



武汉大学 人工智能与软件工程暑期学校

2020-08

演讲人：李必信

1969年8月出生，2001获得南京大学博士学位，2001-2003分别在芬兰、挪威和荷兰等地从事博士后研究工作。东南大学计算机科学与工程学院教授（二级）、博士生导师，东南大学软件工程研究所所长。2006年入选教育部新世纪人才培养计划、2011年度获得中创软件人才奖、2014年获得教育部科学技术奖（自然科学奖）二等奖（排名第一）。江苏省计算机学会软件专委会副主任，第二届江苏省软件工程标准化技术委员会主任委员，中国计算机学会软件工程专委会委员、容错计算机专委会常务委员。研究方向为软件建模、分析、测试与验证、智能软件架构理论和方法、软件演化和软件质量保证等。他从1999年开始，主持各类基金和企业合作项目30多项，先后在《IEEE Transactions on Network and Service Management》、《ACM Computing Surveys》、《Information and Software Technology》、《Software Testing, Verification and Reliability》、《Journal of Systems and Software》、《Science China: Information Science》、《Journal of Computer Science and Technology》、《软件学报》、《计算机学报》、《计算机研究与发展》国内外著名期刊和ICSE、FSE、ASE等重要国际会议发表学术论文180余篇，出版专著3部，教材1部，授权发明专利40多项。相关论文被SCI/EI检索200多篇次，被他人引用累计超过5000篇次。





武汉大学 人工智能与软件工程暑期学校

2020-08

人工智能时代的软件工程

李必信 博士、教授、博导

东南大学计算机科学与工程学院
东南大学软件工程研究所

2020年08月04日 武汉/南京



目录

- 一. 什么是**软件及软件工程**?
- 二. **AI时代的软件工程理论方法**
- 三. **AI时代的软件工程技术手段**
- 四. **AI时代的软件工程产品类型**
- 五. **AI时代的软件工程市场空间**

一、什么是软件及软件工程？

• 1.1 什么是软件？

- 软件 (Software) 是一系列按照特定顺序组织的计算机指令和数据的集合。
- 简单来说：软件=程序+数据+文档

程序：计算机指令/语言代码【加工厂】

数据：程序的加工处理对象【原材料】

文档：记录怎么开发、测试、使用、运维、注释等位置和图表

• 1.2 软件的三种类型

• 编程语言

- C、C++、Java、Python、C#、Visual Basic.NET、PHP、Javascript、SQL、R、Ruby

} 用于编写其他软件的软件

• 基础软件

- 操作系统：Windows、Unix、Linux、Sybain、Andriod、IOS、Windows CE、Windows Mobile、Palm OS
- 开发平台或环境：Visual Studio (VS)、NetBeans、PyCharm、IntelliJ IDEA、Eclipse、Code::Blocks、Aptana Studio 3、CodeLite、Xcode、Komodo
- 数据库：常见的关系型数据库有Mysql、SQL Server、Oracle、Sybase、DB2
- 工具软件：建模工具、测试工具、管理工具、调试工具

用于开发其他软件或者支撑其他软件运维的软件

• 应用软件: 特定领域软件

- 按应用领域划分：财务软件、图书管理软件、各种控制软件、嵌入式软件、办公自动化软件、安全监控软件、智能制造软件、海洋生态健康检测和分析软件、物流管理软件、供应链管理软件、网上拍卖软件系统、竞选投票软件、...

用于处理和管理应用领域业务的软件

1.3 软件工程发展简史

• 1945 to 1968: The Origins (二进制) → Coding (汇编语言) → Software Development(1951) (Fortran)

- 主要特点: Code and fix
- no process

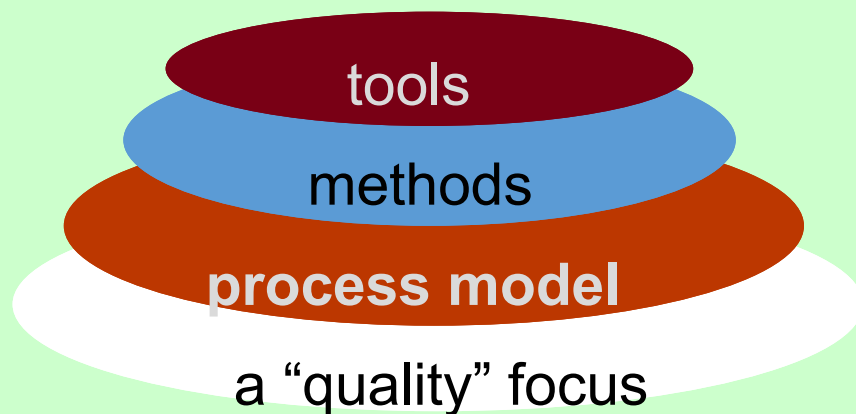
• 1968/1969 to 2001: Software Development → Software Engineering (NATO, 1968\1969) | **No silver bullet**

- 主要特点: Life-cycle、系统化/工程化
- Development model: 瀑布模型、V模型、W模型、快速原型、螺旋模型
- Black-box process: Transform process

•1.3.1什么是软件工程（SE）？

1968/1969以来，一直没有一个人统一的定义，直至...

IEEE Standard 610.12-1990: 软件工程是（1）将系统化的（systematic）、规范化的（disciplined）、可量化的（quantifiable）方法应用于软件的开发、运行和维护中，即：将工程化方法应用于软件开发、运行和维护中；（2）针对（1）中所述方法的研究。



- **SE工具**：各类支持工具(编译、调试、测试、管理、建模...)
- **SE方法**：如何做+基本原则
- **SE基础**：过程
- **SE根基**：质量关注点

软件工程基础是过程(process)!!!

1.3.2 软件工程理论体系的作用

1968/1969,
NATO-.....

软件工程理论体系

指导

指导

► 软件开发实践

1. 软件工程概念
2. 软件开发方法
3. 软件过程：瀑布/V/W
4. 软件模型
5. 编程语言
6. 开发环境与工具
7.

• 软件工程管理实践

1. 风险管理(risk)
2. 人员管理(people)
3. 项目管理(project)
4. 过程管理(process)
5. 产品管理(product)
6. 质量管理与控制
7.

核心关注点：软件产品能力、软件产品质量、软件开发效率、软件运维成本

软件开发主体：软件专家+领域专家

•2001 to 2017 : 敏捷联盟成立→ Agile Development |No silver bullet

- 主要特点: Iteration、Increment、Continuous Integration、Whole life-cycle、Multiple life-cycles
- Agile model: XP、Scrum、FDD、Crystal、 Adaptive Software Development、轻量型RUP
- White-box process: Transform process + Evolution process

•2017 to future: AI+SE|大数据|知识图谱|机器学习| |No silver bullet

- 主要特点: Automation、Intelligence、Smart...
- Learning model: CNN、SVM、NB、KNN、DT、RF/GDBT
- Grey-box process: Transform process + Evolution process + Learning process

2001,敏捷联盟.....

软件工程+敏捷开发理论体系

指导

指导

▶ 软件开发实践

1. 软件工程概念
2. 软件开发方法
3. 软件过程
4. XP/Scrum/RUP/FDD...
5. 软件模型
6. 编程语言
7. 开发环境与工具
8.

• 软件工程管理实践

1. 敏捷开发管理(agile)
2. 风险管理(risk)
3. 人员管理(people)
4. 项目管理(project)
5. 过程管理(process)
6. 产品管理(product)
7. 质量管理与控制
8.

