

经典软件工程

周明辉

zhmh@pku.edu.cn

北京大学

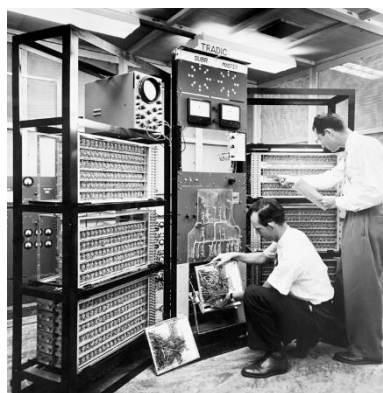
软件工程的历史和定义

软件工程之前的软件开发

个体化的软件开发

1960年代中期之前

通用硬件相当普遍
软件为每个具体应用而专门编写
软件通常是规模较小的程序
编写者和使用者往往是同一组人



第二代晶体管计算机：TRADIC（1954）、IBM 1401（1958）

群体化的软件作坊

1960年代中期~1970年代中期

计算机系统持续发展
软件的通用性不断增强
软件的数量急剧膨胀
软件的复杂度和规模不断增加



IBM公司1963-1966年开发IBM360操作系统

软件工程的提出

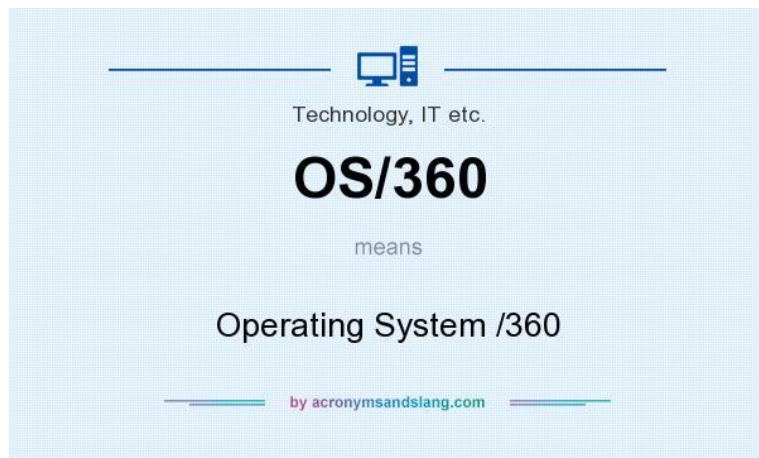
- 1960年代中期软件领域爆发一系列严重问题。1968年，北大西洋公约组织（NATO）在联邦德国的国际学术会议创造**软件危机**一词来描述问题，并提出**软件工程**的概念来解决软件危机。
 - **1st NATO Software Engineering Conference Garmisch, Germany**
 - 软件危机：Software crisis
 - 软件工程：software engineering



什么是软件危机

- 软件危机是指在计算机软件的开发和维护过程中所遇到的一系列**严重问题**。危机表现在几个方面：
 - 项目运行超出预算。
 - 项目运行超过时间。
 - 软件质量低下。
 - 软件不符合需求。
 - 项目无法管理，且代码难以维护。
- 软件危机的主要原因，很不客气地说：在没有机器的时候，编程根本不是问题；当我们有了电脑，编程开始变成问题；而现在我们有巨大的电脑，编程就成为了一个同样巨大的问题。
-- Dijkstra, E. W. The Humble Programmer. Communications of the ACM. Aug 1972.

软件危机实例



IBM公司1963-1966年开发IBM360操作系统，项目花了5000人·年的工作量，最多时有1000人投入开发工作，写出100万行源程序，但发行的每一新版本都是上一版1000个错误的修正。



事后，负责人F.D.Brooks（1931-）总结教训时说：“……正象一只逃亡的野兽落到泥潭中做垂死的挣扎，越是挣扎，陷的越深。最后无法逃脱灭顶的灾难。程序设计工作正像这样一个泥潭，一批批程序员被迫在泥潭中拼命挣扎，……谁也没料到问题竟会陷入这样的困境……”

软件工程的定义

- **1968年, 1st NATO SE会议,**
 - "the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines"—[Fritz Bauer](#)
 - 建立和使用合理的工程原理, 以经济地获得可靠且在真实机器上有效运行的软件
- **1993年~, [IEEE](#) *Standard Glossary of Software Engineering Terminology***
 - The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of [software](#)
 - 将系统的、规范的、可量化的方法应用于软件的开发、操作和维护
- **2010, The Bureau of Labor Statistics—[IEEE](#) Systems and software engineering – Vocabulary**
 - The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software
 - 将科学技术知识、方法和经验系统地应用到软件的设计、实施、测试和文档编制中

