



GOTC 2023

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE, INTO THE FUTURE

「AI编程」专场

软件工程3.0的开发新范式

朱少民 同济大学

目录

CONTENTS

1. 从软件工程1.0到3.0 (SE3.0)
2. SE3.0的新形态
3. SE3.0的新开发范式
4. 新范式下编程将如何展开?
5. 企业如何更好利用新范式
6. 未来展望与挑战

PART 1

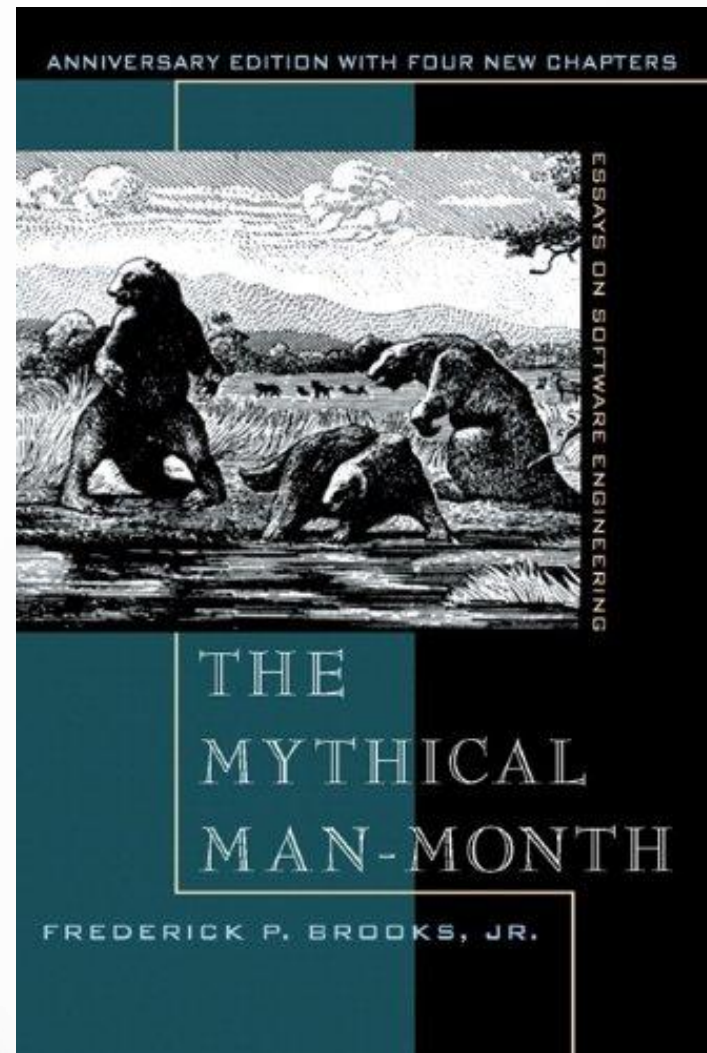
从软件工程1.0到软件工程3.0

软件工程 (Software Engineering, SE) 已走过了55年



上个世纪五、六十年代软件危机四伏

- 对软件开发成本和进度的估计不足，超预算、完成日期一再被推迟的现象屡屡出现
- 不能满足客户需求，用户不满意
- 缺乏质量保证体系，没有成熟的开发流程，产品的质量得不到保证
- 软件不容易被维护，错误难以纠正，新功能难以扩充
- 质量事故经常发生



标志性事件：1968年NATO会议

第一次提出Software Engineering 术语
本次会议标志着进入 **软件工程1.0 (SE1.0)** 时代



SOFTWARE ENGINEERING (SE)

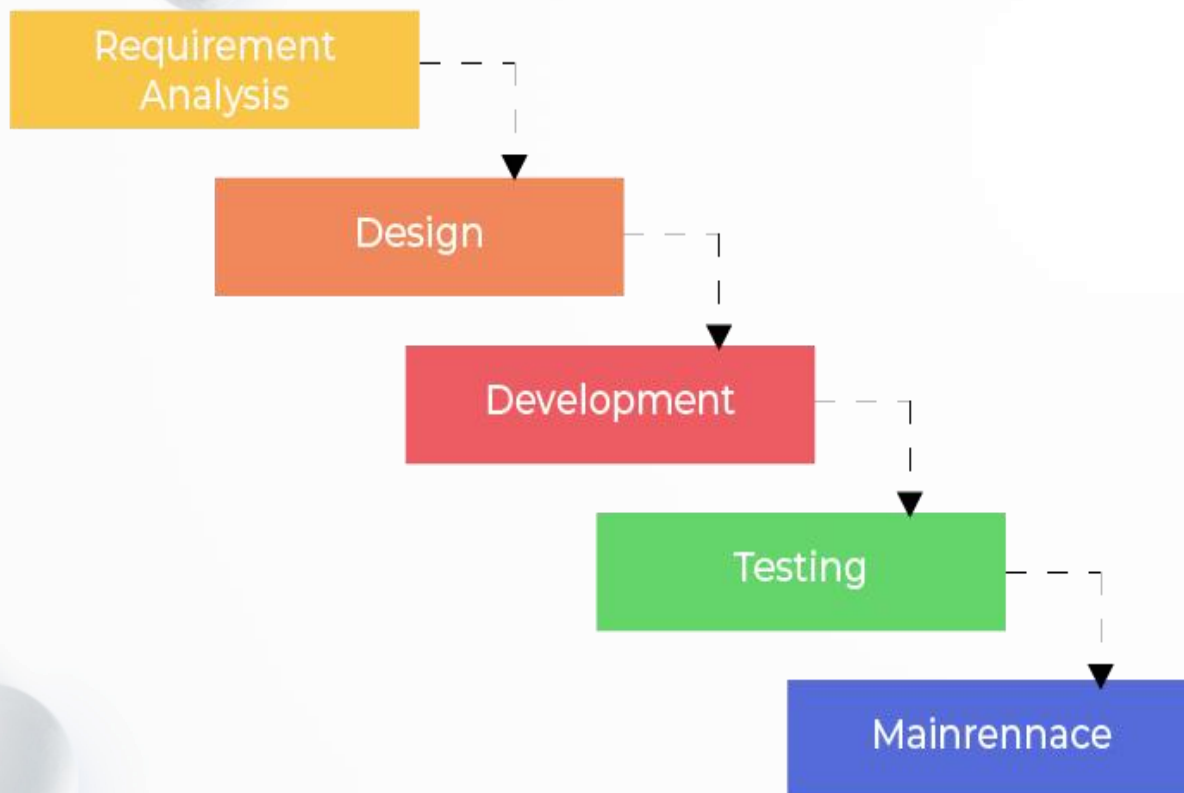
Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

会议输出136页的软件工程论述

SE1.0: 结构化、规范化、工程化



受传统建筑工程方法影响比较大，重文档、重流程，典型代表是CMM model: aims to improve existing software development processes

全球开源技术峰会

D123: DEMO OF CMMI V2.0 (latest version for Dev Model)

Maturity level 3 Sample Document Kit

Total editable documentation package for quick process improvement

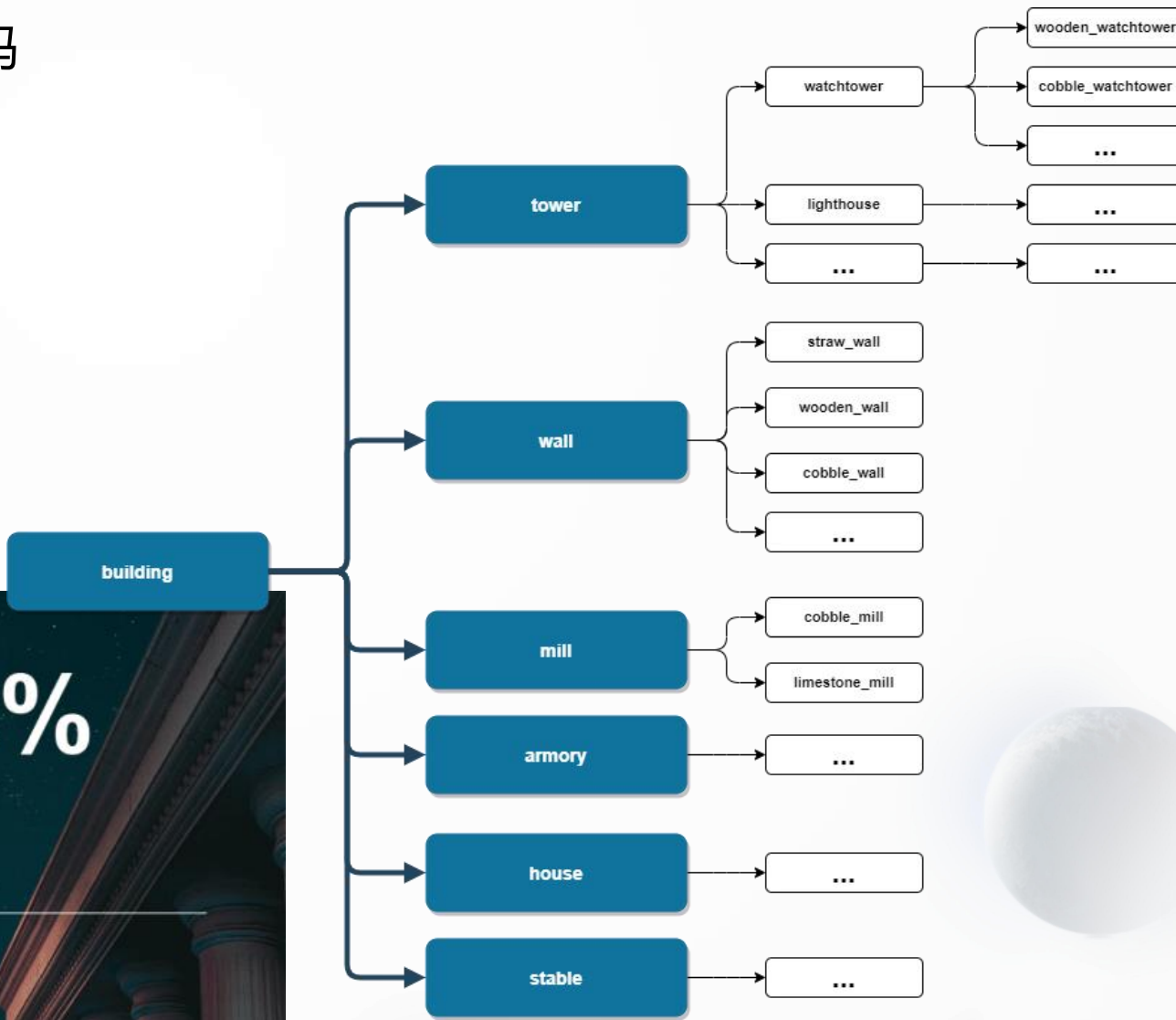
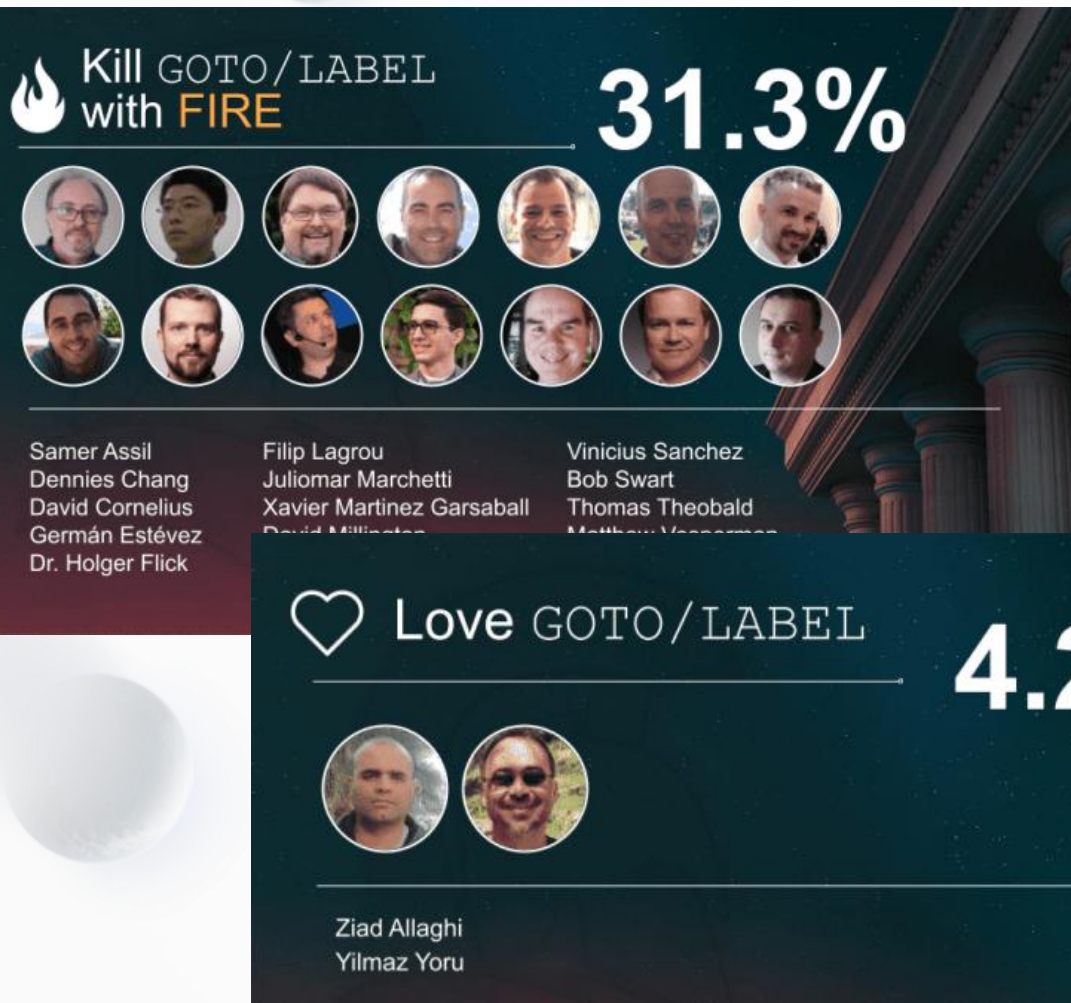
Complete editable document tool kit (Policy manual, process flow chart, procedures, blank forms, filled forms guidelines, coding standards, audit checklist, interview affirmation questions etc.)

