

软件工程

Software Engineering

龙 军

jlong@csu.edu.cn 18673197878

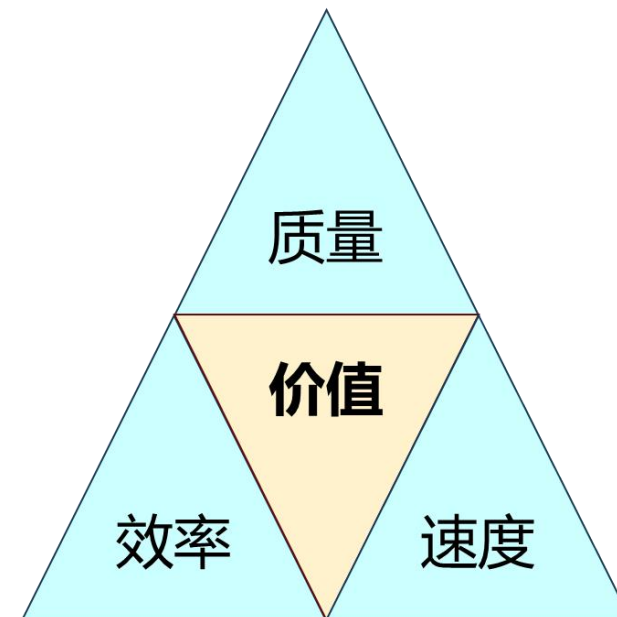
计算机学院 | 大数据研究院

软件定义世界（SDX）

软件是信息技术之魂、网络安全之盾、经济转型之擎、数字社会之基。加强特色化软件产业人才培养，是建设教育强国、制造强国、网络强国，实现中国软件高质量发展的必然要求和重要支撑。

课程定位

- 软件工程是计算机软件相关专业的一门重要课程，它对于培养良好的软件工程师素质，提高**软件研发能力**具有重要的意义。



在**时间**和**成本**约束下，开发出**高质量的**、**有价值** 的软件

How?

系统化的方法

敏捷的过程

抽象和分解

创新

软件工具

团队协同

……

课程目标

- 能够分析**软件需求**，运用软件工具进行软件需求建模。
- 能够根据软件需求，进行**软件设计**，建立软件设计模型，并进行**软件实现**。
- 能够根据需求与对软件进行**测试**，并通过测试评估软件开发的**质量**。
- 能够进行**软件过程和项目的策划与管理**，团队协同，完成软件开发。

先修课程要求

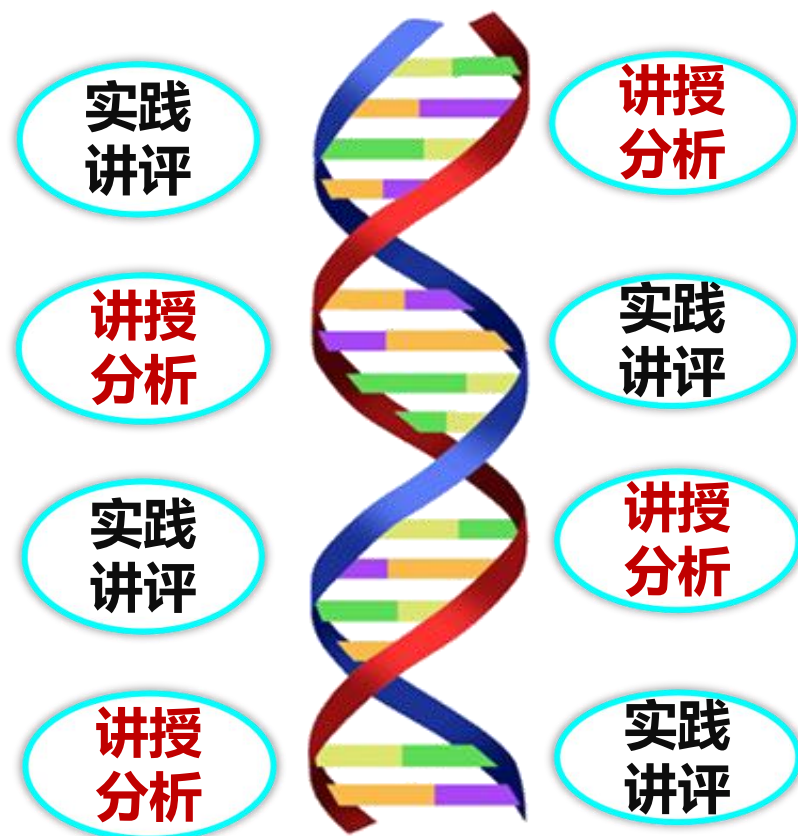
- 熟悉一门高级程序设计语言，例如C++，Java，Python等。
- 熟悉数据结构与算法，具备一定的编程能力。
- 注：不需要应用开发经验。

课程内容

- 第1章 软件工程引论
- 第2章 软件过程
- 第3章 软件工程模型与方法
- 第4章 软件需求
- 第5章 软件架构设计
- 第6章 软件详细设计
- 第7章 用户界面设计
- 第8章 编码
- 第9章 软件测试
- 第10章 软件运维
- 第11章 软件项目管理

案例教学 + 大作业实践

如何学好这门课?



讲授-老师

实践-学生

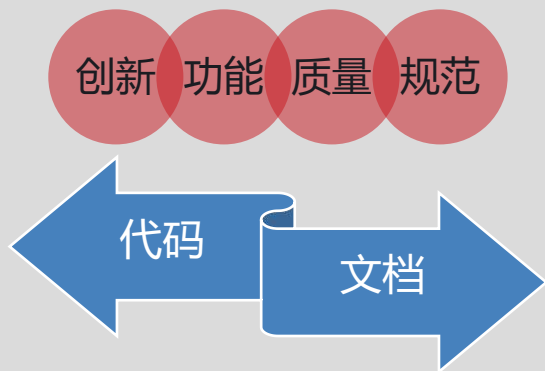
实践、实践、再实践!

考核方法

- 平时成绩 40% (低于10分, 取消考试资格)

- 小组分 20%

- 大作业的课外作业
- 大作业的2次迭代评审
- 大作业的最后验收



小组分按贡献度分配到个人

- 个人分20%

- 单人课堂作业 (5%)
- 考勤 (10%)
- 课堂表现 (5%)

- 期末考试 60%

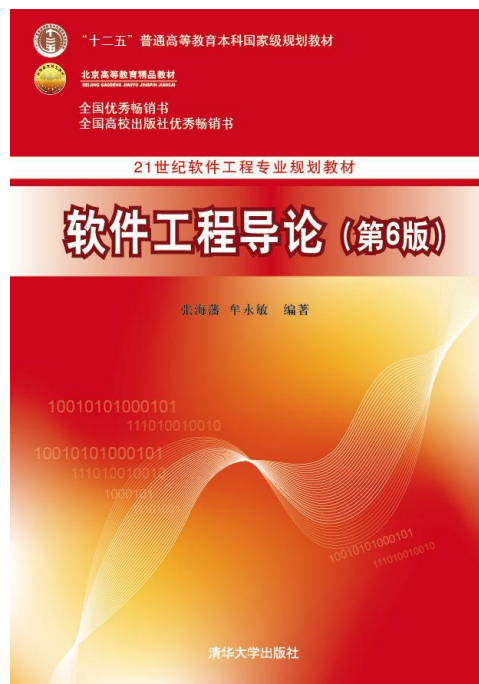
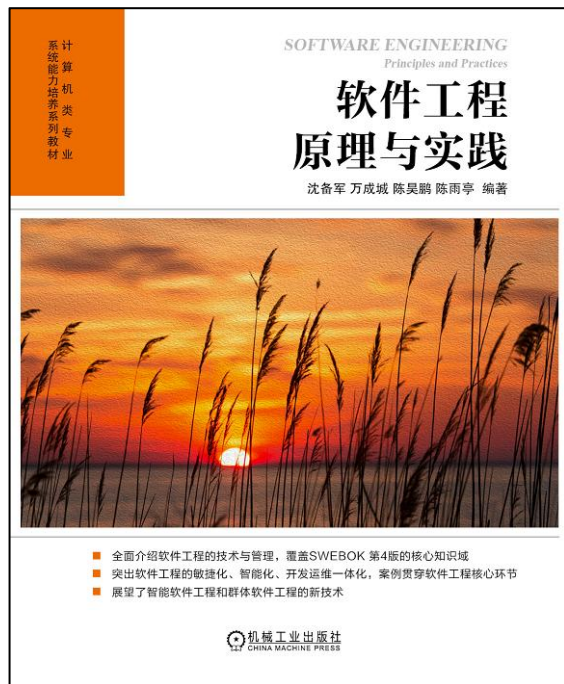
- 开卷笔试

课程教材

□ 课程教材

- 软件工程原理与实践，机械工业出版社，沈备军等编著 ISBN：978-7-111-73944-9
- 软件工程导论(第6版)，清华大学出版社，张海藩等编著，ISBN：978-7-302-33098-

1



大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

☀ 01-什么是软件

软件的定义、特性、作用和发展

挑战和问题

02-什么是工程

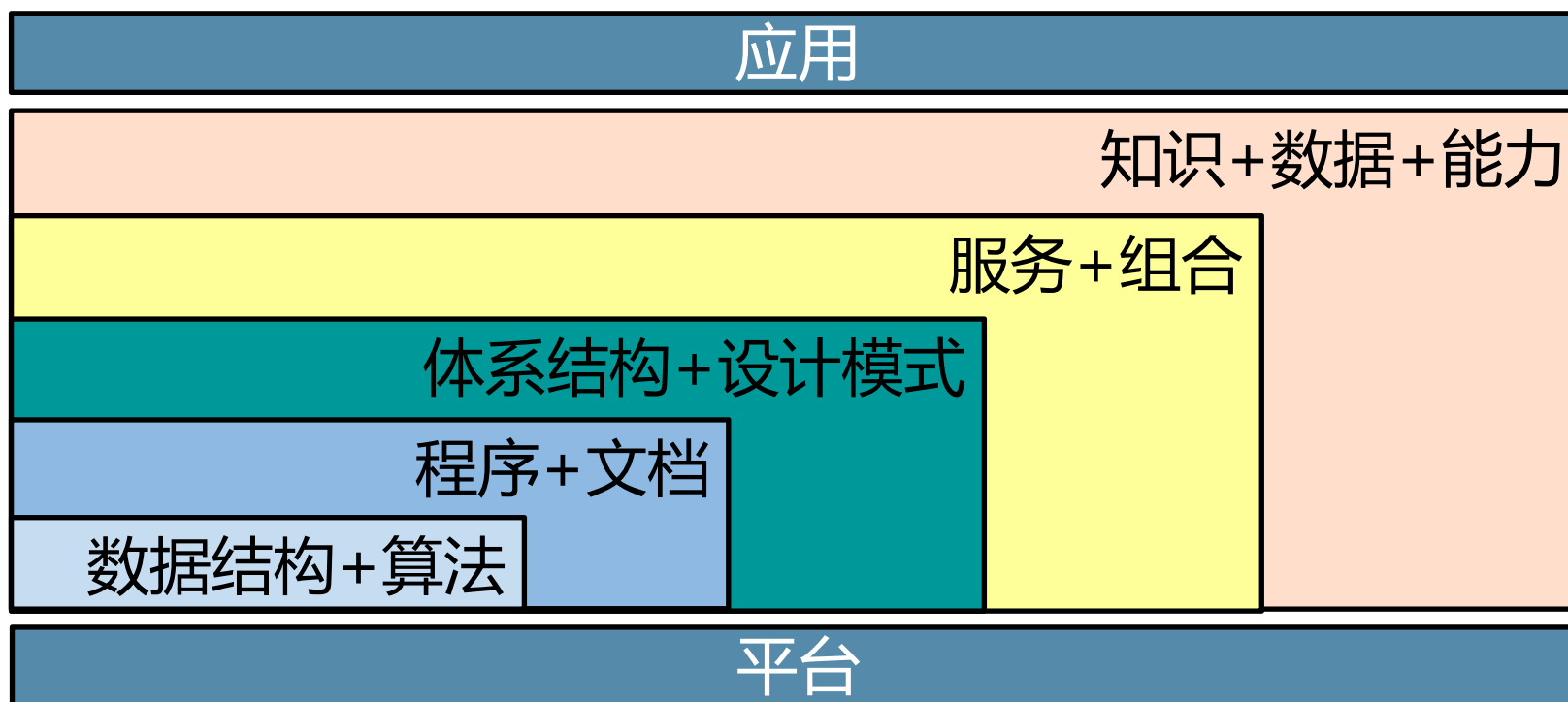
03-什么是软件工程

04-AI时代的软件工程

@第1章.教材

什么是软件

- **定义：** 软件是计算机系统中与硬件相互依存的，运行在硬件之上面向特定应用目标实现的解决方案。
- **特性：** 软件是人类智力活动产品，是逻辑产品而不是物理的产品。
- **组成：** 软件 = 程序（program）+ 相关数据（data）+ 说明文档（document）



平台和应用是软件发展的外在驱动力

软件特征

□ 软件开发不同于硬件设计

- 与硬件设计相比，软件更依赖于开发人员的业务素质、智力，以及人员的组织、合作和管理。同时对硬件而言，设计成本往往只占整个产品成本的一小部分，而软件开发占整个产品成本的大部分。

