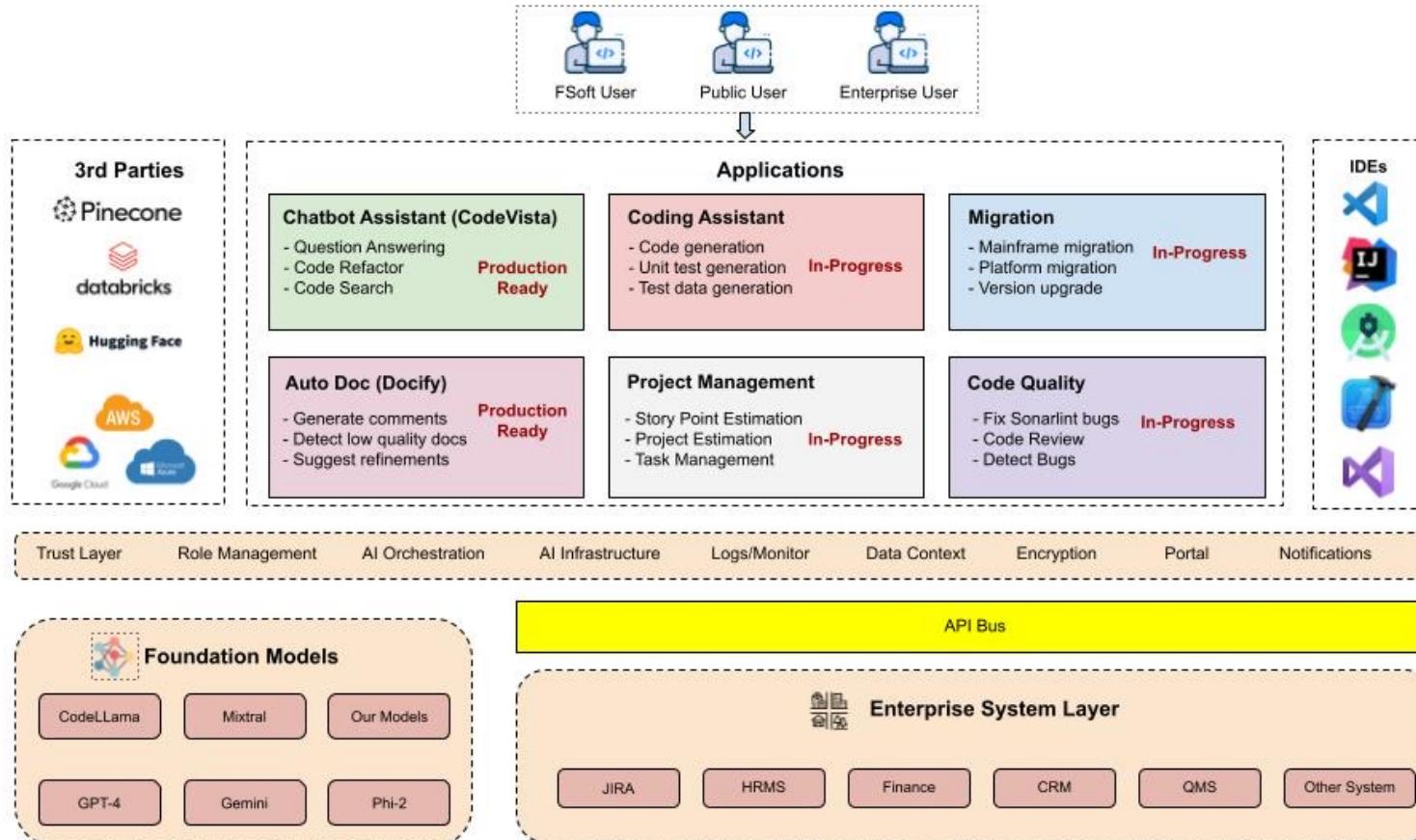


Figure 1: An illustration of how code empowers large language models (LLMs) and enhances their downstream applications as intelligent agents (IAs). While traditional LLMs excel in conventional natural language tasks like document classification and question answering, further pre-training or fine-tuning LLMs with human-interpretable and machine-executable code serves as an additional power-up — akin to equipping wizards with mana-boosting wands. This significantly boosts their performance as IAs through intricately woven operational steps.

# Our AI4Software Ecosystem



# CodeVista

## AI Coding Assistant

AIC – FPT Software

# Developers must often ask for help

Developers seek help from teammates, managers, tools, and of course, the Internet.

Building reliable and efficient applications is difficult, developers are always searching for or build tools that will help them with coding tasks.

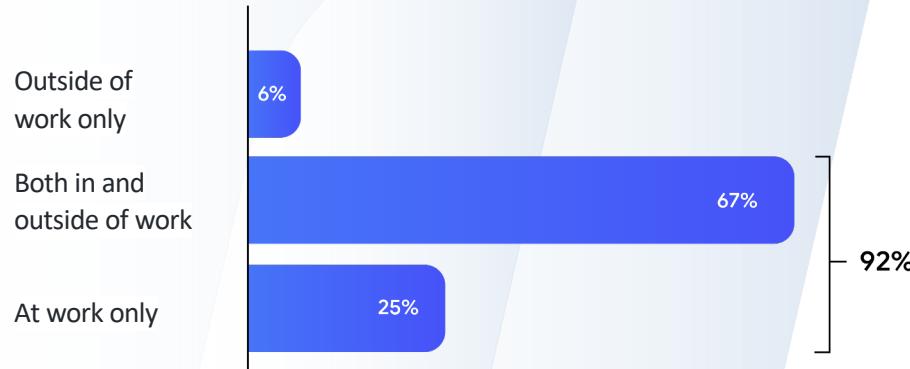
**Generative AI** is ushering in a big wave of helpful developer tools that can boost productivity, quality, and work experience.



## 92% of US Developers are already using AI at work

Where AI coding tools are used

N=500



**Developers believe AI coding tools will enhance their performance**

Where AI can help:

- Meet existing performance standards
- Improve code quality
- Faster outputs
- Fewer production-level incidents.