List of Blank Formats

Sr. No.	Name of Formats	Sr. No.	Name of Formats
Business Development and Analysis (BA)			
1.	Functional Requirement Document	35.	Test Plan
2.	Requirements Sign-Off Sheet	36.	Bug Report
3.	Requirements Traceability Matrix	37.	Test Scenario Report
4.	RM Impact Analysis Checklist	38.	Review Report
Configuration			
5.	Configuration Items List	39.	Software Requirements Specification Template
6.	Baseline Register	40.	Design document
7.	Configuration Management Plan	41.	Peer Review Checklist
8.	CM process audit	42.	Code Review
Organization Process Group (OPG)			
9.	Organization Document List	43.	Integration Plan
10.	QMS Documentation Plan	44.	Minutes of meeting
11.	Organization committee members list	45.	Communication Plan
12.	Change Request	46.	Deployment Instructions
13.	PAL Authorization Request	47.	Installation manual
14.	Document Release Notice	48.	MOM Project Closure
15.	OPG Management Review	49.	User manual
OPG (System improvements and measurement)			
16.	Process improvement proposal form	50.	Release Note
17.	Lessons Learned Checklist	51.	Design review Checklist
18.	Improvement Tracker Register	Software (Planning)	
19.	Process Improvement Plan	52.	Project Plan
20.	Knowledge Transition	53.	WBS Excel
21.	Metrics Report	54.	Estimation Sheet
22.	Organization Measurement Plan	55.	Integration Plan
23.	Casual Analysis and Resolution Report	56.	DAR Report
Purchase (PUR)			
24.	Supplier List	57.	Project schedule
25.	Vendor/ Contractor evaluation & registration form	58.	Project Life Cycle standard process & Tailoring
26.	Supplier Agreement	59.	PMP Sign Off
27.	Supplier Management	60.	Issue log sheet
Quality Assurance (QA)			
28.	Project Name QA Schedule	61.	Risk Register
29.	Audit Findings report	62.	Project Status report
30.	Discrepancy Audit Finding Report sample2	Training (HR)	
Quality Control (QC)			
31.	Verification and Validation (V &V)	63.	Induction training
32.	Integration Test Cases	64.	Competency Report
33.	Unit Test Cases	65.	Job Description Template
34.	Use case Specification	66.	Skill matrix
		67.	Training TT Calendar Template
		68.	Training Feedback Form
		69.	Organizational Training Plan
		70.	Training Report
		71.	Training records
		72.	Training Feedback Form (End User)

从开发角度看SE1.0

创建可读、可维护、分层的代码

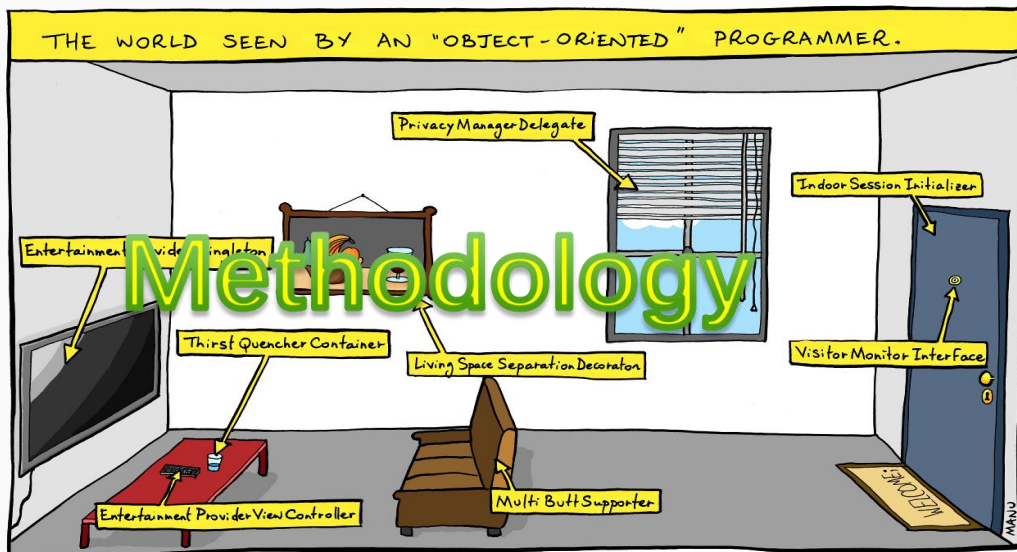


Structural Design Patterns in Python

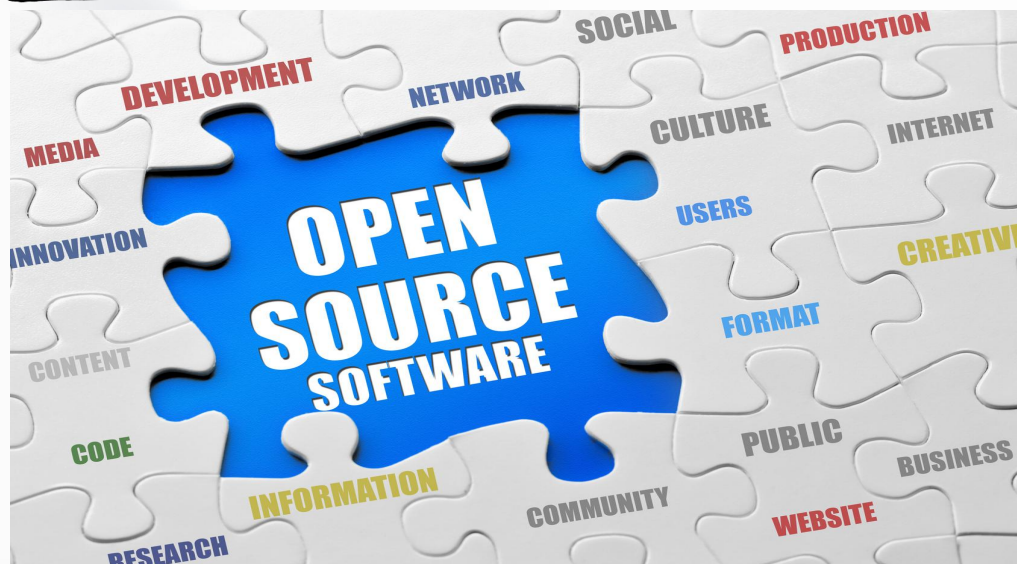
但软件其实不同于硬件

硬件研发与生产	软件研发
产品是一个物理对象	产品是数字、抽象的对象
希望批量生产	希望每个产品都是个性化的
构建/复制成本昂贵	构建/复制成本非常便宜
构建过程缓慢	构建过程非常快
不容易修改、扩展	容易修改、扩展
精确的质量度量	没有精确的质量度量
.....