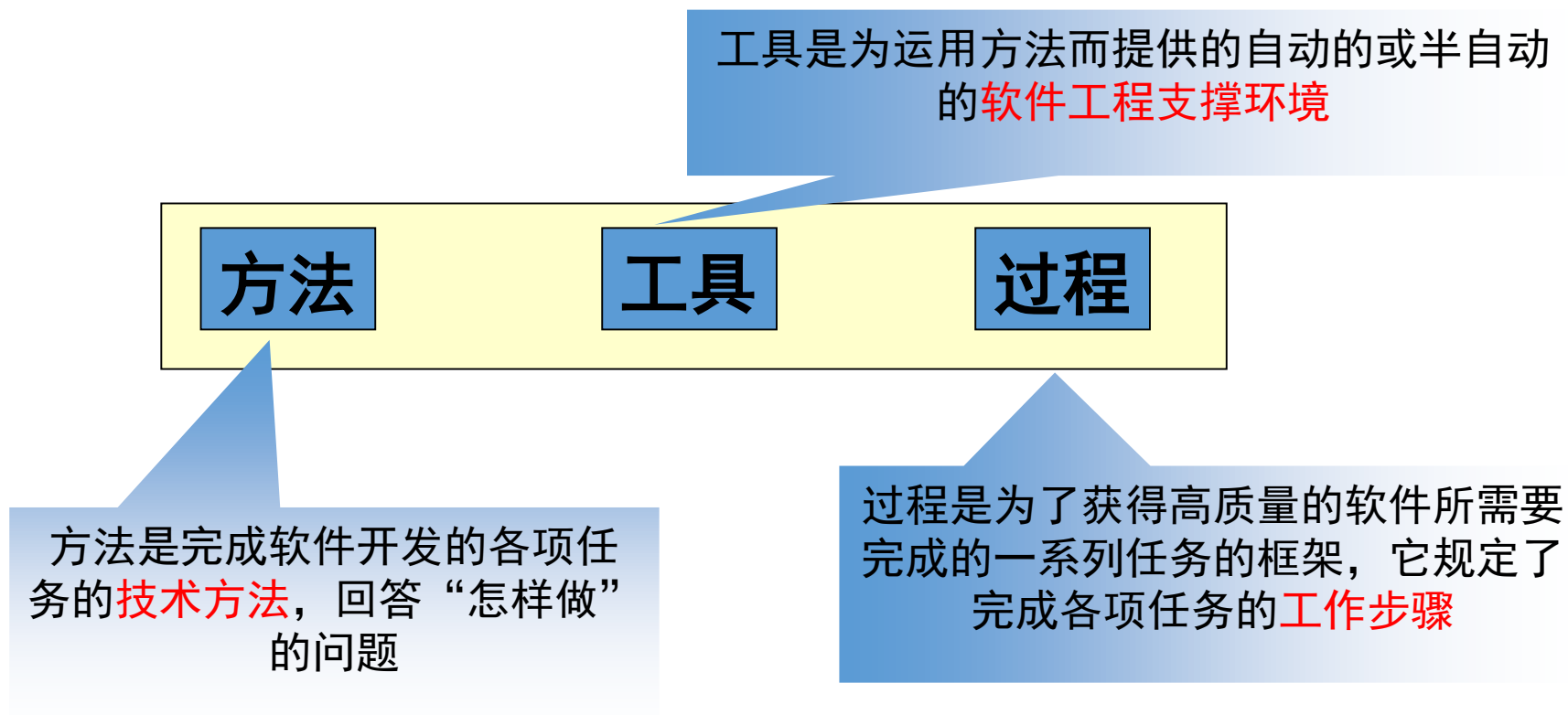
什么是软件工程

[Sommerville, Ian](#) (2007) [1982]. ["1.1.2 What is software engineering?"](#). *Software Engineering* (8th ed.). Harlow, England: Pearson Education. p. 7.

- *Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone into use.* (软件工程是一门工程学科，涉及软件生产的各个方面，从系统规范的早期阶段到系统投入使用后的维护。) *In this definition, there are two key phrases:*
 - *1. Engineering discipline Engineers make things work. They apply theories, methods and tools where these are appropriate [...] Engineers also recognize that they must work to organizational and financial constraints. [...]* 工程师让事情顺利进行的工程学科。他们在适当的地方应用理论、方法和工具。工程师还认识到，他们必须在组织和财务限制下工作。
 - *2. All aspects of software production Software engineering is not just concerned with the technical processes of software development but also with activities such as software project management and with the development of tools, methods and theories to support software production.* 软件生产的所有方面。软件工程不仅涉及软件开发的技术过程，还涉及软件项目管理等活动，以及支持软件生产的工具、方法和理论的开发。

