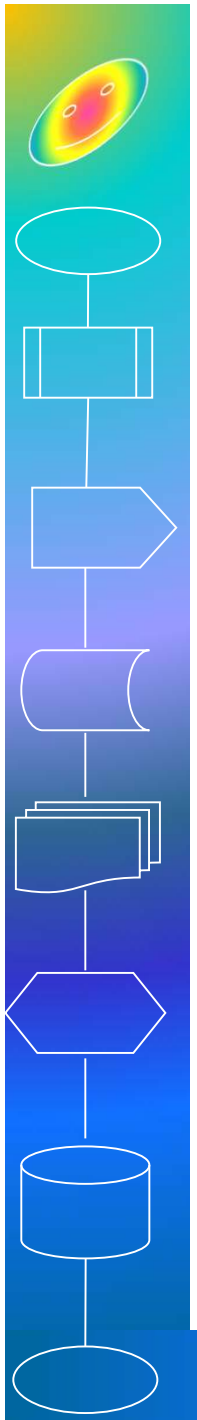
“软件工程课程”的目标

- 软件工程课程与教材是“对如何用工程化的过程、方法、技术、工具等，在时间、资源、成本等约束下，分析、设计、建造、交付高质量软件”的经验和理论总结。
- “No Silver Bullet” ---是软件工程领域的名言。不能像学习“微积分课程”那样，给出每个软件项目的精确解决方案。
- 必须将“软件工程化理论方法”用于实践，并从实践中总结和提炼理论。
- 你不能说读了某本教材或书，就懂得“软件工程”了。
  - 根据对计算机和软件专业的本科毕业生的调查：
    - 在校期间：“软件工程”是感觉最没用的课程之一
    - 两年以后：请告诉师妹师弟们：“软件工程是最有用的课程之一”。
- 本课程是“软件工程导论”，将大家引如软件工程领域，走出“工程化”方法的第一步。

上课要求

- 教材《软件工程化》清华大学出版社，前20章
  - 分为重点讲解和自己阅读
- 签到！
- 每堂课会随机问题，作为平时成绩一部分
  - 抽查出勤率
  - 掌握情况
- 作业要求用手写，作为平时成绩一部分
- 期中考试
- 期末考试