□ 软件生产不同于硬件制造

- 硬件设计完成后就投入批量制造，制造也是一个复杂的过程，其间仍可能引入质量问题；而软件成为产品之后，其制造只是简单的拷贝而已。软件的仓储和运输也非常简单。

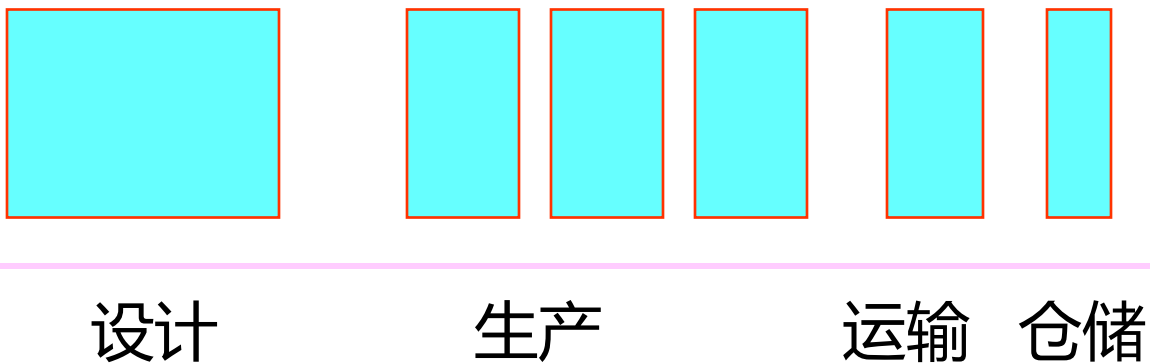
□ 软件维护不同于硬件维修

- 硬件在运行初期有较高的故障率，在缺陷修正后的一段时间中，故障率会降到一个较低和稳定的水平上。随着时间的改变，故障率将再次升高，这是因为硬件会受到磨损等损害，达到一定程度后就只能报废。软件是逻辑的而不是物理的，虽然不会磨损和老化，但在使用过程中的维护却比硬件复杂得多。如果软件内部的逻辑关系比较复杂，在维护过程中还可能产生新的错误。

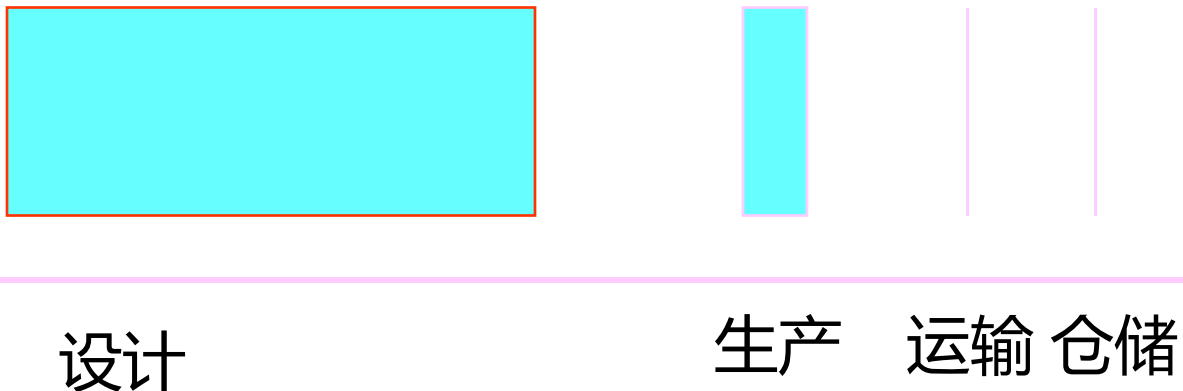
软件是被设计的

做世界上最好的软件产品!

制造业:

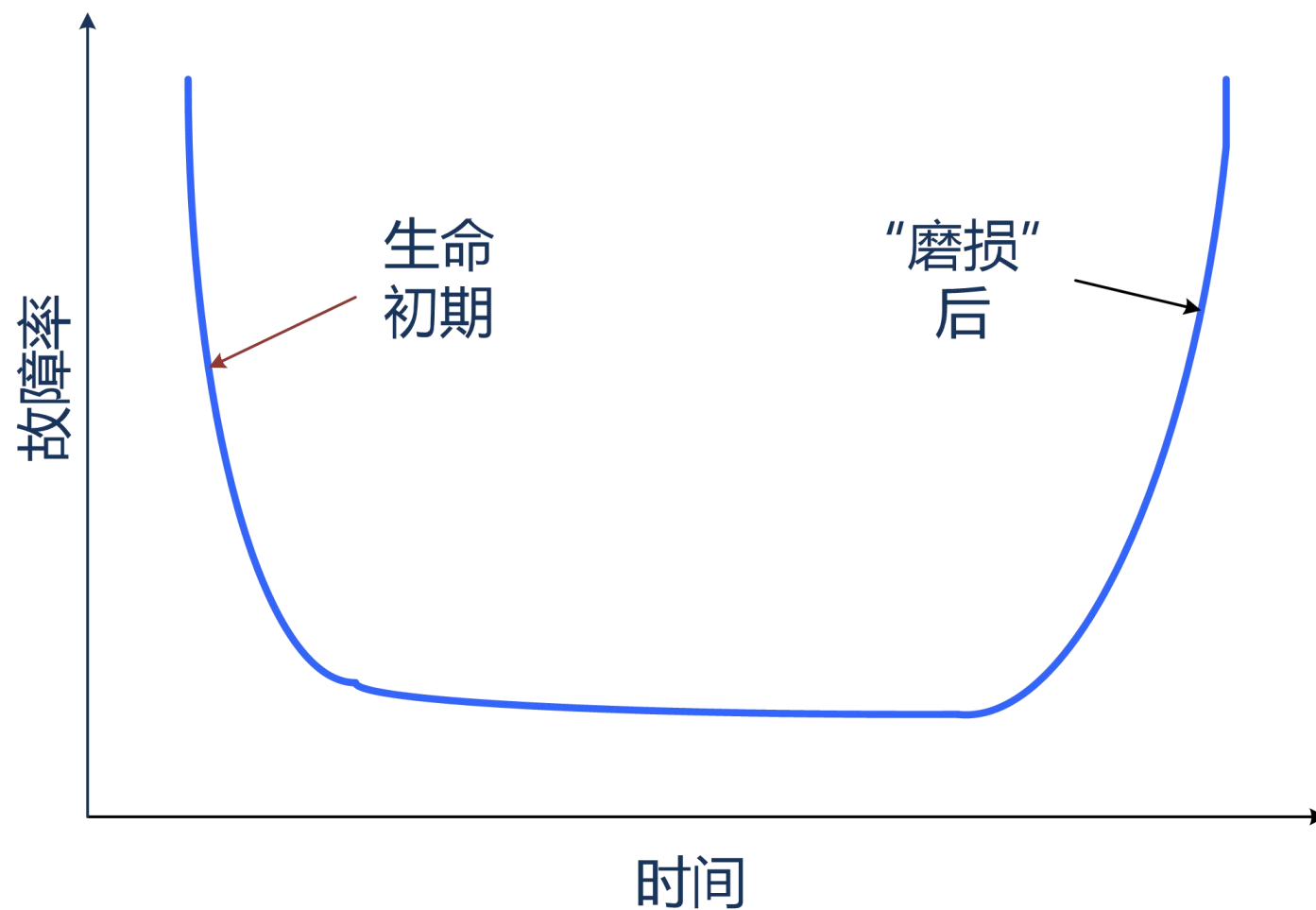


软件业:

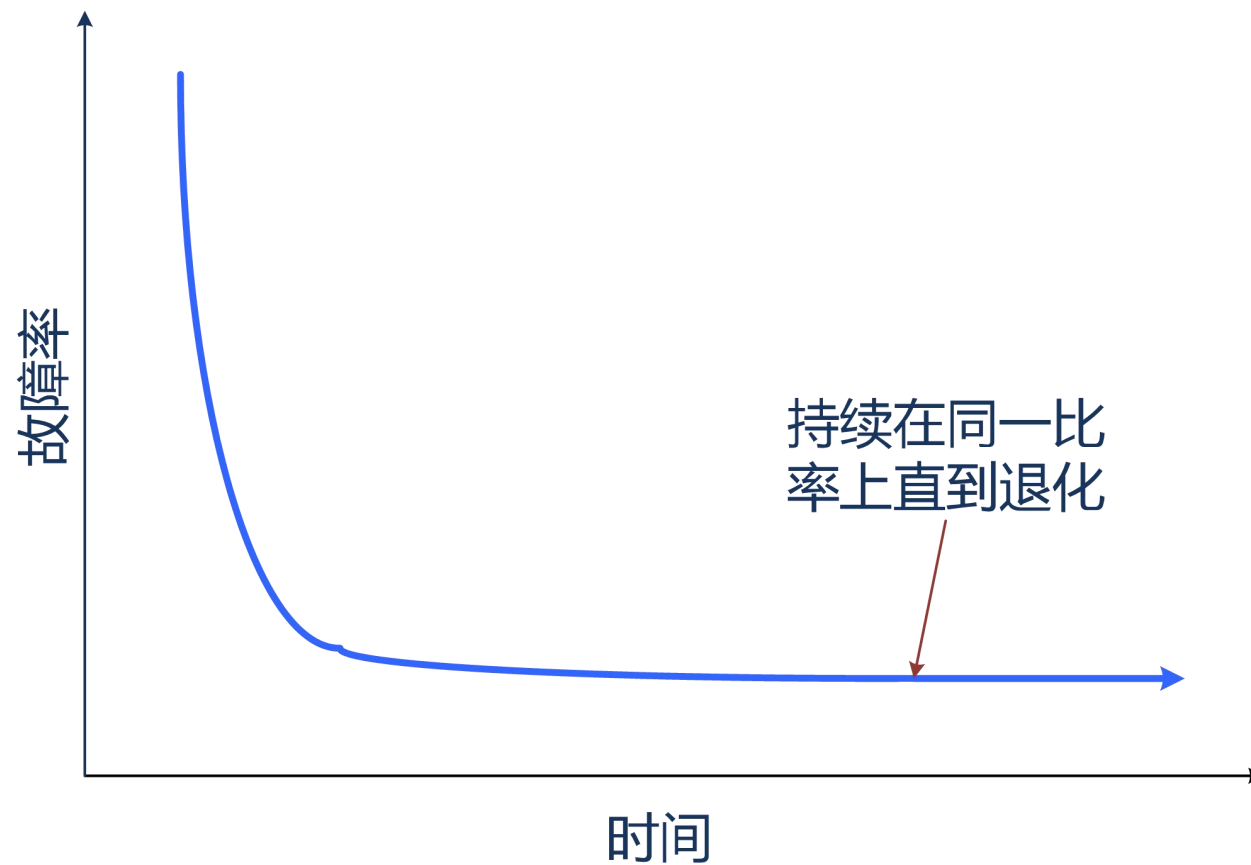


WPS案例

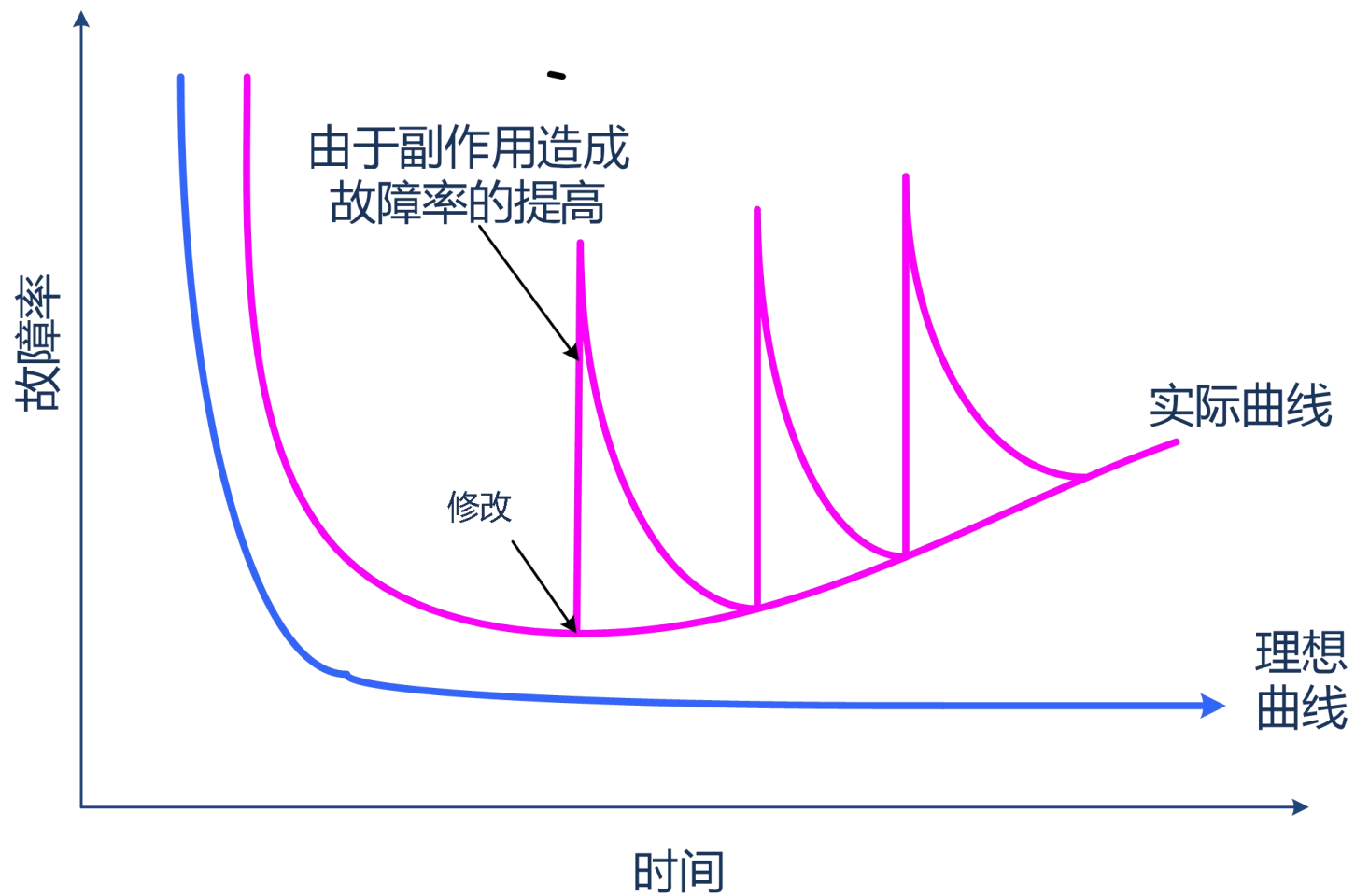
硬件的故障率曲线（浴缸曲线）



软件的故障率曲线(理想情况下)



软件的故障率曲线(实际情况下)



软件发展阶段

软件技术呈现从“工具”和“平台”到引领的转变，软件产品形态和商业模式演进：

□ 硬件附属阶段（大型机和小型机时代）

- 1960s以前，没有“软件”概念，直接通过机器语言和汇编语言编写程序直接操作硬件来运行。
- 1960s以后，软件从硬件分离，成相对独立制品，硬件占据绝对主体地位，软件作为硬件附属产品存在，没有独立商业形态。

□ 独立软件产品阶段（PC机时代，信息化1.0）

- 单机应用为主要特征的数字化阶段，卖软件许可证（license）成为一种新型商业模式。

□ “软件即服务”阶段（“互联网+”时代）

- 软件从单机计算环境，向以网络在线应用为主要特征的网络化阶段（信息化2.0）

□ 软件定义阶段（数字经济时代）

- 对传统物理世界基础设施和社会基础设施进行重塑，通过软件定义方式赋予新的能力和灵活性。

软件无处不在，软件定义一切

软件“吞噬”世界！



Marc Andreessen
NetScape 创始人

1994年第一届WWW大会与Tim-Berners Lee等五人一起被选入第一届万维网 Hall of the Fame



Software **Eats** the World!