# 70% of developers

See a benefit to using AI coding tools at work

# CodeVista at a glance

**CodeVista** is an **AI-powered, context-rich & streamlined** assistant inside the IDE that answers coding questions and helps with coding tasks

AI that provides knowledge about code and programming

*Option to verify AI answers with sources from Google Search*

Pre-built flows + prompt templates to help with coding tasks

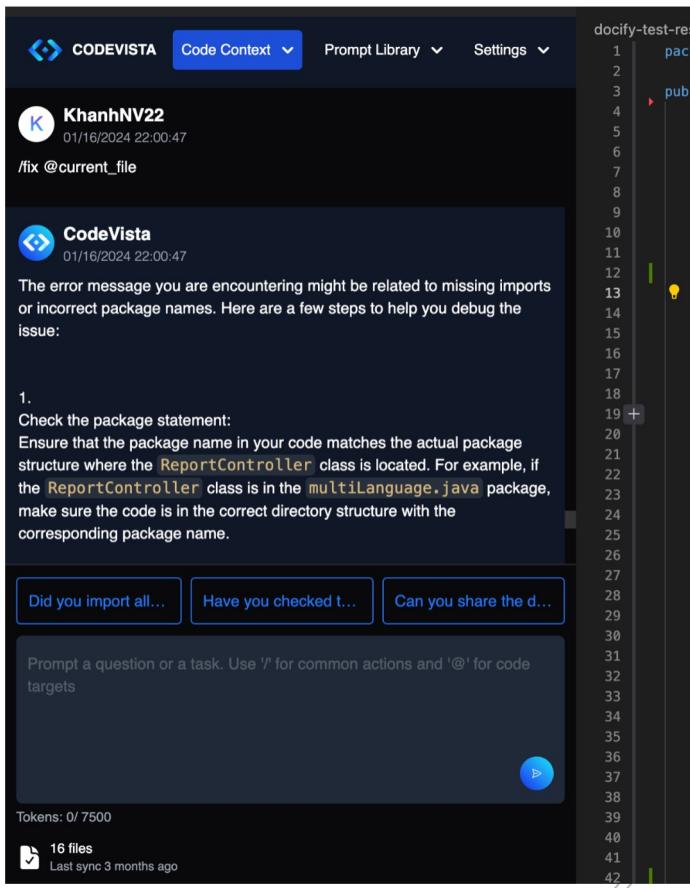
*Code completion, code explain, code refactor, code test, etc.*

Full visibility & control of the data you provide

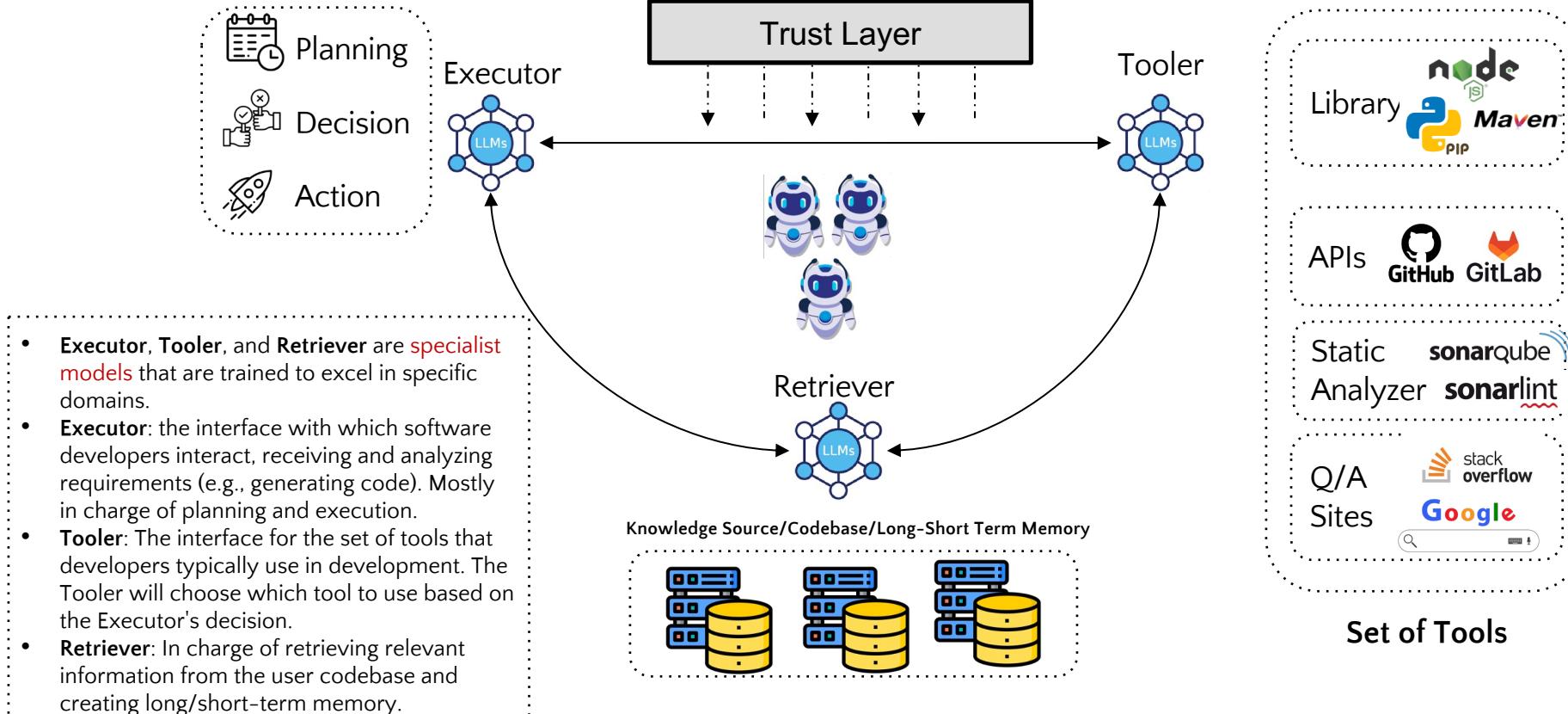
*Upload or remove code that CodeVista can use as context*