第1章 绪论

From Computing Science to Software Engineering

软件无处不在---国防

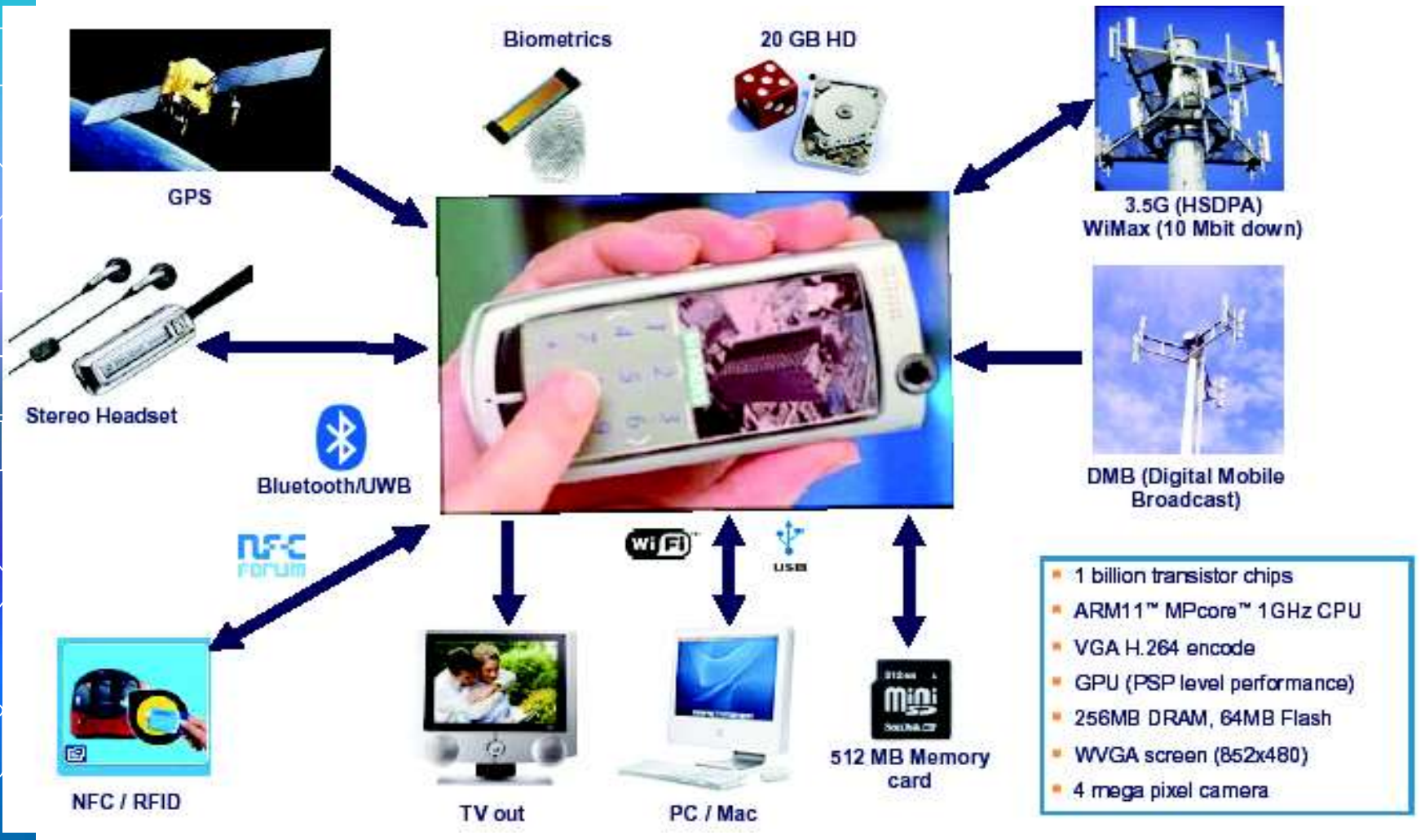
Communication，
Control、
Command、
Information

软件无处不在---航空/航天


图说中国载人航天
2003年
发射神舟五号
航天员上天
2005年
发射神舟六号
多人多天飞行
2008年
发射神舟七号
航天员出舱
2011年11月1日
发射神舟八号
追踪飞行器，
与天宫一号进
行无人对接
2012年
神舟九号、神舟十号
与天宫一号进行有人
对接，建立短期有人
照料空间实验室
1
载人飞船阶段，实现
多人多天飞行、安全
返回和精确落点
2
空间实验室阶段
出舱、对接
2011年9月29日
发射天宫一号
目标飞行器
3
2020年前后
建立一个60吨级的
永久性载人空间站
Chinanews.com

