东南大學 软件工程研究所
INSTITUTE OF SOFTWARE ENGIEERING,SOUTHEAST UNIVERSITY

# 软件工程核心关注点牵引的科学研究选题


李必信

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

http://cse.seu.edu.cn/people/bx.li/

# 提纲

- 一、软件危机和软件工程核心关注点
- 二、软件全生命周期的质量保障问题
- 三、关注点牵引的科研选题

# 一、 软件危机和软件工程核心关注点

# （1）软件工程发展历程概述

- **1945 to 1968**: **The origins （二进制）→ Coding（汇编语言） → software development(1951)（Fortran）**
  - Code and fix
- **1968 to 2001**: **Software development → software engineering（NATO，1968\1969）**
  - Software process: 工程化、规范化
  - Black-box
- **2001 to present**：敏捷联盟成立→ Agile Development |No silver bullet|智能化|大数据|…
  - Iteration
  - Increment
  - Continuous Integration
  - White-box

# A.第一次软件危机（1945 to 1968）

## （A.1）危机表现

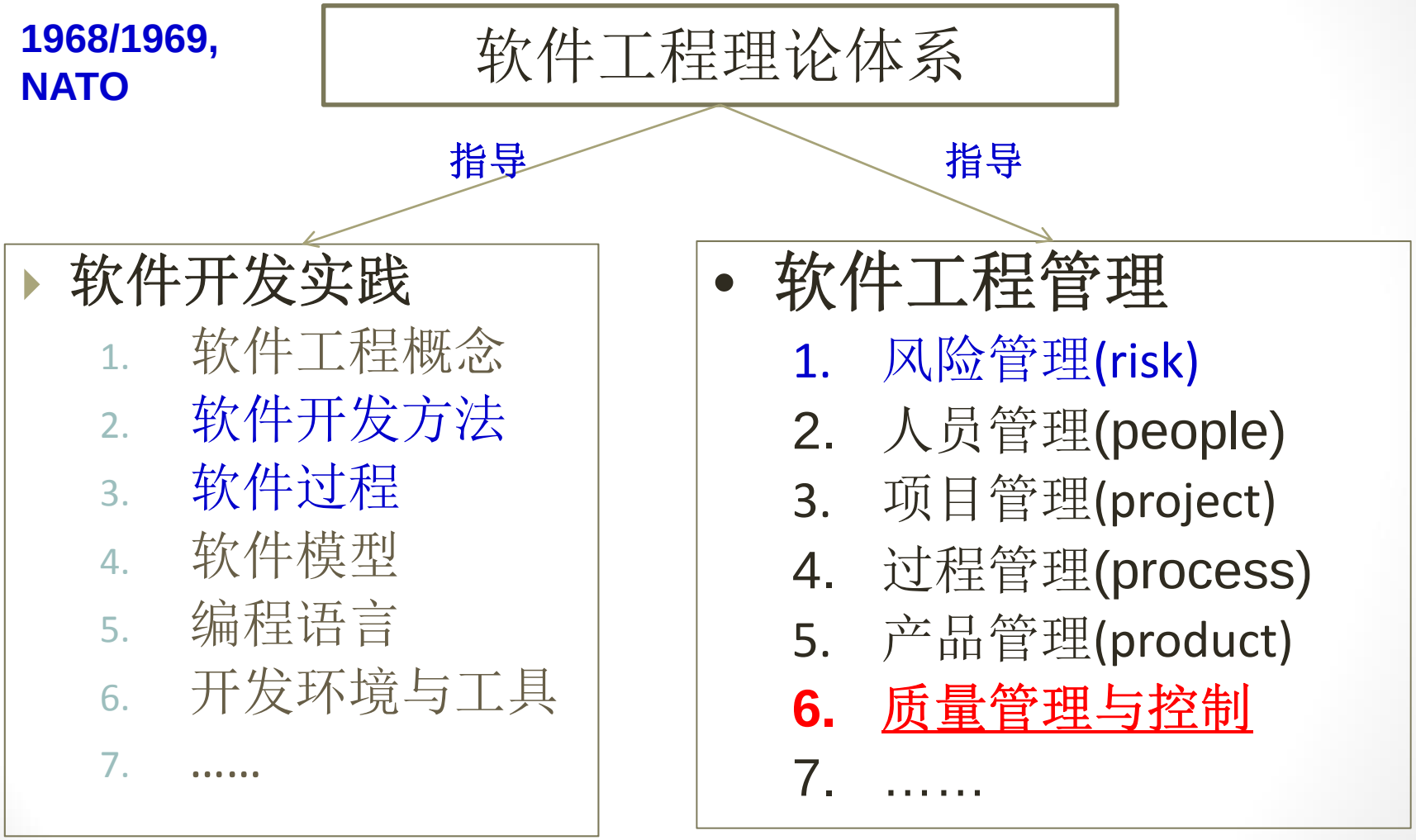
- **软件生产效率低下,软件成本高**: 软件开发成本和进度的估计常常很不准确。
- **软件产品质量差**: 用户对“已完成的”软件系统不满意的现象经常发生。
- **软件产品难以(不能)维护**: 软件通常没有适当的文档资料

## （A.2）危机根源

- 人员知识结构单一
- 手工作坊：code and fix
- 理论体系不成熟[软件开发]: 语言、工具、过程、方法、质量保证.....
- 软件开发管理不完善：规划、预算、可行性分析、风险预测与控制.....
- .....