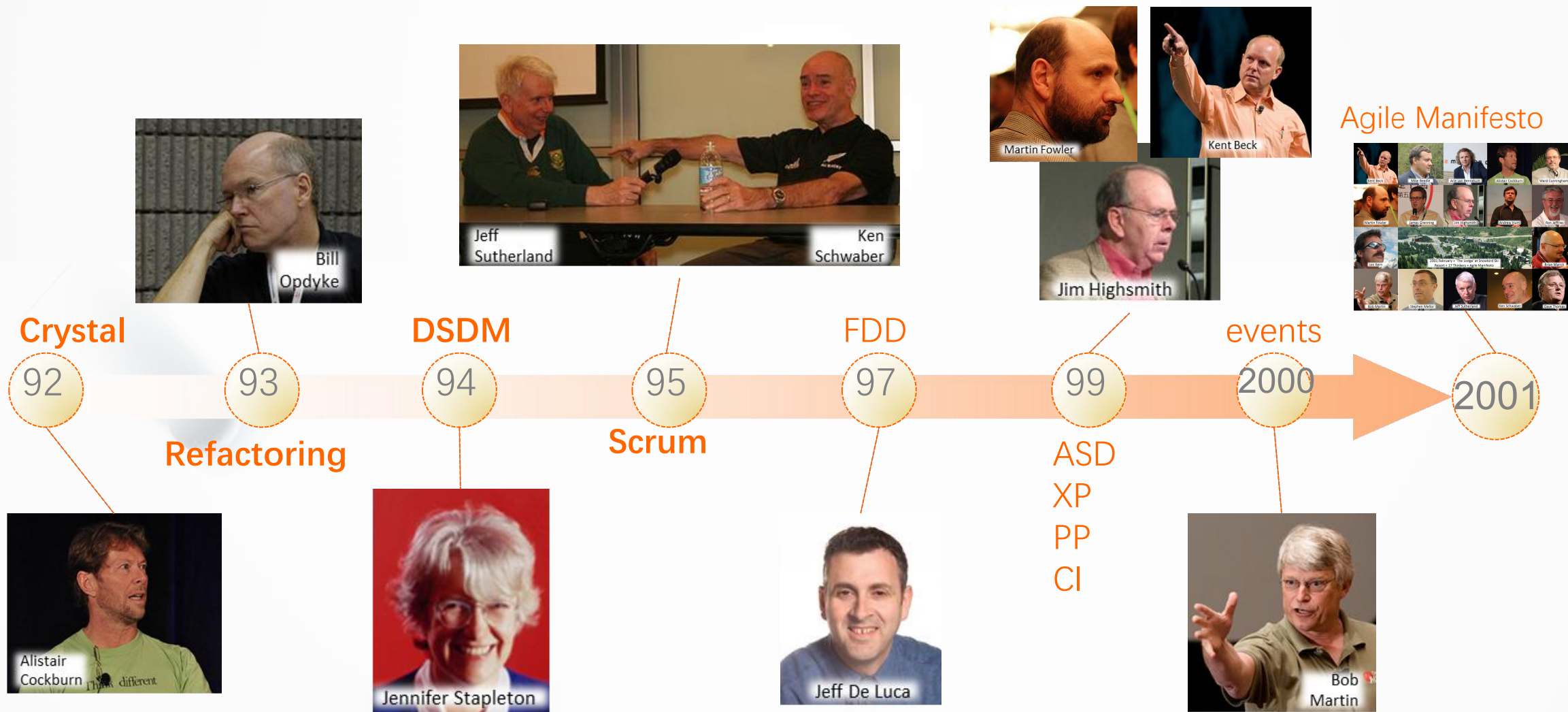
重新认识软件开发、交付



Methodology

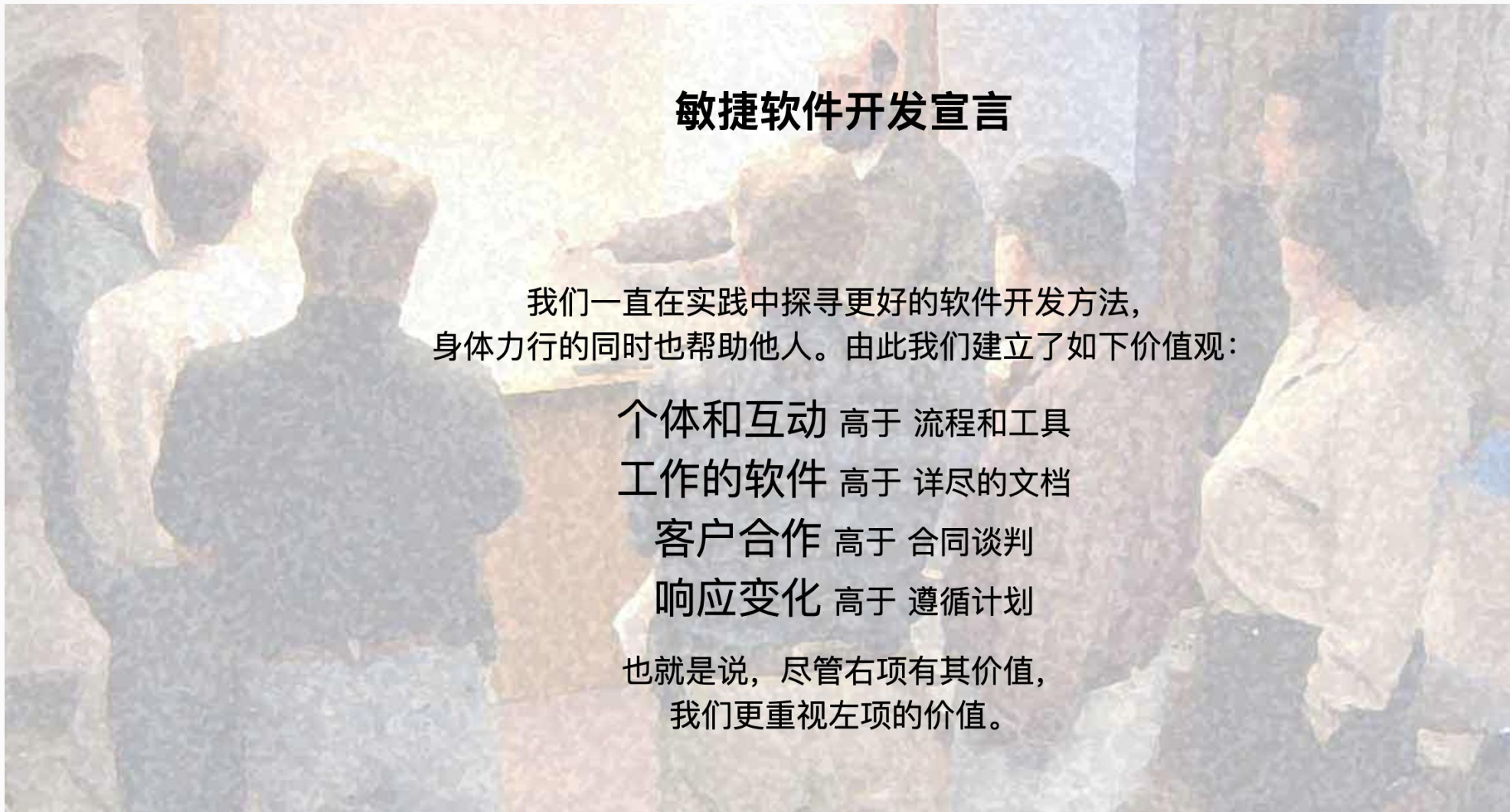


不断思考、探索与实践



DSDM: Dynamic Systems Development Method

PP: Pragmatic Programmer



敏捷软件开发宣言

我们一直在实践中探寻更好的软件开发方法，
身体力行的同时也帮助他人。由此我们建立了如下价值观：

个体和互动 高于 流程和工具

工作的软件 高于 详尽的文档

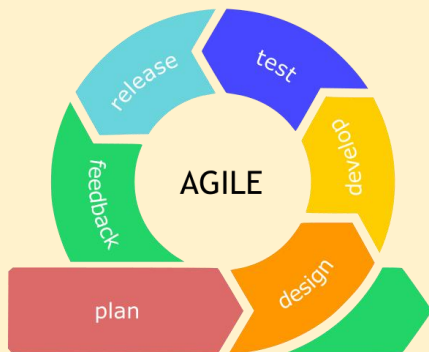
客户合作 高于 合同谈判

响应变化 高于 遵循计划

也就是说，尽管右项有其价值，
我们更重视左项的价值。

<http://agilemanifesto.org/iso/zhchs/manifesto.html>

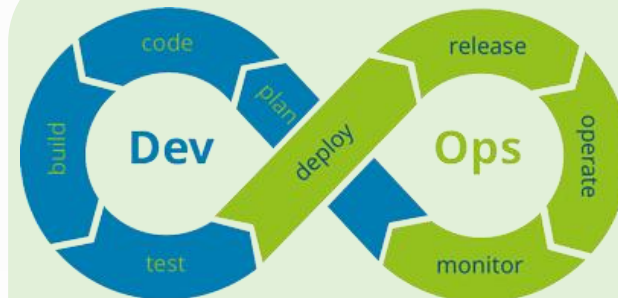
SE2.0开发模式：Agile、CI/CD、DevOps



- 团队协作
- 专注于软件本身
- 拥抱变化以加快产品交付



- 持续获得反馈而改进
- 专注于促进自动化的工具



- 文化是重点，研发和运维融合，形成闭环
- 云原生、平台工程等相继出现

敏捷是银弹吗？

- 效率真的提升了？“快”不代表高效率，多数团队返工更多了、工作更累了
- 口头上常提“价值交付、价值流”，交付的东西都有价值吗？
-

Not a Silver Bullet



Scrum is simple

...implementing Scrum is not



作为一个语言模型，GPT-4 在软件工程方面可能会有以下影响：

1. 自动代码生成：GPT-4 可以通过理解程序员的自然语言输入，自动地生成代码。这可能会加速开发过程，提高软件工程的效率。
2. 基于自然语言处理（NLP）的编码辅助：GPT-4 可能会提供基于自然语言处理的编码辅助，通过建议代码片段、函数和 API 来帮助开发人员。这可能有助于减少错误，提高代码质量，提高生产率。
3. 测试和调试：GPT-4 可以通过根据自然语言输入自动生成测试用例，并更快、更准确地识别 bug 的根本原因，从而改善测试和调试。
4. 设计和架构：GPT-4 可以通过提供基于自然语言输入的建议来协助软件工程师设计和架构软件系统。这可能有助于创建更强大、更高效的软件系统。

GPT-4 发布

带来了曙光

CG

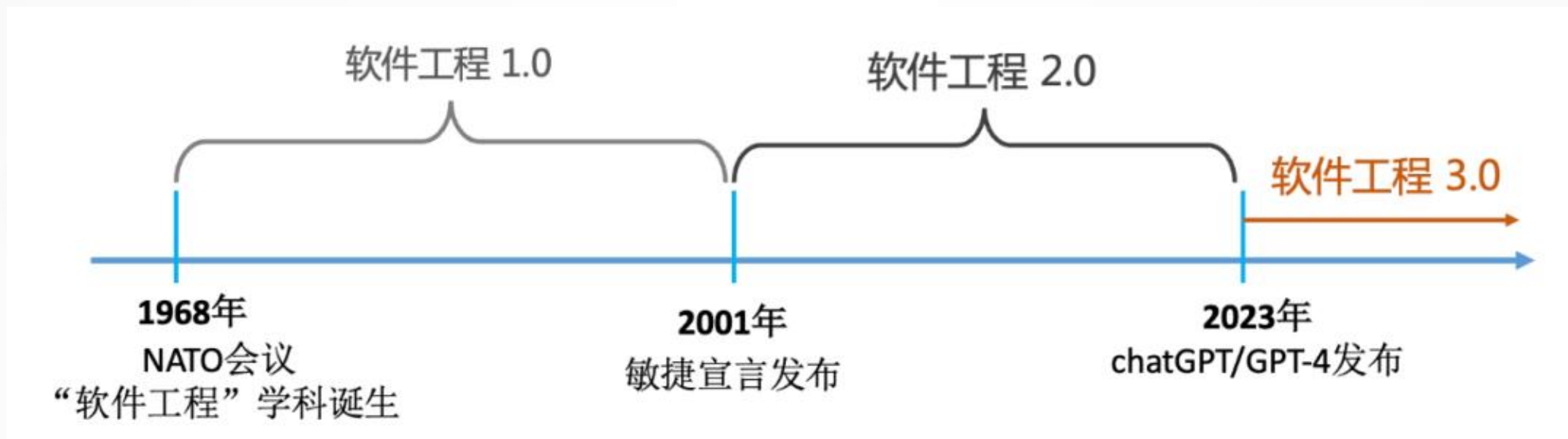
For a specific software requirement, can you generate all test cases?



As an AI language model, I can generate test cases based on natural language inputs, but I cannot guarantee that I can generate all possible test cases for a specific software requirement. Generating all possible test cases is often not feasible due to the large number of possible inputs and combinations that may need to be tested.

However, I can generate a set of test cases that cover a wide range of inputs and scenarios based on the natural language inputs that I receive. This can help to ensure that the software meets its requirements and that common use cases are covered.