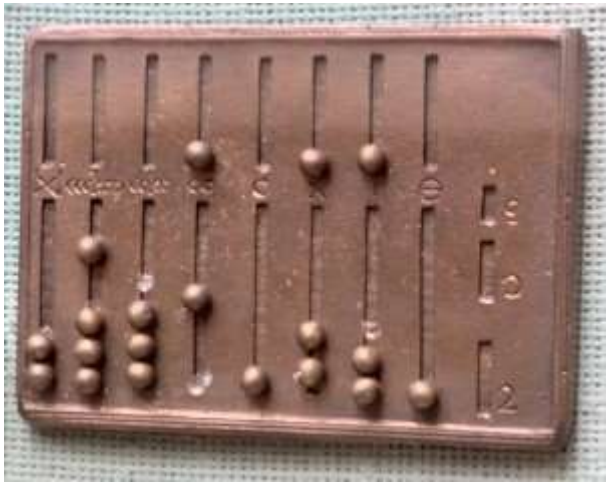
软件无处不在---日常生活

软件无处不在---移动通信

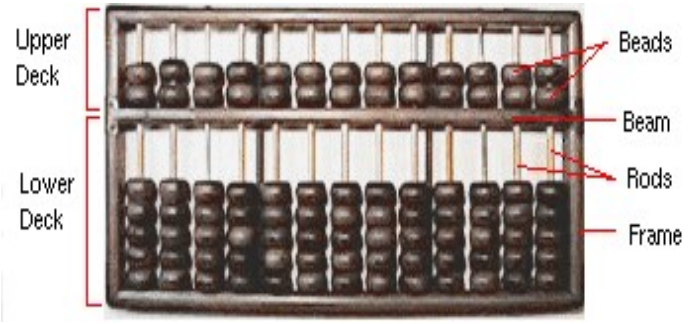

GPS
Biometrics
20 GB HD
3.5G (HSDPA)
WiMax (10 Mbit down)
Stereo Headset
Bluetooth/UWB
DMB (Digital Mobile Broadcast)
NFC / RFID
TV out
PC / Mac
512 MB Memory card
1 billion transistor chips
ARM11™ MPcore™ 1GHz CPU
VGA H.264 encode
GPU (PSP level performance)
256MB DRAM, 64MB Flash
WVGA screen (852x480)
4 mega pixel camera

目录

- 算盘——对一个数学问题，只有确定其可用算盘解算时，此问题才是可解的。——古代中国学者

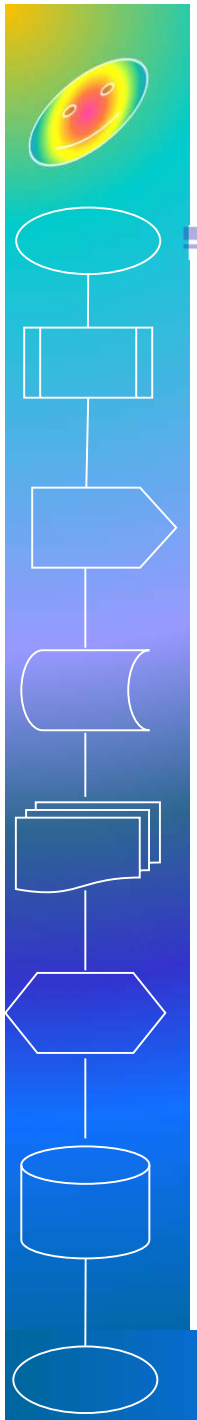


A reconstruction of a Roman abacus in the Cabinet des Médailles, Bibliothèque nationale, Paris.



The Chinese **Standards Abacus**

Japanese soroban

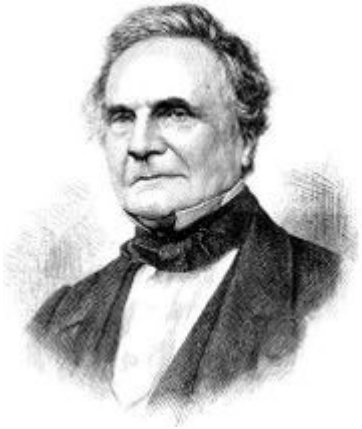


计算机器(computing machinery)的发展历史

- 1275年，西班牙神学家雷蒙德.露利(R.Lullus)发明一种思维机器（“旋转玩具”）
- 1641年法国人帕斯卡(B.Pascal)利用齿轮技术做成加法器
- 1673年，德国人莱布尼茨（G.W.V.Leibniz）在此基础上制造了能加、减、乘、除的计算机器。
- 19世纪30年代，英国人巴贝奇（C.Babbage）设计了用于计算对数、三角函数和其它算术函数的“分析机”；
- 20世纪20年代，美国人布什（V.Bush）研制了能解一般微分方程组的电子模拟计算机。

The work of Babbage

- Charles Babbage(巴贝奇) (1791 –1871)
  - 英国数学家、分析哲学家、机械工程师
  - 原型计算机科学家(proto-computer scientist）
    - 建立可编程计算机 (a programmable computer)思想.
  - 其未完成的计算机存放在 London Science Museum.
  - 1991年，依据 Babbage的原始设计计划，工程师们制造出来差分机(difference engine), 功能完整
  - 验证了在9世纪Babbage的机器是能工作的。