人类文明运行在软件之上！



Bjarne Stroustrup
C++语言发明人

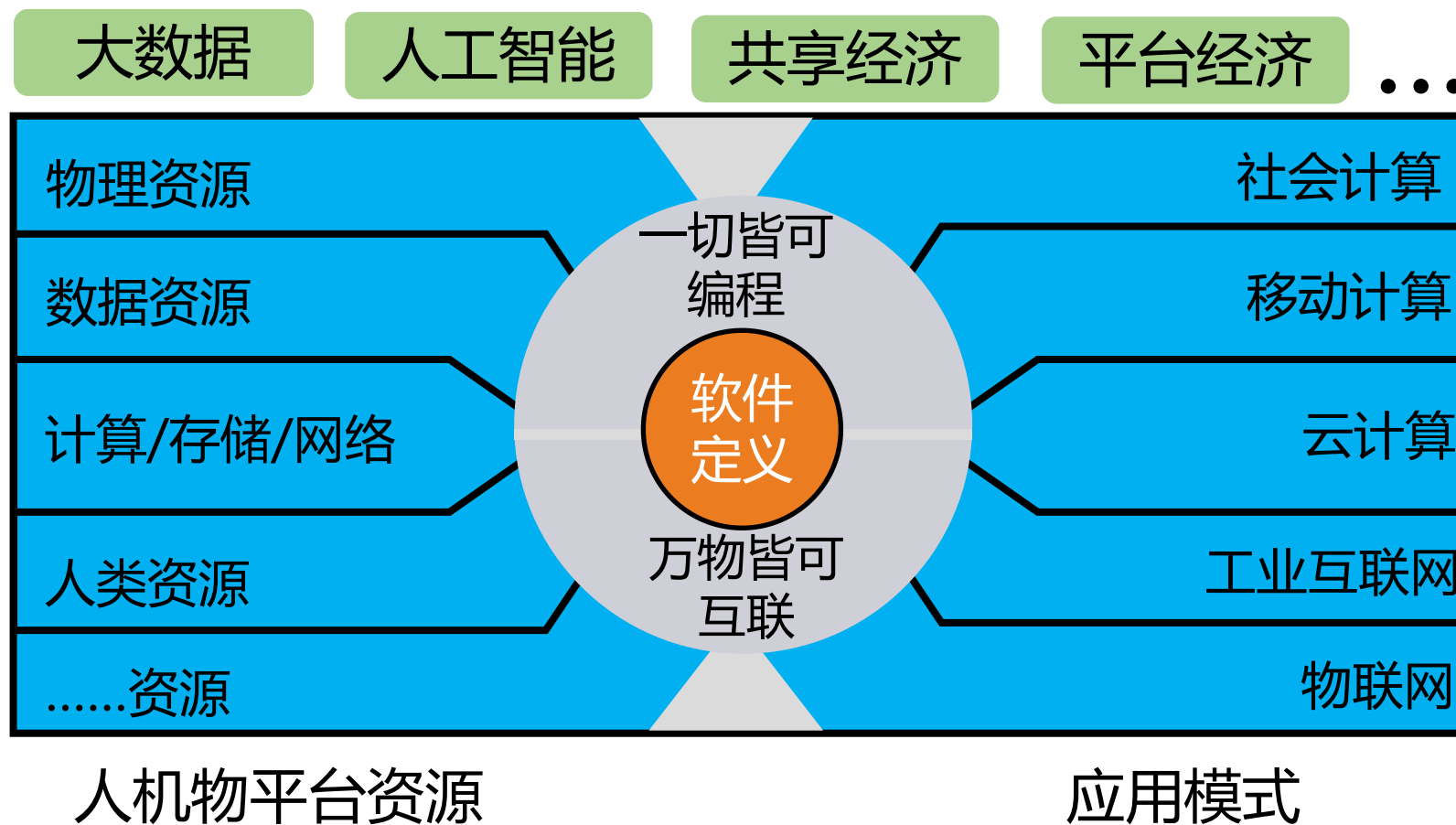


IEEE Computer, January,
2012, pp 47-58

Our **civilization** runs on software

软件成为社会基础设施

无所不在的软件是支撑人类社会运行和人类文明进步中发挥重要的“基础设施”作用。



大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论



01-什么是软件

软件的定义、特性、作用和发展

挑战和问题

02-什么是工程

03-什么是软件工程

04-AI时代的软件工程

@第1章.教材

软件面临的挑战

- ❑ 软件复杂性挑战：网络应用，涉及多种硬件、多种操作系统、多种编程语言，逻辑复杂，新技术层出不穷。
- ❑ 软件大规模挑战：软件规模（代码行）不断扩大，研发人员众多。
- ❑ 软件高质量挑战：高完全性、高可靠性、高性能、高易用性、可配置性。
- ❑ 不确定性挑战：软件项目面临需求和技术的易变性、不确定性、复杂性和模糊性，应用环境从静态、封闭、可控变为动态、开放、难控。
- ❑ 进度和敏捷开发的挑战：项目进度要求紧、需求难以确定。
- ❑ 遗留系统（Legacy System）的挑战：大量遗留系统需要集成和复用。
- ❑ 分散团队的协同挑战：团队地理位置分散。

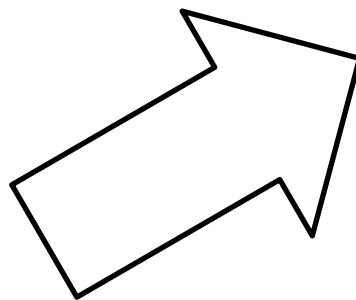
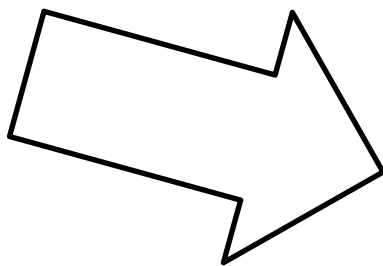
软件危机

开发
手段

作坊式和手工式的
个体编程和创作

实际
需求

大规模和复杂
软件开发要求



软件危机

- 进度经常延迟
- 质量无法保证
- 成本超出预算
- 软件维护困难
- 失败风险很大



软件危机的典型表现

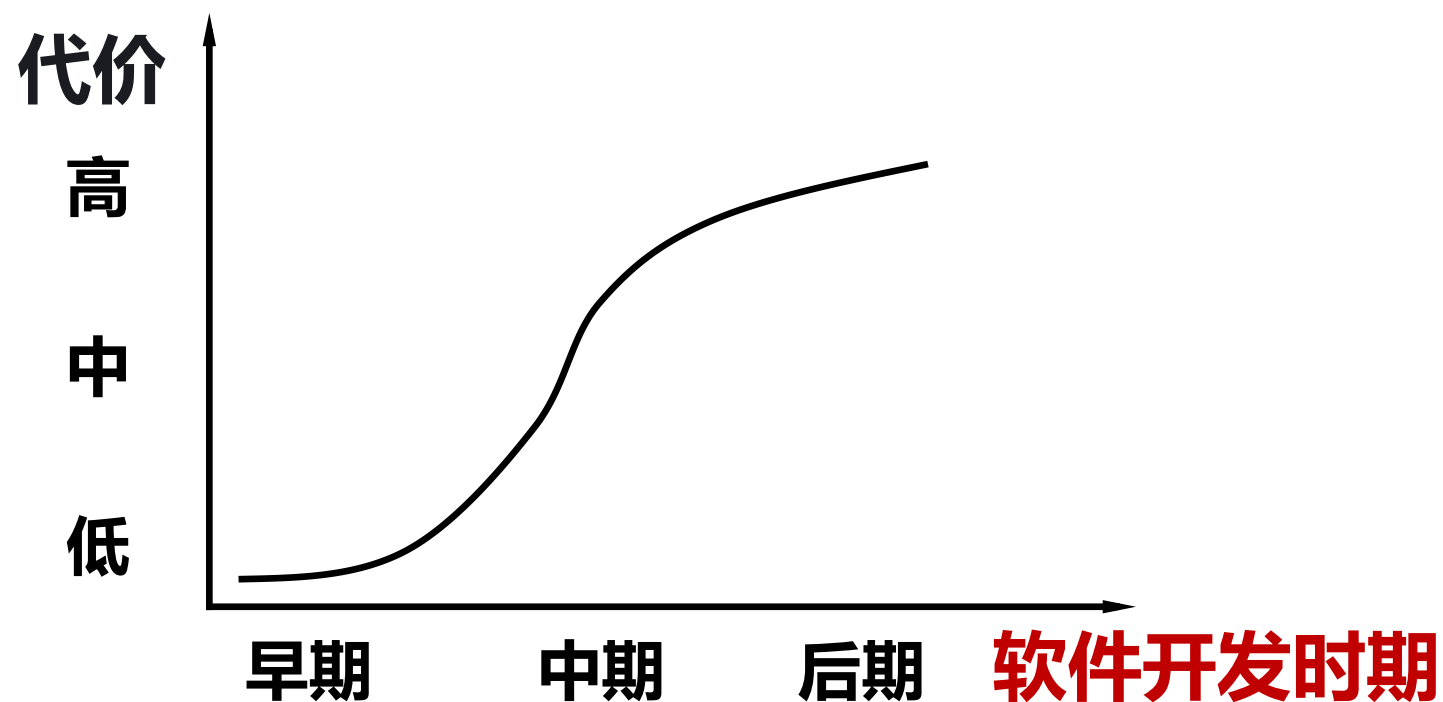
- ❑ 对软件开发成本和进度的估计常常很不准确；
- ❑ 用户对完成的软件系统不满意的现象经常发生；
- ❑ 软件产品的质量往往靠不住；
- ❑ 软件常常是不可维护的；
- ❑ 软件通常没有适当的文档资料；
- ❑ 软件成本在计算机系统总成本中所占的比例逐年上升；
- ❑ 软件开发生产率提高的速度跟不上计算机应用的发展趋势。

产生软件危机的根源

- ❑ **软件本身特点造成** / 对软件这样一类**复杂和特殊系统**的认识不清
 - 软件是新生事物，对其特点、规律性和复杂性认识不够
- ❑ **软件开发与维护的方法不正确** / 没有找到支持软件系统开发的**有效方法**
 - 基础理论、关键技术、开发过程、支撑工具等缺乏
 - 忽视软件需求分析
 - 认为软件开发就是写程序并使之运行
 - 轻视软件维护
- ❑ **缺乏成功软件开发实践**以及相应的**开发经验**
 - 系统总结、认真分析、充分借鉴、吸取教训

如何解决软件危机，化危为机？

软件修改代价随时间变化的趋势



引入同一修改的代价随时间变化的趋势

在软件开发的阶段进行修改需要付出的代价很不相同

当前软件实践的问题

- ❑ **软件直到测试前仅仅是忽略质量的现代技术。** 典型地说，软件工程师
 - 没有计划他们的工作
 - 匆匆地走过需求和设计
 - 在编码时再进行设计
- ❑ **这些实践引入了大量的缺陷**
 - 有经验的工程师每7-10行代码就引入一个缺陷
 - 平均中等规模的系统存在着上千个缺陷
 - 这些缺陷的大多必须靠测试发现
 - 通常要花去一倍以上的开发时间
- ❑ **目前大多数的工作方式还象40年前一样 - - “code-fix”**

软件项目成功率的调查

| | 2011 | 2013 | 2015 | 2017 |
|------------|------|------|------|------|
| Successful | 29% | 31% | 29% | 36% |
| Challenged | 49% | 50% | 55% | 45% |
| Failed | 22% | 19% | 17% | 19% |

Standish Group Chaos Report

软件事故

- ❑ 1981年，由计算机程序改变而导致1/67的时间偏差，使航天飞机上的5台计算机不能同步运行。这个错误导致了航天飞机发射失败。
- ❑ 1986年，1台Therac25机器泄露致命剂量的辐射，致使两名医院病人死亡。原因是一个软件出现了问题，导致这台机器忽略了数据校验。
- ❑ 2016年，区块链业界最大的众筹项目TheDAO遭到攻击，导致300多万以太币资产被盗，原因是其智能合约中splitDAO函数有漏洞。
- ❑ 2018年，印尼狮航一架波音737 MAX 8客机途中坠落，189人罹难，失事原因为软件设计缺陷，飞机的迎角传感器“数据错误”触发“防失速”自动操作，导致机头不断下压，最终坠海。
- ❑ 2018年至今，无人车在自动驾驶时出现了多次交通事故，导致车毁人亡。原因是机器学习模型未能及时正确识别出前方的情况并做出决策。

软件工程师是否需要发执照？

解决软件危机的途径

- 推广使用在实践中总结出来的开发软件的成功技术和方法，并研究探索更有效的技术和方法；
- 开发和使用更好的软件工具；
- 良好的组织管理措施。

※ 为了解决软件危机产生的问题，软件工程与方法学逐渐形成，然后出现了两个相互相承又各有侧重的学科：

1) 软件工程学：主要应用工程的方法和技术研究软件开发与维护的方法、工具和管理的一门交叉学科。

2) 程序设计方法学：主要应用数学的方法研究程序的性质以及程序设计的理论和方法的学科。

软件开发是...?