GPT-4发布标志着软件工程3.0时代的到来



因为软件研发进入真正的数字化时代、开发范式也将发生巨大变化

软件工程3.0刚启动，需要未来几年的探索与实践，不断丰富与完善

人机交互智能 胜于 研发人员个体能力

业务和研发过程数据 胜于 流程和工具

可产生代码的模型 胜于 程序代码

提出好的问题 胜于 解决问题

右边各项有价值，只是左边更有价值

PART 2

软件工程3.0 新形态

软件研发平台开始理解现实世界、理解业务



之前各种智慧系统是人工“智能”



看似智慧，其实系统不能理解业务，完全是靠人工一点一点做出来的



今天，GPT-4发布意味着

GOTC

AI 第一次睁开双眼理解这个世界

计算机全面感知、认知这个世界

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

今天GPT能够理解我们所说的

GOTC

HP

产品需求通过用户故事描述，其中一个用户故事是这样描述的：
“作为京东电商App的一位购物者，我希望能删除购物车中的某些商品，这样当我改变主意时就可以不购买它了”
你能否帮忙写出其验收标准？



当然可以。根据您提供的用户故事，我们可以为这个功能编写以下验收标准：



1. 用户能够在京东电商App中找到并轻松访问“购物车”功能。
2. 在“购物车”页面，用户可以清晰地看到他们添加的所有商品。
3. 用户在每个商品旁边找到一个选项，允许他们删除选定的商品。
4. 在用户选择删除商品后，系统应弹出一个确认对话框，以避免误操作。
5. 用户确认删除后，选定的商品应从购物车中移除。
6. 被删除的商品不应影响购物车中的其他商品。
7. 用户在购物车中删除商品后，购物车中商品的总价应相应地更新。
8. 如果用户删除了购物车中的所有商品，购物车应显示为空，并提示用户添加商品。

这些验收标准旨在确保开发团队充分理解用户故事的需求，从而为京东电商App的购物者提供方便、灵活的购物车商品管理功能。

“正好
Transformer
把这么多知识
压缩在一起，
这是它的最大
突破”

LLM让软件技术成为企业系统的核心力量

从企业运行系统的支撑工具、慢慢融入其中到成为今天这样的形态



信息化



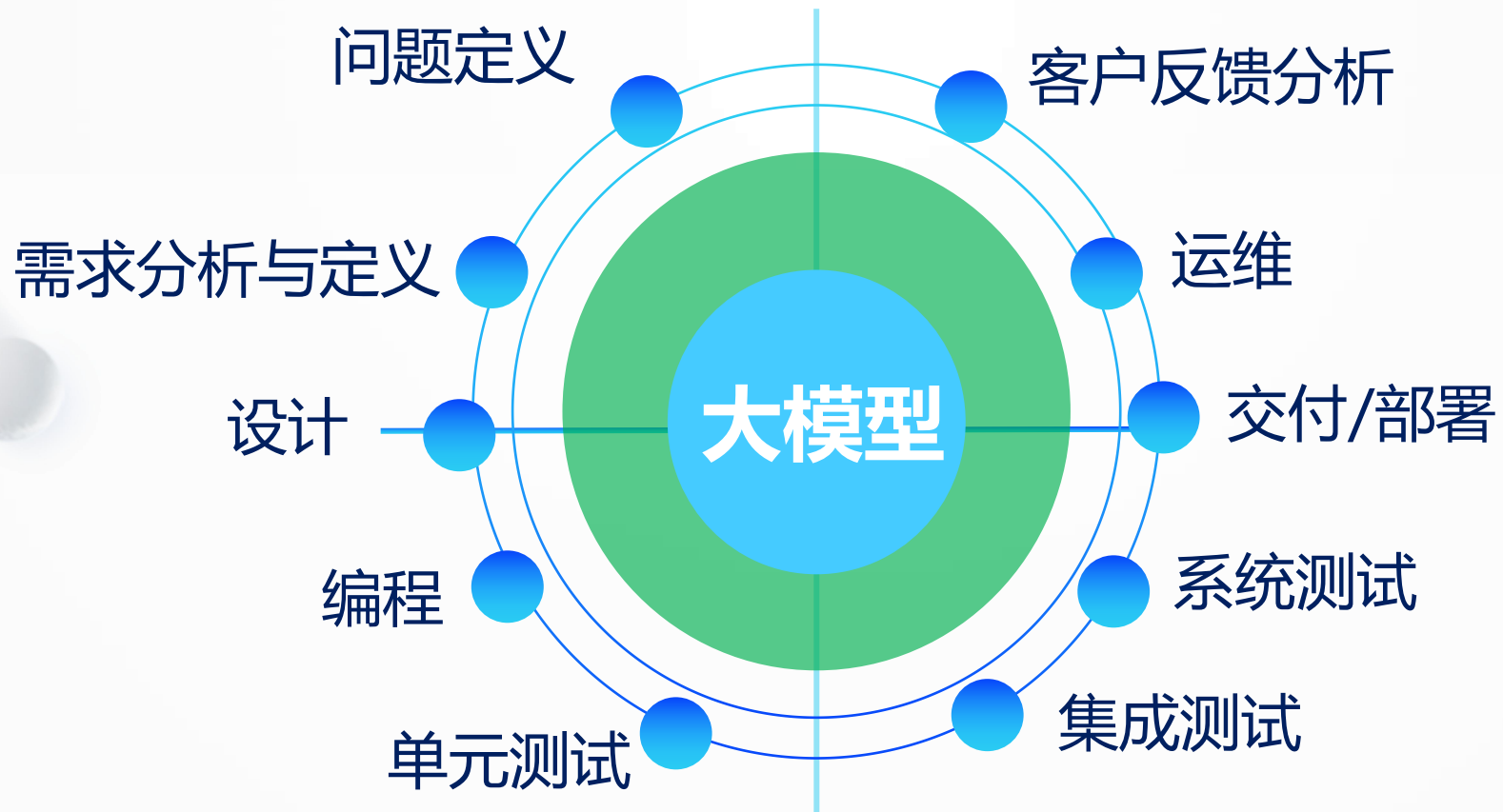
数字化初期



今天的数字化

软件工程3.0新形态：软件即模型

SaaM: Software as a Model



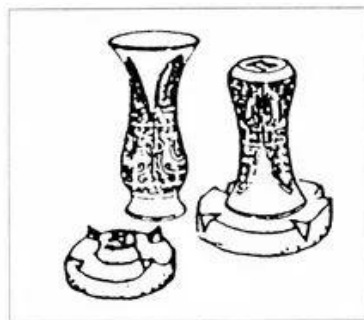
PART 3

软件工程3.0 之开发新范式

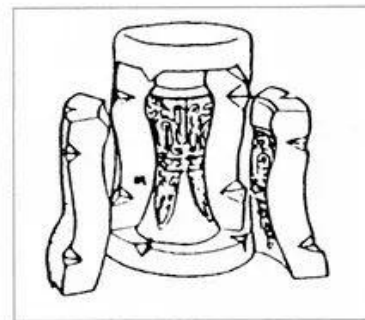
建立在SE2.0之上，但超越SE2.0



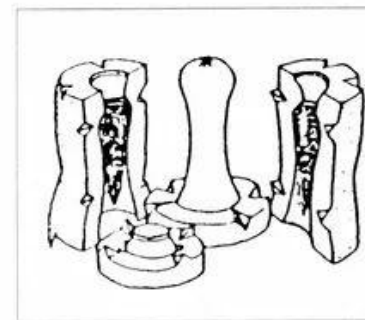
“模”与“范”



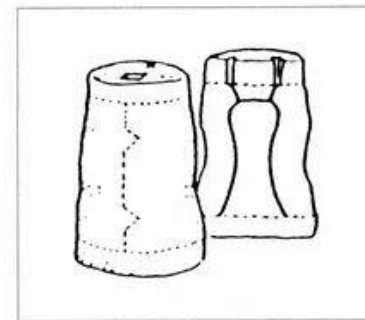
1. 以泥土制出器物之“模”



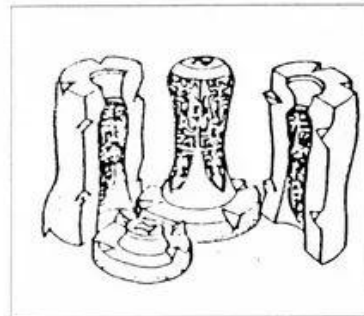
2. 根据“模”翻制出“外范”



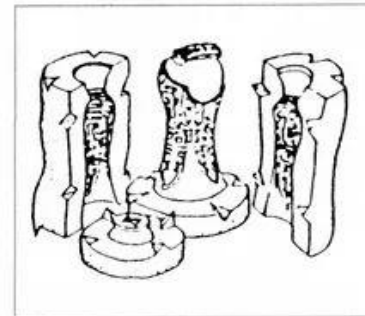
5. “内范”制成



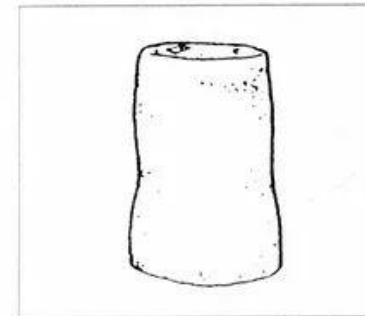
6. 将“范”阴干烤硬后装合



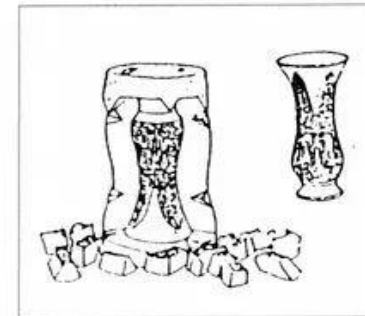
3. 将“外范”切分后除下



4. 削刮“模”制作泥芯



7. 将“范”预热后浇铸铜液



8. 铜液冷却后碎“范”取器

模是事物的大致模样，虽然**范**不是事物本身，但是它决定了事物的模样

基于工艺 (craft) 结构的开发范式 (paradigms)、基于市场的、
基于产品的开发范式; 开源软件范式、分布式开发范式、群智范式

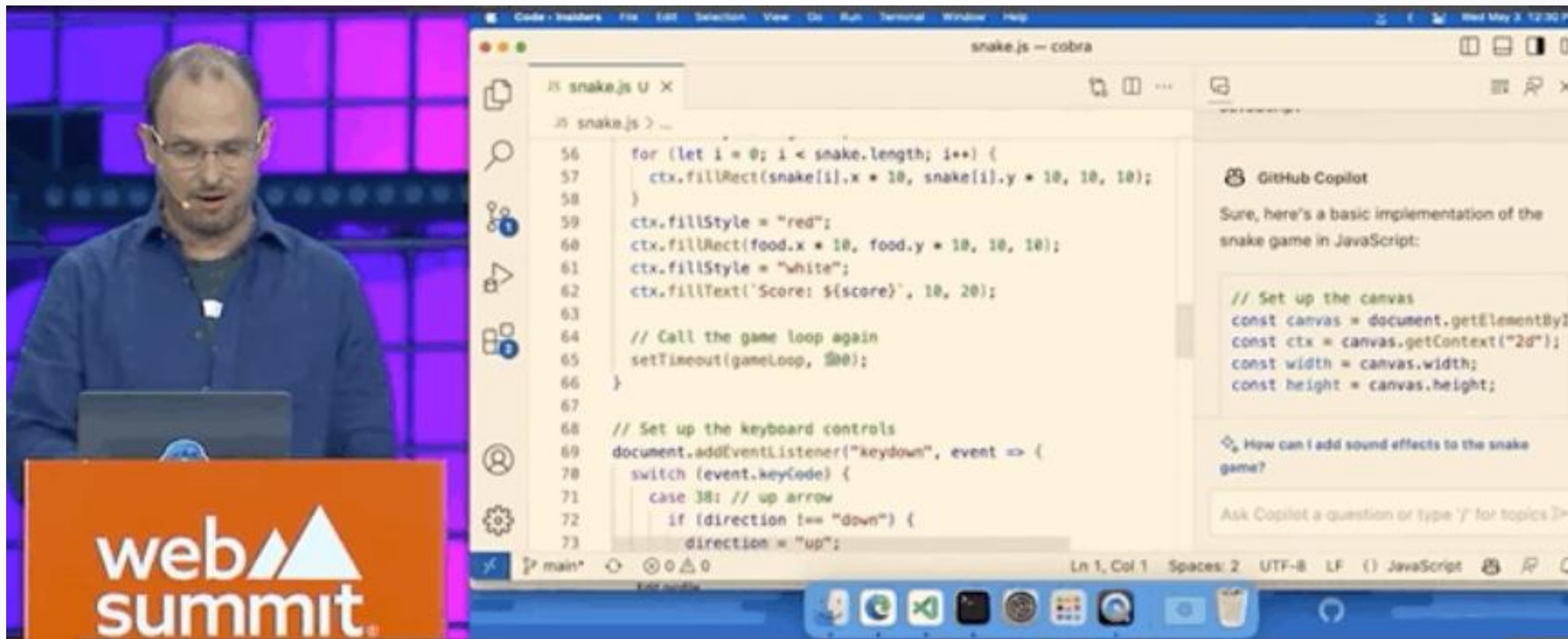


The transition between competing paradigms cannot be made a step at a time, forced by logic and neutral experience. Like the gestalt switch, it must occur all at once (though not necessarily in an instant) or not at all.