Secure & responsible access to the best AI models

*More info in the next slide →*

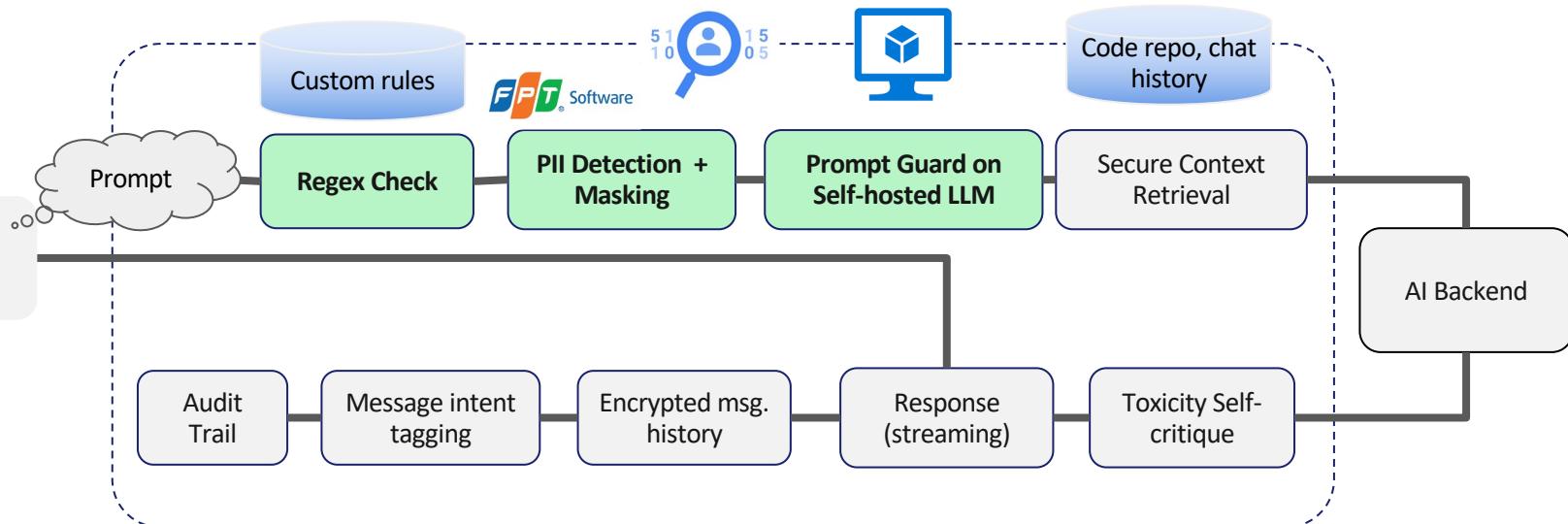


# Multi-Agents Platform for Next-Generation Coding Assistant



# CodeVista Trust Layer

- Additional policies to control user input
  - Detect and mask forbidden key words, PII and credentials data
- Log and notify violations, review audit trail





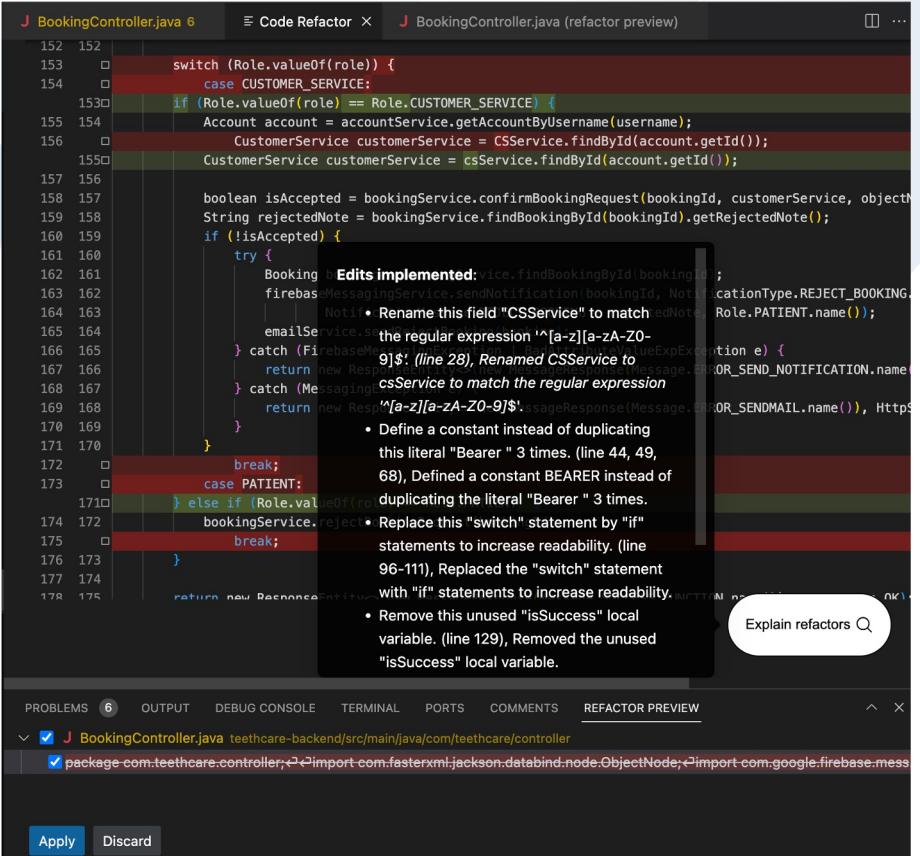
# SonarLint integration

## Current version

- Fetch first 10 suggestions from SonarLint
- CodeVista refactor code + provide explanations
- Refactor Preview + Report View
- Token limit of input file – 6500 tokens

## Next version

- Deselect suggestions to refactor
- Multi-round refactoring
- Flag edits with downstream dependencies
- Suggest tests to verify new code