核心关注点：软件产品能力、软件产品质量、软件开发效率、软件运维成本

软件开发主体：软件专家[含敏捷开发专家]+领域专家

1.4 软件工程发展的目标和核心关注点

- 用户和开发人员关心的永远是这几个方面：
 - 软件产品能力（capability）：能做什么，是否有新功能
 - 软件产品质量（quality）：有多好，例如，性能（performance）如何
 - 软件开发和运维成本（cost）：需要多少钱，多少人力
 - 软件开发效率（efficiency）：开发时间需要多久，使用有多方便
- 软件工程发展的目标和核心关注点就是：如何研究新语言、新方法、新技术、新模型、以及如何培养新人才，便于我们更好地满足用户和开发人员需求：
 - 软件产品能力需求：能力越强越好|数据处理、文档处理、语音处理、图像处理、视频处理…
 - 软件产品质量需求：质量越高越好|功能正确性和完整性、性能、安全、可靠、可信…
 - 软件开发效率需求：速度越快越好|开发效率、维护效率、管理效率、测试效率、…
 - 软件运维成本需求：成本越低越好|开发成本、运行成本、维护成本、更新换代成本、…

二、AI时代的软件工程理论方法

Internet, Internet of Things,
cloud computing, edge
computing,...

万物互联互
通时代！

data

AI时代！

Big data, Machine learning,
ontology, knowledge graph,
Pattern recognition, Face
recognition ...

ways

techniques

软件定义一
切的时代！

SDx, software defined x
SDN, software defined network
SD IOT, software defined iot
SDI, software defined
infrastructure
.....

更好的软件产品

AI for SE

AI时代的软件工程

SE for AI

更好的智能产品

- **AI for SE:** 把人工智能技术、大数据技术和新型计算技术与传统的软件工程技术结合起来，运用到**软件开发**和**管理**过程中，以便提高软件开发的效率、降低软件开发成本，提升最终软件产品的质量和能力等。具体包含智能化软件开发方法与技术、智能化软件调试和测试技术、智能化软件架构和演化技术等等。

- **SE技术理论方法的完善和发展**

- **SE for AI:** 利用软件工程技术理论方法来指导AI系统（例如智能控制系统、智能交通指挥系统、智能诊疗系统等）的软件开发和管理等。

- **SE技术理论方法的应用拓展**

History of AI+SE

• 基于知识的软件工程

- Knowledge-Based Software Engineering, KBSE 1986-1996, 共11届)
- Software Engineering and Knowledge Engineering, SEKE 1989-至今
- Joint Conference on Knowledge-Based Software Engineering, JCKBSE 2004-2014, 共11届

• 自动化软件工程

- Automated Software Engineering Conference, ASE 1997-至今 【前期KBSE 1986-1996】
- Automated Software Engineering Journal, 1994-至今

• 智能系统软件构造

- Intelligent Systems Design and Applications, ISDA 2001-至今
- Complex, Intelligent, and Software Intensive Systems, CISIS 2007-至今

• 自适应软件系统

- Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2006-2017
- Self-Adaptive and Self-Organizing Systems, SASO 2007-至今

• 智能软件工程

- Workshop on Intelligent Software Engineering, WISE³ 2000
- **Workshop on Intelligent Software Engineering, WISE 2017-**

智能化软件工程(ISE): SE家族的成员之一

WISE³: the Third International Workshop on Intelligent Software Engineering with ICSE 2000

There is a growing realization that the design of effective software engineering tools must be smarter. Real world software specs can be very intricate. Manual browsing by a software engineer cannot reveal its subtleties. Automatic tools are required to reflect over business knowledge to identify what is missing or could be effectively changed. At the same time, many AI researchers now realize that software engineering provides the best testbed for AI tools and techniques. While these AI tools are all potentially useful, the core question remains:

Which of these tools, if any, are truly cost-effective?

A sample of these AI tools is listed below. For a further list of techniques, see the proceedings of WISE¹ and WISE².

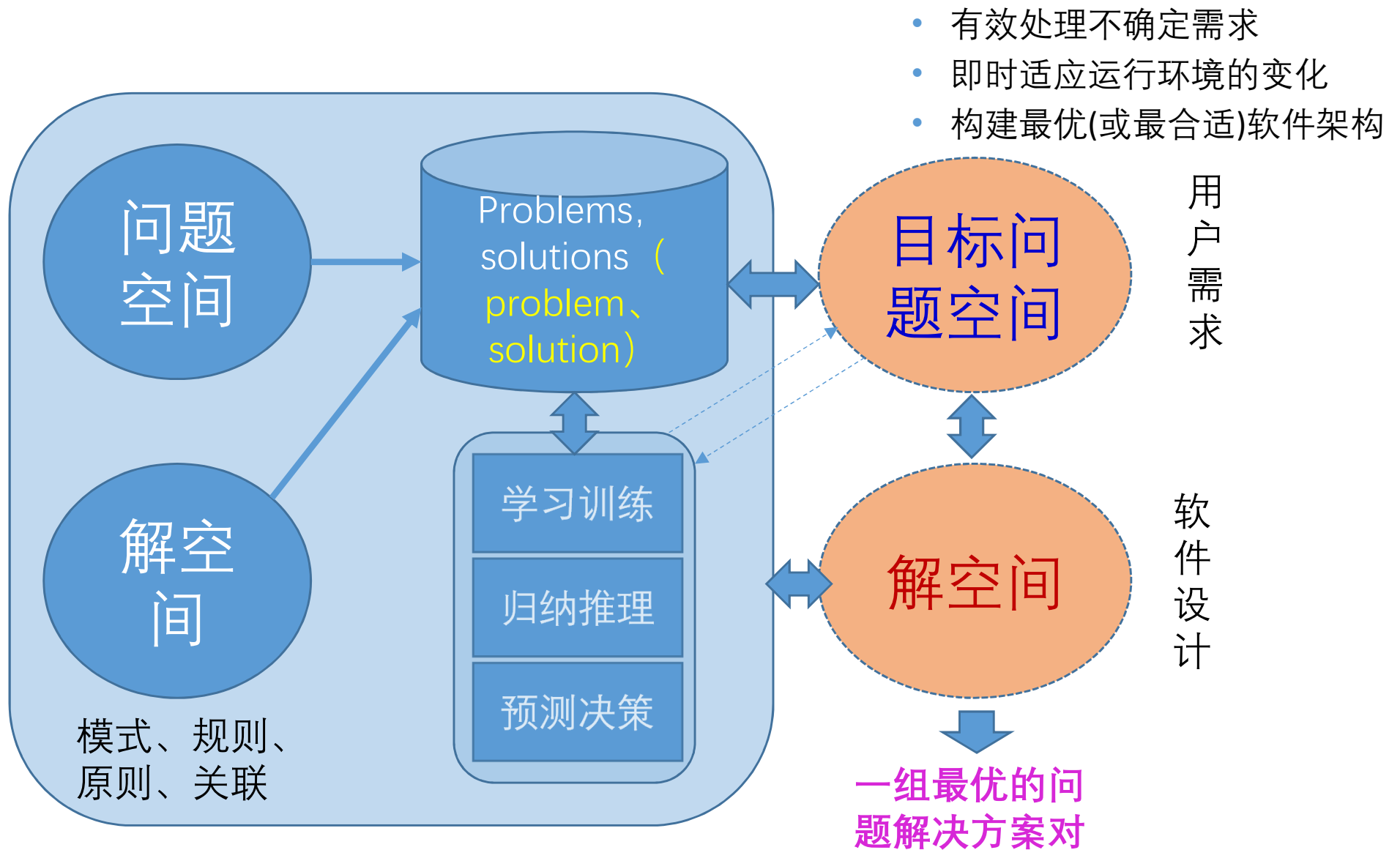
- (1) **During analysis:** Knowledge acquisition methods for requirements elicitation; Knowledge representation methods for the business knowledge; Non-classical logics for requirements engineering.
- (2) **During design and coding:** Knowledge-based program synthesis; Knowledge based techniques; Knowledge-based validation techniques to detect bad semantics; Theorem proving and formal reasoning for managing changing specs.
- (3) **During maintenance:** AI tools to maintain declarative and procedural knowledge; AI tools for program comprehension and reverse engineering.

The purpose of the WISE series is to assess the utility of the above technique, long-term goal (e.g. WISE⁶) is to foster the develop of some rigorous evaluation experiment.