— Thomas Kuhn —

软件开发范式的变化快，常常伴随着标志性的事件

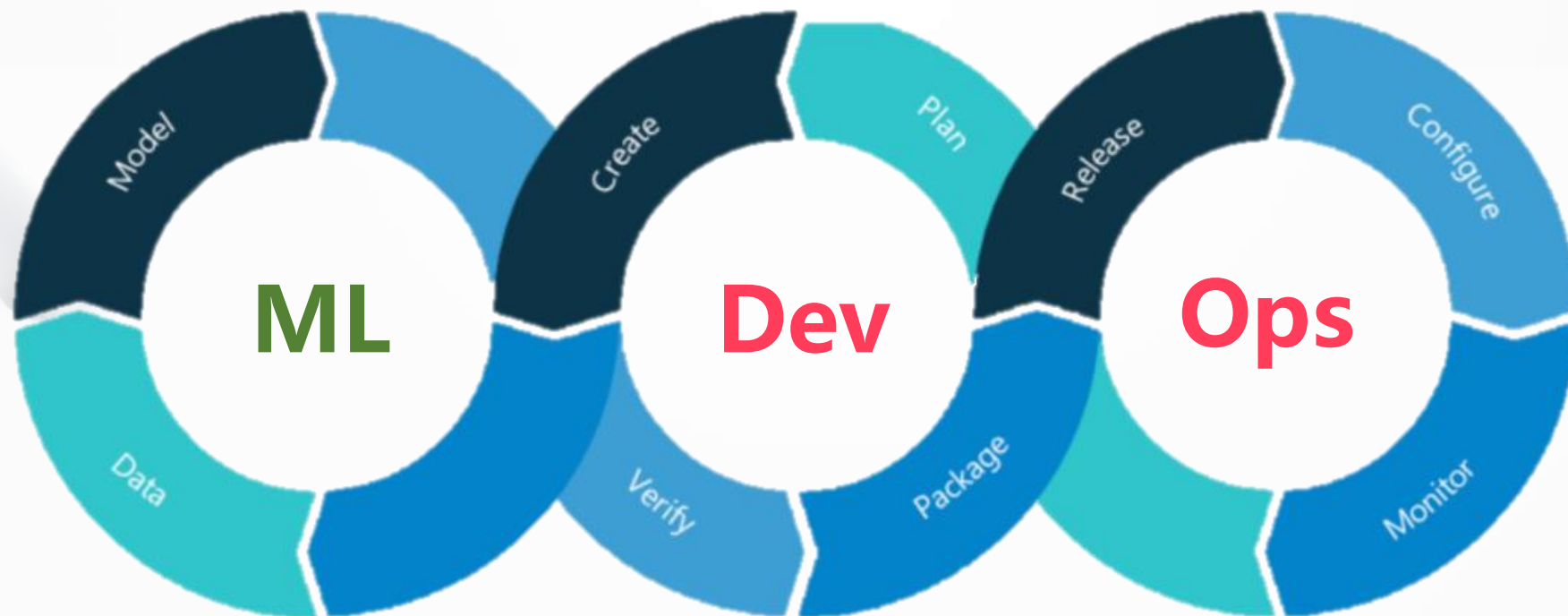
今天大模型能生成UI、代码、测试脚本等



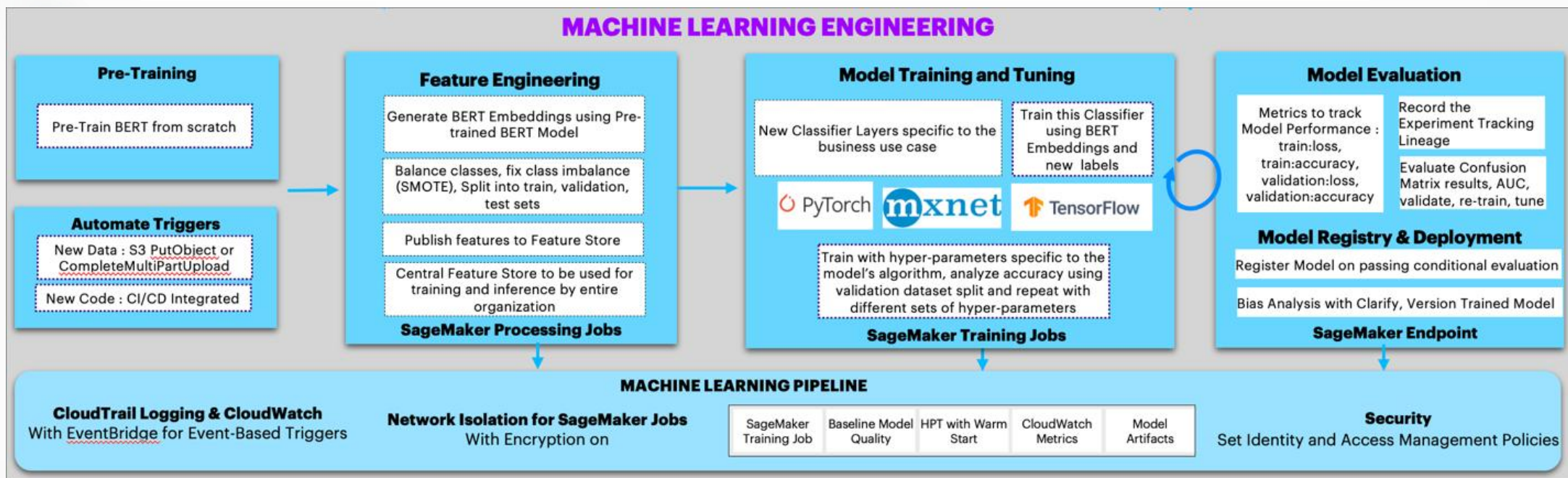
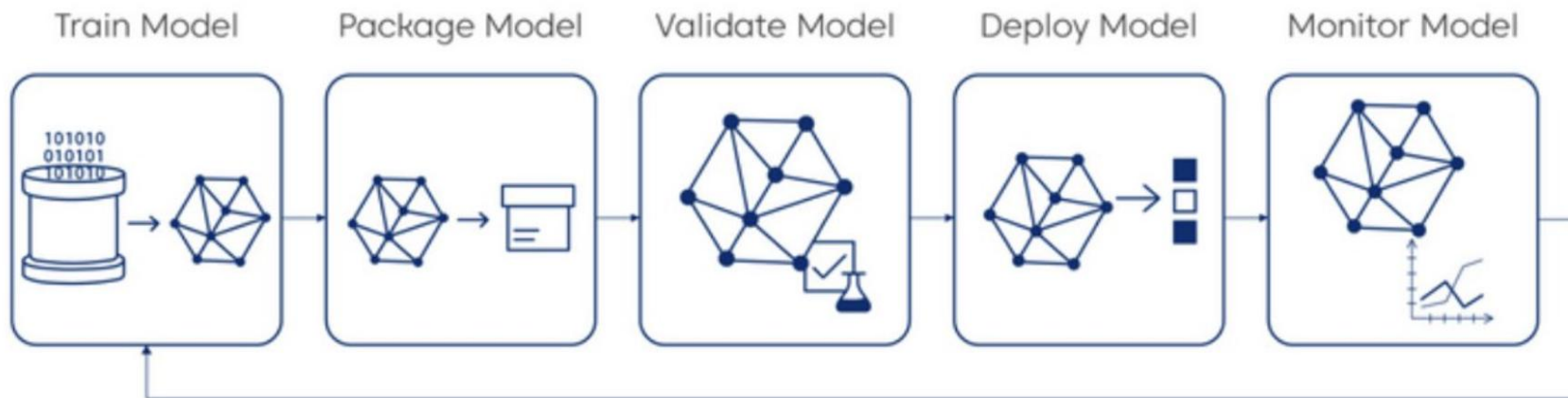
GitHub CEO Thomas Dohmke 挑战：用AI18分钟完成一个小游戏的编程。结果只用了不到15分钟成功上线部署，这边他还在讲台上，全球网友已经能玩上了。

软件开发新范式：ML-DevOps

模型驱动开发、模型驱动运维

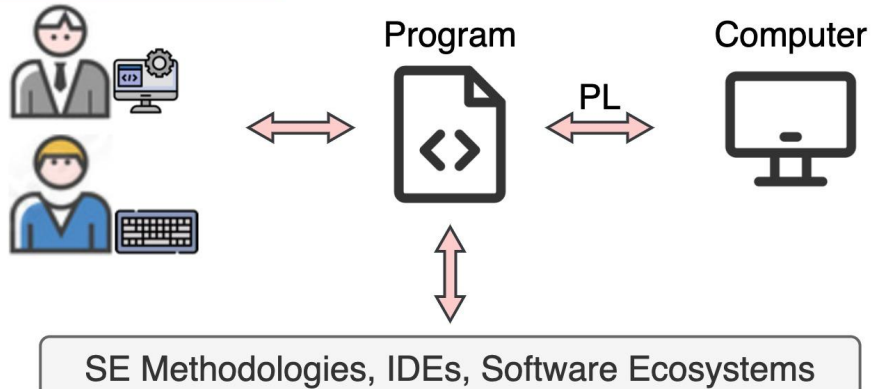


新范式：一切围绕着“数据、模型、提示”开展工作



案例：AI Chain

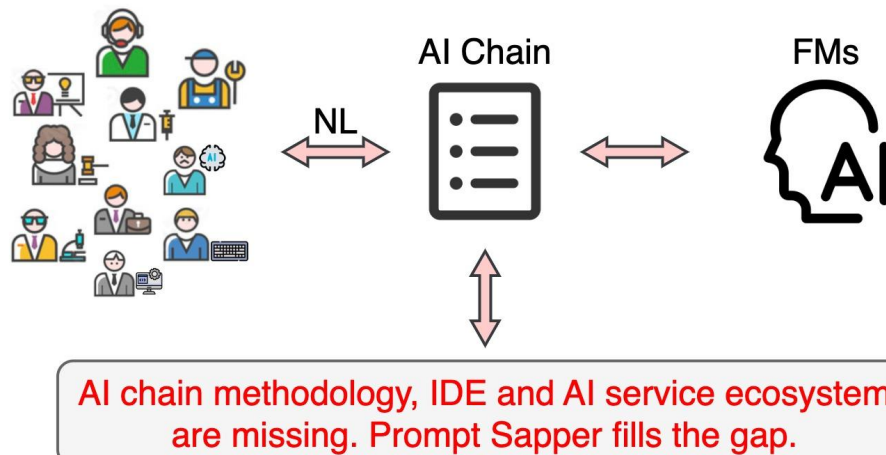
Software 1.0/2.0



Human-Computer Interaction in Software 1.0/2.0

- Human is forced to interact with computer
 - in some computer language
 - focus on "how to do it"
- For only those
 - with programming and AI training