软件工程三要素

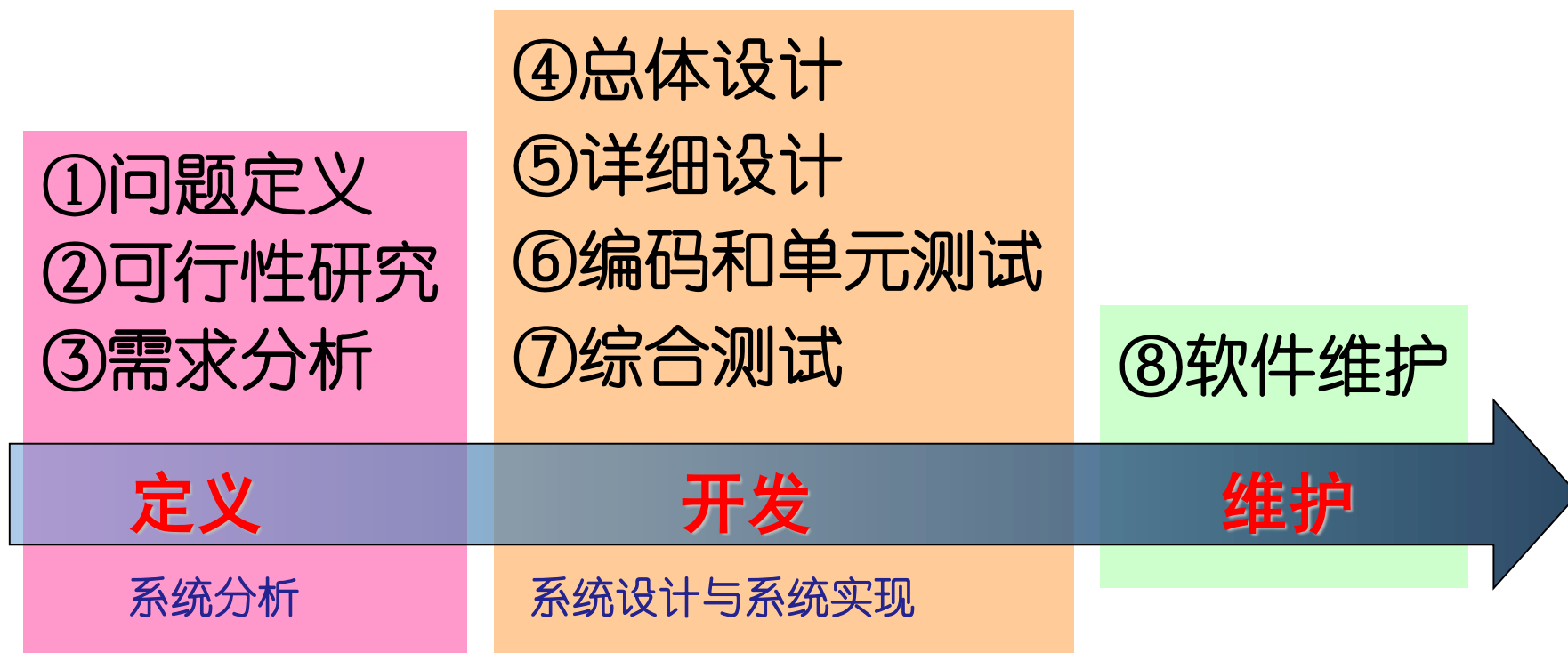
软件工程三要素：方法、工具和过程



- 软件工程方法学methodology：在软件生命周期全过程中使用的一整套技术方法的集合，也称为范型(paradigm)
- 传统方法学 和 面向对象方法学

软件生命周期

- 一般来说，软件生命周期由软件定义、软件开发、和软件维护 3 个时期组成，每个时期又进一步划分成若干个阶段



软件生命周期

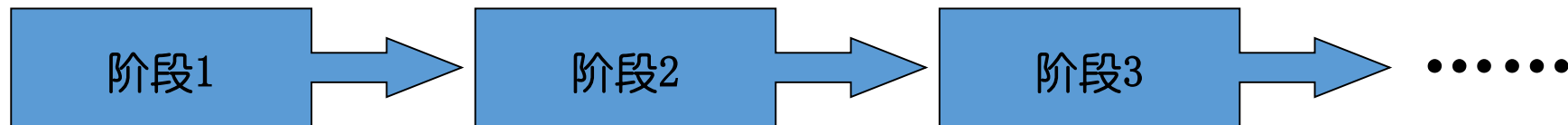
阶段	关键问题	结束标准
问题定义	问题是什么？	关于规模和目标的报告书
可行性研究	有可行的解吗？	系统的高层逻辑模型数 数据流图 成本/效益分析
需求分析	系统必须做什么？	系统的逻辑模型 数据流图 数据字典 算法描述
总体设计	概括地说，应该如何解决这个问题？	可能的解法 系统流程图 成本/效益分析 推荐的系统结构 层次图或结构图
详细设计	怎样具体地实现这个系统？	编码规格说明 HIPO图或PDL
编码和单元测试	正确的程序模块	源程序清单 单元测试方案和结果
综合测试	符合要求的软件	综合测试方案和结果 完整一致的软件配置
维护	持久地满足用户需要的软件	完整准确的维护记录

1. 传统方法学

传统方法学也称为生命周期方法学或结构化范型。

它采用结构化技术完成软件开发的各项任务，并使用适当的软件工具或软件工程环境来支持结构化技术的运用。

这种方法学把软件生命周期的全过程依次划分为若干个阶段，然后顺序地完成每个阶段的任务



审查两阶段之间的文档

审查的目的：前一阶段的结束 后一阶段的开始

审查的内容：高质量的文档

文档的作用：

①通信的工具。

它们清楚准确地说明了到这个时候为止，关于该项工程已经知道了什么，同时奠定了下一步工作的基础。

②备忘录的作用。

如果文档不完整，那么一定是某些工作忘记做了，在进入生命周期的下一个阶段之前，必须补足这些遗漏的细节。

但软件需求经常是 模糊 的，或者 随时间而变化。

2. 面向对象方法学

对象

- 把**对象**作为融合了数据及在数据上的操作行为的统一的软件构件。程序是由对象组成的，程序中任何元素都是对象，复杂对象由比较简单的对象组合而成。也就是说，用对象分解取代了传统方法的功能分解。

类

- 把所有对象都划分成**类**。每个类都定义了一组数据和一组操作，类是对具有相同数据和相同操作的一组相似对象的定义。

继承

- 按照父类(或称为基类)与子类(或称为派生类)的关系，把若干个相关类组成一个层次结构的系统(也称为类等级)。在类等级中，下层派生类自动拥有上层基类中定义的数据和操作，这种现象称为**继承**。

封装

- 对象彼此间仅能通过发送消息互相联系。对象与传统数据有本质区别，它不是被动地等待外界对它施加操作，相反，它是数据处理的主体，必须向它发消息请求它执行它的某个操作以处理它的数据，而不能从外界直接对它的数据进行处理。也就是说，对象的所有私有信息都被**封装**在该对象内，不能从外界直接访问，这就是通常所说的封装性。