Difference engine of Babage

Part of Babbage's Difference engine, assembled after his death by Babbage's son, using parts found in his laboratory

The London Science Museum's replica difference engine, built from Babbage's design.

http://en.wikipedia.org/wiki/Difference_engine

Babage in British Science&History Museum

目录

- 手工计算到机械化计算机器
- 图灵理论计算机
- 电子实现的自动计算机器
- 集成电路
- 从程序到软件
- 软件艺术
- 软件危机与工程侧面
- 软件产业
- 计算科学与软件工程
- 软件工程历程

数学形式化体系

- 德国数学家康托尔（G.Cantor,1845～1918）从1874年开始，发表一些列的集合论的著作，创立了集合论。
- 希尔伯特纲领
  - 将每一门数学的分支形式化，构成形式系统和理论，并在此为对象的元理论即元数学中，证明每一个形式系统的相容性，从而导出数学的相容性。
  - 实质是寻找通用的形式逻辑系统，该系统应当是完备的，即，该系统中可以机械地判断任意给定命题的真伪。
  - 于1900年8月8日在巴黎第二届国际数学家大会上，提出了新世纪数学家应当努力解决的23个数学问题

大卫·希尔伯特

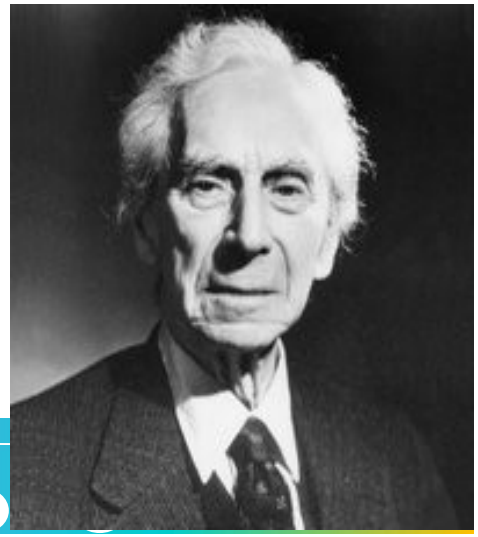
D.（David Hilbert，1862～1943）

罗素悖论

- 1901年罗素（B.Russell）在集合论概括原则的基础上发现“罗素悖论”——数学史上的第三次危机。
  - $S=\{x \mid x / \in S\}$，定义S中元素x不属于S。
  - 理发师宣布：给且只给村里那些不自己刮胡子的人刮胡子
  - => 理发师自己给自己刮胡子<=>理发师自己不给自己刮胡子

In 1950, Russell was made a Nobel Laureate in Literature(诺贝尔文学奖)
贡献："in recognition of his varied and significant writings in which he champions humanitarian ideals and freedom of thought".

Bertrand Arthur William Russell, 3rd Earl Russell, OM, FRS (18 May 1872 – 2 February 1970), was a British philosopher, logician, and mathematician, working mostly in the 20th century.



哥德尔(Kurt Gödel)

- Kurt Gödel (IPA: [kurt gø:dl]) (April 28, 1906 Brno, then Austria-Hungary, now Czech Republic – January 14, 1978 Princeton, New Jersey) was a logician, mathematician, and philosopher of mathematics.

- Gödel is best known for his two incompleteness theorems, published in 1931 when he was 25 years of age, and only one year after finishing his doctorate at the University of Vienna.