The International Workshop on Intelligent Software Engineering (WISE 2017), co-located with ASE 2017, is to bring together academics *working on algorithms, methods, and techniques* for automated software engineering, with practitioners, interested in developing *more intelligent tool support to address important problems* in software engineering practices. The workshop's purpose is to make researchers aware of industry's problems, and practitioners aware of research approaches **in the broad area of intelligent software engineering.**

Some example questions to be explored by the workshop are the following: *how can AI improve the efficiency of software development? How can AI reduce the maintenance cost of software services? Can AI shift the software development model? Can AI even fully overtake coding tasks?*

2.1 AI for SE



现在→将来

软件工程+人工智能

指导

指导

► 软件开发实践

1. 软件工程概念
2. 人工智能原理
3. 软件开发方法
4. 软件过程/CNN...
5. 软件模型
6. 编程语言
7. 开发环境与工具
8.

• 软件工程管理实践

1. 风险管理(risk)
2. 人员管理(people)
3. 项目管理(project)
4. 过程管理(process)
5. 产品管理(product)
6. 质量管理与控制
7.

核心关注点：软件产品能力、软件产品质量、软件开发效率、软件运维成本

软件开发主体：软件专家+领域专家+人工智能专家

2.1.1 AI for Software Development

- **AI Programming Languages:** lisp, planner, cinniver, qlisp, pop-2, sail, fuzzy // java、c++、python、...
- **AI SD models:** transform models、 evolution models、 learning models、 ...
- **AI SD methods:** 基于搜索、基于学习、基于预测、基于推理
- **AI SD techniques:** 云计算、海量数据存储、大数据、机器学习、自然语言处理、语音识别、图像识别、视频处理、动态环境感知、...
- **AI SD management:** 海量复杂数据管理、非结构化数据管理、4P、质量管理、...
- **AI SD platforms and tools:** AI Studio、aiXcoder、Google ML kit 工具包、Tensorflow、Infosys Nia、H2O、...

- **案例一：** aiXCoder智能编程助手 <https://www.aixcoder.com/#/>
- **核心功能**
 - aiXcoder代码智能补全AI引擎预测您要输入代码，只需“一键确认”，即可完成一行或多行代码编写。
 - aiXcoder相似代码推荐自动搜索并推荐可供参考的优质代码样例。
- **产品优势**
 - 智能代码搜索： aiXcoder能够在程序编程的过程中，智能地搜索并推荐与当前程序功能相似的规范程序代码，为您提供有力的编程参考。
 - 代码补全： 让您一键完成一整行或多行代码编写。
 - 完成相同代码，按键次数大大减少： 把重复的劳动交给aiXcoder,让开发者专注于逻辑、业务的编写。



案例二：华为AI开发平台ModelArts

- 繁多的AI工具安装配置、数据准备、模型训练慢等是困扰AI工程师的诸多难题。为解决这个难题，将一站式的AI开发平台（ModelArts）提供给开发者，从数据准备到算法开发、模型训练，最后把模型部署起来，集成到生产环境。一站式完成所有任务。ModelArts的功能总览如下图所示。

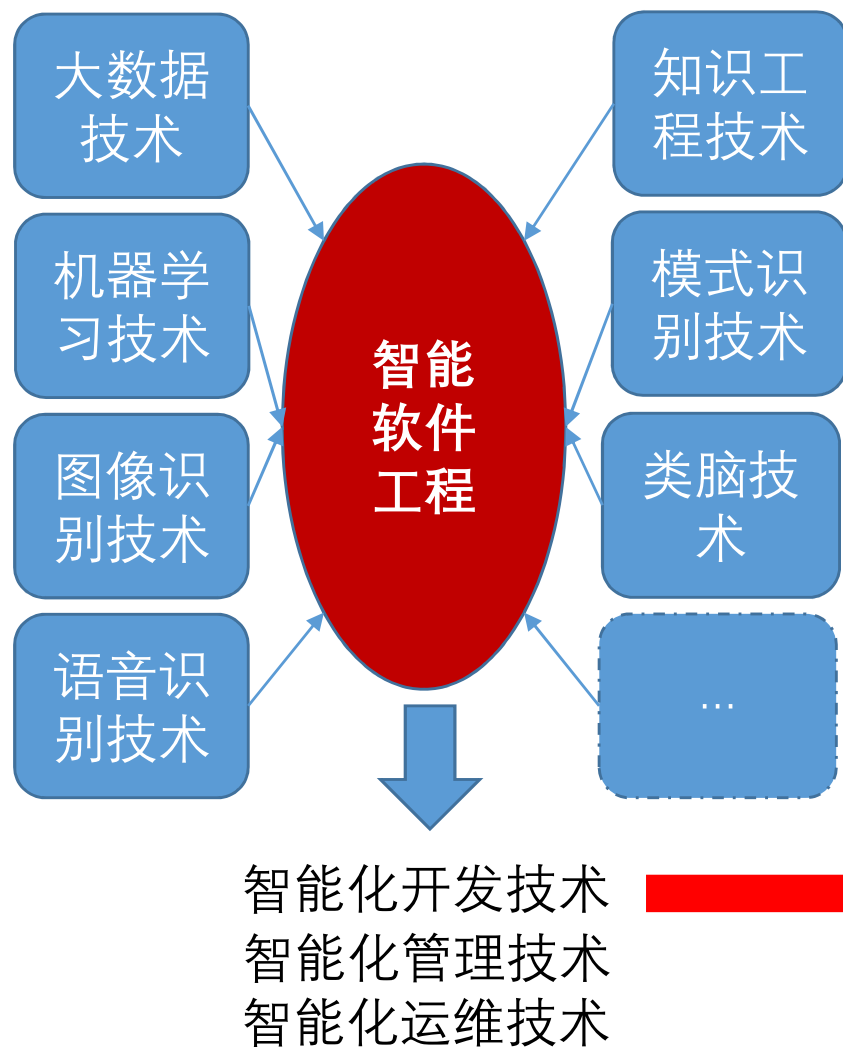


案例三： 百度AI Studio

- AI Studio是基于百度深度学习平台飞桨的一站式AI开发平台，提供在线编程环境、免费GPU算力、海量开源算法和开放数据，帮助开发者快速创建和部署模型。



AI时代的软件工程技术



- **是否提高软件能力?**
 - 能检测更多类型缺陷(语法的、语义的、逻辑的...)
 - 能检测更多地方的缺陷(文档的、模型的、代码的、数据的...)
- **是否提高软件质量?**
 - 查全率
 - 查准率
 - F-value
- **是否提高软件性能?**
 - 检测时间
 - 资源占用率
- **是否降低开发成本和运维成本?**
 - 时间成本
 - 资源成本
 - 经济成本

哪些没有发生变化? 哪些发生了变化?