用面向对象方法学 开发软件的过程

是一个 主动地多次反复迭代的 演化过程

① 面向对象方法普遍进行的对象分类过程

支持从特殊到一般的归纳思维过程

② 通过建立类等级而获得的继承性

支持从一般到特殊的演绎思维过程

传统方法学强调：

自顶向下 顺序地 完成软件开发的各阶段任务

促进软件重用——面向对象方法的一个优点

用面向对象方法学开发软件

最终软件产品由许多较小的、相对独立的对象组成

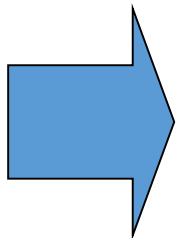
每个对象相当于一个微型程序

而且大多数对象都与现实世界中的实体相对应

因此，降低了 软件产品的复杂性

提高了 软件的可理解性

简化了 软件的开发和维护工作



① 对象是相对独立的实体

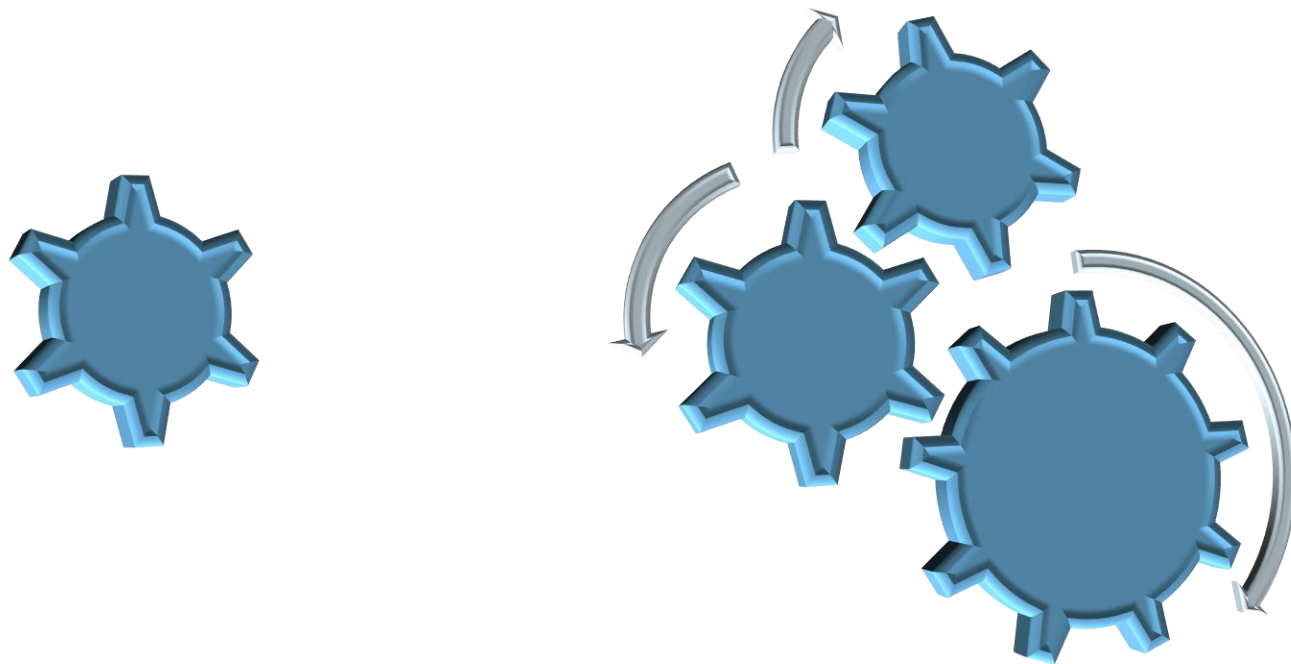
容易在以后的软件产品中重复使用

② 面向对象方法特有的继承性和多态性