Software 3.0



Human-AI Interaction in Software 3.0

- Human interacts with AI
 - in natural language
 - focus on "what to do"
- For everyone
 - who understands task instructions, examples and workflows
 - who can express their intentions in (vague) natural language

PART 4

新范式下编程将如何展开？

建立在SE2.0之上，但超越SE2.0



垂直路径

- 针对特定领域，进行模型的训练或精调，并给出具体的提示文本（序列）集，构建领域LLM。
- 能帮助企业快速地发布符合市场的产品、获取用户、收集反馈并持续改进

水平路径

- 将生成式AI功能引入到通用的开发云平台之中，实现全新的或重构已有的开发 workflow、开发IDE、测试框架与工具、运维工具。
- 重构项目管理、需求管理、代码管理、测试管理等

- ① **AI+人类共同创造工具**：通过文本、语音或图像提示进行资产（代码、UI、文档等）生成的共同创作工具
- ② **提示工程**：AI可以帮助开发者进行语义搜索，帮助开发者找到正确的提示，或在团队之间分享成功的提示脚本.....
- ③ **创建新颖的功能、机制**：有了生成式AI，开发者可以让GPT生成创新的功能，并通过一些竞争机制提升代码或测试的质量。
- ④ **大企业**构建自己的大模型，**中小企业**通过开放API调用，使用大模型服务商（类似今天的云服务商）提供的服务。

Coding Buddy (AIGC) : 生成代码、代码补全

如GitHub Copilot in VS Code, 基本不改变原有编程方式, 但效率会提高一倍

```
parse_expenses.py write_sql.go sentiment.ts addresses.rb
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, va
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

```
Tabnine > Python > s3_bucket.py > ...
import boto3

bucket_name = 'experiments'
file_name = 'results.json'
region = 'us-east-1'

# read file from s3 bucket

# upload a file to s3 bucket
```

<https://www.tabnine.com/>

// q: What is a class in object-oriented programming?

测试更简单 // test calculator function

```
Files | index.js x
index.js
1 // @ts-nocheck
2 const fs = require("fs");
3 const https = require("https");
4
5 function saveImageToDisk(url, localPath)
6   const file = fs.createWriteStream(localPath);
7   const request = https.get(url, function(response) {
8     response.pipe(file);
9   });
10 }
11
12 function colorString(fill) {
13   return `rgba(${Math.round(fill.color.r * 255)},
14     ${Math.round(fill.color.g * 255)}, ${Math.round(
15     fill.color.b * 255
16   )}, ${fill.opacity ? fill.opacity : fill.color.alpha})`;
17 }
18 function dropShadow(effect) {
19   return `${effect.offset.x}px ${effect.offset.y}
20   px ${effect.radius}px ${colorString(effect)}`;
21 }
```

```
function tempConvert(temp, scale) { // Vague function name
  var result = 0; // 'var' is so es2015
  if (scale == "C") { // Is this the correct way to compare strings?
    result = (temp - 32) * (5 / 9); // Magic numbers
  } else {
    result = (temp * 9 / 5) + 32;
  }
  return result;
}

/**
 * Converts a temperature between Celsius and Fahrenheit.
 * @param {number} temp - The temperature value to convert.
 * @param {string} scale - The temperature scale ('C' or 'F') to convert to.
 * @return {string} The converted temperature value, formatted with 1 decimal place.
 */
const tempConvert = (temp, scale) => {
  const isValidScale = scale === "C" || scale === "F";
  if (!isValidScale) {
    throw new Error("Invalid scale: must be 'C' or 'F'.");
  }

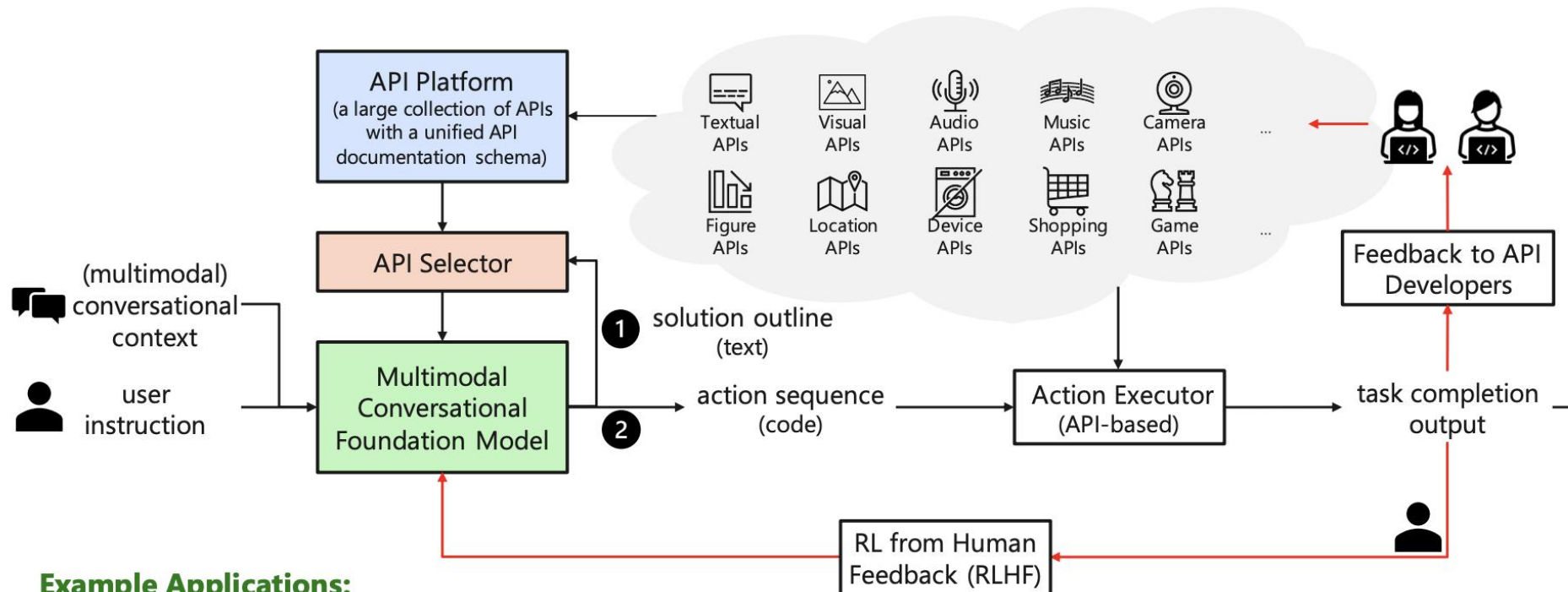
  const conversionFactor = 5 / 9; // <-- Note: This is not correct for Fahrenheit
  const result = scale === "C"
    ? (temp - 32) * conversionFactor
    : (temp * conversionFactor) + 32;