2.1.2 AI for Software Defect Detection

• 数据缺陷检测

- 结构化数据类型缺陷
- 半结构化数据类型缺陷
- 非结构化数据类型缺陷

规范性、完整性、准确性、一致性、时效性、可访问性、多样性

• 文档缺陷检测

- 用户需求文档
- 软件需求文档
- 架构设计文档
- 模块设计文档
- 接口设计文档
- 数据库设计文档
- 各种测试文档
- 帮助文档
- 用户使用说明书

规范性、完整性、准确性、一致性、无冗余、即时更新、模棱两可、遗漏、不确定性、有用性

• 模型缺陷检测

- 需求模型
 - 用例模型
 - 领域模型
- 设计模型
 - 交互模型
 - 功能模型
 - 接口模型
- 实现模型
 - 算法模型
 - 数据模型

一致性、互操作性、兼容性、完整性、准确性、易理解性、可扩展性、可修改性、可维护性、复杂性、循环依赖

• 代码缺陷检测

- 不同语言源代码
- 中间码
- 二进制码

规范性、BUG、ERROR、Bad Smell、Technical Debt、代码漏洞、冗余代码, 等等

• 算法缺陷检测

D:\360MoveData\Users\Administrator\Desktop

打开

读入需求文本

读入UML

缺陷内容	缺陷位置	缺陷名	规范性 权值
☐ 3.2.2.m.1输入数据实体	3.2.2.m.2过程算法或公式	文本描述内容遗漏	完整性 5
☐ 3.2.2.m.2过程算法或公式	3.2.2.m.3受影响的数据实体	文本描述内容遗漏	完整性 5
☐ 3.2.2.m.3受影响的数据实体	3.2.3数据构建规范	文本描述内容遗漏	完整性 5
☐ 3.2.3.p构建p	3.2.3.p.1记录类型	文本描述内容遗漏	完整性 5
☐ 3.2.3.p.1记录类型	3.2.3.p.2组成字段	文本描述内容遗漏	完整性 5
☐ 3.2.3.p.2组成字段	3.2.4数据词典	文本描述内容遗漏	完整性 5
☐ 3.2.4.q数据元素q	3.2.4.q.1名称	文本描述内容遗漏	完整性 5
☐ 3.2.4.q.1名称	3.2.4.q.2表示法	文本描述内容遗漏	完整性 5
☐ 3.2.4.q.2表示法	3.2.4.q.3单位/格式	文本描述内容遗漏	完整性 5
☐ 3.2.4.q.3单位/格式	3.2.4.q.4精确度/准确度	文本描述内容遗漏	完整性 5
☐ 3.2.4.q.4精确度/准确度	3.2.4.g.5范围	文本描述内容遗漏	完整性 5
☐ 3.2.4.g.5范围	3.3性能需求	文本描述内容遗漏	完整性 5
☐ 本部分一定可能应当提供整个SRS的概述	4附录12	文本模糊	歧义性 5
☐ 这些信息也许肯定可以通过引用附录或引用其他文	1.4引用文件	文本模糊	歧义性 5
☒ a)系统接口：这是一个系统的接口。/a)系统接口...	2.2产品描述/2.1.1系统接口	文本不一致	一致性 10
☒ 注：有时此条规定作为用户界面的一部分。/注：...	2.1.7操作/3.3性能需求	文本不一致	一致性 10

文本检测

UML检测

对照检测

● 全选

生成缺陷报告

检查出的总数量

29

检查正确的数量

27

生成检查结果

实际总数量

28

完整性的缺陷数量

19

实际数量

20

无歧义性的缺陷数量

2

实际数量

2

无冗余性的缺陷数量

3

实际数量

4

一致性的缺陷数量

0

实际数量

0

正确性的缺陷数量

1

实际数量

2

读入UML

生成缺陷报告

生成检查结果

2

2

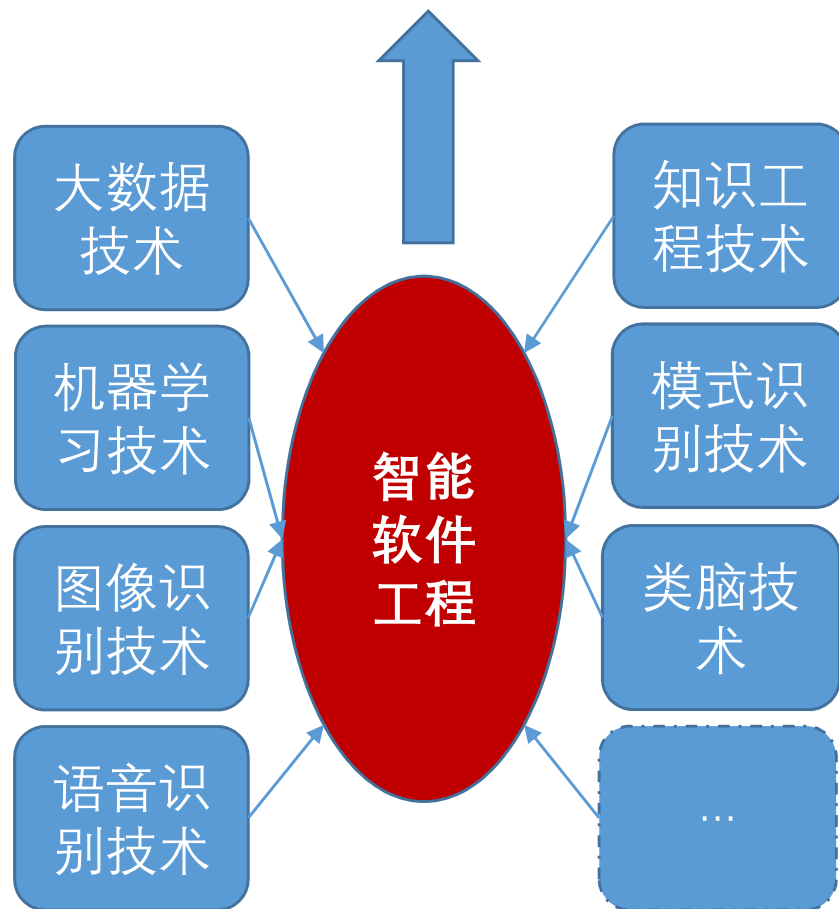
AI for defect detection and defect detection for AI

AI for DD

智能化质量保障技术
智能化缺陷检测技术



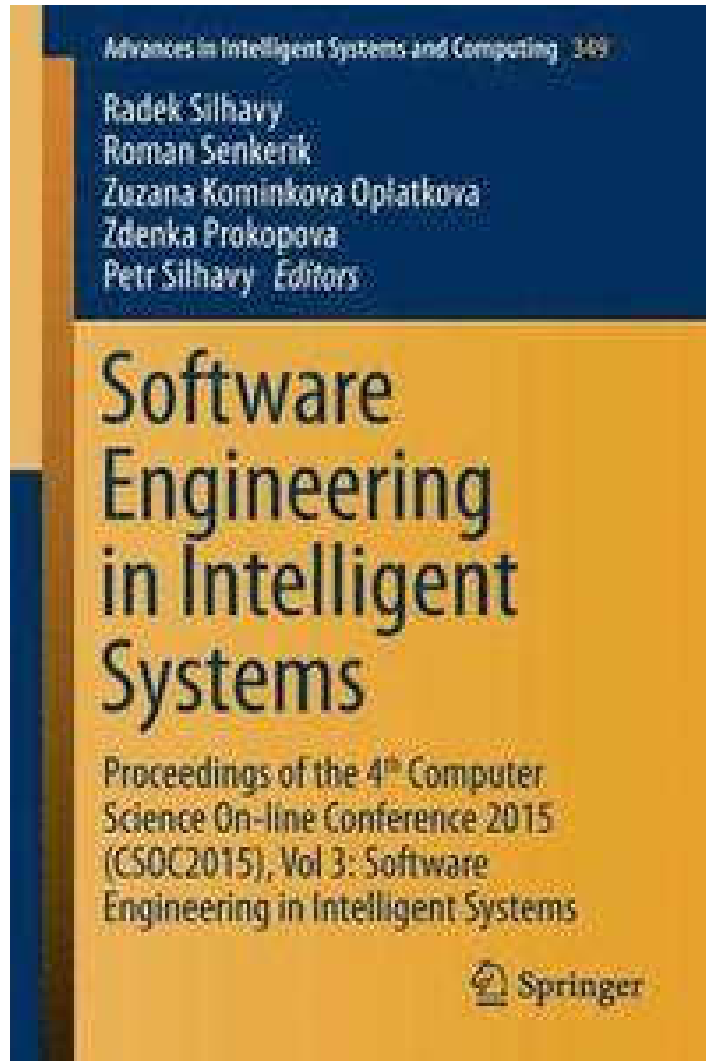
AI时代的软件工程技术



- **是否提高软件能力?**
 - 能检测更多类型缺陷(语法的、语义的、逻辑的...)
 - 能检测更多地方的缺陷(文档的、模型的、代码的、数据的...)
- **是否提高软件质量?**
 - 查全率
 - 查准率
 - F-value
- **是否提高软件性能?**
 - 检测时间
 - 资源占用率
- **是否降低开发成本和运维成本?**
 - 时间成本
 - 资源成本
 - 经济成本

哪些没有发生变化? 哪些发生了变化?