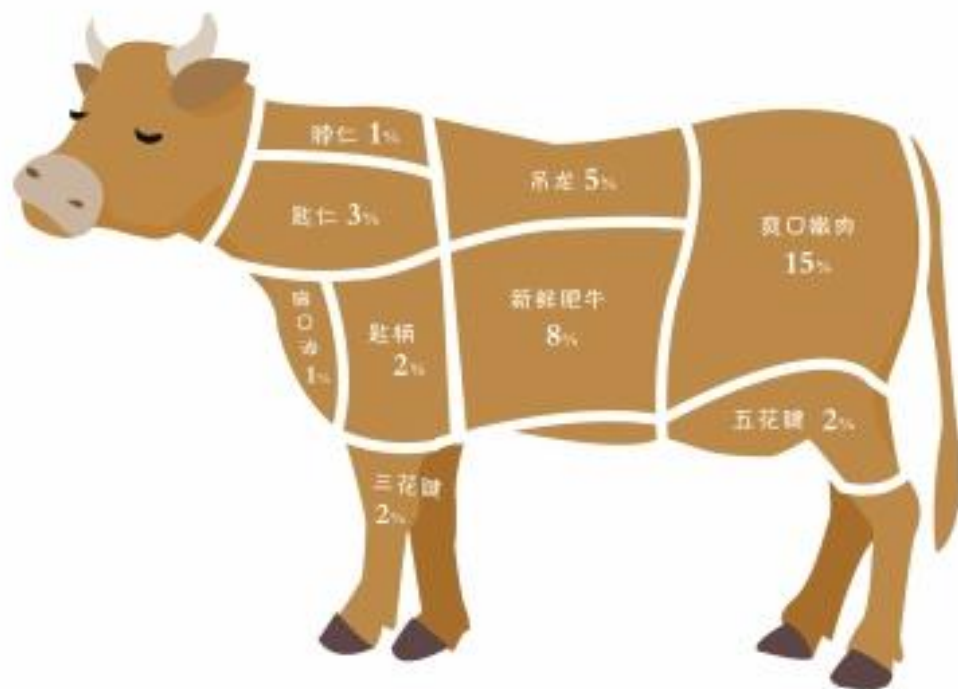
进一步提高了面向对象软件的可重用性

软件工程的本质特性

1. 软件工程关注于大型软件系统的构造



2. 软件工程的中心课题是控制复杂性

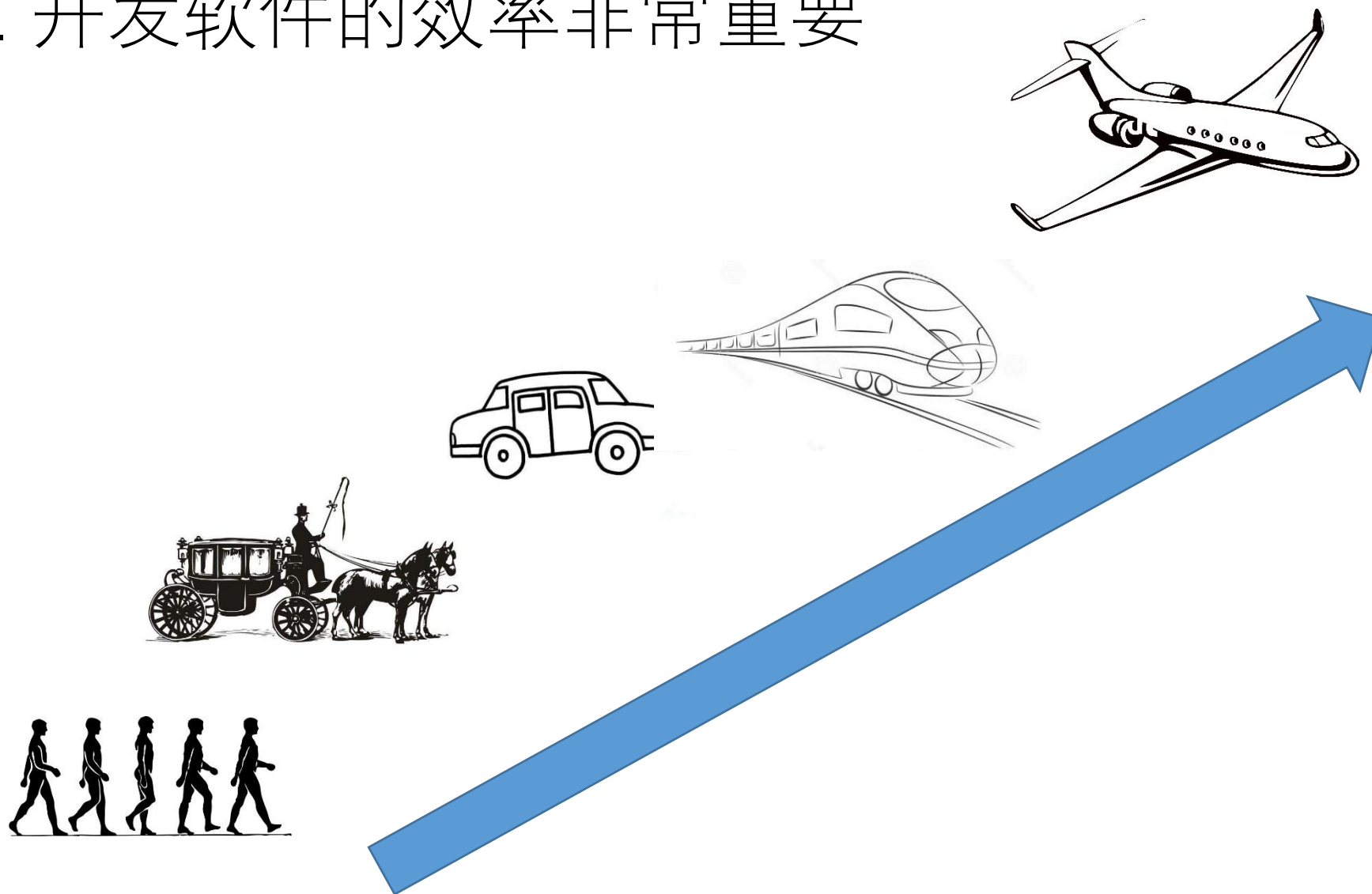


注意：许多软件的复杂性主要不是由问题的内在复杂性造成的，而是由必须处理的大量细节造成的。

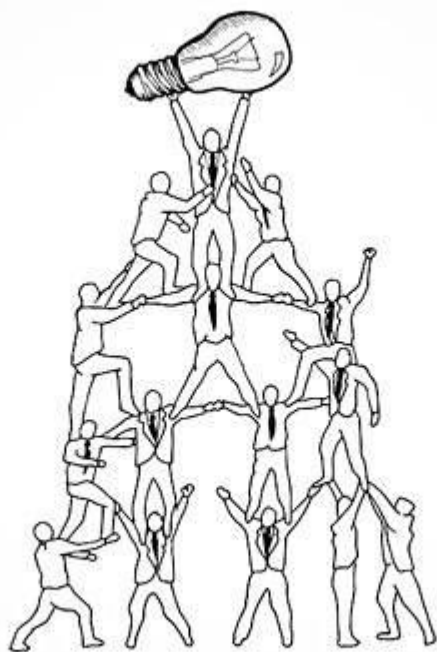
3. 软件经常变化



4. 开发软件的效率非常重要



5. 群体合作是开发软件的关键



6. 软件必须有效地支持它的用户

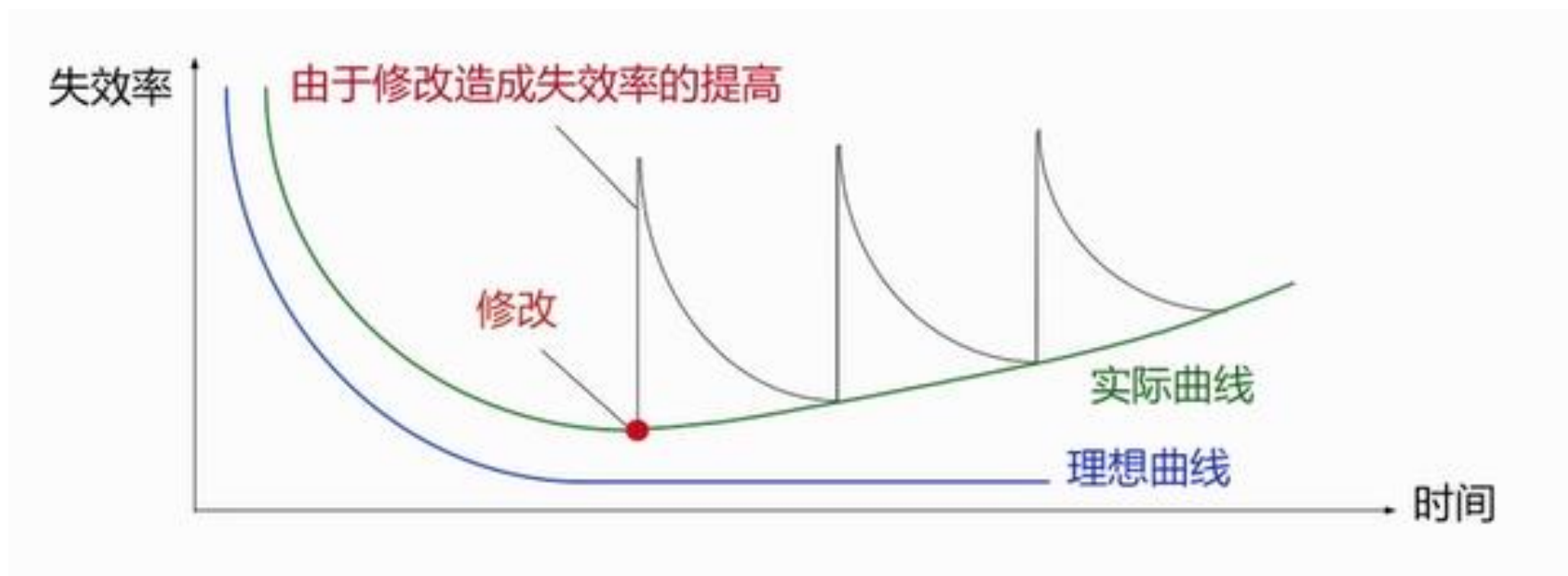


7. 在软件工程领域中是由具有一种文化背景的人为具有另一种文化背景的人提供应用系统

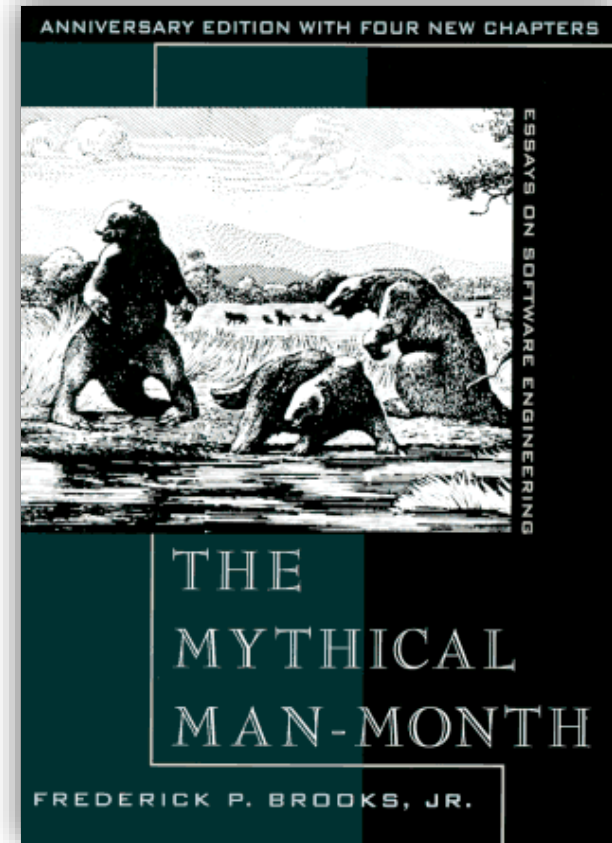