□ 软件开发

■ 是一门艺术?

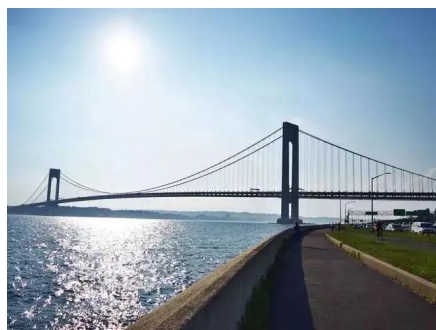
追求美：优秀的软件好用且美，像艺术品。

■ 是一门科学?

追求真：具有数理性质，可以用数学的方法进行分析和细化。

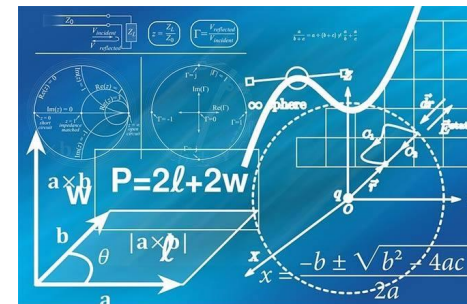
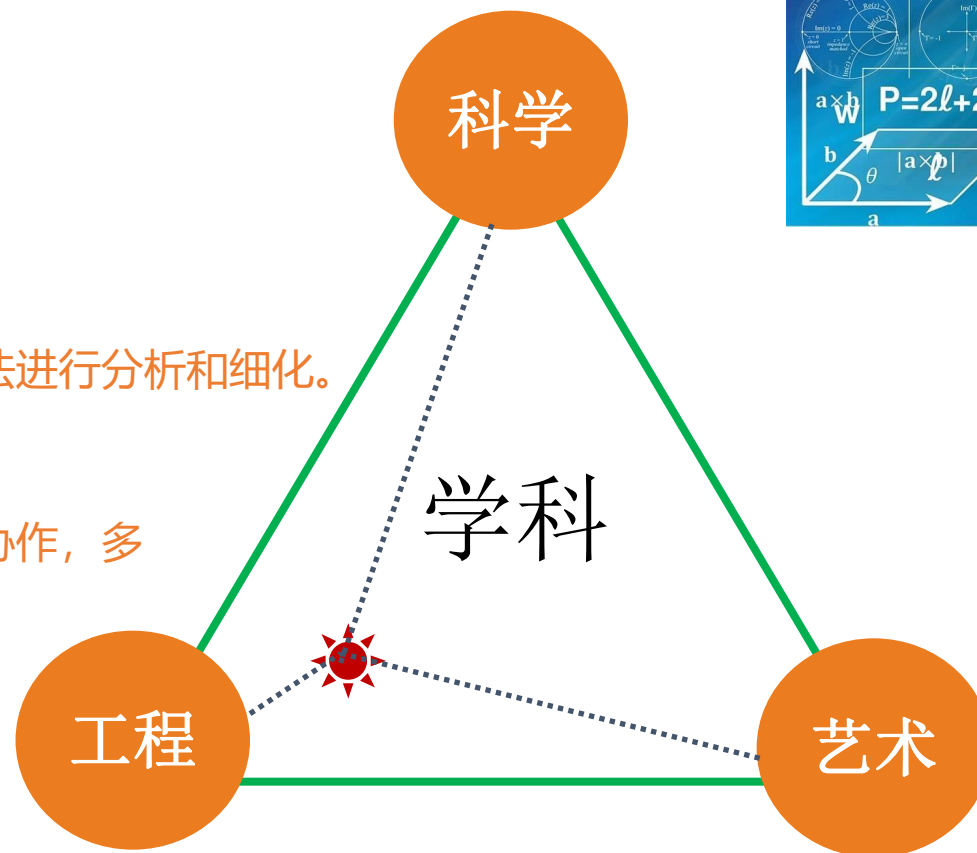
■ 是一门工程?

追求善：软件开发具有工程性，需要多人协作，多学科技术综合。



追求Benefit(善)

追求Truth(真)



追求Beauty(美)

什么是有价值的软件？

- 满足用户需求
 - 解决用户的**痛点**
 - 点燃用户的**兴奋点**
- 和竞争产品相比，具有**创新**和**竞争**优势
 - 业务创新和技术创新
 - 技术、政策和资源的优势

大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

01-什么是软件

☀ 02-什么是工程

03-什么是软件工程

04-AI时代的软件工程

@第1章.教材

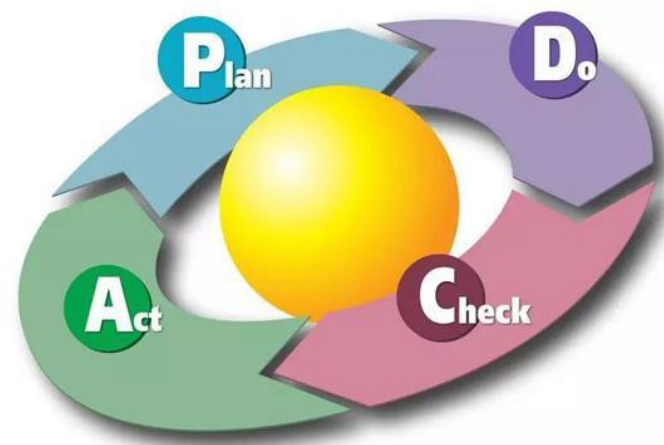
工程的定义和特性

- 工程是对技术（或社会）实体的分析、设计、建造、验证和管理。
- 工程是一种组织良好、管理严密、各类人员协同配合、共同完成工作的学科。
- 它具有以下特性：
 - **以价值为目标：** 获得最大效益，给社会或组织带来价值。
 - **高度的组织管理性：** 各类人员为了共同的愿景和目标协同配合，将互相作用和互相依赖的若干组成部分结合成具有特定功能的有机整体。
 - **多种学科的综合：** 综合使用多门学科的知识和技术。
 - **高度的实践性：** 从实践中总结知识，再应用于实践中。

工程过程

■ 工程过程四个核心步骤：

- 理解真正的问题：采用根源分析技术来发现需要解决的真正的潜在问题是什么。在软件工程领域，就是软件需求分析；
- 设计方案。设计多个技术上可行的解决方案，从技术、财务、进度等维度进行评估，从中选出最佳方案，确保方案能很好地满足需求。在软件工程领域，就是软件设计；
- 实施方案。在软件工程领域，就是软件编码；
- 监控方案的效能。实施过程中监督和评估它的实际效能。在软件工程领域，就是软件测试和质量保障。



- 工程过程是迭代的。前一个迭代中获得的知识将为后一个迭代提供指导和改进方向。

大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

01-什么是软件

02-什么是工程

☀ 03-什么是软件工程 **软件工程的知识和知识体系**
软件工程技术
软件工程管理

04-AI时代的软件工程

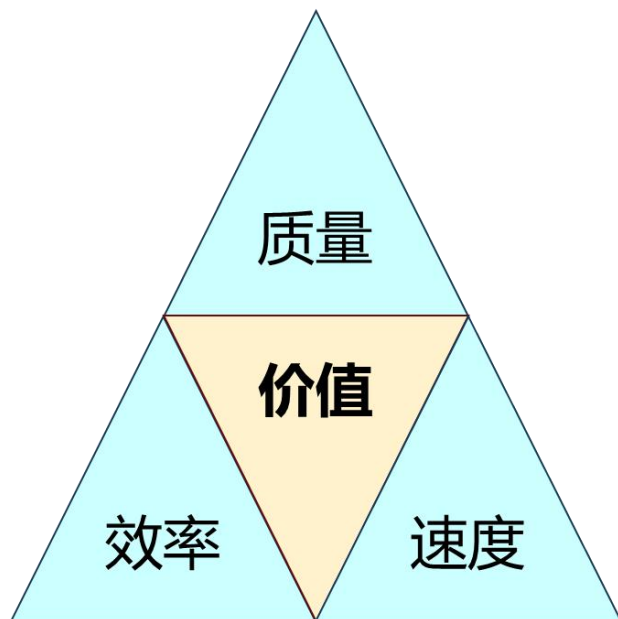
@第1章.教材

软件工程概念

面对软件开发和维护面临的挑战和问题：

- 以价值为核心，将工程的方法应用于软件开发；
- 用现代软件工程的观念、原理、技术和方法进行计算机软件开发、运营和维护；
- 将经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来；
- 以经济地开发出高质量的软件并进行有效的维护。

软件价值工程



- 软件工程是一项价值工程，根本目标就是创造价值，其生产力就是持续快速交付价值的能力，即以价值为中心，追求软件开发的质量、效率和速度，就是“多块好省”地开发和运维有价值的软件。
- 价值是“事物的作用、重要性或实用性”，提高软件价值就是提高效益、降低成本，对不同对象含义不同。
 - 软件开发方，高价值就是低开发成本；
 - 软件开发投资方，价值就是投资效益；
 - 软件使用者，价值就是效益。
 - 软件需求的根本目标就是创新，解决“提高效益”问题，即让“产品好卖”；软件设计的根本目标是软件模块与复用，致力于解决“降低成本”。
- 软件工程的每个环节和每个活动都是为了价值交付。

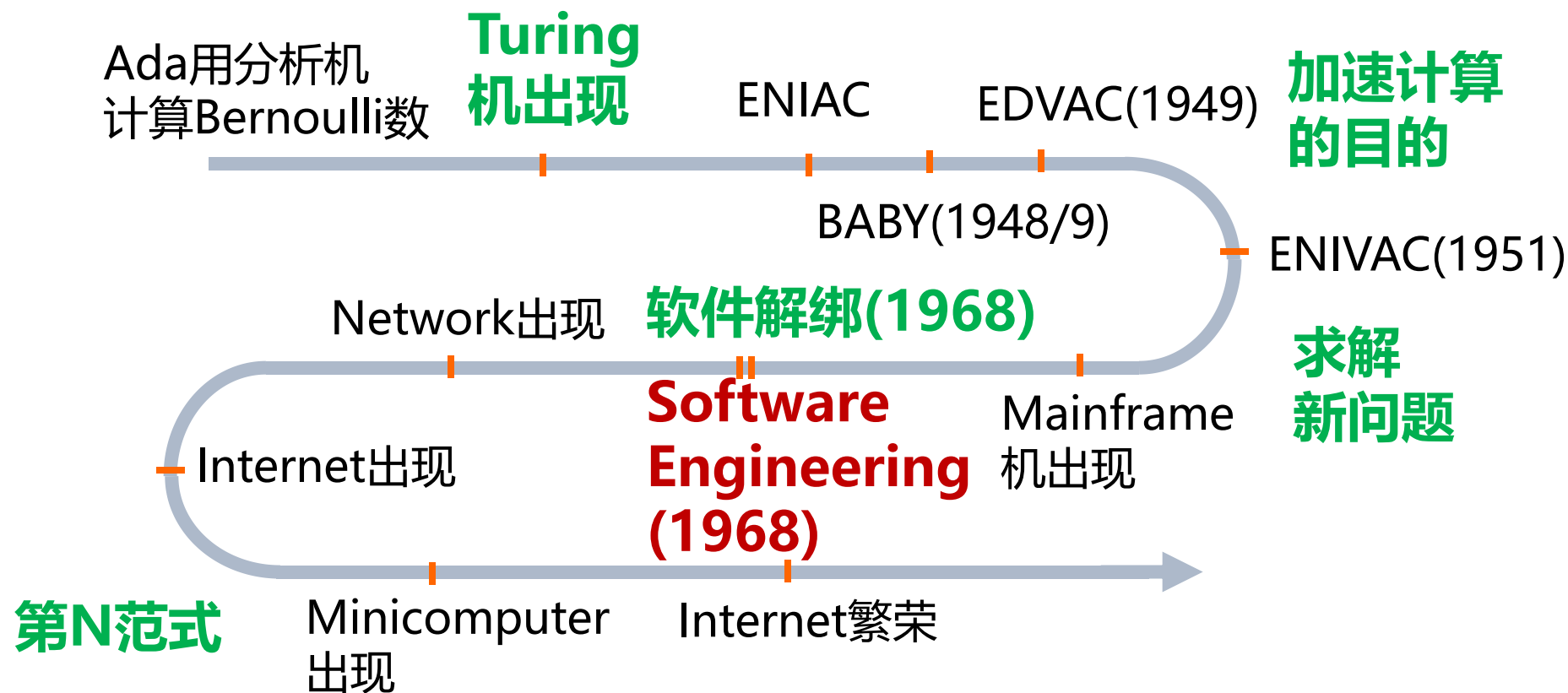
软件系统工程

- 软件总是处于复杂的系统，包括基础软硬件以及用户和软件所处的物理和社会环境。对用户和干系人，真正有价值的是完整的系统而非其中的软件，**需要在系统范围内思考软件定位和作用，明确软件需求。**
- **软件工程的系统思维：**
 - 1) 全局思维：分析软件系统内各模块之间关系与外部系统交互，以及关系背后的运行逻辑；
 - 2) 综合思维：社会系统和技术系统相互作用而形成的社会技术系统；
 - 3) 工程思维：以资源、条件不足为前提，去实现“现实世界”的目标；
 - 4) 抽象思维：抽象提炼，对软件系统建模。

软件工程是一个国家的战略性学科

- 软件工程的发展受到了美国DOD的极大推动
 - CMU SEI
- 如果一个国家的软件工程不强，那么这个国家就不会强大
- 软件工程要做强，关键在于**创新**，在于**规范**化的软件开发

软件工程概念的提出



软件工程的经典定义

- “The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works on real machines.”
[Fritz Bauer]

1968年NATO会议：软件工程就是为了经济地获得可靠的且能在实际机器上高效运行的软件而建立和使用的工程原理。

- “The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” [IEEE 1990]

1993年IEEE：软件工程是（1）将系统的、规范的、可量化的方法应用于软件的开发、运行和维护，即将工程化应用于软件。（2）研究（1）中提到的途径。

软件工程的经典定义(Cont.)