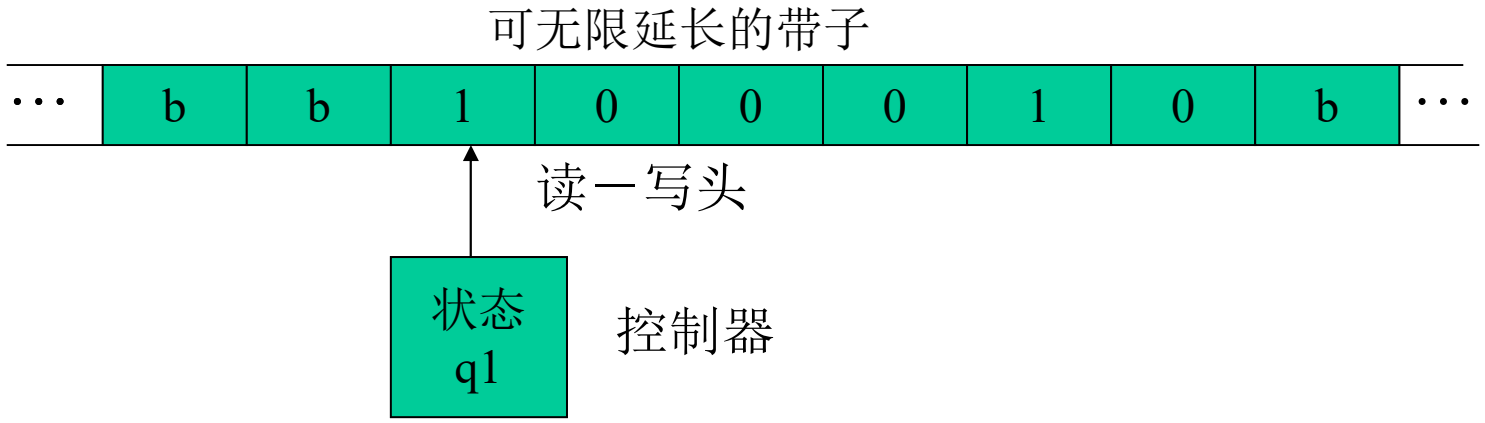
- The more famous incompleteness theorem states that for any self-consistent recursive axiomatic system powerful enough to describe the arithmetic of the natural numbers (Peano arithmetic), there are true propositions about the naturals that cannot be proved from the axioms.

- To prove this theorem, Gödel developed a technique now known as **Gödel numbering**, which codes formal expressions as natural numbers.

- 希尔伯特纲领的研究基础是逻辑和代数——19世纪英国数学家乔治.布尔（G.Boole）的逻辑代数体系
  - “真”、“假”，和“与”、“或”、“非”
- 1931年，奥地利25岁的歌德尔（K.Godel）提出的关于形式系统的“不完备性定理”中指出，这种形式系统是不存在的。宣告了著名的“希尔伯特纲领”失败。
- “希尔伯特纲领”
  - 保存全部古典数学的前提下去排除集合论的悖论，给数学基础问题带来了全新的转机。
  - 希尔伯特纲领的提出使元数学得到了发展
  - 对计算机科学而言，希尔伯特纲领的失败启发人们应避免花费大量的精力去证明那些不能判定的问题，而应把精力集中于解决具有能行性的问题。

图灵理论计算机

- 图灵对计算本质的揭示
  - 在歌德尔研究成果的影响下，20世纪30年代后期，图灵(A.M.Turing)从一般计算过程入手，对计算的本质进行了研究。



一个有穷字母表:$\{S_0,S_1,S_2,\ldots,S_p\}$

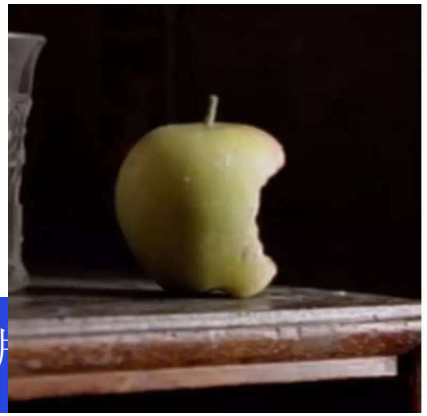
机器的控制状态为: $\{q_0,q_1,q_2,\ldots,q_m\}$

Alan Turing

- Alan Mathison Turing (June 23, 1912 – June 7, 1954) was a British mathematician, logician, and cryptographer. Turing is often considered to be the father of modern computer science.
- In 1952, Turing was convicted of acts of gross indecency after admitting to a sexual relationship with a man in Manchester. He was placed on probation and required to undergo hormone therapy.
- When Alan Turing died in 1954, an inquest found that he had committed suicide by eating an apple laced with cyanide.

http://en.wikipedia.org/wiki/Alan_Turing

Alan Turing is often considered the father of modern computer science.