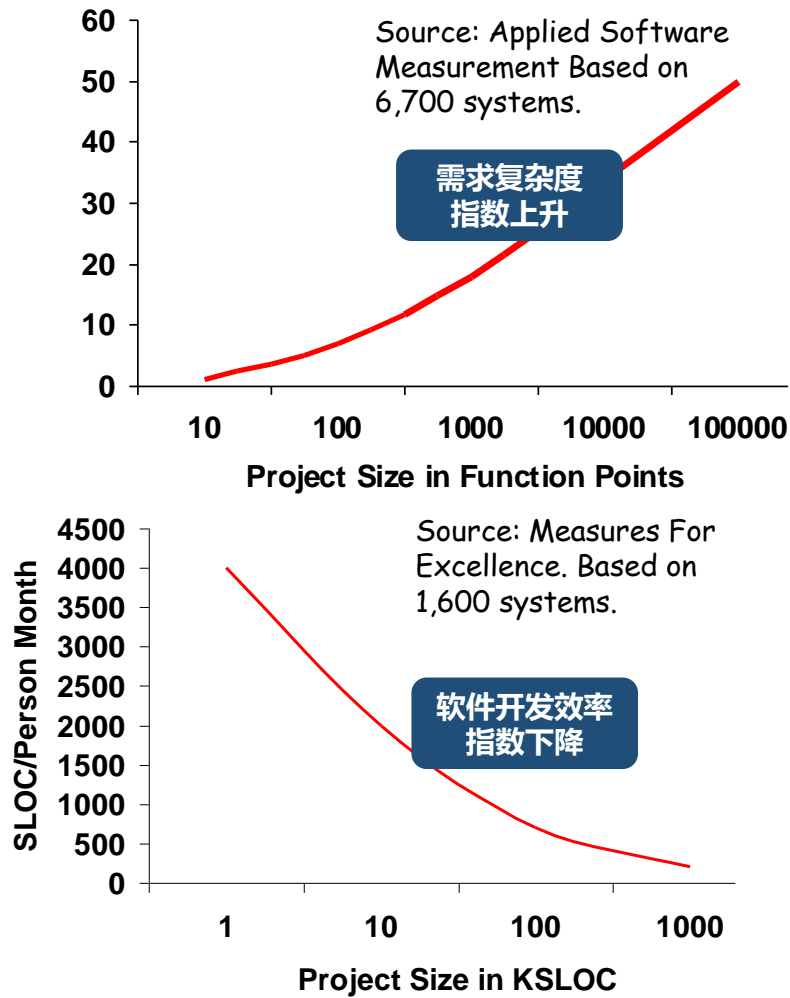


从制造产业到服务产业

- 软件在一开始被认为是制造产业，但是，难以用销售价值决定软件价值，因为75%的资源都体现在维护上。
- 软件逐渐被认为是服务产业。



软件开发的本质复杂性



“No Silver Bullet”

需求复杂、群体协同复杂 – 社会技术系统的根本复杂性

软件的复杂性：需求的复杂性



1990年4月10日，在伦敦地铁运营过程中，司机还没上车，地铁列车就驶离车站。当时司机按了启动键，正常情况下如果车门是开着的，系统就应该可以阻止列车启动。当时的问题是司机离开了列车去关一扇卡着的门，但当门终于关上时，列车还没有等到司机上车就开动了。

2021年7月20日，郑州暴雨，地铁启动后发现积水暴涌，“列车长尝试把车开回海滩寺，但因为地铁的自动保护设计，开不了，铁轨上都爆出火花了。”