- “Software engineering is that form of engineering that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems.” [CMU/SEI-90-TR-003]

软件工程就是应用计算机科学和数学的原理来经济有效地解决软件问题的一种工程。

软件工程的本质特性

- ❑ 软件工程关注于大型程序的构造；**(关注大型程序)**
- ❑ 软件工程的中心课题是控制复杂性；**(控制复杂性)**
- ❑ 软件经常变化；**(考虑软件可能变化)**
- ❑ 开发软件的效率非常重要；**(关注软件开发效率)**
- ❑ 和谐地合作是软件开发的关键；**(团队开发)**
- ❑ 软件必须有效地支持它的用户；**(满足用户需求)**
- ❑ 在软件工程领域中是由具有一种文化背景的人替具有另一种文化背景的人创造产品。**(缺乏相关领域知识)**

软件工程的基本原理

1. 用分阶段的生命周期计划严格管理;
2. 坚持进行阶段评审;
3. 实行严格的产品控制;
4. 采用现代程序设计技术;
5. 结果能清楚地审查;
6. 开发小组的人员应该少而精;
7. 承认不断改进软件工程实践的必要性。

软件生命周期组成

1. 软件定义

A.问题定义 B.可行性研究 C.需求分析

2. 软件开发

D.总体设计

E.详细设计

F.编码和单元测试

G.综合测试

3. 运行维护。

软件生命周期各个阶段

1. 问题定义;
2. 可行性研究;
3. 需求分析;
4. 总体设计（概要设计）;
5. 详细设计;
6. 编码与单元测试;
7. 综合测试;
8. 维护。

软件工程知识体系SWEBOK



- 软件工程知识体系(Software Engineering Body of Knowledge) 简称SWEBOK
 - 由IEEE和ACM等国际组织联合推出
 - 促进世界范围内对软件工程的一致观念，阐明软件工程相对其他学科（如计算机可行、项目管理、数学、工程学）的定位和分界，刻画软件工程学科的核心知识。
 - 版本：当前版本 V4 (2024)
 - 网址： <http://www.swebok.org>

- 国际软件工程师证书：
 - CSDA，针对大学应届生
 - CSDP，针对有经验的工程师

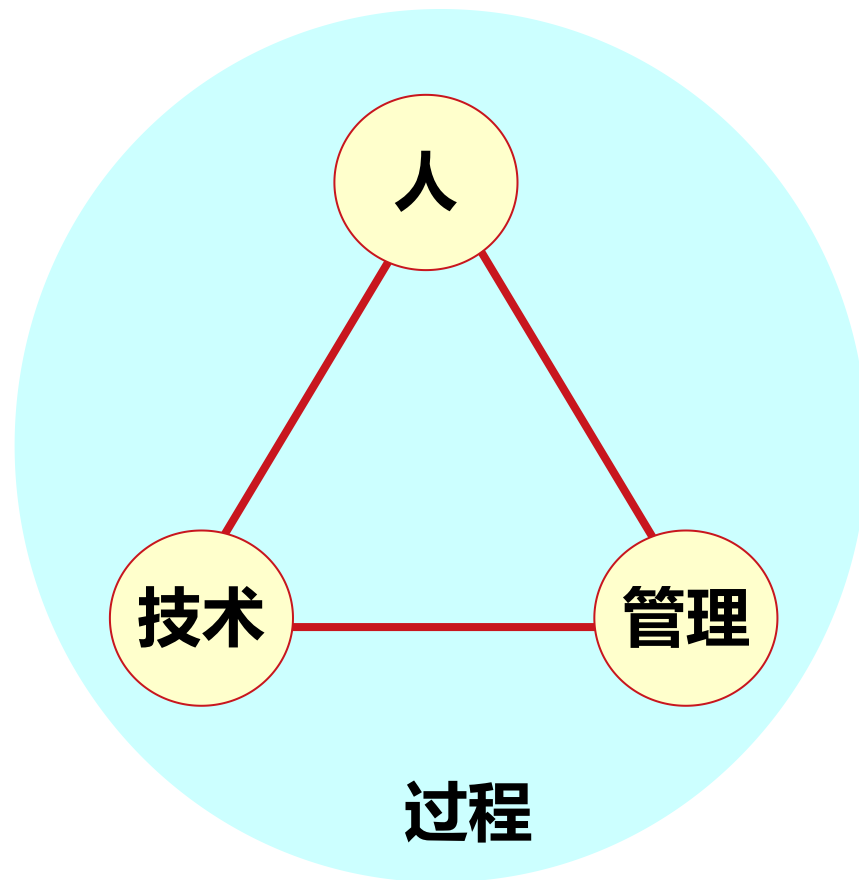
SWEBOK V4的18个知识域

□ 软件工程专业知识域

- 软件需求、软件架构、软件设计、软件构造
- 软件测试、软件运营、软件维护、软件配置管理
- 软件工程管理、软件工程过程
- 软件工程模型与方法、软件质量、软件安全
- 软件工程经济、软件工程职业实践

□ 软件工程通用知识域

- 工程经济基础、计算基础
- 数学和工程基础



软件工程的金三角

大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

01-什么是软件

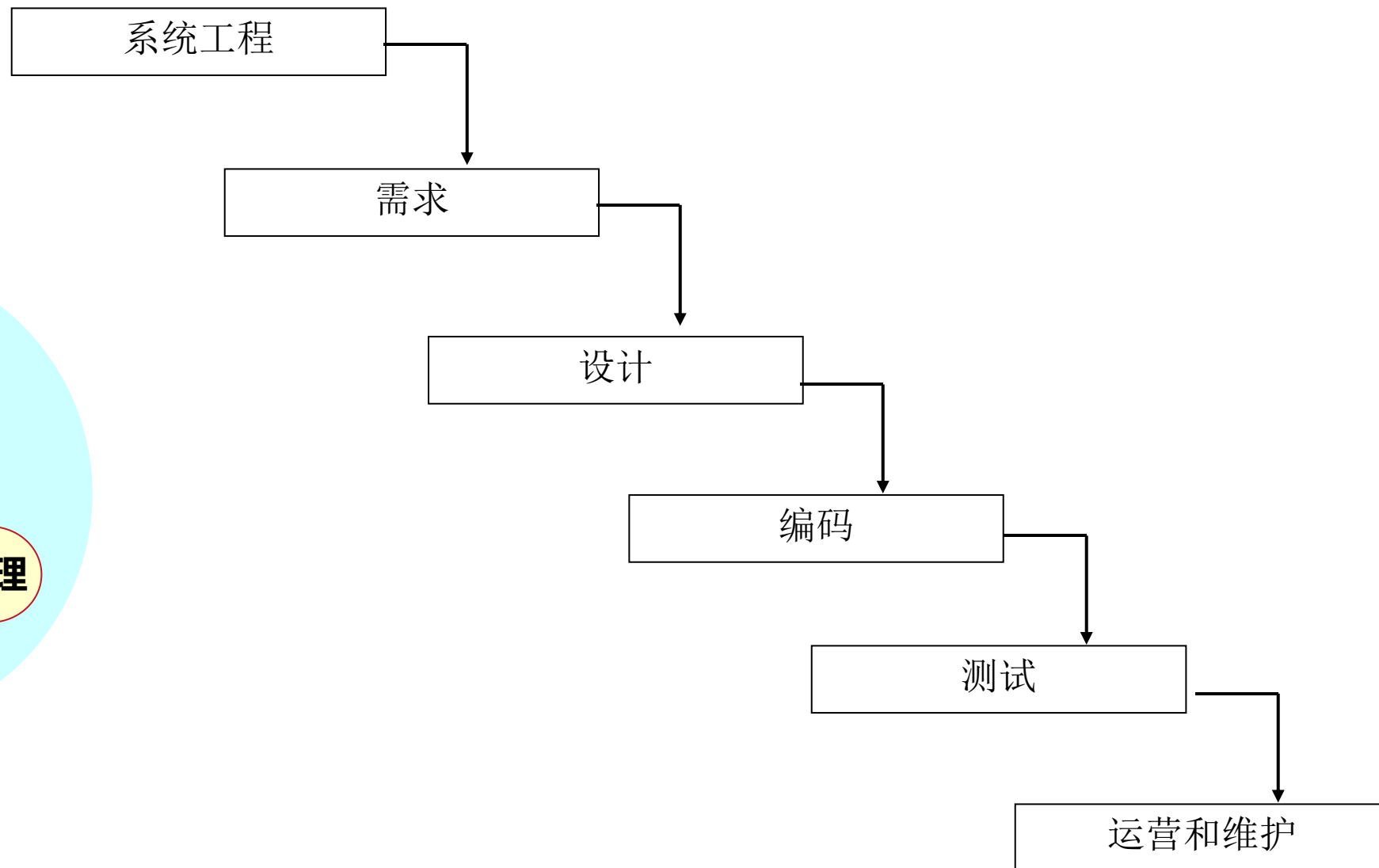
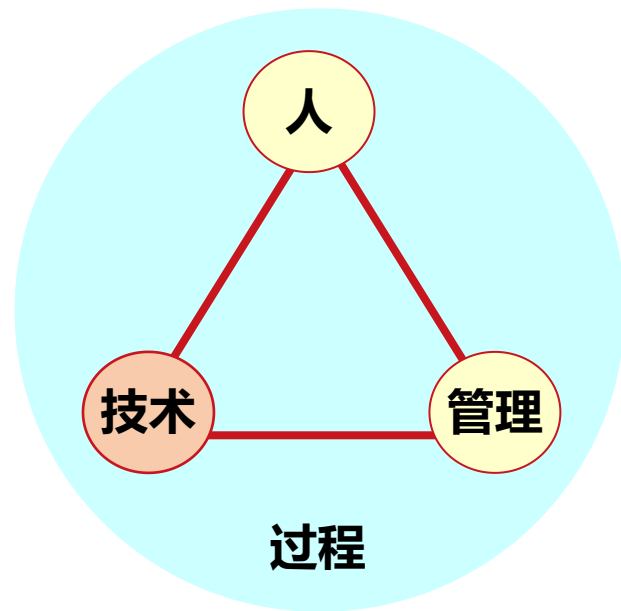
02-什么是工程

☀ 03-什么是软件工程 软件工程的知识和知识体系
软件工程技术
软件工程管理

04-AI时代的软件工程

@第1章.教材

软件工程技术



系统工程 (System Engineering)