J BookingController.java 6    Code Refactor    J BookingController.java (refactor preview)

```

152 152
153 153   switch (Role.valueOf(role)) {
154 154     case CUSTOMER_SERVICE:
155 155       if (Role.valueOf(role) == Role.CUSTOMER_SERVICE) {
156 156         Account account = accountService.getAccountByUsername(username);
157 157         CustomerService customerService = CSservice.findById(account.getId());
158 158         CustomerService customerService = csService.findById(account.getId());
159 159
160 160         boolean isAccepted = bookingService.confirmBookingRequest(bookingId, customerService, object);
161 161         String rejectedNote = bookingService.findBookingById(bookingId).getRejectedNote();
162 162         if (!isAccepted) {
163 163           try {
164 164             Booking b Edits implemented: ervice.findBookingById(bookingId);
165 165             firebaseMessagingService.sendNotification(bookingId, NotificationType.REJECT_BOOKING,
166 166             Note);
167 167             emailService.sendEmail(account.getEmail(), "Booking Rejected", "Your booking request was rejected");
168 168           } catch (FirebaseException e) {
169 169             return new ResponseEntity<String>(e.getMessage(), HttpStatus.EXPECTATION_FAILED);
170 170           }
171 171         }
172 172         break;
173 173       case PATIENT:
174 174         } else if (Role.valueOf(role) == Role.PATIENT) {
175 175         bookingService.reject();
176 176       }
177 177     }
178 178   }
179 179   return new ResponseEntity<String>(message, HttpStatus.OK);

```

PROBLEMS 6    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    COMMENTS    REFACTOR PREVIEW

✓ J BookingController.java teethcare-backend/src/main/java/com/teethcare/controller

✓ package com.teethcare.controller; import com.fasterxml.jackson.databind.node.ObjectNode; import com.google.firebaseio.mess

Apply   Discard

## What it is

- AI assistant tool
- Tool developed in house by FPT
- Free to use to FSoft
- Evolving features from R&D
- Output depends on prompts

## What it is NOT

- Senior developer replacement
- 3rd party vendor product
- Paid product
- Static features
- Magic understanding

CODEVISTA: CHATBOT

Code Context ▾ Prompt Library ▾ Settings ▾

CodeVista

Hello! How can I assist you today?

KhanhNV22

What can you help me with?

CodeVista

I can help you with a wide range of programming-related problems. Whether you need help with syntax, debugging, algorithm design, or any other coding concept, I'm here to assist you. Just provide me with the necessary information and I'll do my best to guide you in the right direction. So, feel free to ask me anything coding-related!

Prompt a question or a task. Use '/' for common actions and '@' for code targets

Tokens: 0/ 7500

10 files Last sync 19 days ago

go\_parser.py javascript\_parser.py 4 python-visualize.py 8 python-visualize0.py 4

```
1 import matplotlib.pyplot as plt
2 from matplotlib import dates as mpl_dates
3 import pandas as pd
4
5 # Define project tasks and their start and end dates
6
7 tasks = {
8     #Add more mock data to have 10 entries
9     'Task 1': {'owner': 'John', 'start': '2023-01-01', 'end': '2023-01-15', 'priority':
10     'Task 2': {'owner': 'Jane', 'start': '2023-01-16', 'end': '2023-02-01', 'priority':
11 }
12
13 # Convert start and end dates to datetime objects
14
15 # Draw the figure with axis
16 +
17 # Plot the start and end dates for each task
18
19 # Show plot
20
21
```

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS ... Filter (e.g. text, \*\*/ts, !\*\*/n...)

python-visualize0.py demos 4

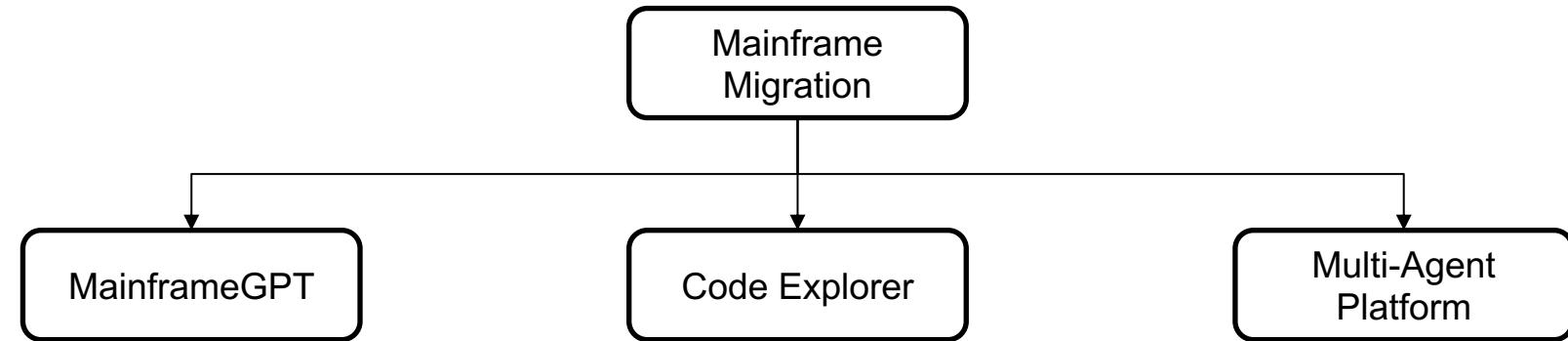
- ⚠ Import "matplotlib.pyplot" could not be resolved from source Pylance(reportMissingModuleSource) [Ln 1, Col 8]
- ⚠ Import "matplotlib" could not be resolved from source Pylance(reportMissingModuleSource) [Ln 2, Col 6]
- ⚠ Import "matplotlib.dates" could not be resolved from source Pylance(reportMissingModuleSource) [Ln 2, Col 24]
- ⚠ Import "pandas" could not be resolved from source Pylance(reportMissingModuleSource) [Ln 3, Col 9]

# Migration Platform

## Copilot for Code Migration

AIC – FPT Software

# Mainframe Migration



Foundation Model Specialized for  
Mainframe Migration (to replace with  
OpenAI GPT)

Platform to explore Cobol codebase,  
understand and generate specs

Multi-Agent migration platform to  
replace traditional migration tools  
(most innovative + breakthrough)