  const resultUnit = scale === "C" ? "F" : "C";
  return `${result.toFixed(1)}°${resultUnit}`;
}
```

LLM和其它AI模型集成: TaskMatrix.AI



借助API平台, 和基础设施、现有模型的无缝集成, 输入多样性、结果更具解释性, 系统能力易扩展



Example Applications:



AI for Robotics



AI for Office Automation



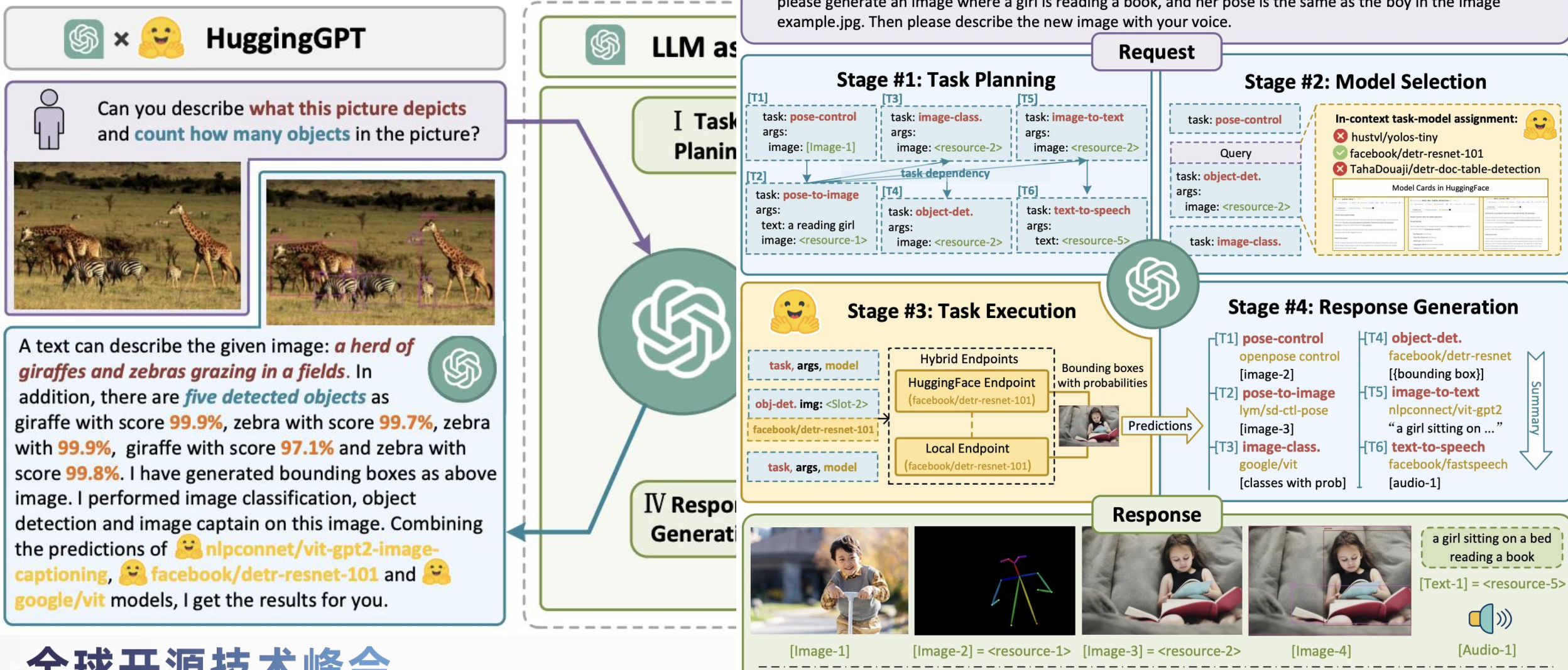
AI of Things (AIoT)



AI Copilots

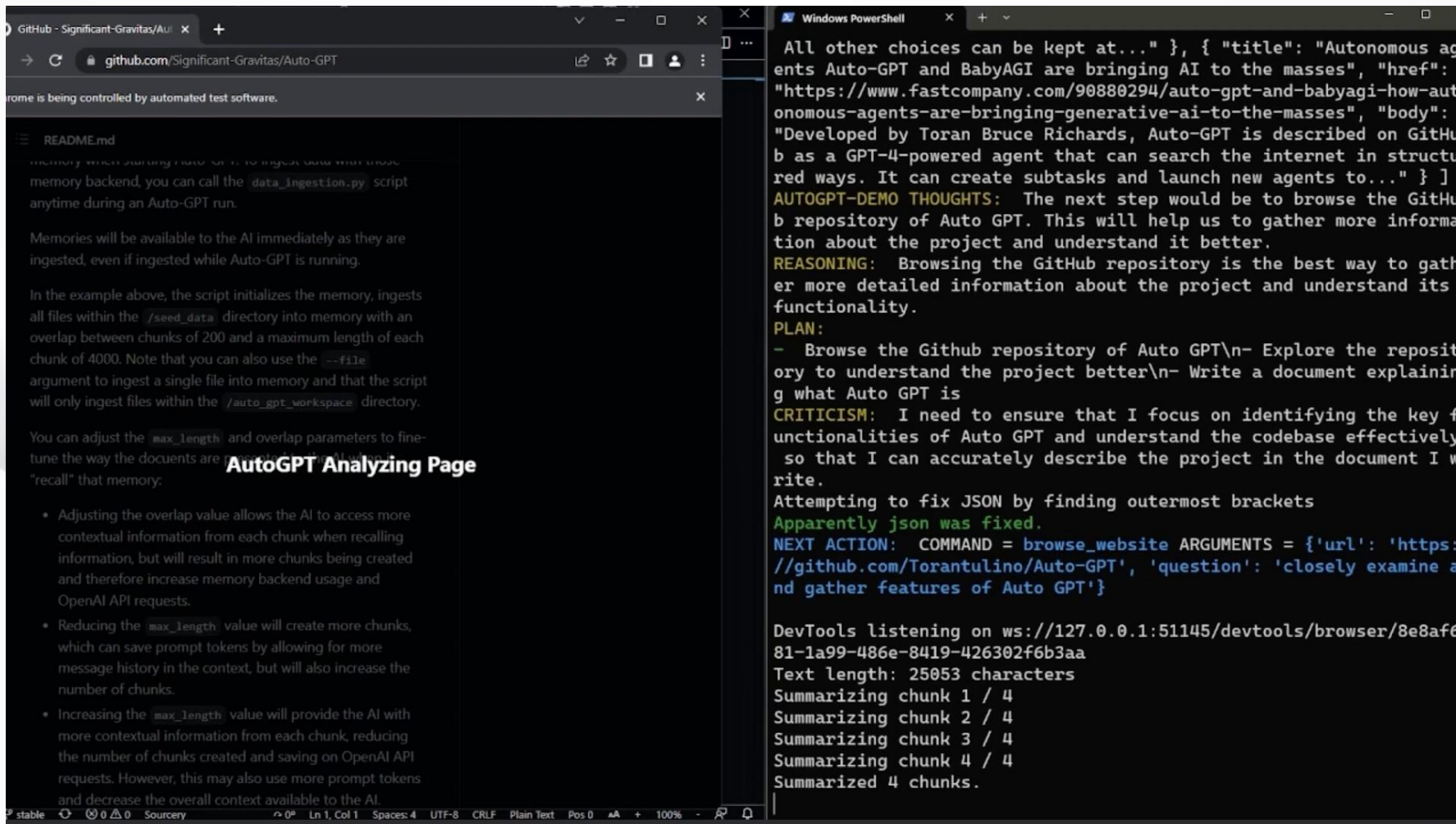
管理LLM模型之间的合作: HuggingGPT

LLM作为一个控制器, 连接众多AI模型, 管理和组织这些模型, 如计划、选择模型、任务分配等



LLM驱动任务目标实现：Auto-GPT

实验性的开源应用程序, 由GPT-4驱动, 将LLM “思想” 链接在一起, 以自主实现我们设定的任何目标



PART 5

企业如何更好利用新范式？

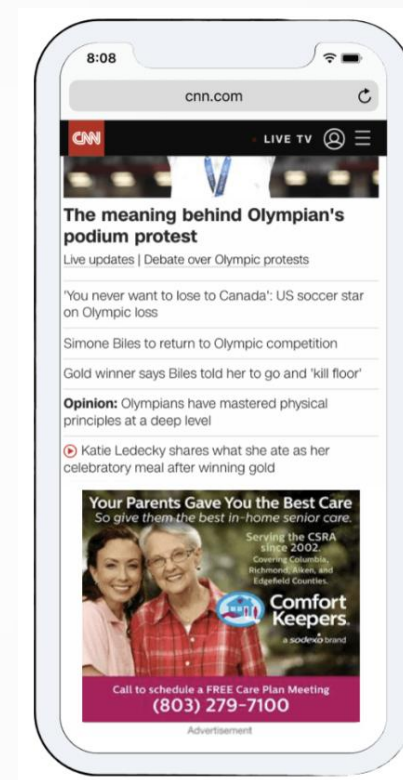
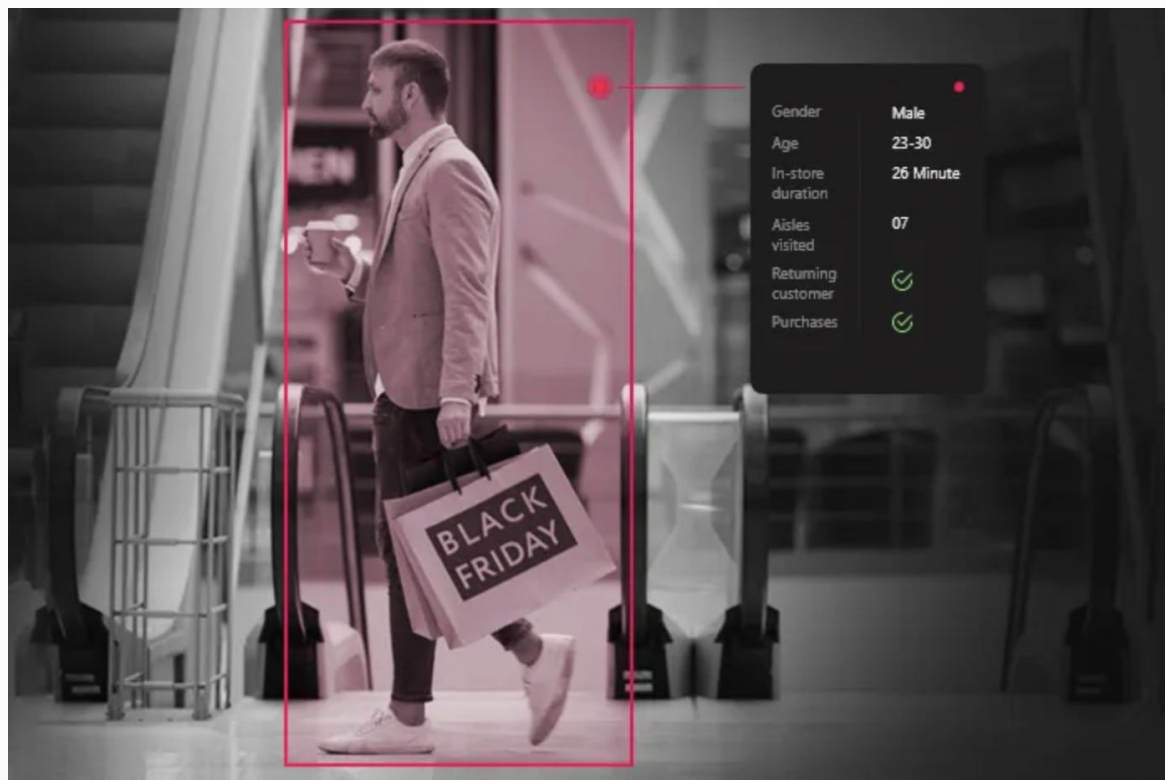
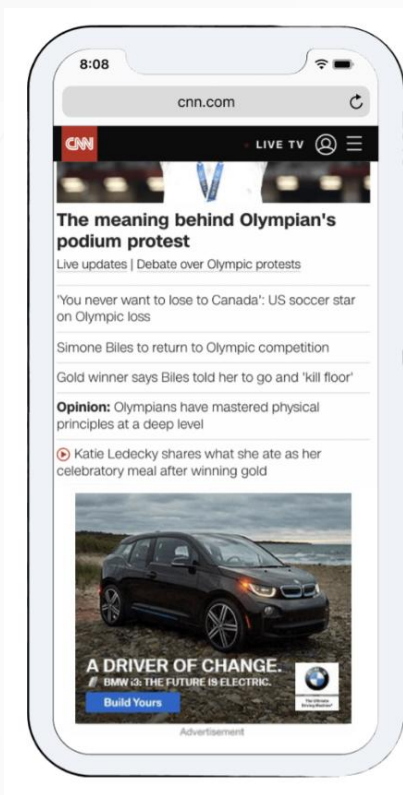
拥抱新范式，和大模型一起成长



定义正确的问题



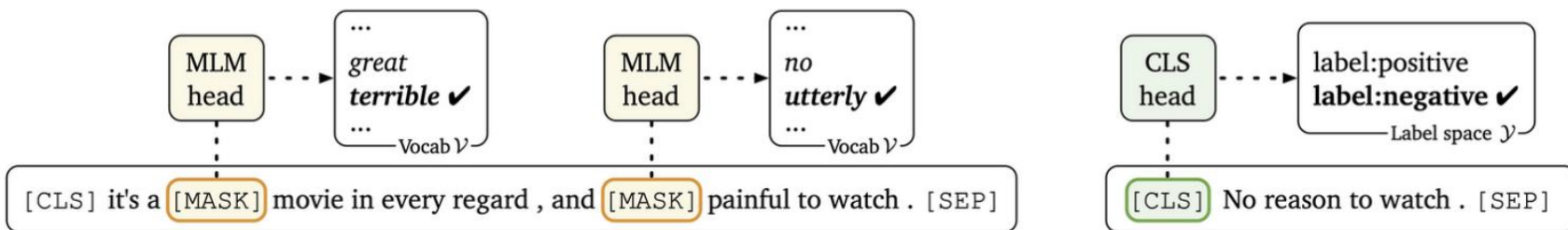
定义问题由产品经理做决定，但AI能在客户画像、挖掘用户痛点、挖掘应用场景上提供有力的帮助



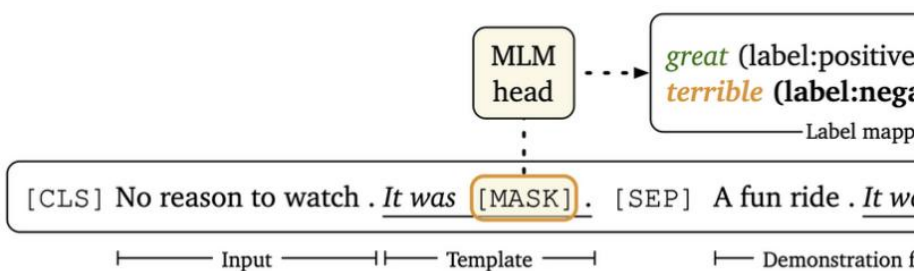
善于提问：逻辑思维、批判性思维

从项目开始到设计、编程、测试等过程，都依赖人机交互完成，善于提问，循序渐进，不断引导大模型、及时纠错、优化开发过程

Why Prompts?



(a) MLM pre-training



(c) Prompt-based fine-tuning with

```
prompt = """The following is a conversation with an AI assistant.
The assistant is typically sarcastic and witty, producing creative
and funny responses to the users questions. Here are some examples:
```