# （A.3）软件工程理论体系形成

**1968/1969, NATO**




No silver bullet

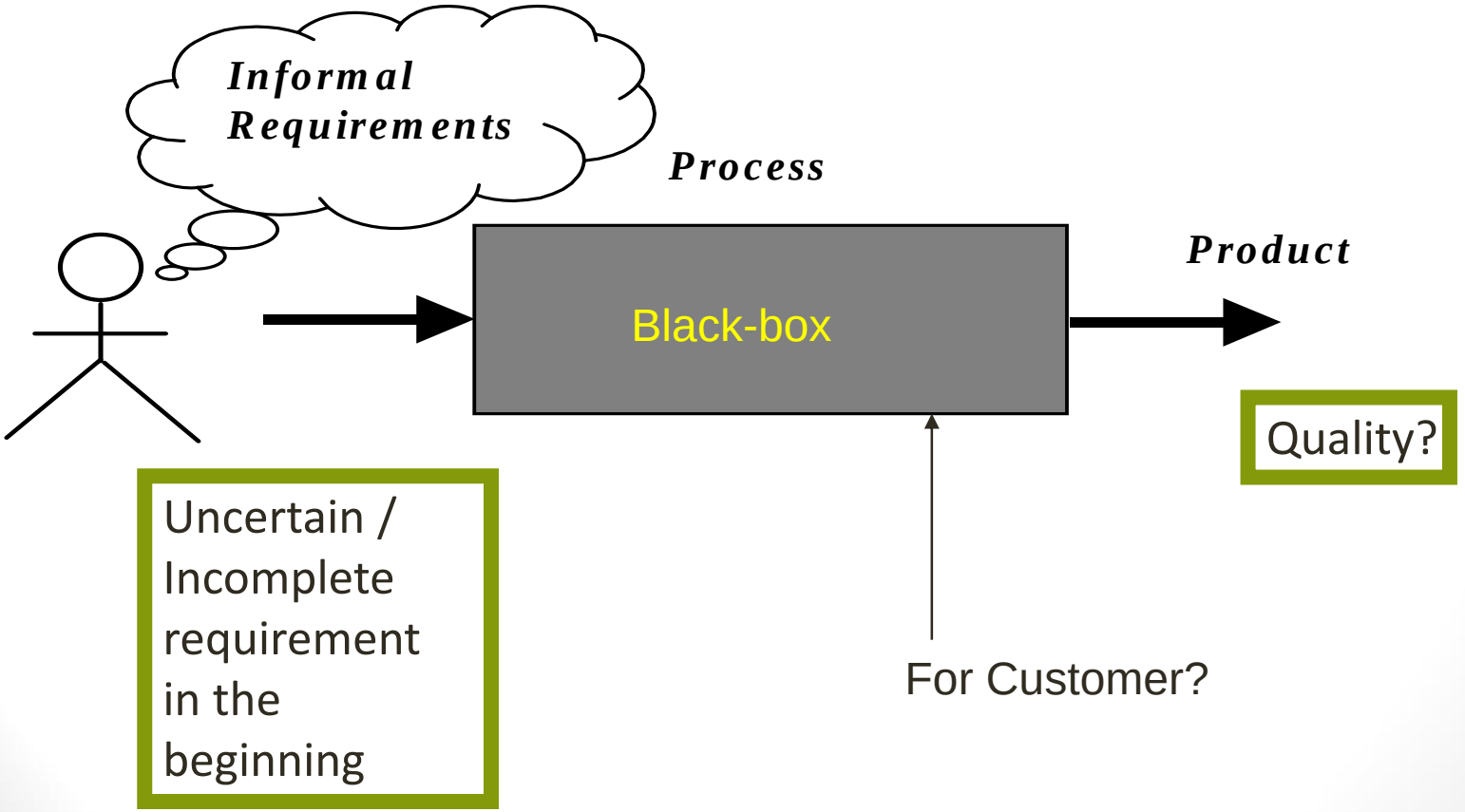
# （B）第二次软件危机（ 1968 to 2001 ）

## （B.1）危机表现

- The project will produce the wrong product[有错的产品].
- The project will produce a product of inferior quality[低质量的产品].
- The project will be late[延迟].
- We'll have to work 80 hour weeks[每周工作80小时].
- We'll have to break commitments[违约].
- We won't be having fun[没有休闲时间].

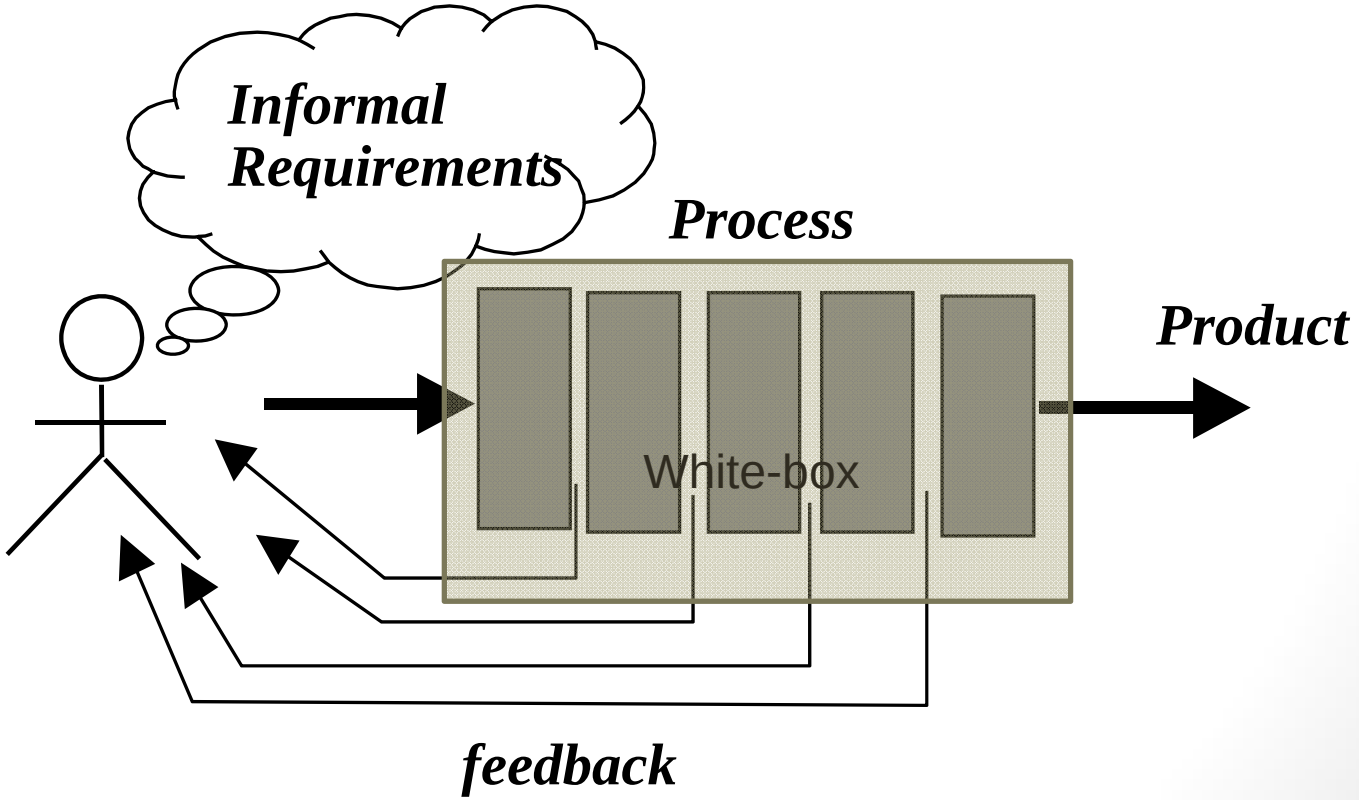
# （B.2）危机根源

- The assumption is that requirements can be fully understood prior to development
- Interaction with the customer occurs only at the beginning (requirements) and end (after delivery)
- Unfortunately the assumption almost never holds