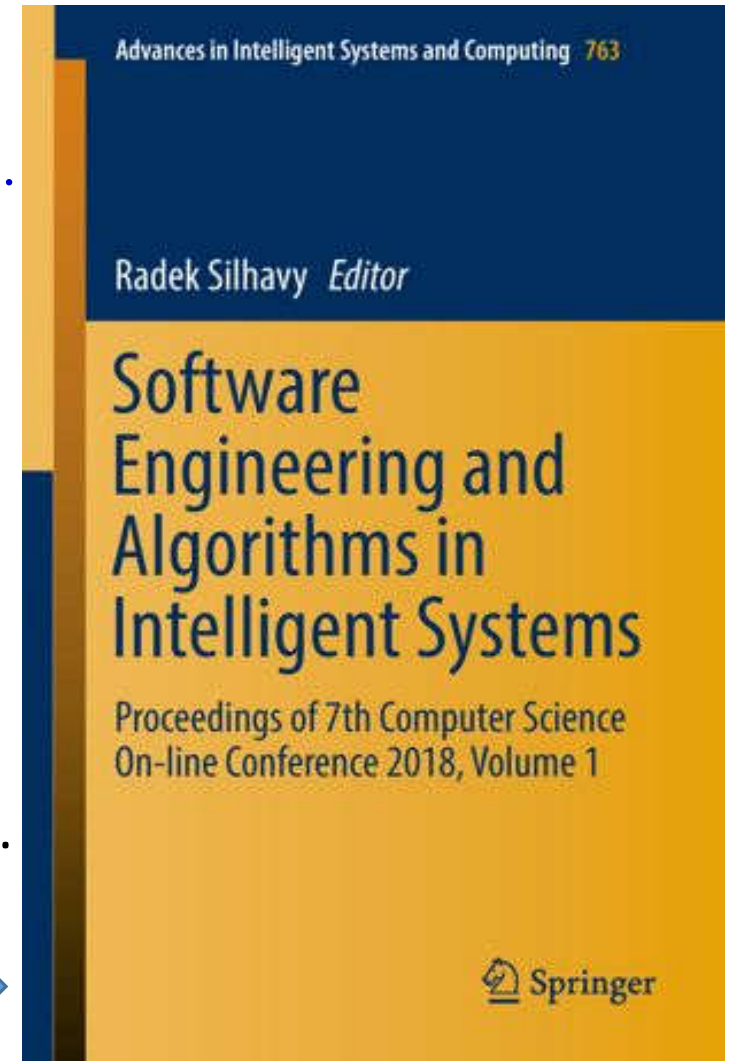
2.2 SE for AI: SE for Intelligent System



- Software Engineering in Intelligent Systems. CSOC2015, Springer



- Software Engineering and Algorithms in Intelligent Systems. CSOC2018, Springer



三、AI时代的软件工程技术

- **Intelligent SE** concentrates on algorithms, methods, and techniques for *automated software engineering* by developing more intelligent tools to support to address important problems in software engineering practices.
- 在软件工程中AI能做什么？
 - Can AI improve the efficiency of software development?
 - Can AI reduce the maintenance cost of software services?
 - Can AI shift the software development model?
 - Can AI even fully overtake coding tasks?
 - Can AI capture certain or uncertain requirement?
 - Can AI find feasible and best solution? For example, Local optimal solution or global solution.
 - Can AI improve the quality of software artifacts and final products?
 -

软件开发具有智能了：技术手段、理论方法

四、AI时代的软件产品

- 软件产品就是软件，软件根据它的用途分成如下三种类型：
 - 编程语言：lisp, planner, cinniver, qlisp, pop-2, sail, fuzzy///java、c++、python, ...
 - 基础软件
 - 操作系统：Andriod、IOS、Windows CE、Windows Mobile...
 - 开发平台：Xcode、aiXcoder
 - 工具软件：SHC, 智能测试工具
 - 应用软件：各行各业
 - 智能制造
 - 智慧家居
 - 自动驾驶
 -

开发出来的软件产品具有智能了：感知能力、学习能力、推理能力、预测能力、....

基础软件： 用于帮助开发其他软件的软件；用于支持其他软件运行的软件

- 智能化软件开发环境（平台）
- 智能化软件开发工具
 - 智能需求获取工具软件
 - 智能仿真工具软件
 - 智能架构工具软件
 - 代码自动生成和修复工具软件
- 智能化软件开发管理工具
 - 过程管理工具软件
 - 项目管理工具软件
 - 产品管理工具软件
 - 人员管理工具软件
 - 质量管理工具软件
 - 版本管理工具软件
 -
- 智能化质量保障工具
 - 智能测试工具软件
 - 智能监控工具软件
 - 智能路径剖析工具软件
- 软件健康检测平台软件
 - 基于大数据的SHC软件
 - 基于知识图谱的SHC软件
 - 基于人工智能SHC软件---各种软件检测机器人
- 代码深度剖析工具软件
 - 代码度量工具软件
 - 代码依赖关系分析工具软件
 - 代码漏洞检测工具软件
 -

应用软件： 帮助解决领域业务问题的各种软件

图像识别软件、语音识别软件、机器翻译软件、各种手机APP、智慧交通管理软件、智慧城市、智能家居、自动驾驶、智能办公...



- **案例一：** aiXCoder智能编程助手 <https://www.aixcoder.com/#/>

- **核心功能**

- aiXcoder代码智能补全AI引擎预测您要输入代码，只需“一键确认”，即可完成一行或多行代码编写。
- aiXcoder相似代码推荐自动搜索并推荐可供参考的优质代码样例。

- **产品优势**

- 智能代码搜索： aiXcoder能够在程序编程的过程中，智能地搜索并推荐与当前程序功能相似的规范程序代码，为您提供有力的编程参考。
- 代码补全：让您一键完成一整行或多行代码编写。
- 完成相同代码，按键次数大大减少：把重复的劳动交给aiXcoder,让开发者专注于逻辑、业务的编写。