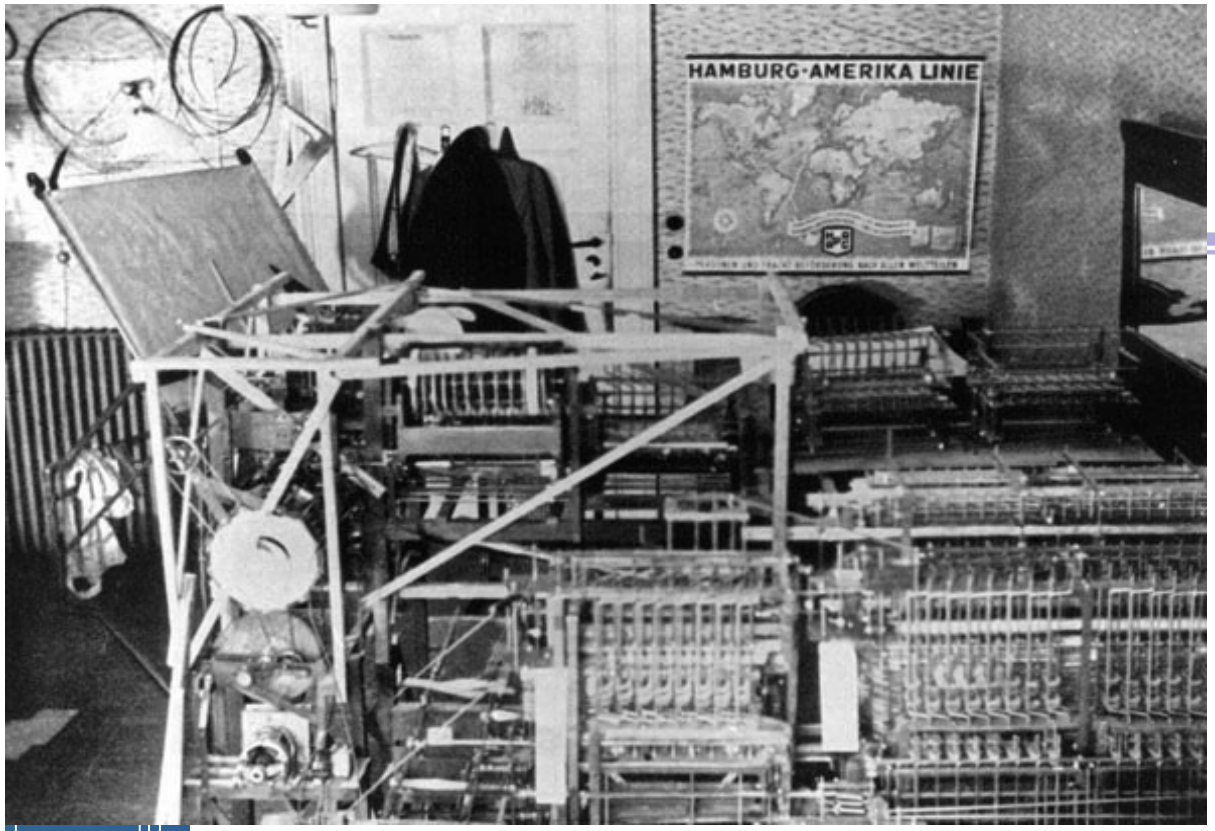
a (one-tape) Turing machine can be formally defined as a 7-tuple

- $M = \langle Q , \Gamma , b , \Sigma , \delta , q\,0 , F \rangle$, where
- Q is a finite, non-empty set of states;
- Γ is a finite, non-empty set of tape alphabet symbols;
- $b \in \Gamma$ is the blank symbol (the only symbol allowed to occur on the tape infinitely often at any step during the computation);
- $\Sigma \subseteq \Gamma \setminus \{ b \}$ is the set of input symbols, that is, the set of symbols allowed to appear in the initial tape contents;
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{ L , R \}$ is a partial function called the transition function, where L is left shift, R is right shift. (A relatively uncommon variant allows "no shift", say N, as a third element of the latter set.) If δ is not defined on the current state and the current tape symbol, then the machine halts;
- $q\,0 \in Q$ is the initial state;
- $F \subseteq Q$ is the set of final states or accepting states. The initial tape contents is said to be accepted by M if it eventually halts in a state from F.
- -------Following Hopcroft and Ullman (1979, p. 148),