# （B.3）软件工程理论体系完善

- Reduce risks by improving visibility—**white-box**
- Allow project **changes** as the project progresses
  - based on feedback from the customer




No silver bullet

# （C）第三次软件危机（2001 to present）

- **（C.1）危机表现**
  - 变更/演化/重用（质量、效率、成本）
  - 知识产权问题、安全问题
- **（C.2）危机根源**
  - 大量遗产系统的挖掘和重用
  - 大量开源软件/代码的使用
  - 普遍存在的动态/不确定需求
- **（C.3）软件工程理论体系发展/重建**
  - 自动化
  - 智能化/智慧化
    - 自适应、自测试、自修复、自主演化
  - 自主软件（混源软件、开源软件、闭源软件）

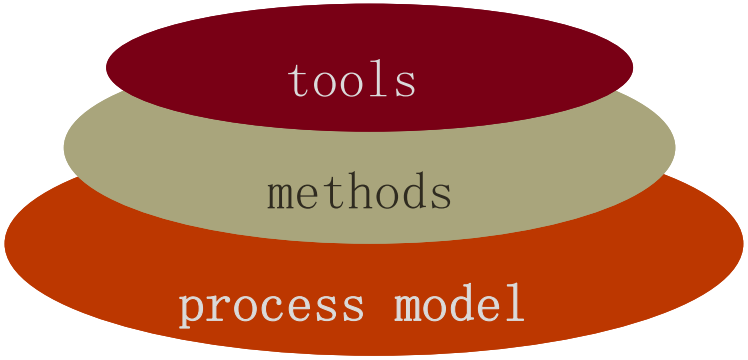
# （2）什么软件工程核心关注点

| 关注点 | 问题 | 手段 |
| --- | --- | --- |
| 质量<br>效率<br>成本<br>变更<br>…… | 如何提高产品质量 | 软件质量保障技术 |
| | 如何提高管理效率，降低成本，缩短上市时间 | 软件工程过程模型 |
| | 如何提高开发效率，解决维护难的问题 | 软件开发方法/语言/工具 |
| | 如何适应用户需求变更 | 演化维护技术 |