平原公子赵胜

7-21 11:04 来自 荣耀20 PRO

+关注

#郑州地铁五号线#有媒体发了一篇郑州地铁五号线被困人员的口述报道，基本能还原整个过程：

- 1、大部分人20日下午出门时，并没有意识到会发生如此猛烈的降水和洪涝，很多人觉得地上积水也不多，就正常出门了。
- 2、口述者下午乘坐地铁回家，一路都有人上地铁，所有没有觉得有什么问题。
- 3、事故在海滩寺站和沙口站之间发生的，乘客能从车窗看到车外水急剧上涌，列车长尝试过把车开回海滩寺，但因为地铁的自动保护设计，开不了，铁轨上都爆出火花了。
- 4、水开始涌进车厢，当时的路段是倾斜的，车尾地势高，大家聚集在车尾，后来列车长尝试过组织自救，指挥大家往车头走，打开第一节车厢的门，通过列车内部的人行通道出去.....一部分人走了出去。
- 5、但是人行通道非常窄，人又多又挤，很多人就被迫退回了车厢，列车长只能关闭车门打电话求救，大家待援。
- 6、大家虽然有人恐慌、哭泣，但是还是保持了很好的组织和秩序，有个姑娘不断安慰大家，维持秩序，安抚情绪，劝大家不说丧气话，保存体力。还有人一直在通报救援队的情况，说他们已经到了外面，正在垒沙袋，拉绳子。
- 7、水位越来越高，车窗外已经一人多高，车厢里的水也渐渐漫过胸口、脖子，很多人体力不支、缺氧、低血糖，车厢里还有老人、孩子、孕妇.....很多人已经在发消息交代身后事了。
- 8、有人冲动要砸开车窗，被一位大叔制止，因为外面的水位更高，如果水涌进来，伤亡会更多。
- 9、出现了转机，因为水流的冲击，车厢发生了偏移，一边高一边低，往上翘的一面，车窗露出了水面，于是大家提议，砸开那块玻璃透气，在大家的努力下砸开了车窗，缺氧状况得到了明显的好转。
- 10、正在大家积极用灭火器继续破窗的时候，救援队出现在车厢外面。他们最先通过凿开的那处玻璃窗，将破窗器递到车厢内。车厢内里面大家也在接力将破窗器向后传递。救援人员把车长车厢打开了，大家排队陆续出去，孕妇、女人、孩子先走。
- 11、救援人员在外面拉，车厢里的群众互相帮助，一个个把同伴从水中托起来.....脱离险情后走了十来米，就到了安全的地方，大家互相搀扶着往前走.....但是还看到许多穿着制服的人向着反方向逆行。

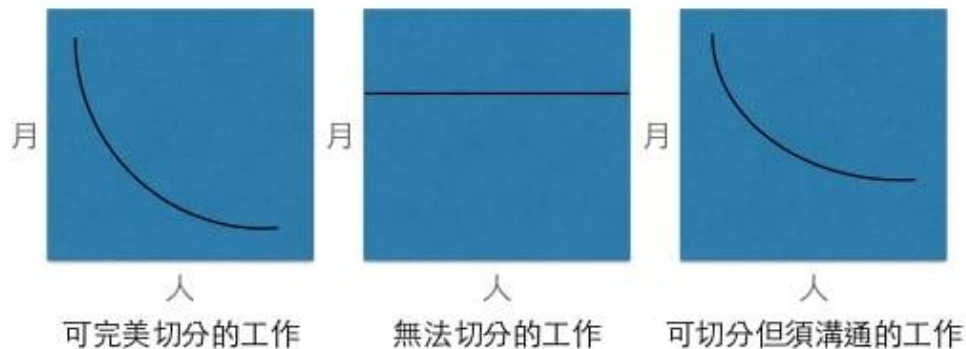
总结一下，五号线被困地铁上的乘客、车长其实努力过自救，只是情况实在是太复杂太危急，才会选择原地待援.....这趟列车上的群众表现出来极高的组织力和整体素质，但还是不幸有12人死亡，5人受伤。

Brooks法则： 协同的复杂性

- 向进度落后的项目中增加人手，只会使进度更加落后。 - 《人月神话》

人、月 之外的變因

1. 是否可完全切分：
生小孩要9個月，10個媽媽生也是要9個月
2. 是否需要溝通：
訓練成本與交流成本也應計入工作量



人月神话

什么是“人月神话”

- 一个人在一个月内完成的工作→“人月”
- 想通过增加人手来缩短工程时间→只是一个“神话”

如何处理“人月神话”的困境

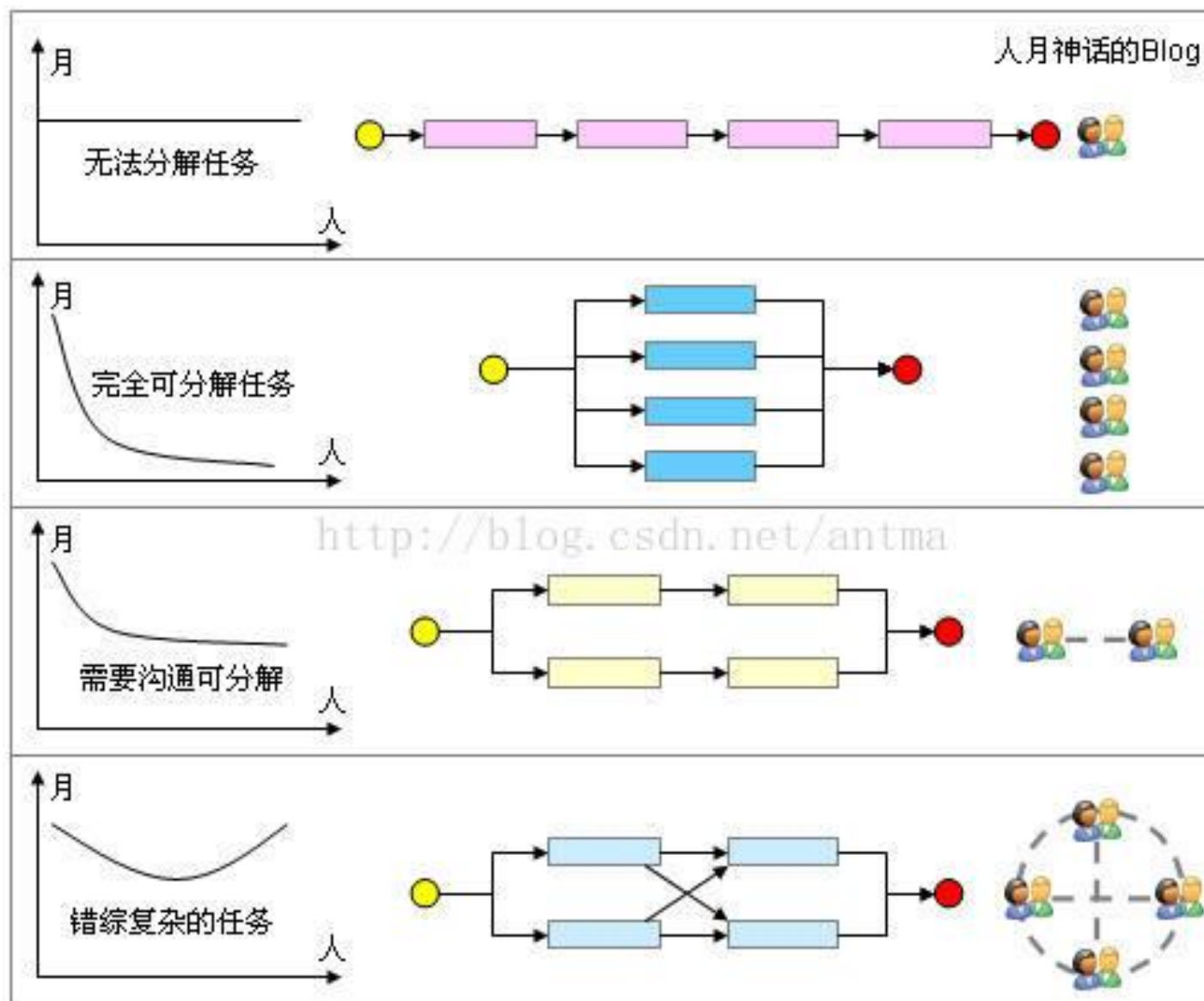
- 以小团队的方式开展工作
- 进行行之有效的沟通
- “防微杜渐”
 - 设立“里程碑事件”
 - 设立专门的规划和控制小组
 - 灵活设计系统
 - 提前集成测试