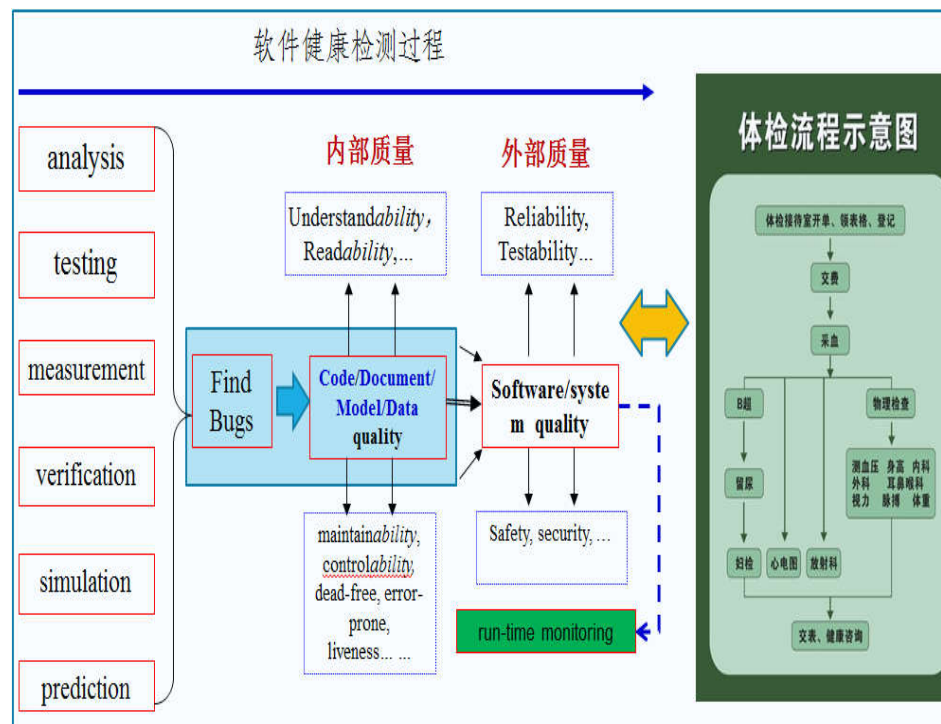
• 案例二：SHC 1.0 软件全方位健康检测平台

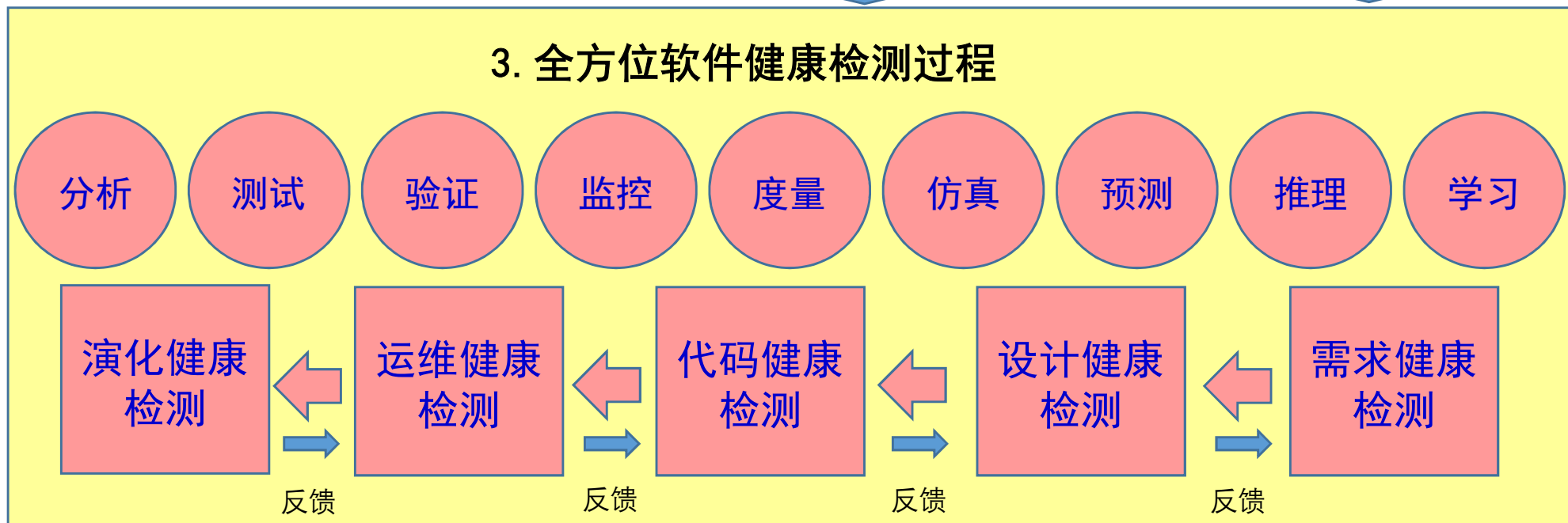
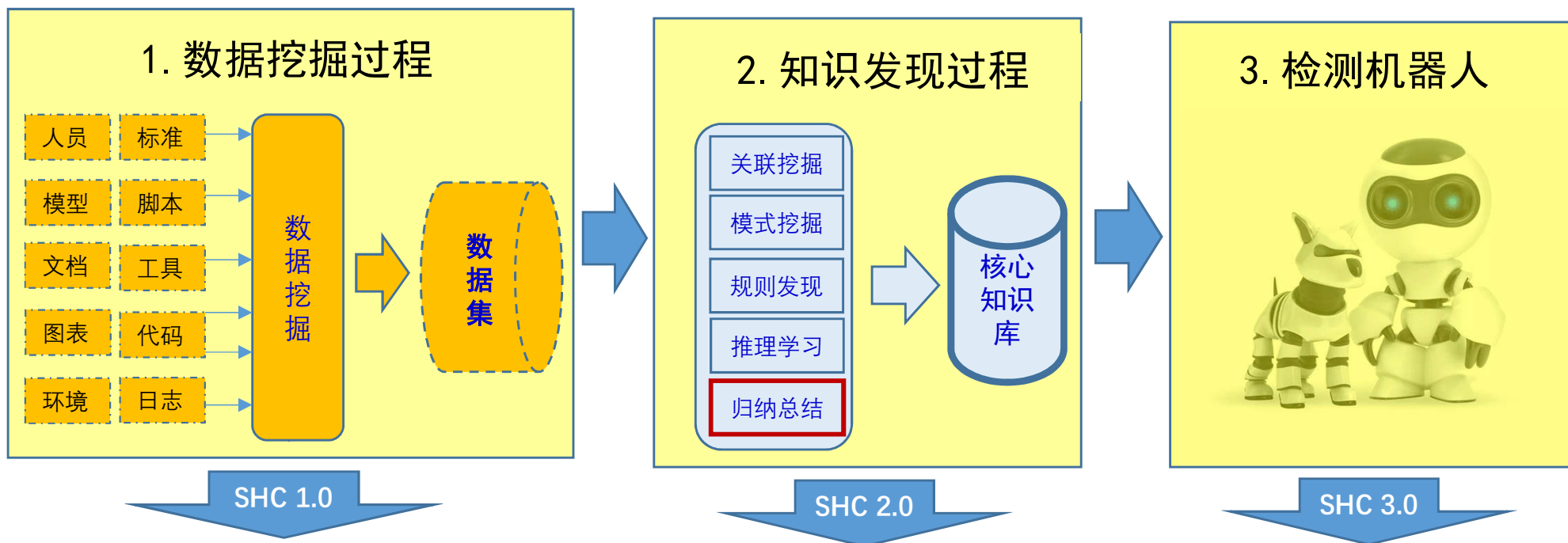
• 如何解决软件健康问题？

- 第一步：软件健康检测！（如同人们需要体检、诊疗一样）
- 第二步：软件缺陷定位！
- 第三步：软件缺陷修复！

• SHC平台主要手段

- 分析（望闻问切、专家会诊）
- 测试（运动平板实验、抽血化验）
- 监控（透视、B超、心电图）
- 度量（量身高、体重、血压）
- 验证（切片、肠镜、胃镜检查）
- 仿真（核磁共振）
- 推理（病例分析）
- ...