**持续关注点：（质量、效率、成本）变更**

# IEEE Standard 610.12-1990

Software Engineering




a “quality” focus

SE工具：（半）自动化支持（CASE）

SE方法：如何做；基本原则

**SE基础：过程**

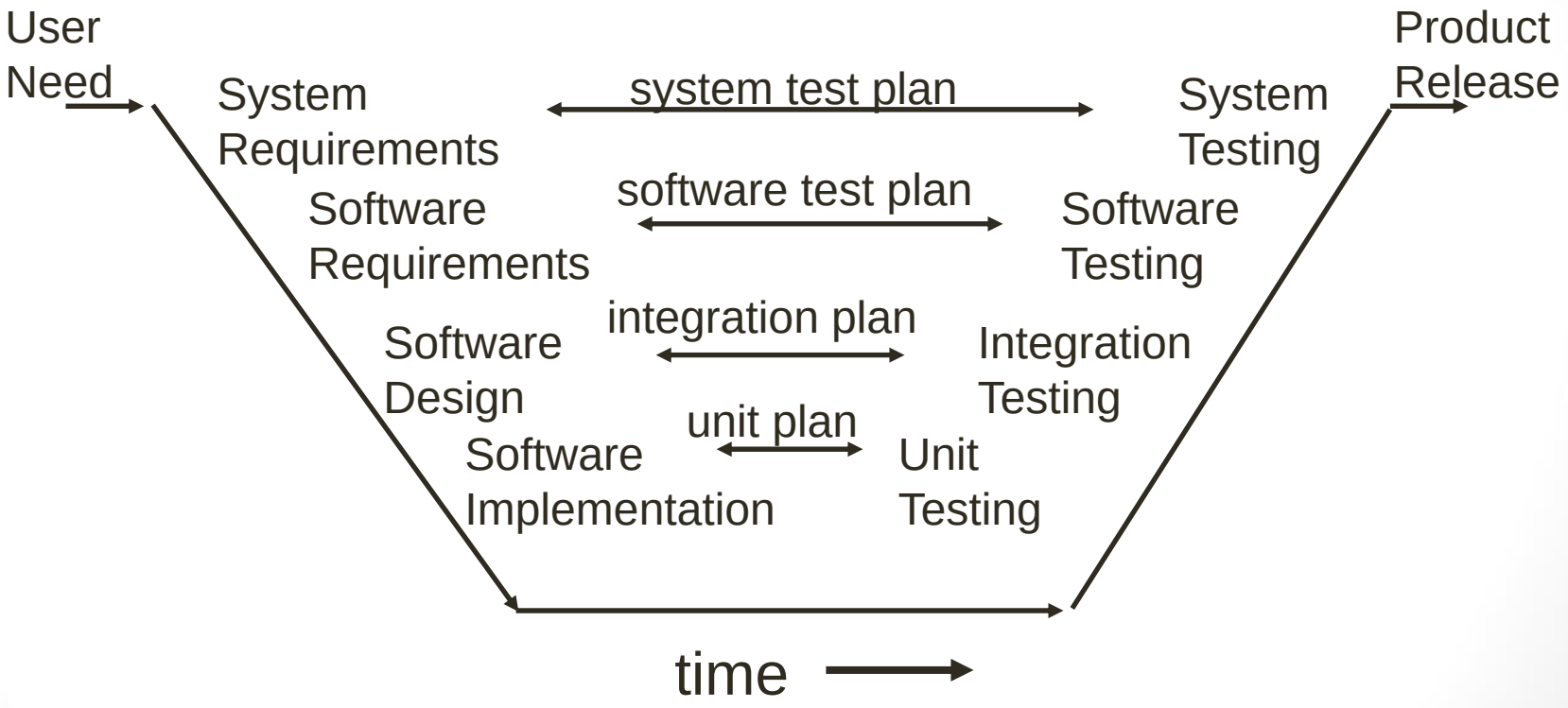
**SE根基：质量关注点**

软件工程基础是过程(process)！！！

# 二、 软件全生命周期质量保障问题

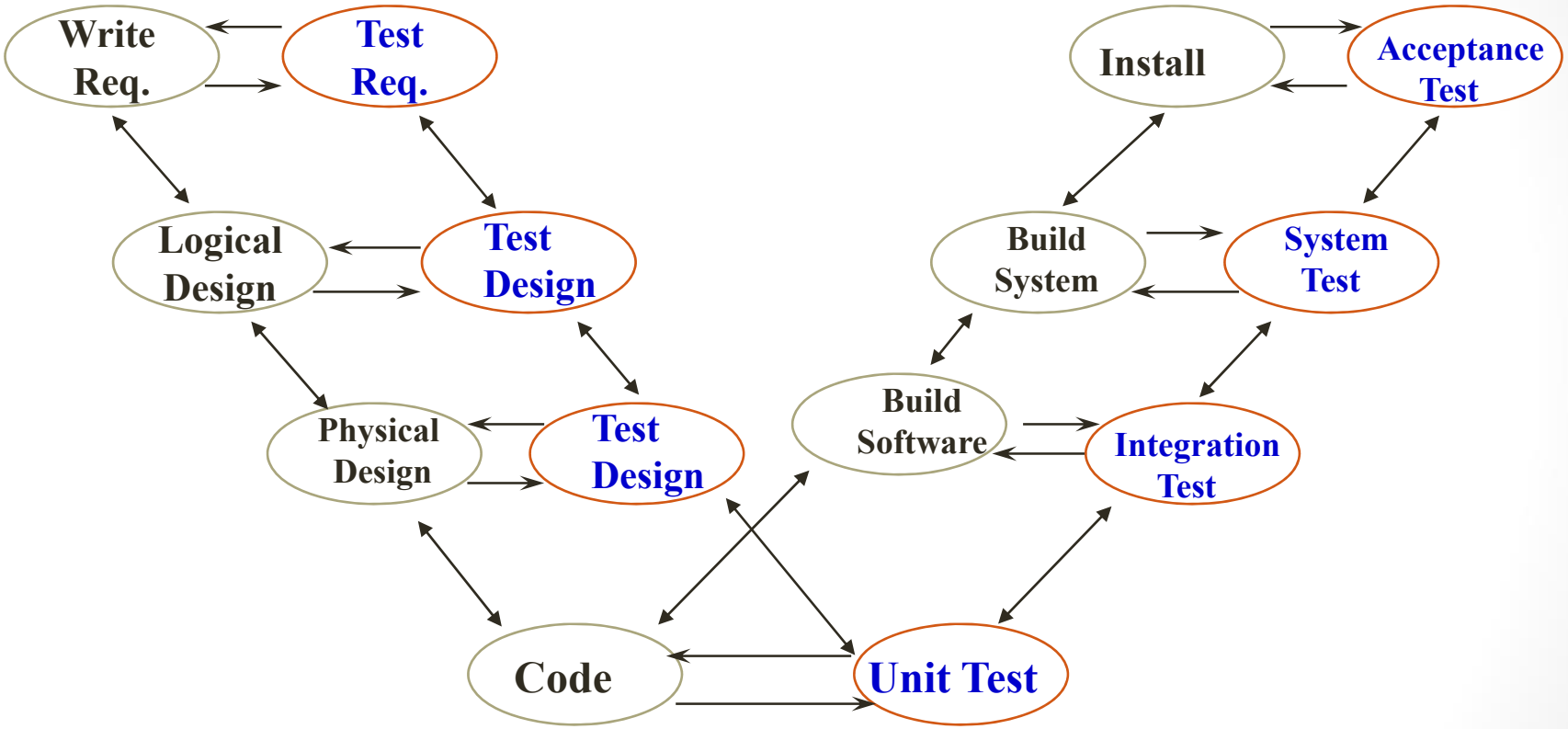
# （1）何谓全生命周期软件质量保障？

•If we rely on testing alone, defects **created first** are **detected last**