“人月神话”为何无法彻底解决

- 计算机技术的发展太快了
- 软件研发本身的内在特性
 - 需求的复杂度
 - 组织管理的复杂度
 - 计算机技术的复杂度

改善软件开发根本困境的建议

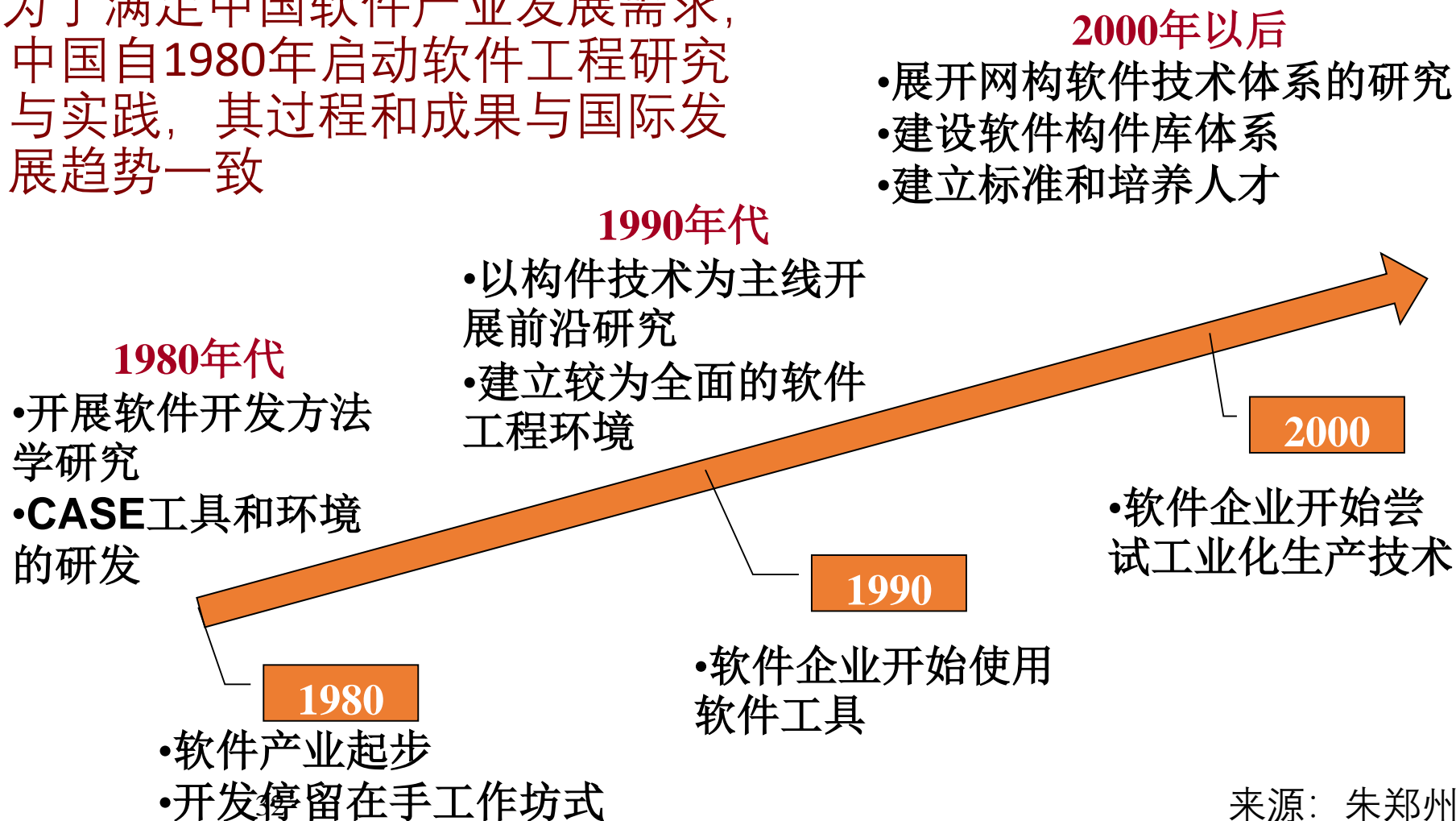
- 采用新技术
- 快速开发原型
- 聘用卓越人才



来源网上

中国软件工程的发展历程

为了满足中国软件产业发展需求，中国自1980年启动软件工程研究与实践，其过程和成果与国际发展趋势一致



鸣谢：
朱郑州
王涛