项目管理

项目管理

知识库管理

项目管理

clone

1.1.1.1

开发视图: 文件目录树

架构视图: 架构恢复 编译构建

度量视图: 度量基本信息 架构演进性及原则达成性度量 坏味检测

重构视图: 变更检测 架构重构 面向持续演进的重构 架构模式识别

filezilla

openbbs

MyBlog

freecol

terrier

mirror

zxing

git

+添加项目

版本详情

添加版本

项目详情

共98个项目 分10页 当前第1页

首页 上一页 下一页 尾页

首页

项目管理

基本度量信息

McCall质量模型

导航菜单

项目信息

+添加项目

项目名称

创建日期

VUE_TEST

2019.10.15

-

☒ 1.1.1.1

版本详情

☐ 1.1.1.2

版本详情

项目信息：基本度量信息 项目依赖信息

添加版本

质量评估：McCall质量模型 Boehm质量模型 ISO质量模型 参考质量模型

缺陷预测：缺陷倾向排序预测 缺陷数量预测 缺陷类型预测 缺陷严重性预测

项目详情

风险分析：许可证介绍 冲突识别 不一致性检测 合规性分析 风险评估 风险传播

tomcatVersions

2019.10.17

+

risk_analysis

2019.10.17

+

risk_spread

2019.10.17

+

inconsistency

2019.10.18

+

共5个项目 分1页 当前第1页

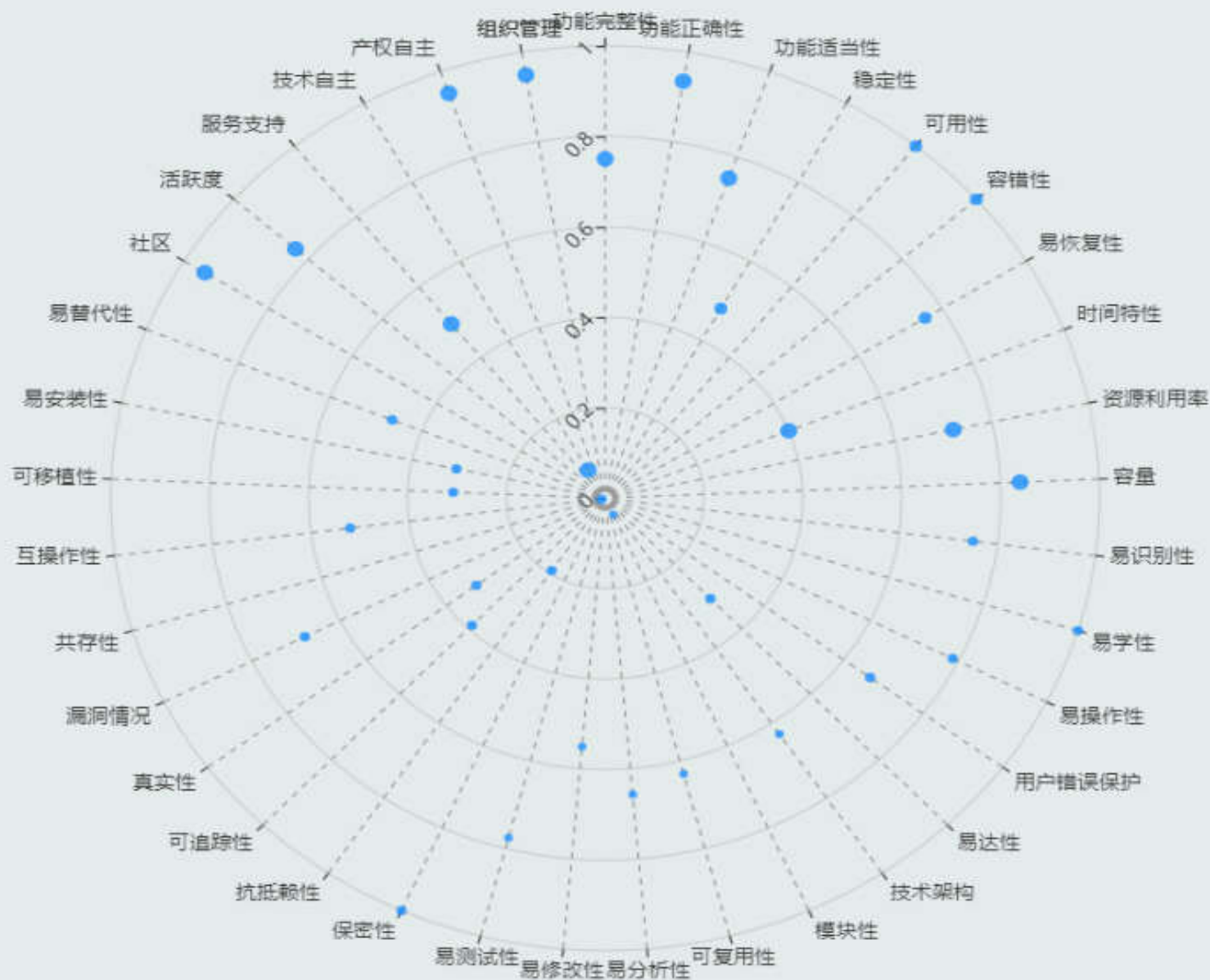
首页

上一页

下一页

尾页

二级指标权重与度量值



用户可在二级指标权重与度量值展示部分[查看](#)所选择的二级指标的权重以及度量值，

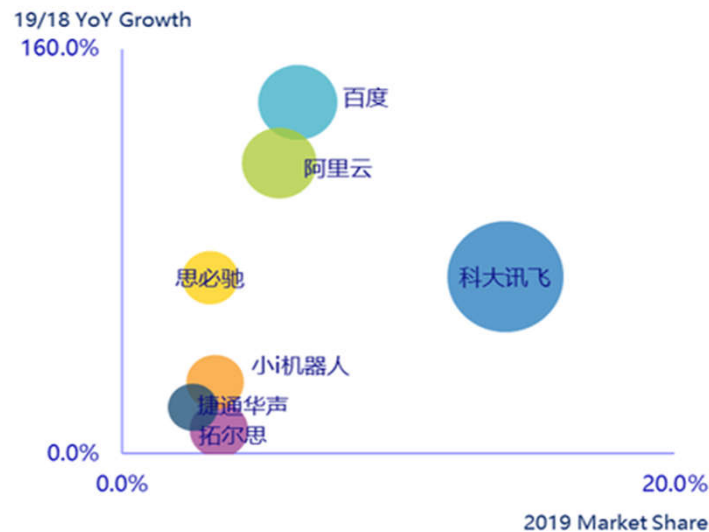
其中，[点越靠近圆心](#)表示该一级指标的度量值越小，越靠近外圈表示该一级指标的度量值越大；

[点的面积](#)越小表示该一级指标的权重越小，点的面积越大表示该一级指标的权重越大。

五、AI时代的软件产品市场



中国人工智能市场之语音语义应用主流供应商，2019

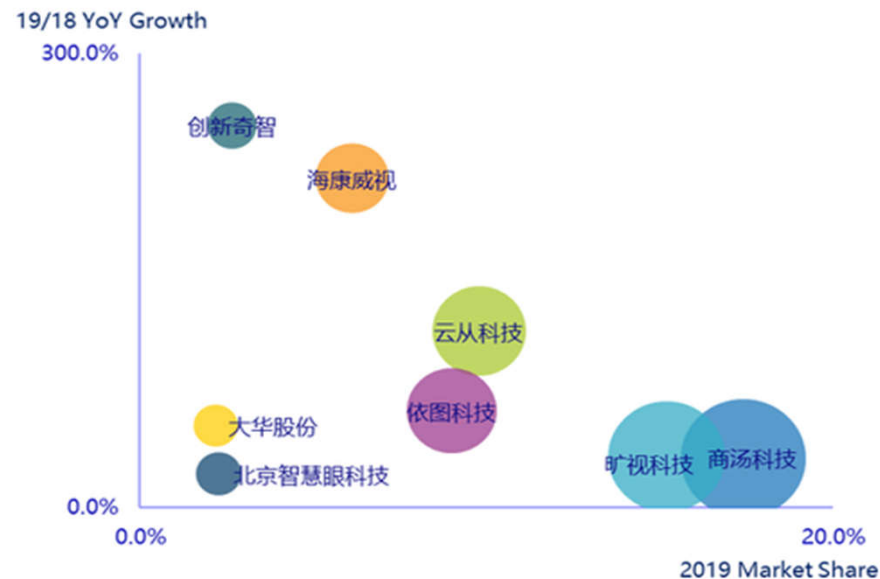


来源：《IDC中国人工智能市场软件及应用半年度研究报告，2019H2》

备注：此图部分公司仅提供语音技术或仅提供语义技术，考虑到语音语义技术融合应用为大势所趋，因此本次研究将所有参与者放到一起统一统计并展示。欲了解详细数据，请咨询IDC。



中国人工智能市场之计算机视觉应用主流供应商，2019



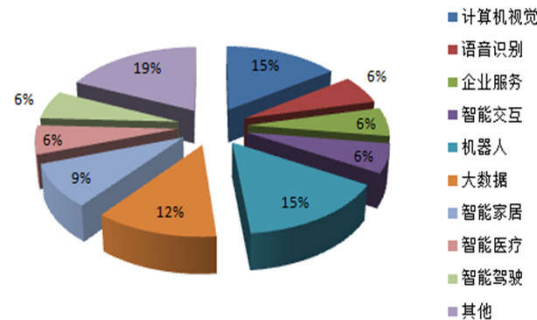
来源：《IDC中国人工智能市场软件及应用半年度研究报告，2019H2》

2020年中国AI企业分地区占比情况

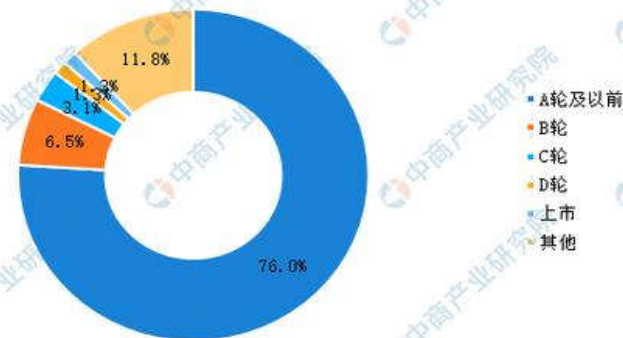


制图：中商情报网 (http://www.askci.com/)