```
User: What is the meaning of life?
```

```
AI: """
```

```
openai.temperature = 1.0 # increase creativity/randomness of output
```

```
print(openai(prompt))
```


- Standard Prompting
- Zero-Shot Prompting
- Few-shot Prompting
- Chain-of-Thought Prompting
- Zero-Shot CoT
- Self-Consistency
-

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

提示工程: Synergizing Reasoning + Acting

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch

(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control ...

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ...

Act 3: Search[Front Row (software)]

Obs 3: Front Row is a discontinued media center software ...

Act 4: Finish[yes]

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: Search[Apple Remote]

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: Search[Front Row]

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software) .

Act 3: Search[Front Row (software)]

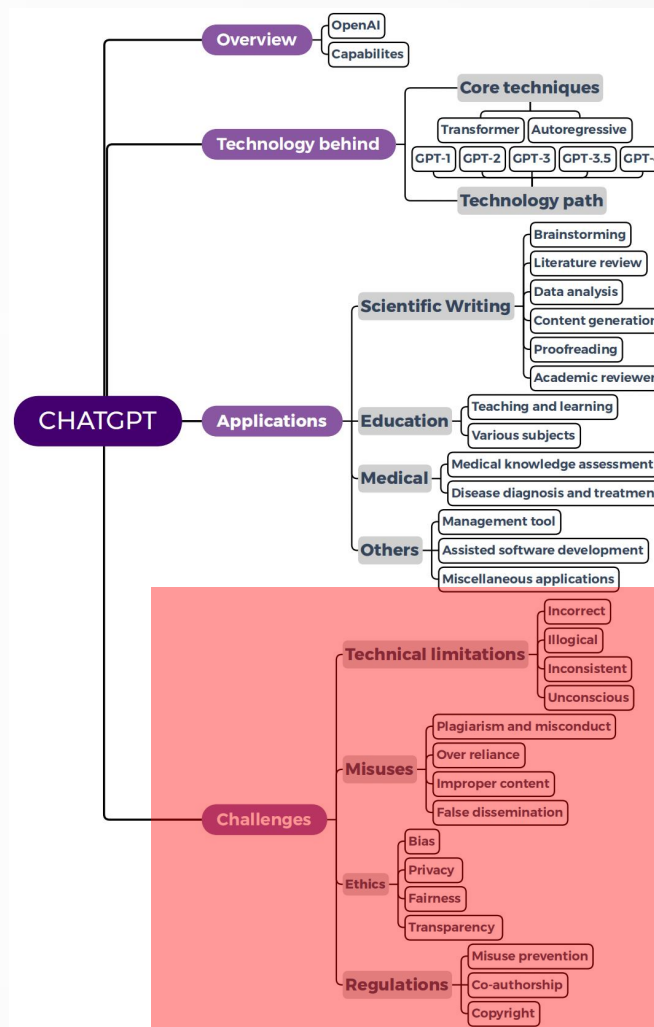
Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: Finish[keyboard function keys]

结果验证依旧重要

- ① **AGI**是基于概率做出决策，具有较大的随机性，甚至会“编造”事实，并进行错误的推理
- ② 各项业务有一定的特殊性，更大的问题是其复杂性
- ③ 软件系统的复杂性也是前所未有的，软件工程主要目的之一就是解决其复杂性
- ④



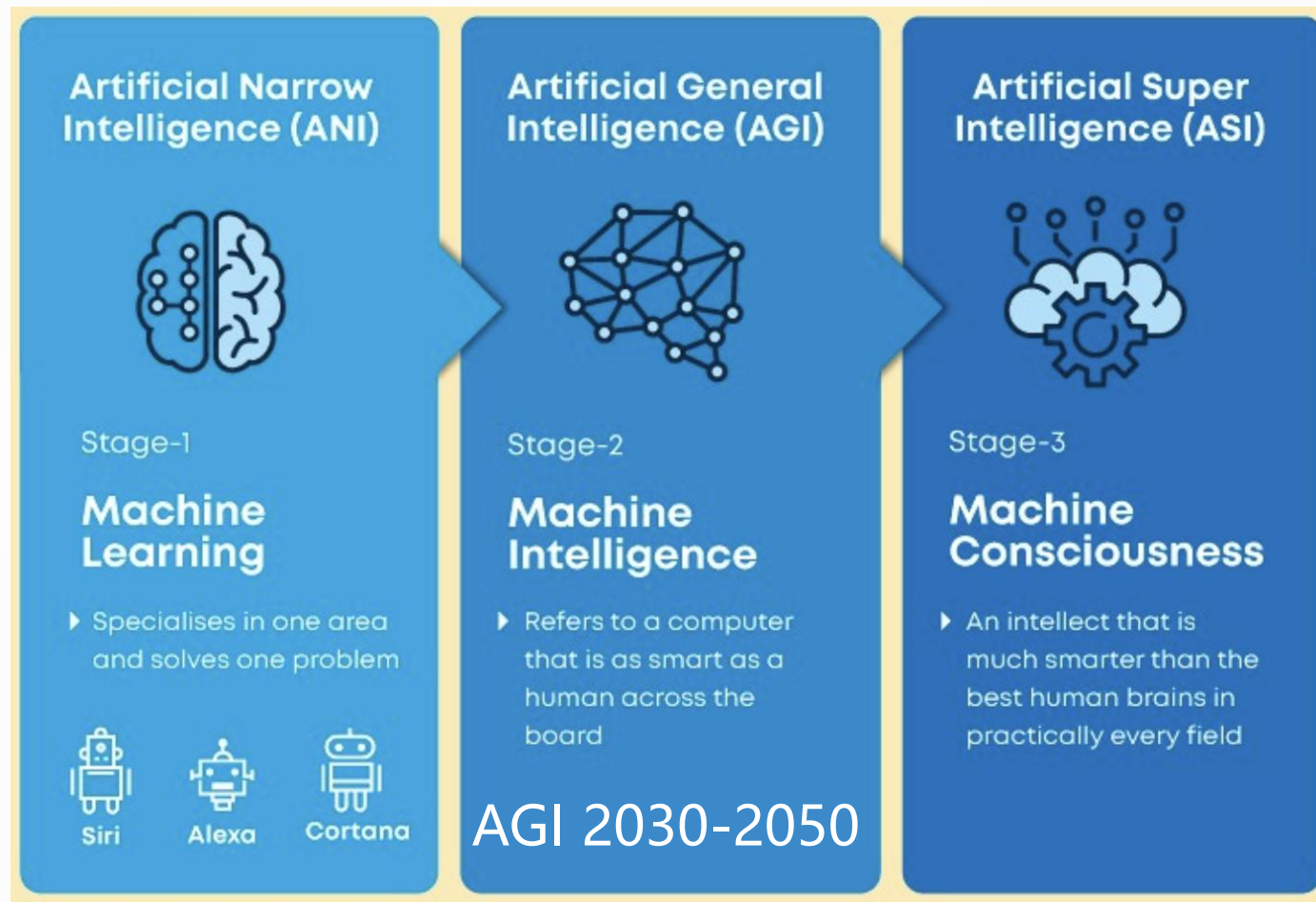
PART 6

未来机遇与挑战

更多的是惊喜



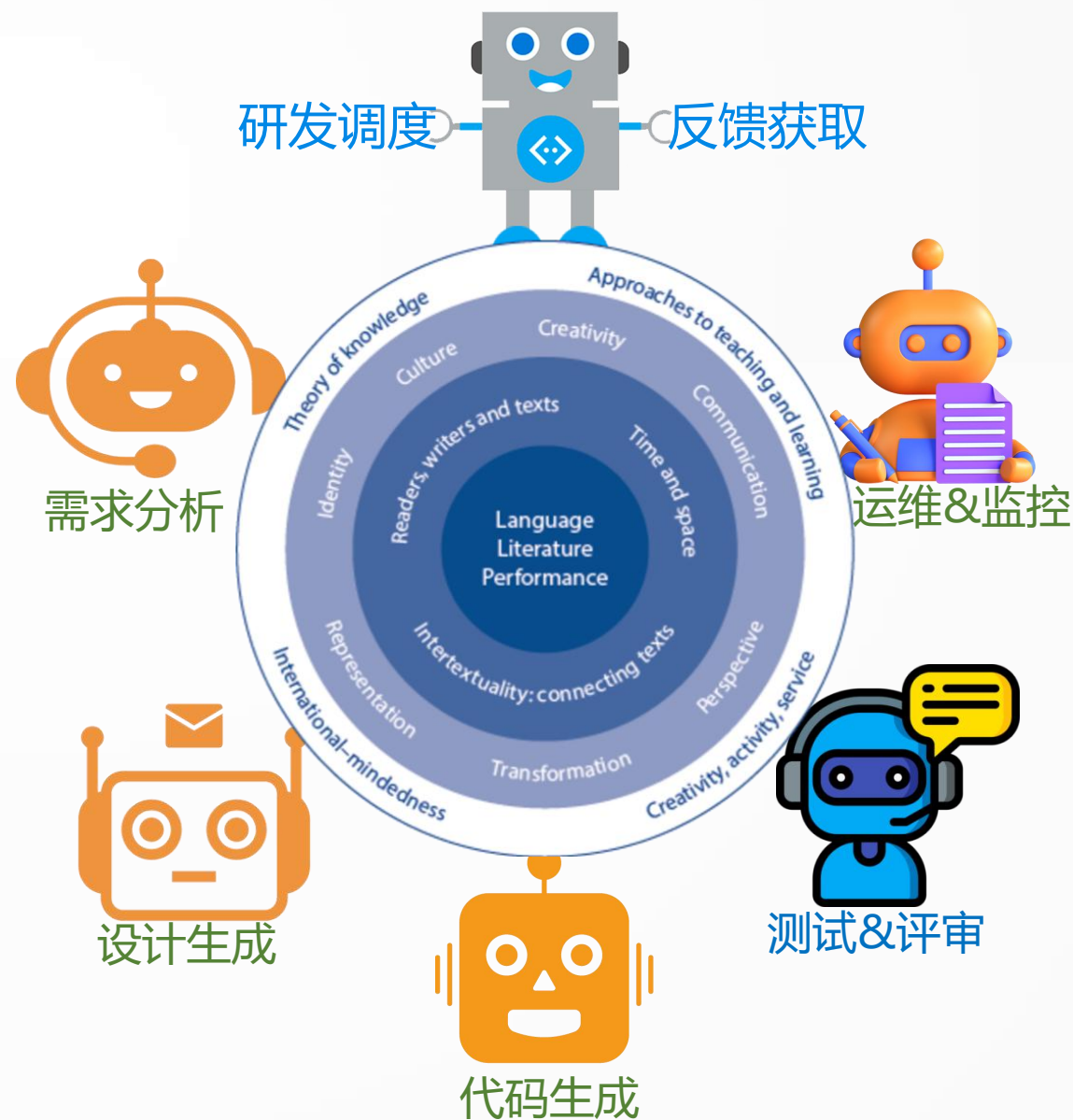
- 自动推理
- 知识表示
- 自动规划能力
- 学习能力
- 多模态能力
- 泛化能力
- 更多的涌现



Ray Kurzweil的著作《奇点临近:当人类超越生物学》：一旦奇点到来，机器智能将比所有人类智能的总和强大得多。

展望：SE3.0的机遇

- 持续交付得以真正实现
- 极大地提高研发效率
- 提升用户体验和满意度
- 个性化与定制化
- 对非技术人员更友好
- 创新领域拓宽



- 大规模复杂性的处理
- 多人多机协调
- 可解释性与可信赖性
- 知识与技能更新
- 数据安全性与隐私保护
- 法律与道德问题



THANKS