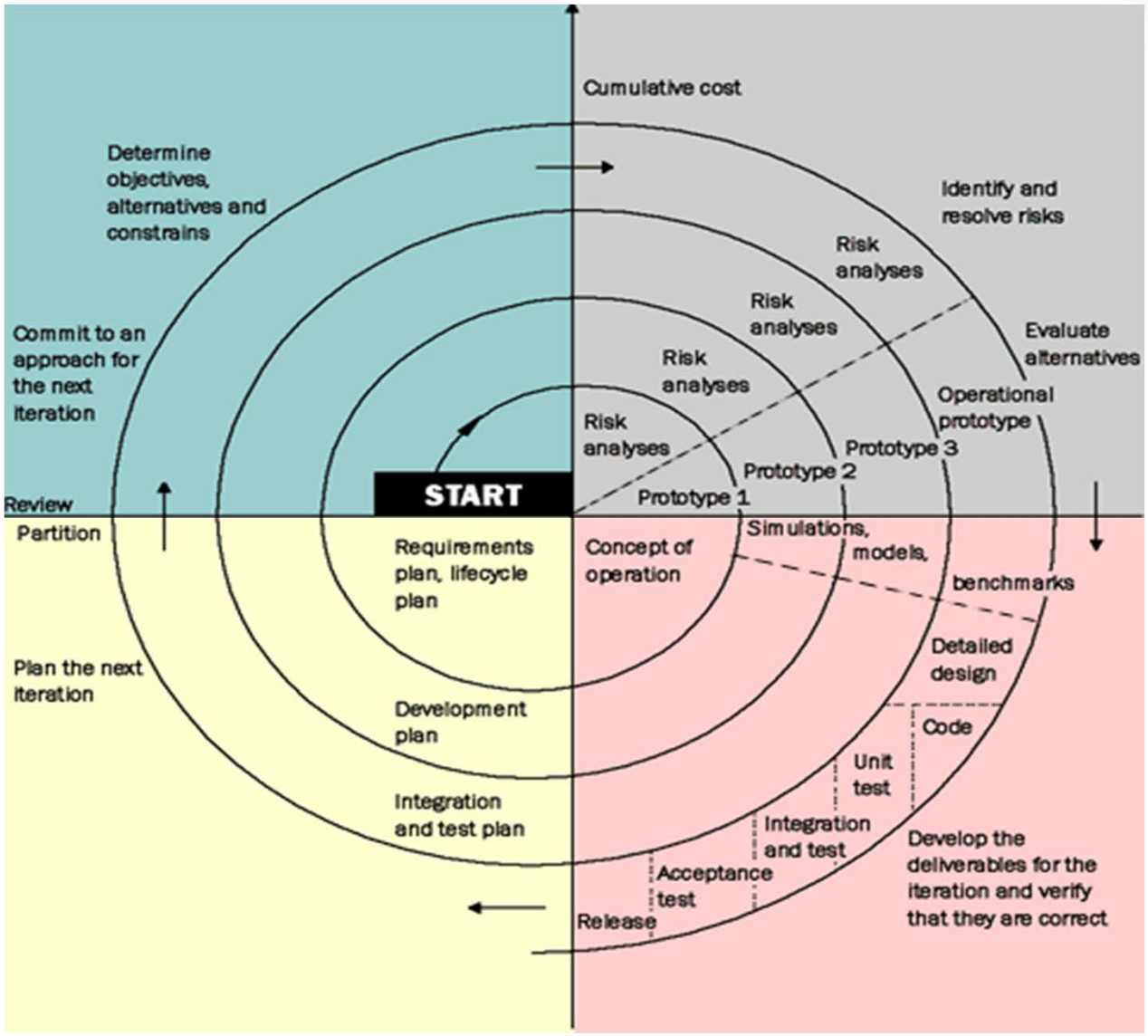



这是软件生命周期后期阶段的质量保障！

# W - Whole Life Cycle

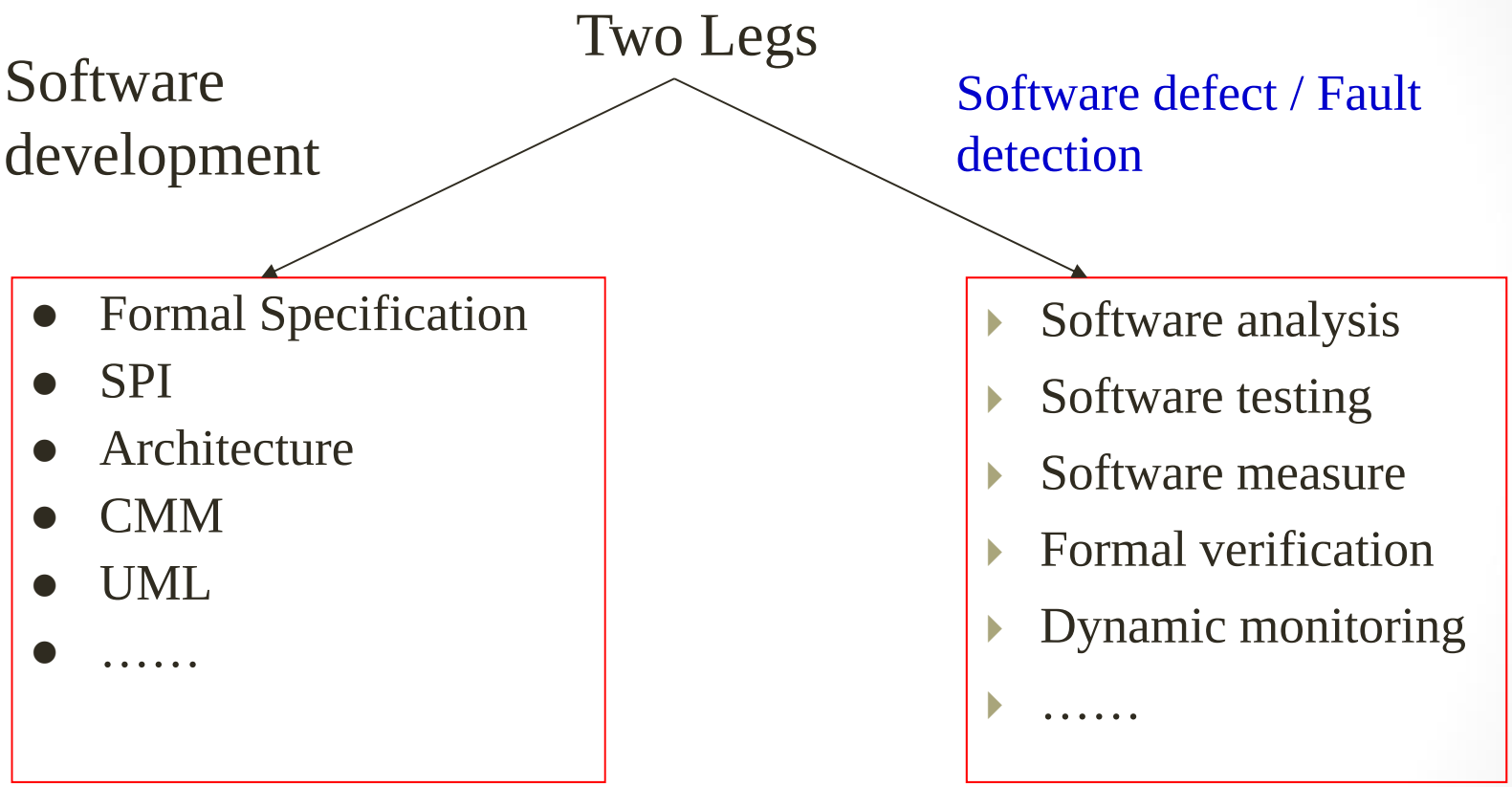



单个生命周期中每个阶段的质量保障！【传统线性开发】

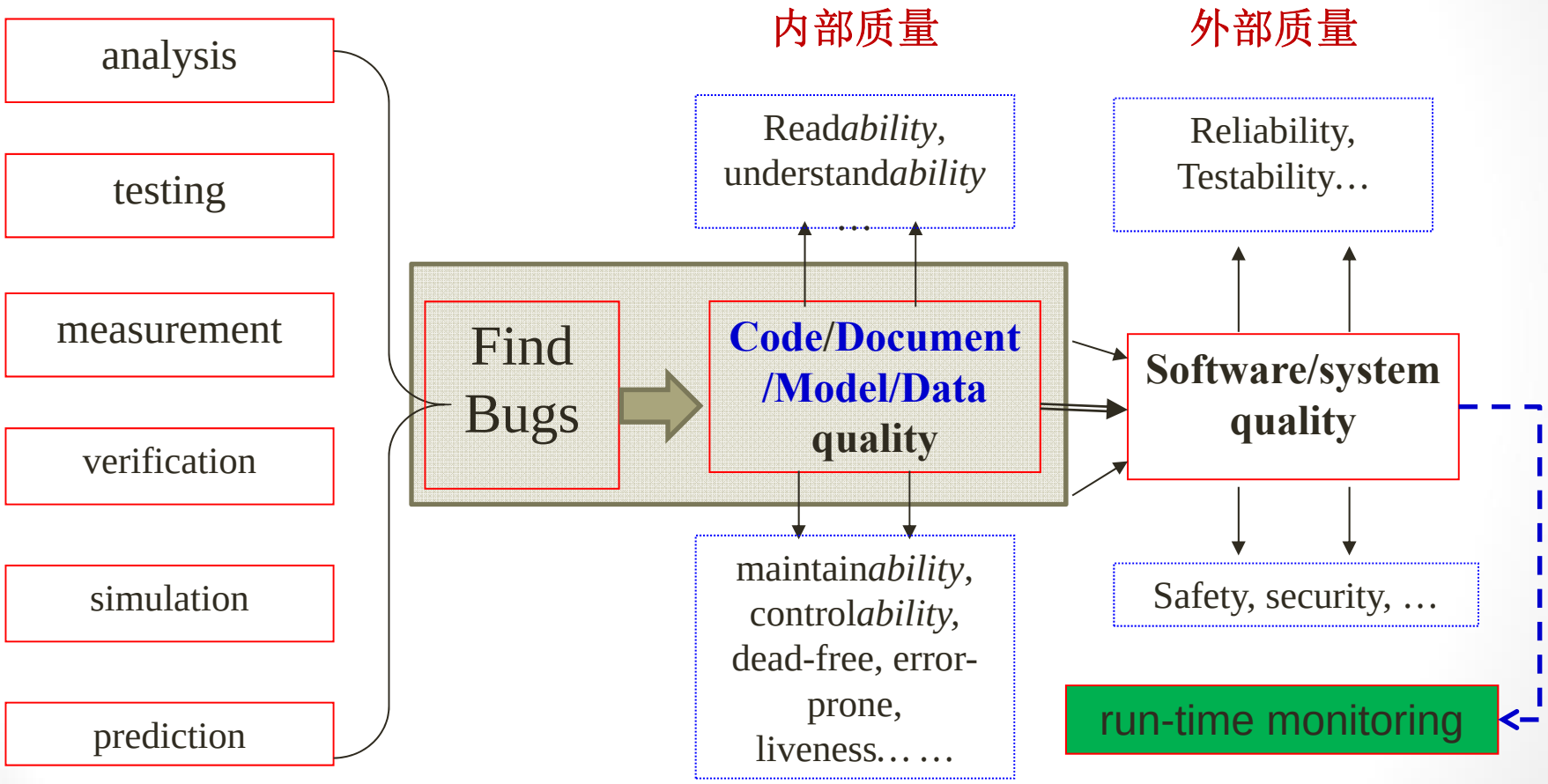



多生命周期中每个生命周期的每个阶段的质量保障！
（非线性开发，敏捷开发的基础理念）

# （2）全生命周期软件质量如何保证？

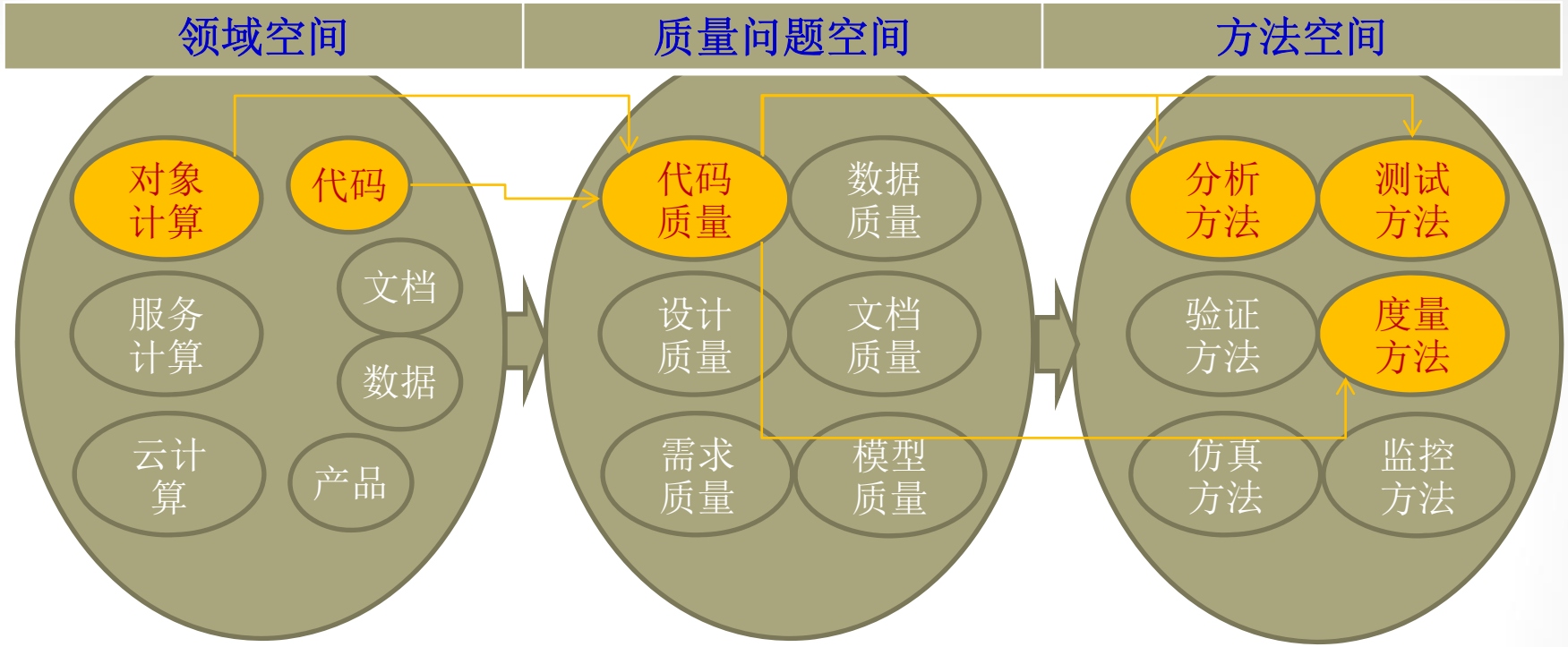


# （3）软件质量与软件缺陷的关系-root cause analysis



# 三、关注点牵引的科研选题

# （1）关注点牵引的科学研究选题




确定选题的一般步骤：