9月融资事件行业分布图



2020年AI公司融资阶段占比情况



制图：中商情报网 (https://www.askci.com/)

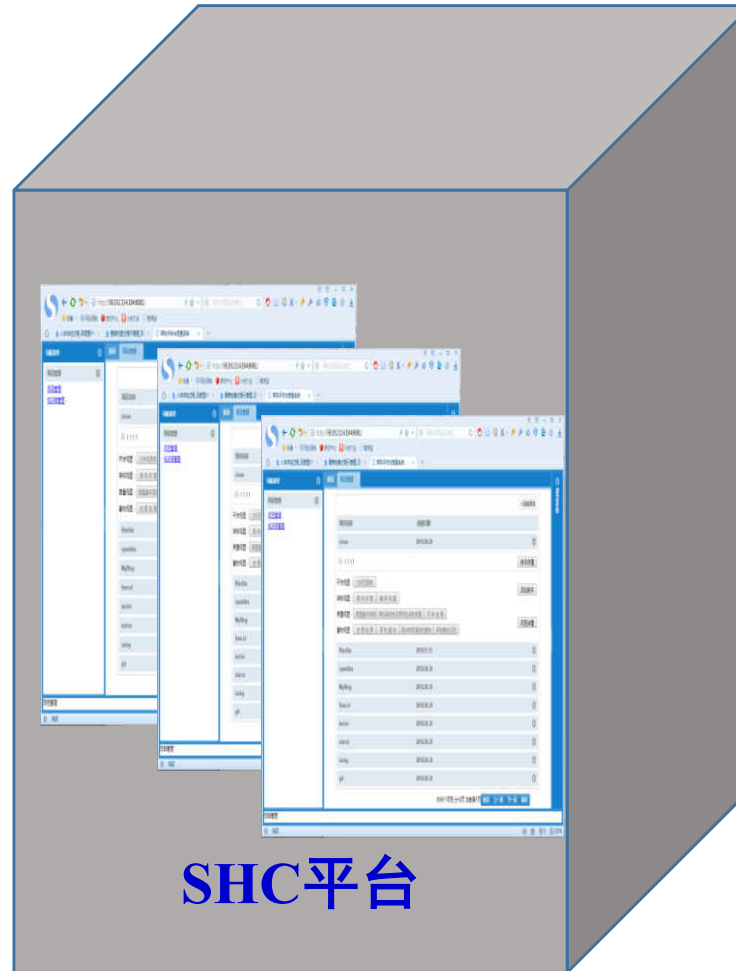
2020-08-04

案例分析：SHC产品

功能安全单位(企业)
需求：高质量、高可靠、高安全、高可信。例如，航空航天、轨道交通、军事...

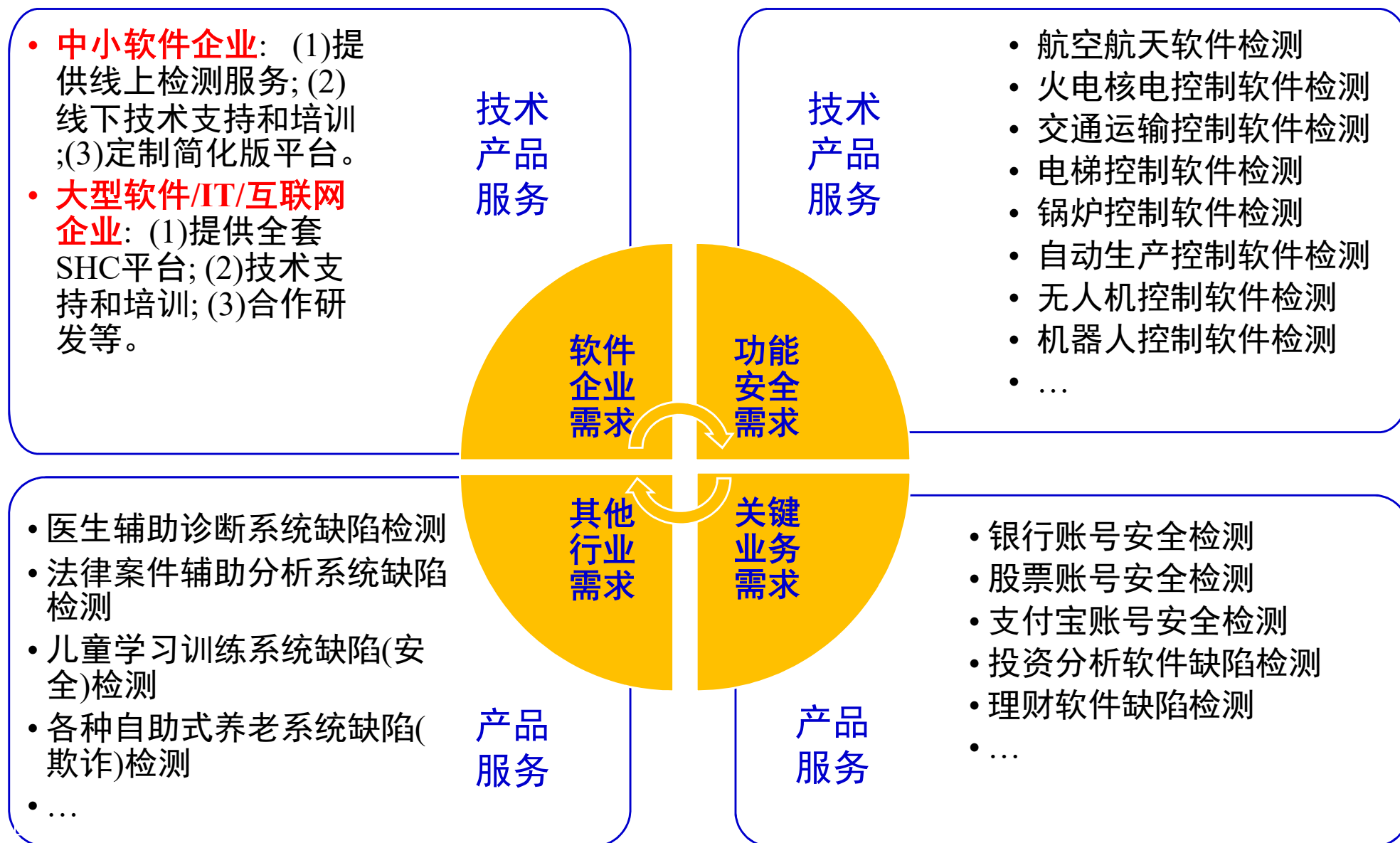
关键业务企业需求：
高质量、高可靠、高安全、高可信。
。例如，银行、保险、证券、在线支付...

软件企业需求：它们很难在软件健康检测方面投入大量人力、物力和财力...



软件检测费用占软件成本的45%左右：

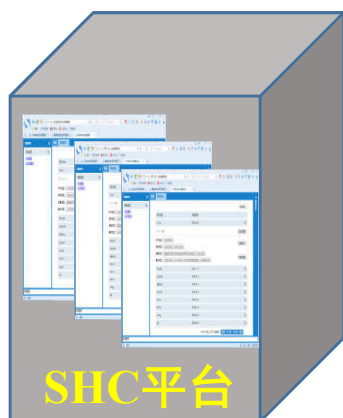
- 全球万亿市场空间
- 中国数千亿市场空间
- 江苏省百亿的市场空间



数据 ↔ 知识 ↔ 产品 ↔ 服务



- 数据库、知识库的使用或者购买
- 加工后的软件开发、运维和升级过程大数据以API形式接入客户内部系统，按使用量收费



- 购买SHC平台和检测机器人：单机版、网络版、精华版
- 线上使用SHC平台[Web版]按使用次数、或代码行收费。或者可以申请账户，包月或包年



- 根据不同客户需求，进行平台定制
- 不同等级技术培训：**初级**是学会怎么用，**中级**是学会怎么维护，**高级**是学会怎么改进



武汉大学 人工智能与软件工程暑期学校

2020-08

**感谢您的关注！
感谢邀请！**