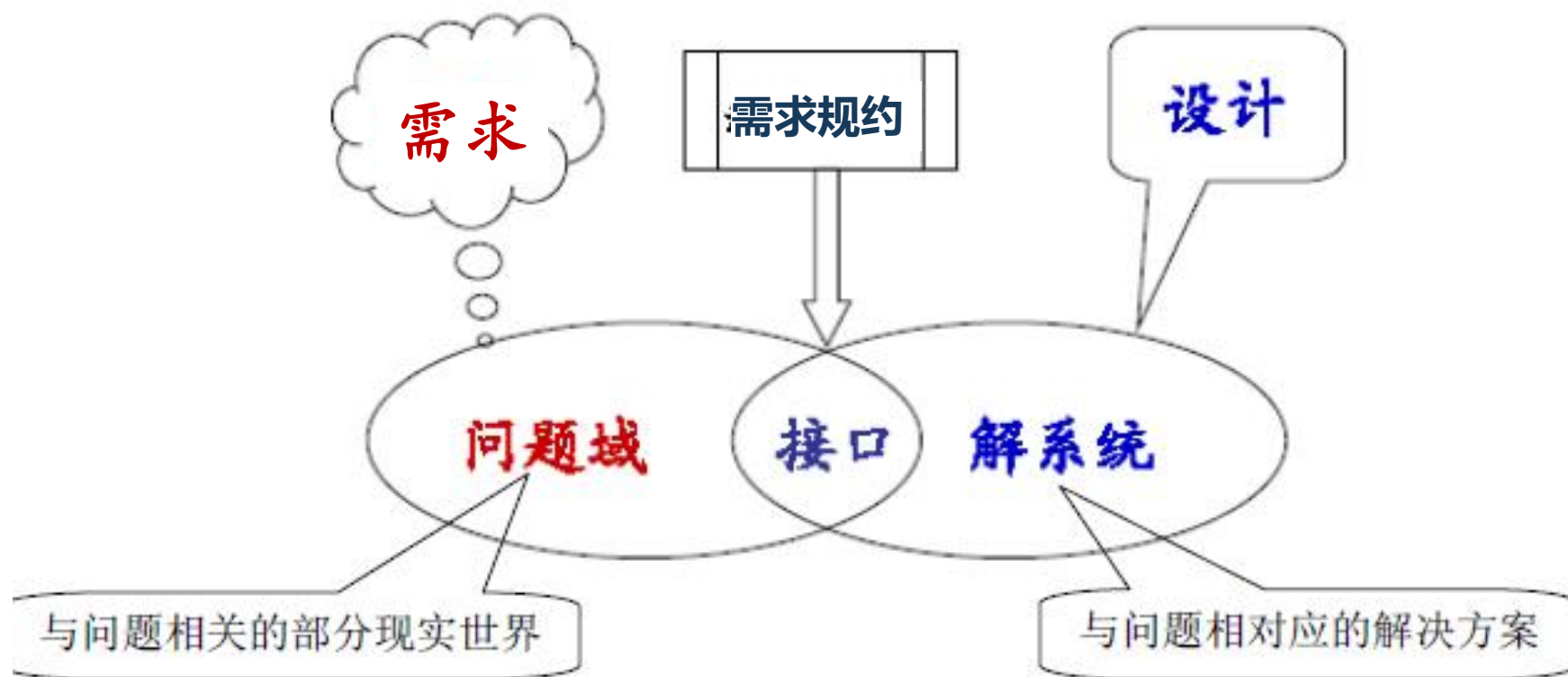
- 在软件开发之前，必须了解该软件所外的外部“系统”。
- 计算机系统包括计算机硬件、软件、人员、数据库、文档、规程等系统元素。
- 系统工程的任务：
 - 系统建模 - - 系统模型
 - 系统仿真
- 系统工程的表现形式
 - 信息系统，关注企业，业务过程工程 - - 业务模型
 - 嵌入式系统，关注产品，产品工程 - - 产品模型
 - 多媒体系统，关注内容，内容工程 - - 剧本

需求 (Requirement)

- 目的：澄清用户的需求。
- 描述：软件人员和用户沟通，充分理解和获取用户的需求，并进行分析，形成《软件需求规约》文档和分析模型。
- 任务：
 - 需求获取
 - 需求分析和建模
 - 需求定义
 - 需求确认
 - 需求管理

重点在What

需求与设计



设计 (Design)

- 目的：建立软件的设计蓝图，是需求到代码的桥梁。
- 任务：软件人员依据软件需求，进行设计，形成《软件架构》文档和设计模型。
- 内容
 - 架构设计
 - 详细设计

重点在How

编程 (Coding)

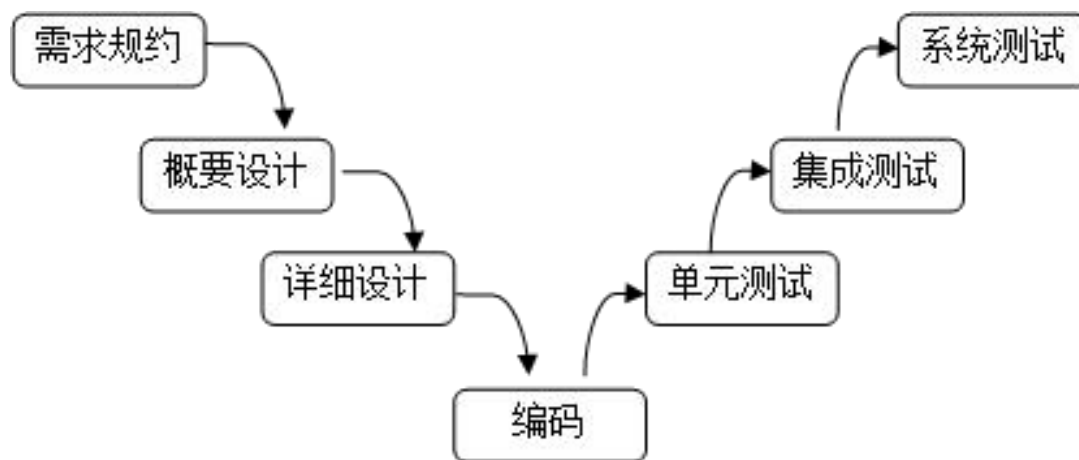
- 任务:依据设计说明书为每个模块编写程序
- 交付:程序
 - 程序符合编程规范, 含有必要的注释
 - 不含有语法错误
- 交付这样的程序就是编程阶段完成的里程碑

编程的误区

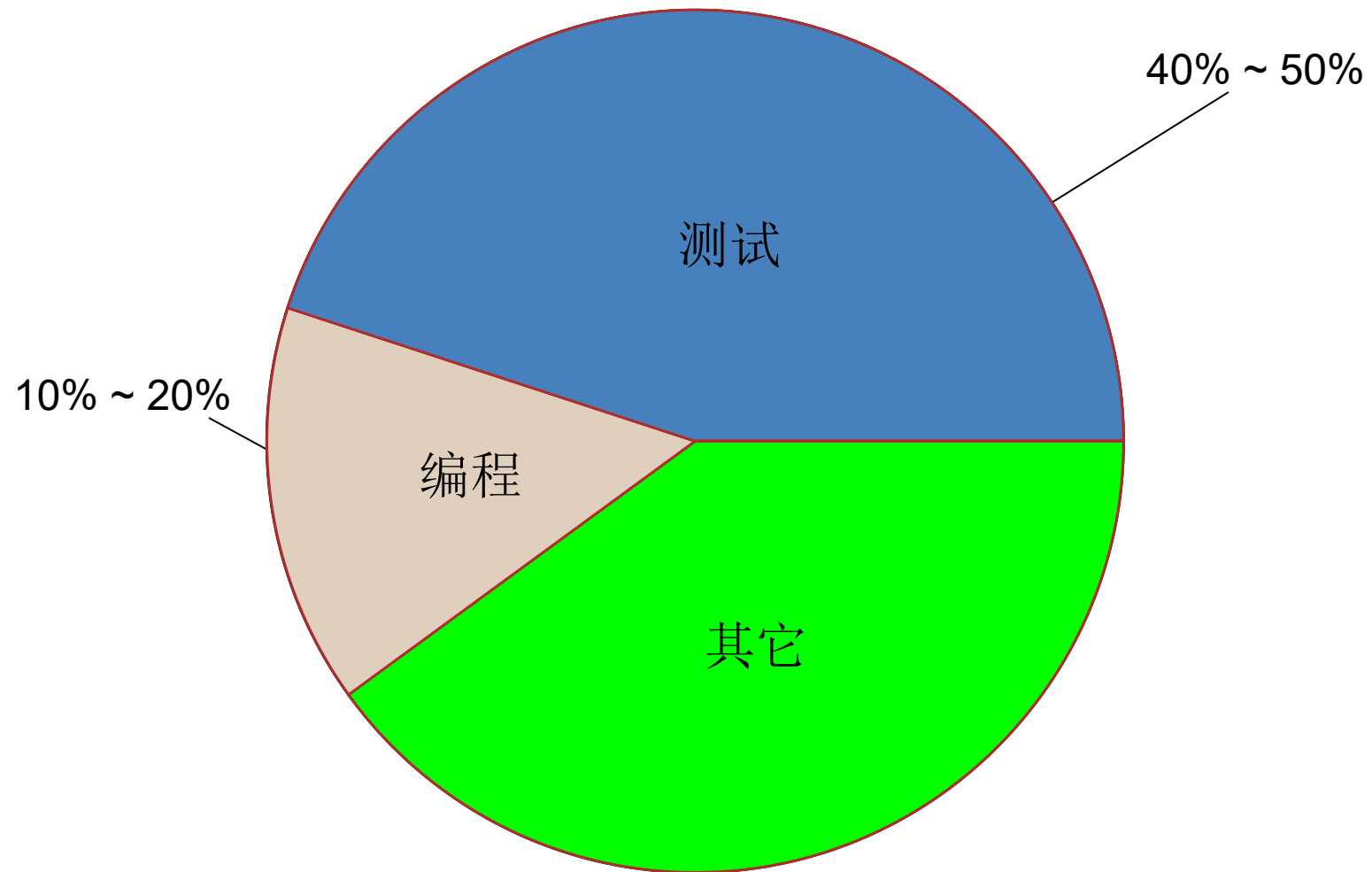
- ❑ 软件人员应该克制急于去编程的欲望
 - 先经过分析阶段确定用户要求,
 - 再经过设计阶段为编程制订蓝图,
 - 至此编程的条件才具备, 可进入编程阶段
- ❑ 不要让compiler代你找代码的错误
 - 那样会让语义bug隐藏得更深
 - 先要自查代码的错误, 再编译

测试 (Testing)

- ❑ 任务：发现并排除软件中的缺陷
- ❑ 不同层次的测试：
 - 单元测试 (Unit testing)
 - 集成测试 (Integration testing)
 - 系统测试 (System testing)



软件开发工作量分配比例



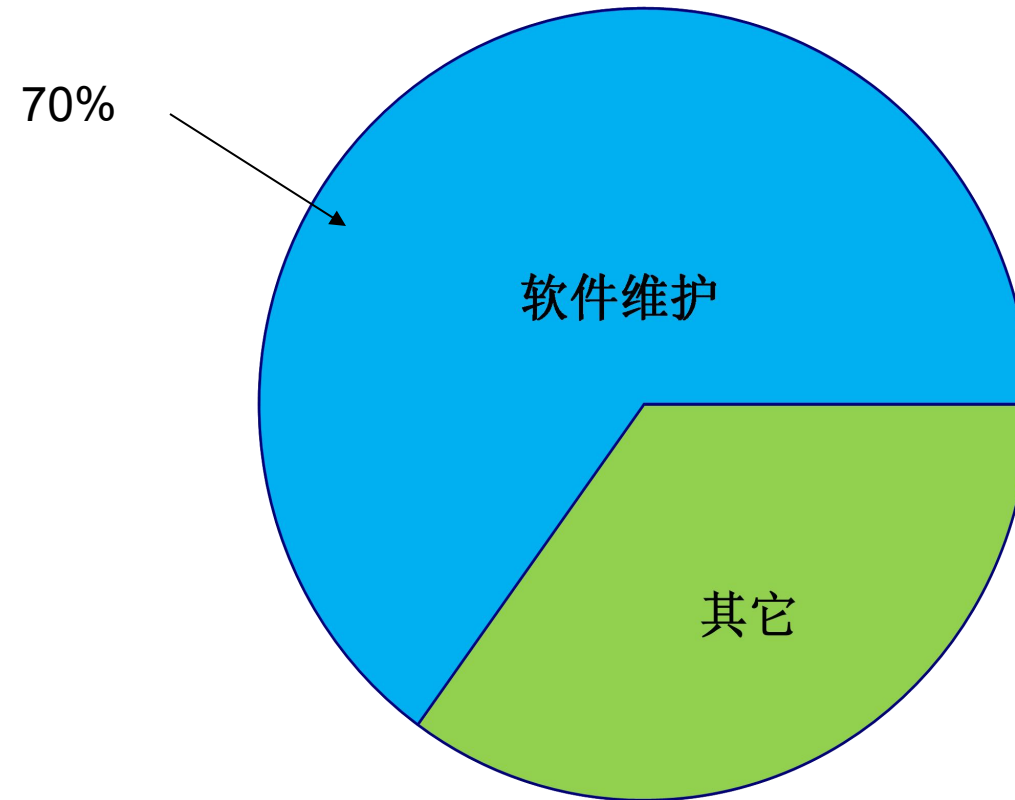
软件运营 (Operation)

- 软件运营，是指在目标环境中部署和配置软件，以及在软件运行时（直到停用）对其进行监控和管理的过程
- 目标
 - 快速部署
 - 配置优化
 - 高可靠运行，高质量服务

软件维护 (Maintenance)

- 软件维护，是指软件系统交付运营后，为了改正错误或满足新的需要而修改软件的过程。
 - 举例：一个中等规模的软件，如果开发过程要一年时间，它投入使用后，其运行时间可能持续五年，这段时间是维护阶段。
- 与开发相比，维护阶段的工作量和成本要大得多。
- 人们对软件维护的认识远不如软件开发，因为开发具有主动性和创造性，易被人们所重视。