1）找到您最熟悉也是最感兴趣的领域

2）分析最需要、也是您最有可能解决的问题

3）寻找您最有把握的问题解决方法

4）确定最终选题

**问题**：1）有价值；2）未（完全）解决

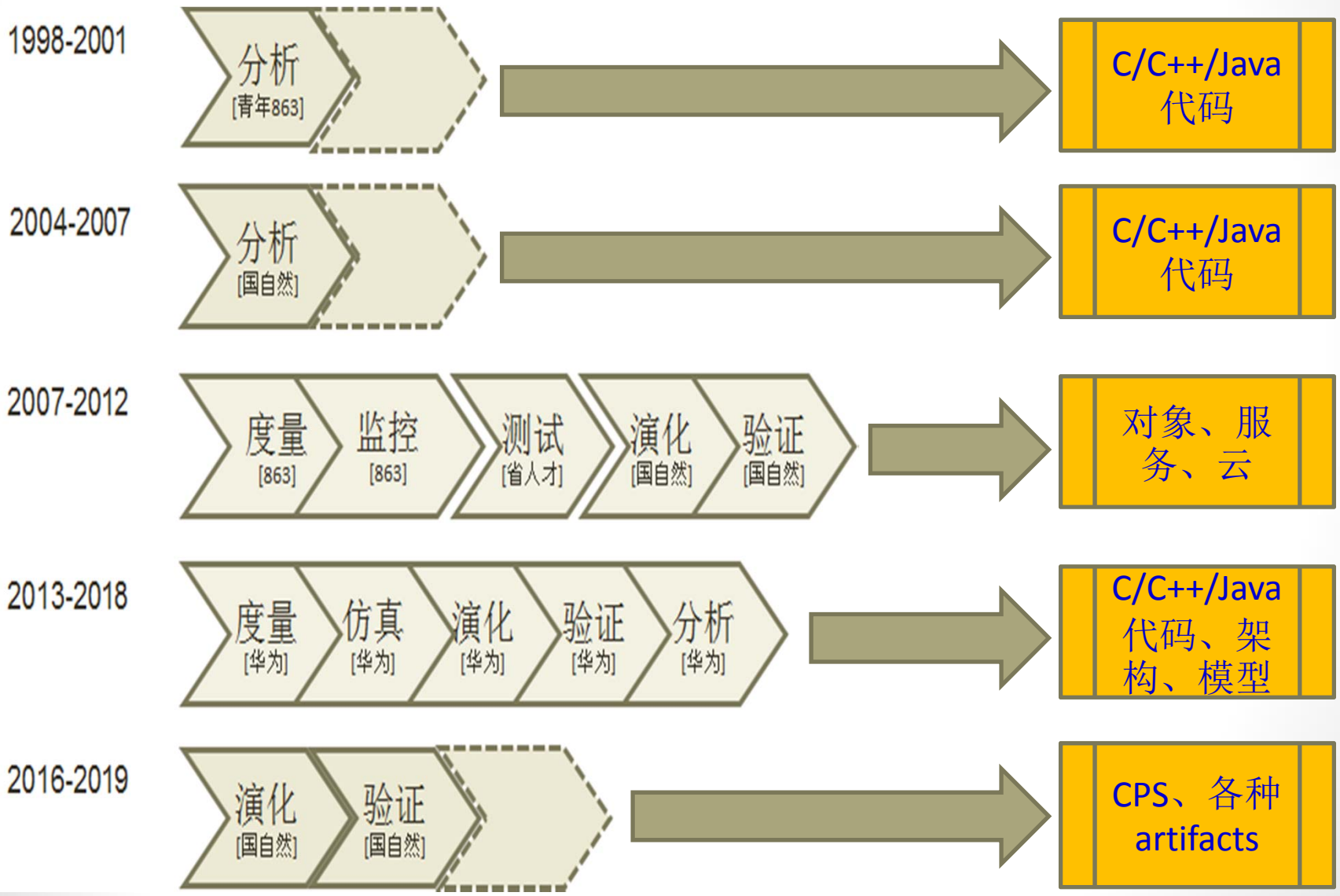
**问题求解**：1）老方法解决老问题；2）新方法解决老问题；3）老方法解决新问题；4）新方法解决新问题

# 代码质量问题的细化—举例

| 代码质量问题 | 问题发现方法举例 | 问题解决方法 |
| --- | --- | --- |
| 复杂性问题 | 圈复杂度、Halstead复杂度 | Modification<br>Modularize<br>Refactor<br>Re-structure<br>Re-architecture<br>Re-configuration<br>Re-compile<br>Reverse Engineering<br>Re-engineering<br>Usage of patterns<br>…… |
| 一致性问题 | Conformance testing，consistency checking | |
| 兼容性问题 | 数据一致性、接口一致性检查 | |
| 可维护性问题 | 可修改性分析、修改影响分析 | |
| 可理解性问题 | 编码规范性检查 | |
| 可扩展性问题 | 接口测试、内聚性度量 | |
| 可集成性问题 | 接口测试、兼容性测试 | |
| 信息流安全问题 | 信息流安全测试 | |
| 可测试性问题 | | |
| ……. | | |