# AI4Software: Mainframe Migration

Input	Discovery and Assessment	Definition	Reverse engineering	Designs	Migration	Testing	Deployment
<b>Legacy System</b> Application, Middleware, 3rd party, Peripheral equipment,....	1. Define Migration Strategy 2. Analyze Application Complexity 3. Analyze Code Quality 4. Analyze Dependencies 5. Analyze Infrastructure 6. Baseline Scope 7. Estimate for Migration (v1) 8. Identify Migration issues	1. Define Migration Strategy 2. Define Requirements 3. Define TO-BE architecture 4. Perform PoC 5. Define Migration Process 6. Define Migration Design 7. Define Test Strategy 8. Plan for TO-BE infrastructure	1. Source code exploration ( Summarization, Input/output, flow chart ) 2. Section similarity analysis: present similarity in logic and feature 3. Control flow graph: present interaction between files in COBOL source 4. Dead code Analysis.	1. Business flow (VTOC) 2. Program design	1. Prepare source code 2. Migrate source code 3. Implement enhancements 4. Build source code 5. Data file (SAM/VSAM) migration 6. Encoding migration 7. Database Migration (DB2 to Oracle, etc.)	1. Measure similarity between source and target 2. Generate and Execute UT 3. Test and validate functionality <ul style="list-style-type: none"> <li>• Test scenario create</li> <li>• Test data prepare</li> </ul> 3. Test non-functional <ul style="list-style-type: none"> <li>• Performance</li> <li>• Security</li> <li>• Scalability</li> </ul>	1. Prepare for deployment 2. Deploy 3. Cut over

♦ AI can be applied into 5 phases with high accuracy (verified by Cobol experts)
 

- Code Explanation: ~60%
- Flow understanding with Code Graph: ~70%
- Test case assessment: ~50%
- Total effort saved (estimation): ~30%

 ♦ Piloting with Honda project.

# Mainframe Migration

### Matching

The screenshot shows a comparison interface between Java and COBOL. On the left, a sidebar lists files like getaccountdetails.java, account.java, and transaction.java. The main area shows Java code on the left and COBOL code on the right, with a matching progress bar at 95%.

```
import java.util.Scanner;
public class Account {
    public static void main(String[] args) { Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Account Number: ");
        long accountNumber = scanner.nextLong();
        getAccountDetails(accountNumber);
    }
    // Implementing a stub method to getAccountDetails public
    static void getAccountDetails(long accountNumber) { // Logic for fetching account details would go here
        System.out.println("Fetching details for account
number: " + accountNumber); }
```

```
DATA DIVISION.
WORKING-STORAGE SECTION.
77 WS-S1 PIC S9(04) COMP-3.
77 WS-S2 PIC S9(04) COMP-3.
77 WS-S3 PIC S9(04) COMP-3.
77 WS-S4 PIC S9(04) COMP-3.

COPY *HEADING.CRT.

COPY *WS.WS*.

COPY FUNCTION.WS*.

LINKAGE SECTION.

COPY *CHAIN.LS*.

77 WS-OPTION PIC X(01).

01 L-MESSAGE PIC X(48).
01 L-LINE PIC 9(02).

SCREEN SECTION.
01 OPTLINE 02 BACKGROUND-COLOR Cyan 03.
```

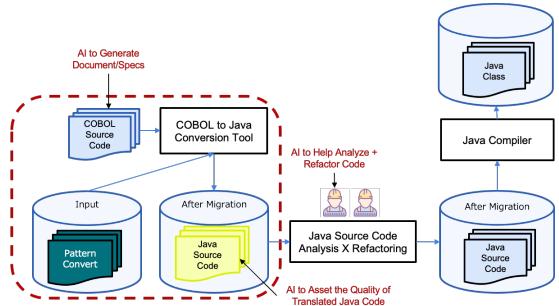
### Complexity

A circular gauge meter indicates a complexity level of 100%. Below it, a chart titled 'COBOL CODE / JCL / ASM' displays various metrics:

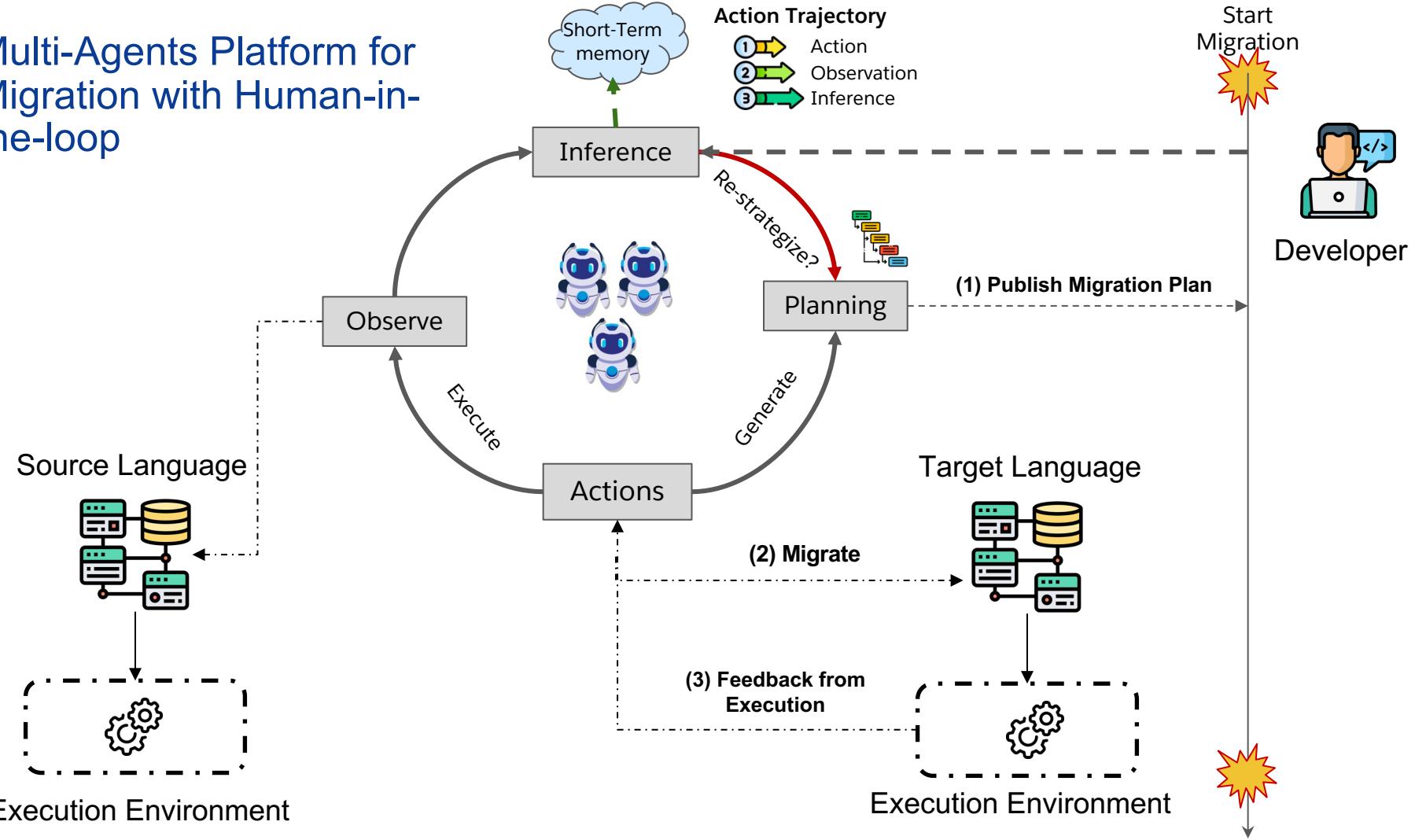
Metric	Value
Avg. LOC	618.62
Number of files	71
Cyclomatic Level	47.23
Avg. Number of tokens	3511.69
Avg. Number of operators	842.94