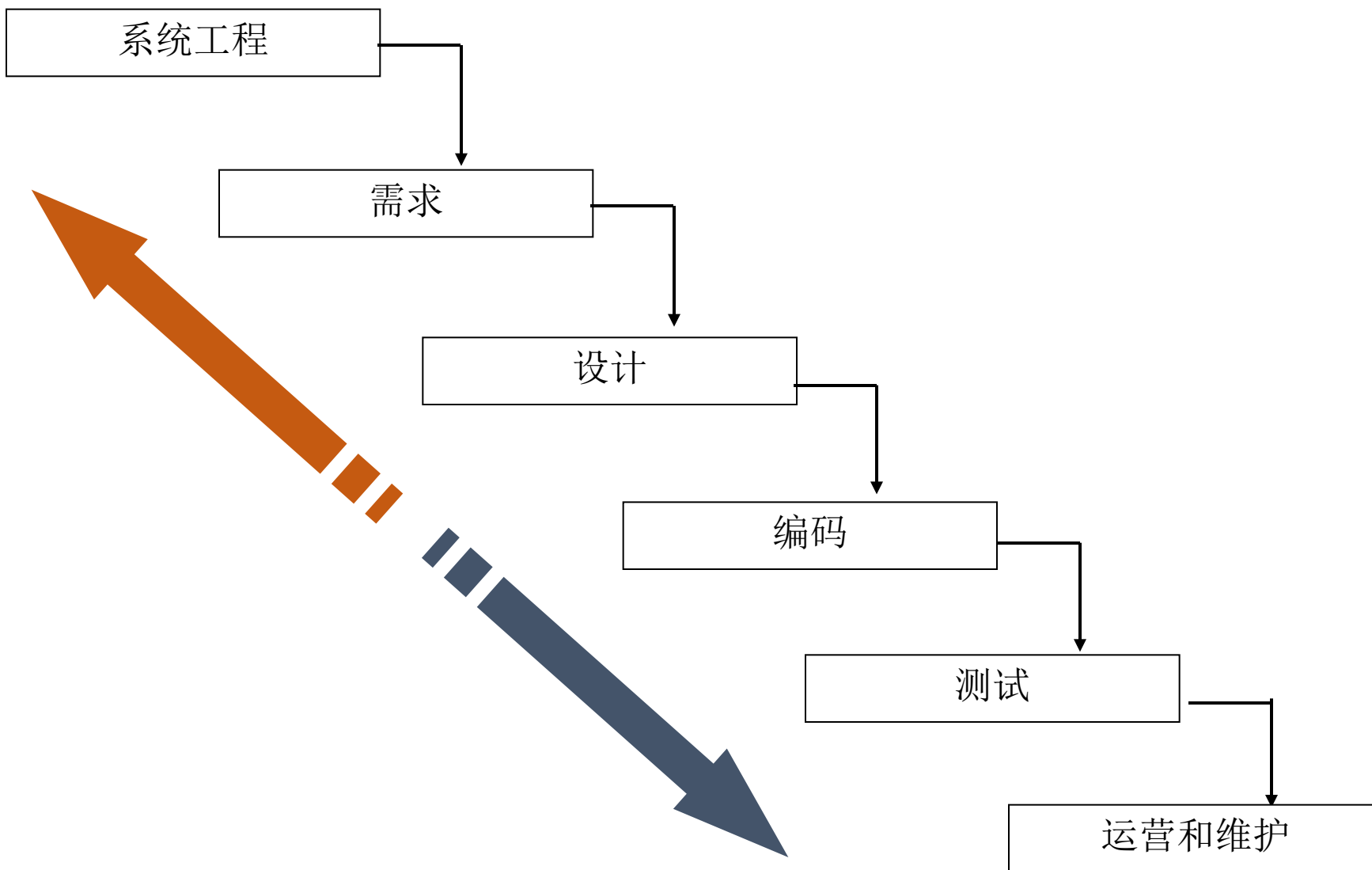
费用分配比例



维护类型

- 纠错性维护 (corrective maintenance)
 - 为修复所发现的问题而进行的反应式维护
- 预防性维护 (preventive maintenance)
 - 在软件产品中的潜在错误成为实际错误前所进行的预防性修复，例如设计重构和代码优化等
- 适应性维护 (adaptive maintenance)
 - 为适应环境的变化而修改软件的维护
- 补充性维护 (additive maintenance)
 - 添加新功能或新特性以增强产品使用的维护
- 完善性维护 (perfective maintenance)
 - 为用户提供功能增强、程序文档改进、软件性能和可维护性等质量属性的提升而做的维护。
和补充性维护不同的是，它并不增加新功能和新特性，所做的修改较小
- 紧急维护 (emergency maintenance)
 - 计划外的应急修改，以暂时保持系统运行，等待后续纠错性维护

软件工程的发展



大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

01-什么是软件

02-什么是工程



03-什么是软件工程

软件工程的知识和知识域

软件工程技术

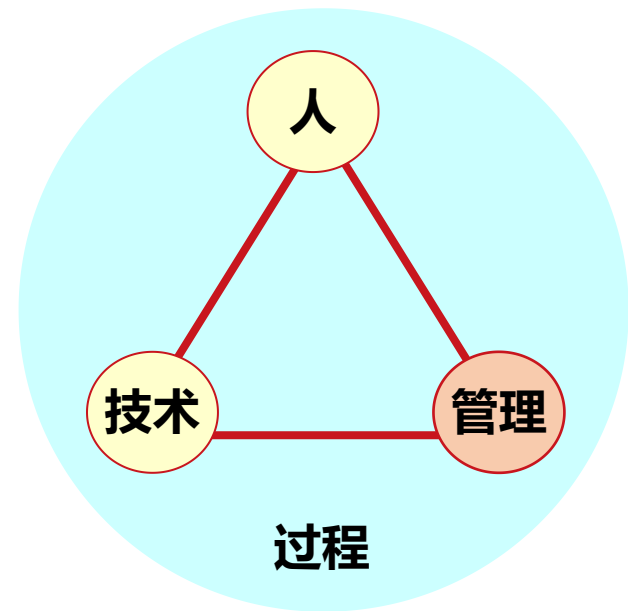
软件工程管理

04-AI时代的软件工程

@第1章.教材

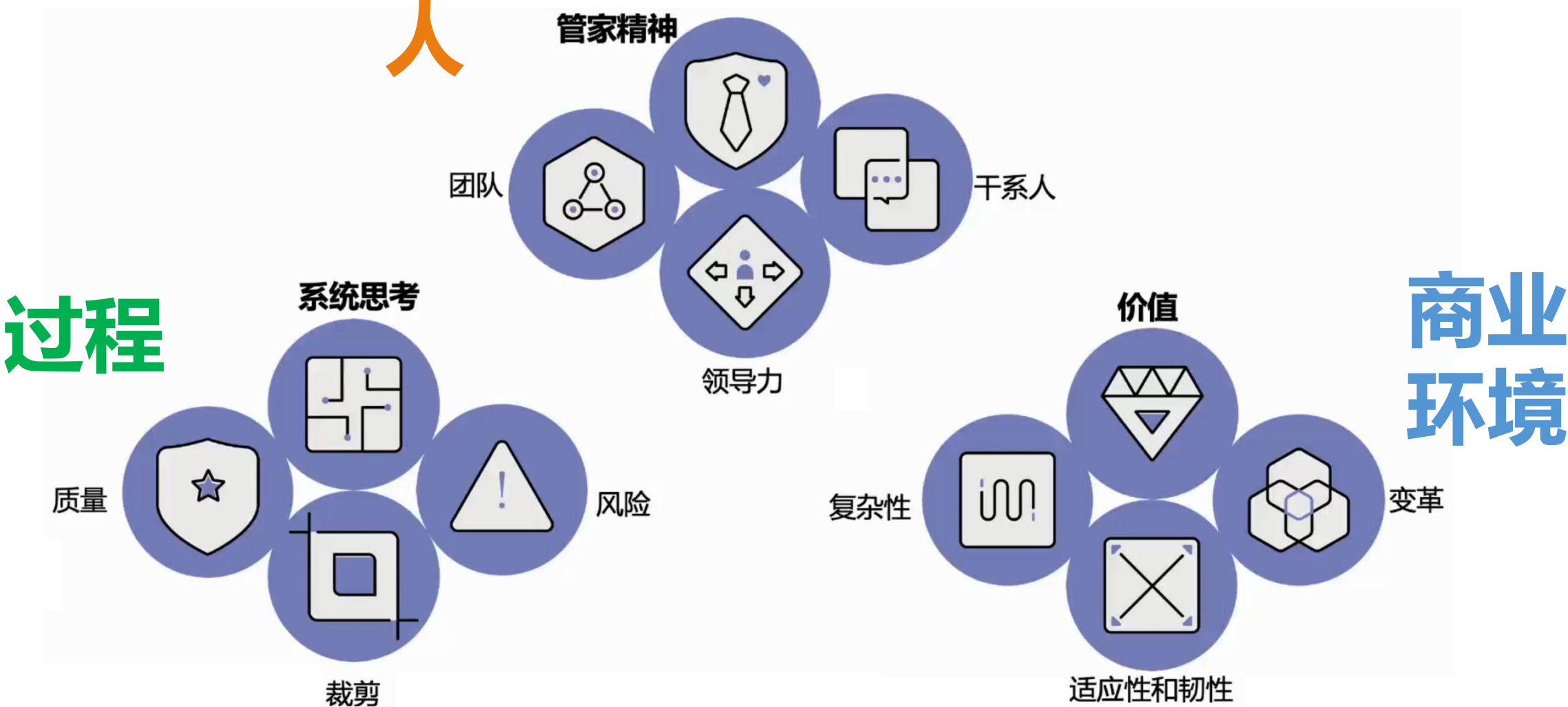
软件工程管理

- ❑ 项目的知识体系PMBOK (Project Management Body of Knowledge)
- ❑ 由美国项目管理学会 (PMI) 提出
- ❑ 1996年第1版, 目前最新版本为2021年第7版
- ❑ 核心内容
 - 十二大原则
 - 八大绩效域
- ❑ PMI项目管理专业人员资格认证PMP
- ❑ ISO以PMBOK为框架制订了 ISO10006标准