| | |
|---|---|
| Understanding the API usage in Java | IST |
| A new method to encode calling contexts with recursions | SCIENCE CHINA |
| Efficient online cycle detection technique combining with Steensgaard points-to information | SPE |
| Incremental verification of evolving BPEL-based Web composite service | CJE |
| Regression Testing of Web Service: A Systematic Mapping Study | ACM Computing Surveys |
| Profiling Selected Paths with Loops | SCIENCE CHINA |
| PHAT: A Preference and Honesty Aware Trust Model for Web Services Trust Model for Web Services | IEEE TNSM |
| Verifying the Concurrent Properties in BPEL Based Web Service Composition Process | IEEE TNSM |
| FCA-CIA: An Approach of Using FCA to Support Cross-level Change Impact Analysis for Object Oriented Java Programs | IST |
| A Survey of Code-based Change Impact Analysis Techniques | STVR |
| Using Water Wave Propagation Phenomenon to Study Software Change Impact Analysis | ADES |
| Combining Concept Lattice with Call Graph for Impact Analysis | ADES |
| Profiling All Paths: A New Profiling Technique for Both Cyclic and Acyclic Paths | JSS |
| Automatic Test Case Selection for Regression Testing of Composite Service Based on Extensible BPEL Flow Graph | JSS |
| Combining Control Structure and Composition Condition for Web Services Reliability Prediction | CJE |
| HSM-based Change Impact Analysis of Object-Oriented Java Programs | CJE |
| Generating Test Case of Composite Services Based on OWL-S and EH-CPN | IJSEKE |
| A Classification and Comparison of Model Checking Software Architecture Techniques | JSS |
| Timed Property Sequence Chart | JSS |
| Static change impact analysis techniques: A comparative study | JSS |
| MSR4SM: Using topic models to effectively mining software repositories for software maintenance tasks | IST |
| SE-FCA: A Model of Software Evolution with Formal Concept Analysis | CJE |
| Analyzing Impact Rules of Different Change Types to Support Change Impact Analysis | IJSEKE |
| Change impact analysis and changeability assessment for a change proposal: An empirical study | JSS |
| Model and Verification of WS-CDL Based on UML Diagrams | IJSEKE |

# （2）关注点驱动的项目申报



# 東南大學 软件工程研究所

INSTITUTE OF SOFTWARE ENGIEERING,SOUTHEAST UNIVERSITY

东南大学软件工程研究所（ISEU，Institute of Software Engineering, Southeast University）是一个专注于软件工程理论和实践的学术机构，由李必信教授于 2012 年 1 月 8 日发起成立。ISEU 是在软件工程理论与实践研究室的基础上、融合软件分析与测试和面向服务的软件工程等研究室共同建立起来的，现有教师 10 名，博士后 2 名，博士生 10 名，硕士生 30 名，外国留学生 6 名。

## 研究所的任务

全面开展软件工程及其相关领域的学术研究，出高水平的研究成果；全面开展软件工程实践探索，使之直接服务于社会、造福于人民；培养软件工程领域的高级研究人才和工程技术人才，为我国软件技术研究和软件产业发展做出我们的贡献。

东南大学软件工程研究所近年来承担着 20 多项国家级和省部级的科研项目（包括国家 863 高技术计划项目、国家自然科学基金项目、江苏省自然科学基金项目、教育部博士点基金项目等），同时也承担了 10 多项横向合作项目的研究与开发工作。研究所近年来取得了一系列令人满意的成果：出版高水平学术专（译）著 5 部，发表论文 160 余篇，被 SCI/EI 检索超过 150 篇次，授权发明专利 30 项，软件著作权登记 14 项；为社会输送博士毕业生 10 人，硕士毕业生 80 多人。

研究所承担了计算机科学与技术学科、软件工程学科大量的基础课和专业课的教学工作，通过教学与科研的紧密结合，促进教学质量不断提高。**ISEU 希望和世界范围内软件工程领域的研究者和实践者建立广泛的交流与合作。**

## 主要研究方向

（1）软件工程的基础理论和实践；（2）软件分析、测试和验证；（3）软件演化、维护和变更影响分析；（4）软件设计模式和架构模式识别；（5）软件架构演化、评估和重构；（6）软件安全性和可靠性，等等。

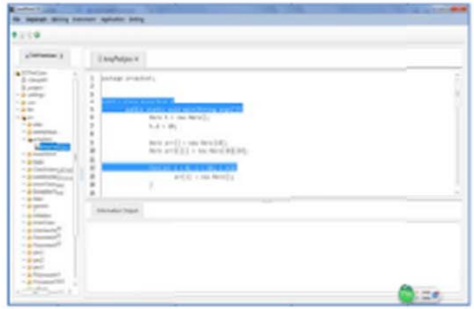
## 代表性原型工具简介

**1.软件架构可演进性评估系统 v1.0**

由 ISEU 开发的软件架构可演进性评估原型系统，用于辅助分析当前架构的演进能力，对软件架构整体指标和组件指标进行评估，模拟多层次指标的状况和趋势，指示软件架构的演进能力和演进效果。具体功能：面向架构的逆向工程（从四个维度自底向上进行架构恢复）、架构演进性度量、架构演进原则达成性度量、架构坏味道识别、架构模式识别、质量驱动的面向模式架构重构等。

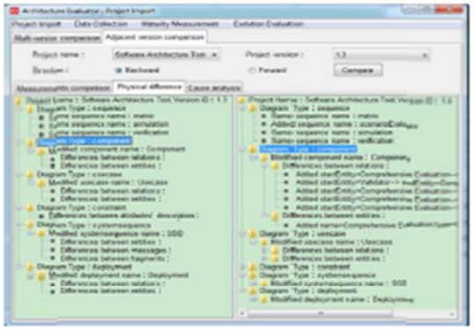
**2.切片工具 MyJavaSlicer v1.0**

由 ISEU 开发的一项自动化 Java 程序切片工具，能够对 JDK 7 开发的面向对象、并发和分布式的 Java 工程进行静态和动态切片；同时，也具备增量切片、有条件切片、分层切片等功能。另外，基于 MyJavaSlicer，还可以进行 Java 故障定位、修改影响分析、可复用功能提取以及 API 度量等。



**3.软件演化度量评估系统 v1.0**

由 ISEU 开发的用于软件演化度量和评估的原型系统。该系统目前具有软件架构演化度量和评估、源代码演化度量和评估两大功能，具体包括：架构演化度量比较、物理差异性分析、引起度量结果差异的原因分析和定位；软件演化对代码复杂度（圈复杂度/Halstead 复杂度）的影响、对软件可集成性、可替换性、可兼容性的影响，演化成本预测等。



**4.软件架构仿真、评估与验证系统 v2.0**

由 ISEU 开发用于软件架构仿真、评估和验证的原型系统，按照自定向下方式对基于 UML 设计模型的软件架构结构化文档进行仿真、评估和验证，为架构设计提供直观可靠的评价结果，指导架构设计与维护。具体包括：架构合格性检查、架构属性度量、架构仿真、架构验证、场景评估、知识库构建和综合评价等。


谢谢大家！