Legend: Low (blue), Medium (yellow), High (red).

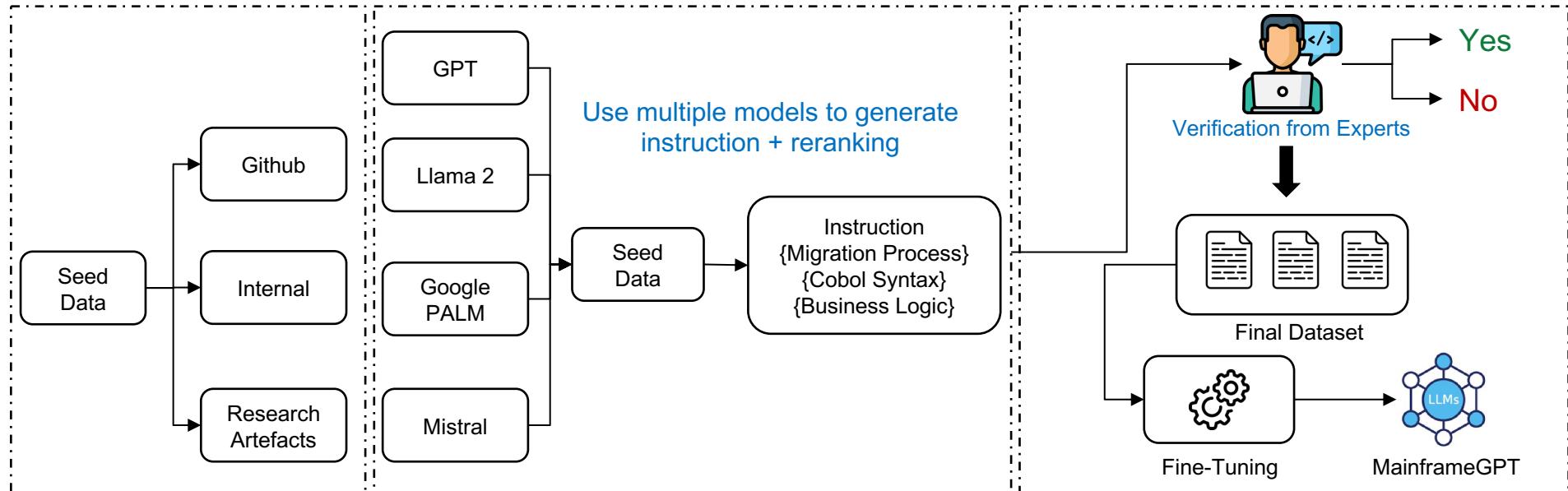
- Developing Cobol Explorer Platform for Cobol code understanding, assessment, business/logic understanding.
- Piloting on Honda project.
- Accuracy in business logic understanding: >70%.
- Accuracy of code explanation: >90%.
- Impact + Accuracy are verified by Cobol experts.