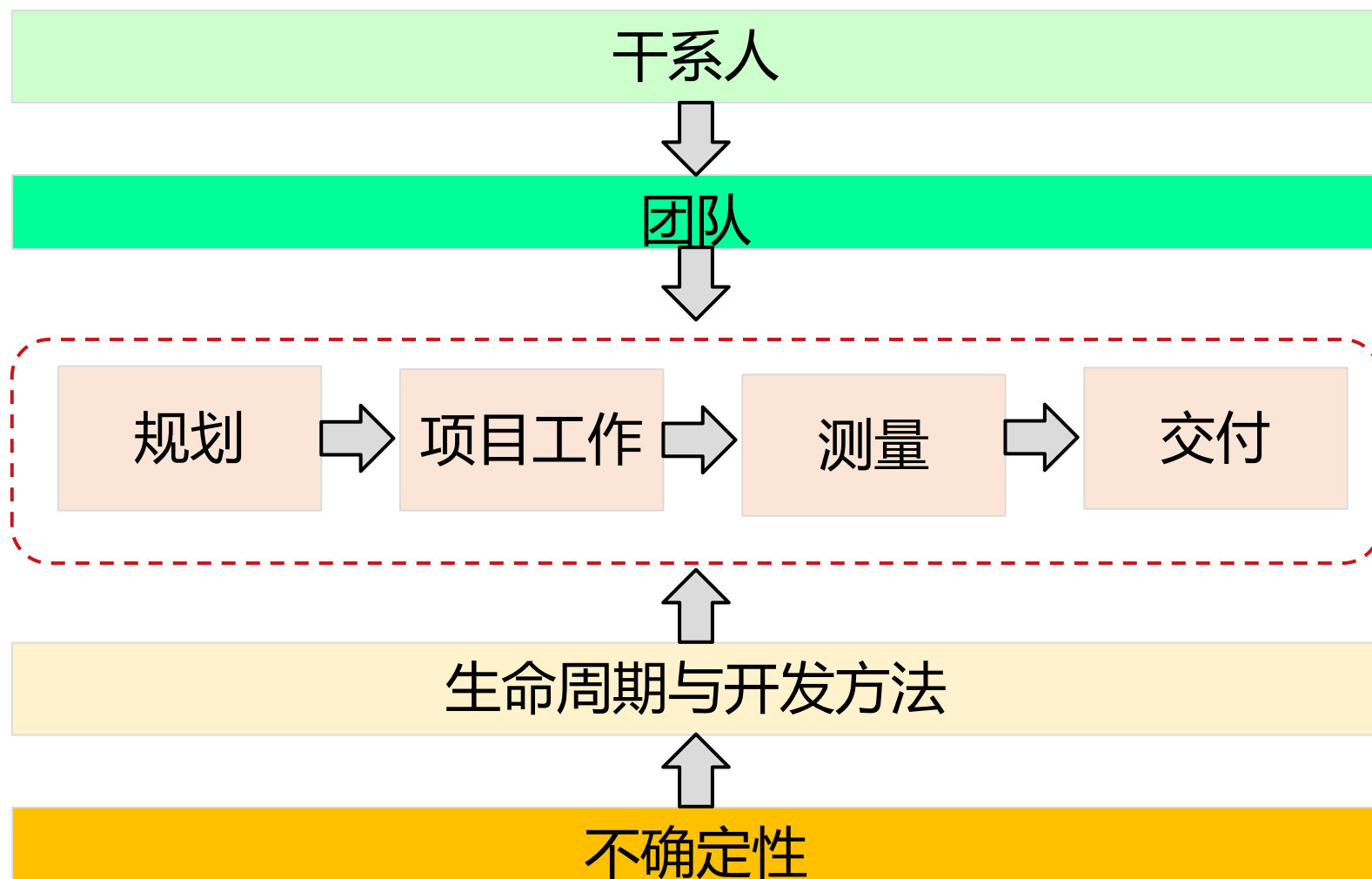


<http://www.pmi.org>

PMBOK的项目交付十二大原则

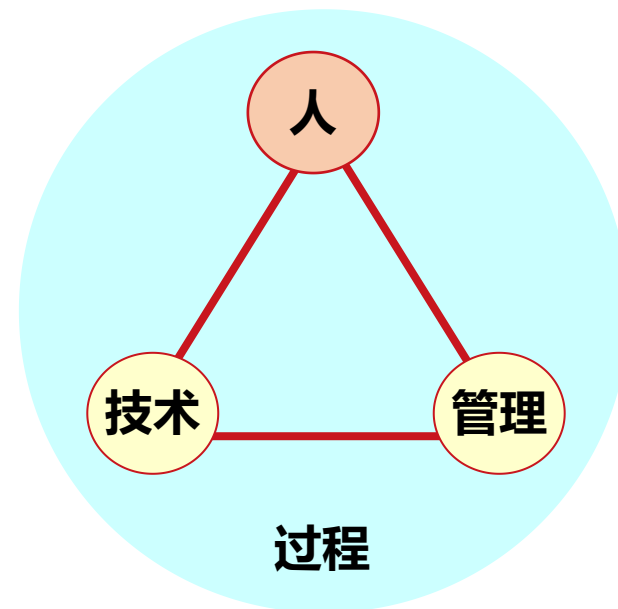
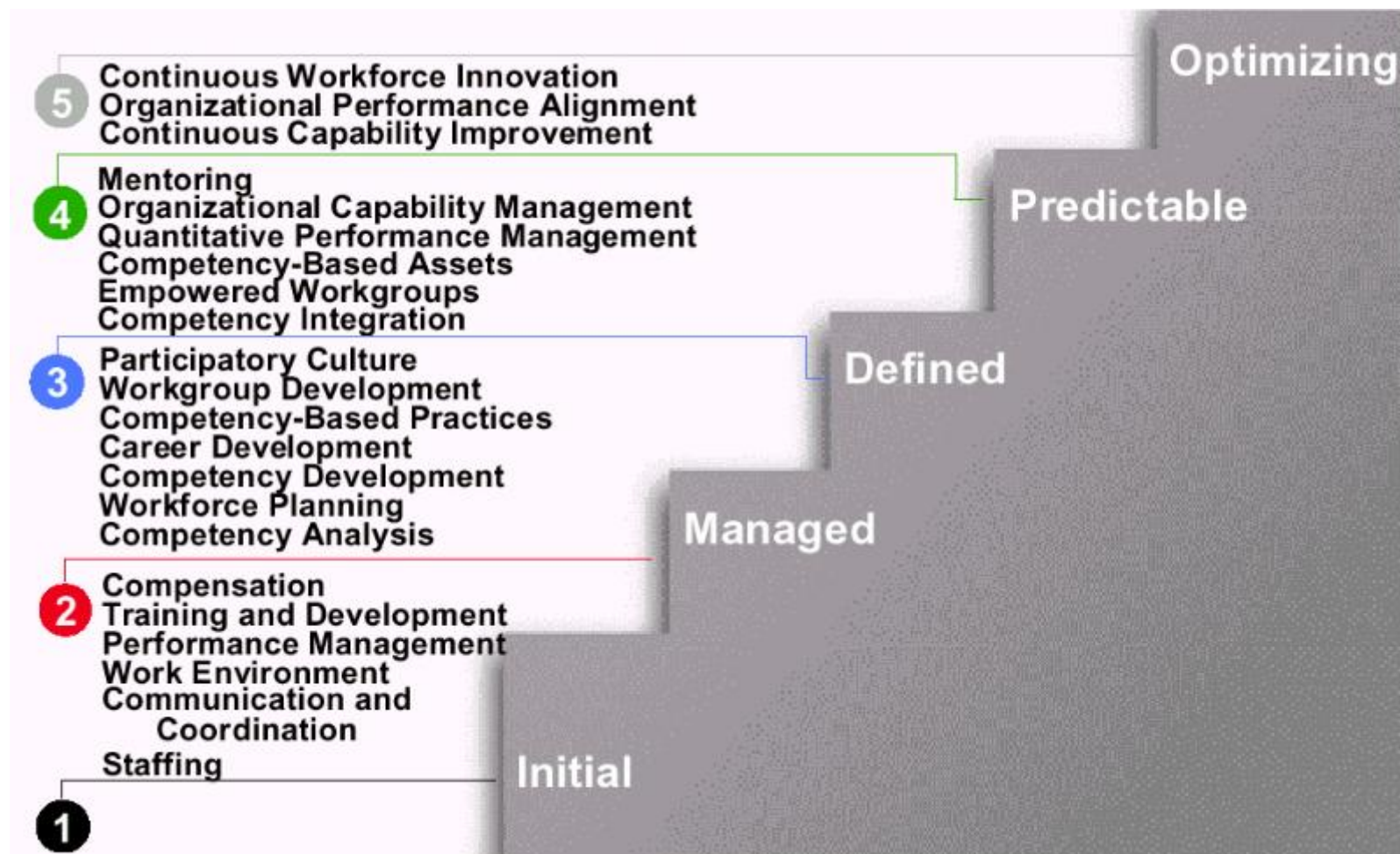


PMBOK的八大绩效域



软件工程中人的管理

□ People CMMI V2



软件工程

职业道德规范和实践要求

IEEE-CS 和ACM软件工程师道德和职业实践联合工作组推荐
(5.2版)
居德华 朱三元 译

软件工程师应当作出承诺，使软件的分析、规格说明、设计、开发、测试和维护等工作对社会有益且受人尊重。基于对公众健康、安全和福利的考虑，软件工程师应当遵守以下8条原则：

1. 公众感——软件工程师应始终与公众利益保持一致；
2. 客户和雇主——软件工程师应当在与公众利益保持一致的前提下，保证客户和雇主的最大利益。
3. 产品——软件工程师应当保证他们的产品以及相关的修改尽可能满足最高的行业标准。
4. 判断力——软件工程师应当具备公正和独立的职业判断力。
5. 管理——软件工程管理者和领导者应当维护并倡导合乎道德的有关软件开发和维护的管理方法。
6. 职业感——软件工程师应当弘扬职业正义感和荣誉感，尊重社会公众利益。
7. 同事——软件工程师应当公平地对待和协助每一位同事。
8. 自己——软件工程师应当毕生学习专业知识，倡导合乎职业道德的职业活动方式。

<http://seeri.etsu.edu/OfficalChineseTranslation.pdf>

软件工程师的社会责任

社会需要
国家需求

伦理道德
社会影响

现实世界



理解

应用

软件定义一切

职业道德
工匠精神

可靠可信
自主可控



交付



软件工程师

软件系统

大纲



中南大学
CENTRAL SOUTH UNIVERSITY

第一章 软件工程引论

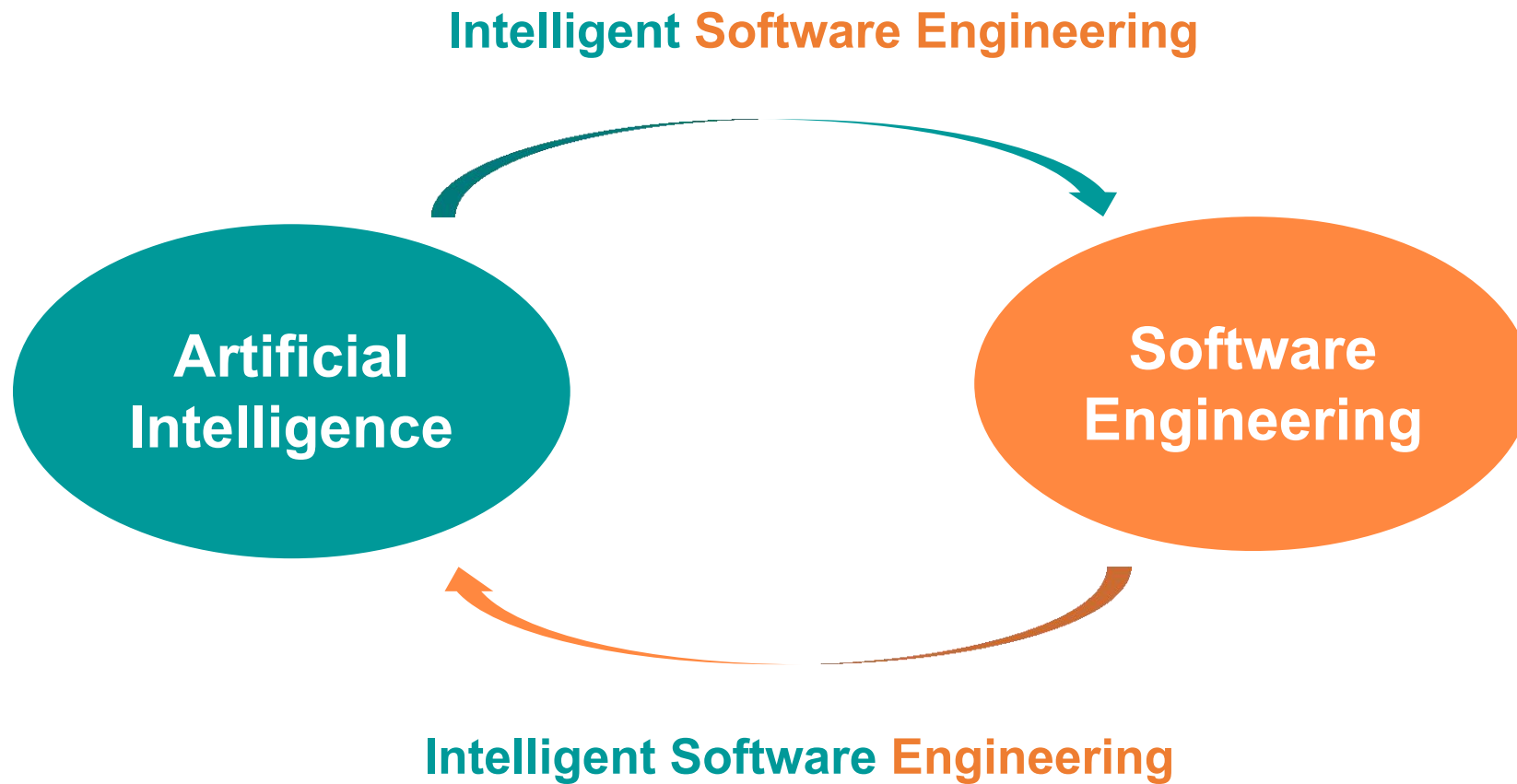
01-什么是软件

02-什么是工程

03-什么是软件工程

☀ 04-AI时代的软件工程

@第1章.教材



大模型(Large Language Model, LLM)的崛起

❑ DeepMind 的代码生成工具AlphaCode

- 采用Transformer模型，用大规模代码语料进行预训练
- 2021年底参加了 Codeforces竞赛平台组织的10场实时编程比赛，总体排名位于前54.3%，击败了 46% 的人类参赛者。



❑ 微软的AI结对编程助手GitHub Copilot

- 采用OpenAI 的Codex模型，一个基于GPT-3 的代码预训练模型
- 2022年8月份结束了公测，开始商业化运作



❑ OpenAI 的对话式大型语言模型ChatGPT和GPT-4

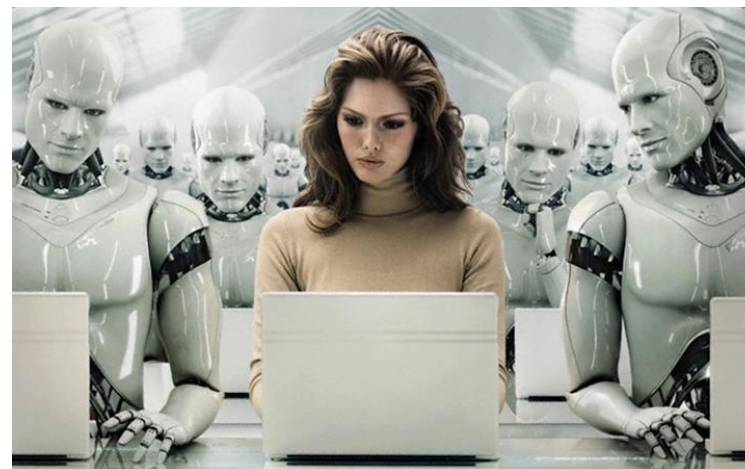
- ChatGPT 最擅长的就是回答用户提出的问题。
- ChatGPT 通过了 2022 年的 AP computer science A 考试，得分 32 分，满分 36。这门考试的难度等同于大学 CS专业课第一学期难度。



智能开发助手

工具示例：

- Github Copilot
- Amazon CodeWhisperer
- Google Colab
- Aixcoder
- CodeArts Snap



编程助手

- 代码生成 NL2PL
- 代码补全 Code Complete
- 代码翻译 PL2PL
- 代码理解与调试

质量助手

- 代码评审与代码重构
- 静态代码检查
- 缺陷修复
- 测试用例生成

技术助手

- 技术问答
- 需求分析
-

挑战还是机遇？



大模型将在3年内终结编程。程序员会被淘汰，未来只有产品经理和代码审查员。

-- 谷歌工程主管 Matt Welsh



局部效率提升

**基础编码能力
的知识平权**

LLM 提高了编程的效率和质量，使我们可以在更短的时间内完成更多的工作，因而能留出更多的时间让我们思考。

LLM时代，软件工程什么变了？

对软件工程的影响

- LLM 的作用
 - 开发的局部辅助（pair programming），提升局部效率
 - 有利于控制研发团队规模，保持小团队的优势
- 软件工程的内在复杂性和可变性等特性并没有发生本质上的变化
- 需求和设计的重要性变得更为重要

对软件工程师的影响

更需要从以下方面来强化自己的能力

- 抓住问题本质的能力
- 需求理解和需求分析的能力
- 架构设计和设计取舍的能力
- 代码审查的能力

思考：如何控制软件开发的复杂性？

□ 软件开发常常是相当复杂、不可预测、难以计划的。

□ 软件开发的复杂性主要来源于：

■ 技术的复杂性

- 技术不断发展
- 使用多项技术

■ 需求的复杂性

- 需求模糊
- 需求不断变化

■ 人的复杂性

□ 虽然有了LLM和工具，但是软件开发的复杂性仍需要人去控制



如何控制复杂性？

- ✓ 抽象
- ✓ 分解
- ✓ 迭代