# Multi-Agents Platform for Migration with Human-in-the-loop



# MainframeGPT: Large Language Model for Mainframe Migration



Data Collection

Dialogue-Style Data Enrichment

Manual Verification + Fine-tuning

# Agile Copilot

## Copilot for Project Manager

AIC – FPT Software

# Agile Copilot: Multi-Agent Platform for Software Project Management

## Backlog Grooming

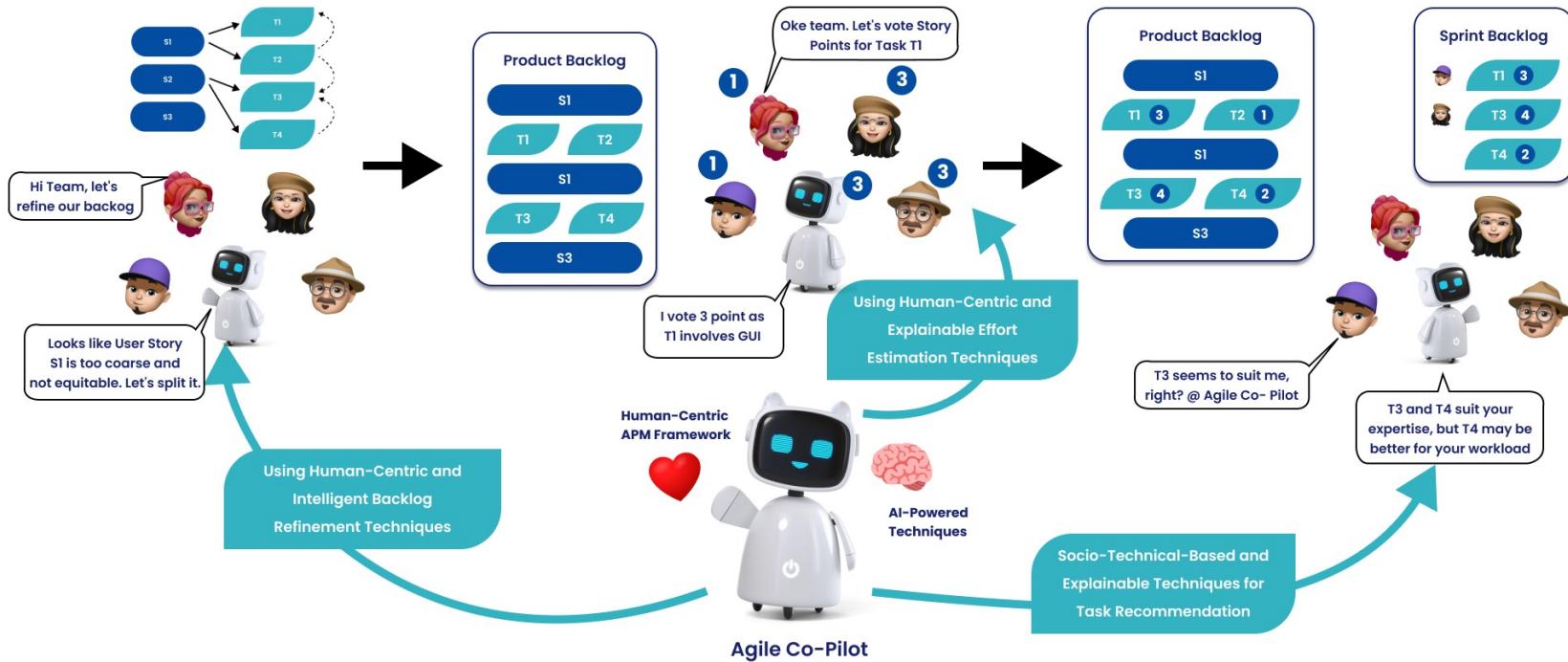
(Better Task Breakdown and Dependencies Refinement)

## Effort Estimation

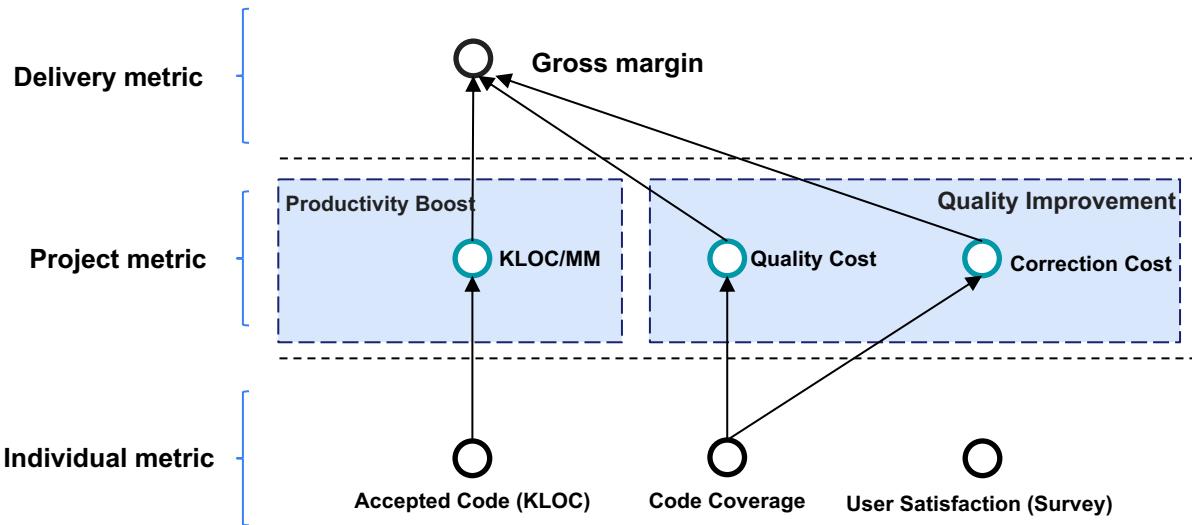
(Improving Accuracy)

## Task Allocation

(More Humane and Sustainable)



# Impact Measurement



## Delivery Metric

Measure the effectiveness at large scale to address the correlation between AI adoption and financial impact (gross margin growth)

## Project metric

AI adoption helps boosting productivity, improving quality of the output and results in *KLOC/MM* increase, deduction of *Quality Cost & Correction Cost*

The project level metrics would be measured in the set of project in which AI solution applied and compared to the norm of Fsoft projects in QMS system

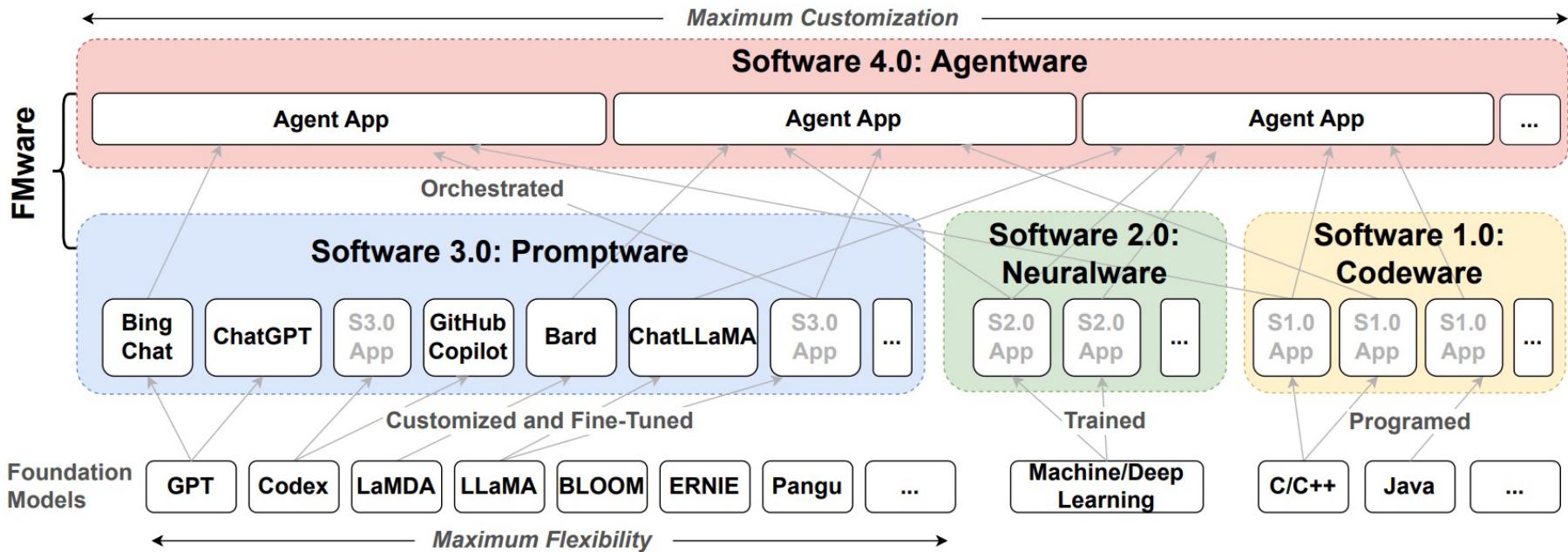
## Individual metric

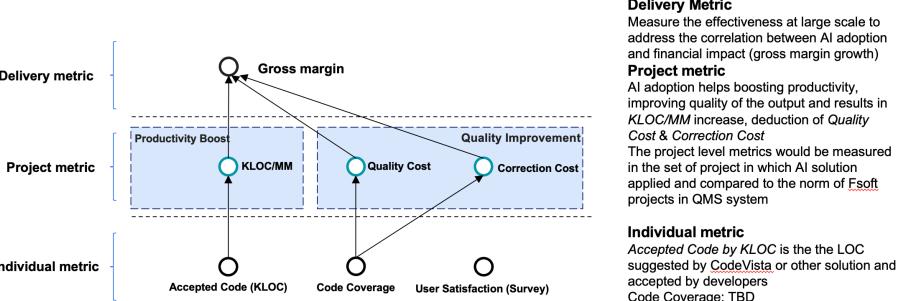
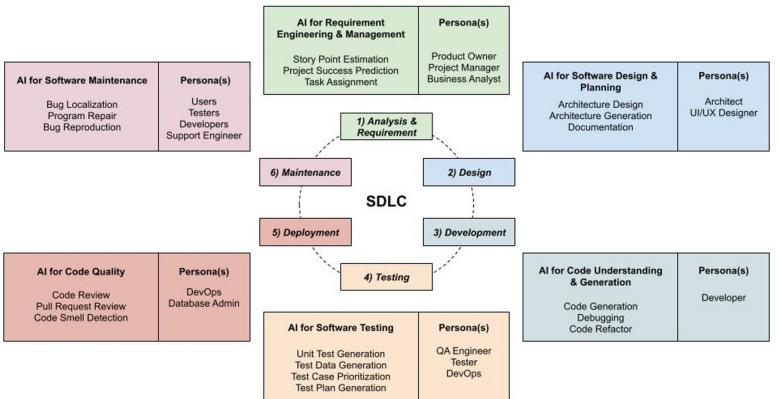
*Accepted Code by KLOC* is the LOC suggested by CodeVista or other solution and accepted by developers

Code Coverage: TBD

User Satisfaction is measured by surveying users on regular basis

# What Could Be The Future of Software? (\*)



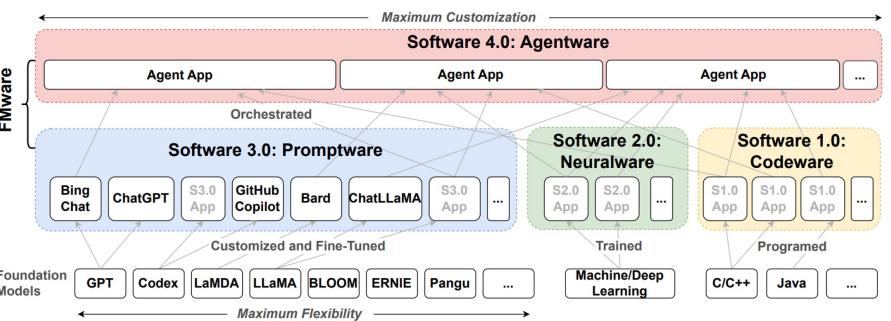
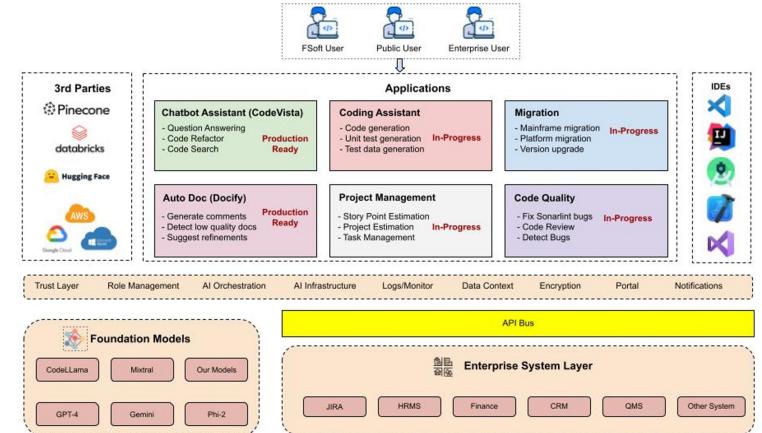


© Corp

© Copyright FPT Software AI Center – Confidential

## Our AI4Software Ecosystem

## What Could Be The Future of Software? (\*)



©

© Copyright FPT Software AI Center – Confidential

(\*) Rethinking Software Engineering in the Foundation Model Era, Hassan et al.

## **CONTACT US**

FPT Software AI Center,

10 Pham Van Bach Street, Dich Vong Hau Ward,  
Cau Giay District, Ha Noi City, Vietnam

Contact Person: Mr/Ms.ABC

Head of ABC

Email: [abc@fpt-software.com](mailto:abc@fpt-software.com)

# THANK YOU.