目录

- 手工计算到机械化计算机器
- 图灵理论计算机
- 电子实现的自动计算机器
- 集成电路
- 从程序到软件
- 软件艺术
- 软件危机与工程侧面
- 软件产业
- 计算科学与软件工程
- 软件工程历程

- 图灵的成果
  - 可计算性＝图灵可计算性
  - 算法（也称为能行方法或能行过程），是对解题（计算）过程的精确描述，由一组定义明确，且能机械执行的规则（语句、指令）组成。
  - 任一过程是能行的（能表现在一个算法中），当且仅当它能被图灵机实现。
- 其它计算机器模型
  - 递归函数论，
  - λ 演算，
  - Post规范系统（E.L.Post）
- The Church-Turing Thesis. *Every effectively computable number-theoretic function is recursive(computable) and vice versa.*
- 这些计算模型在计算能力上是等价的。



Z1---- the first electro-mechanical binary programmable computer

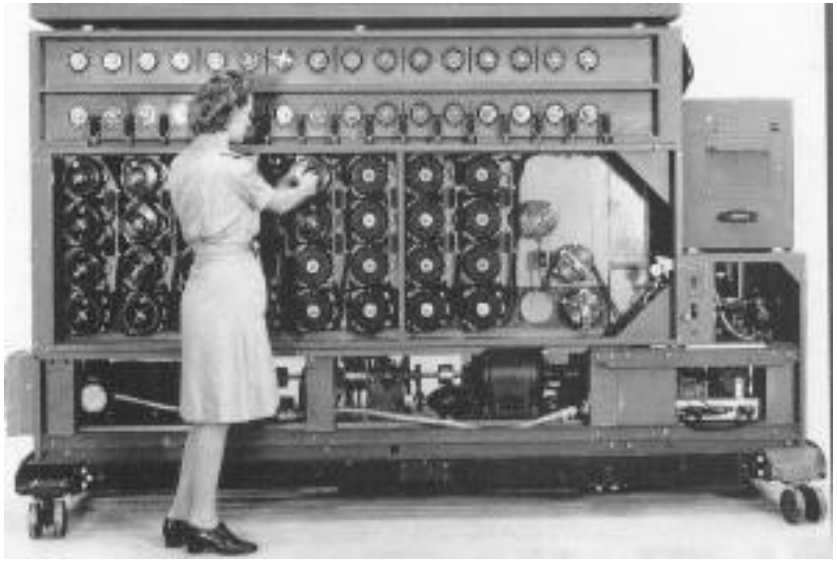
- 1936年，德国人Konrad Zuse 在父母的居住室开发了Z1机器，是第一台 electro-mechanical binary programmable computer.
- The Z1 had 64-word memory (each word contained 22 bits) and a clock speed of 1 Hz.
- Both programming and output were generated using punch tape with a specific reader.

机电计算机

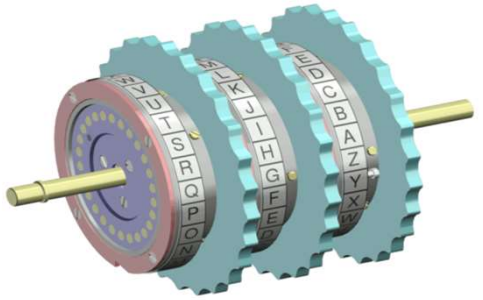
•In 1939, the German military commissioned Zuse to build the Z2, which was largely based on the Z1.
•Later, he completed the Z3 in May of 1941, the Z3 was a revolutionary computer for its time and is considered the first electromechanical and program-controlled computer.

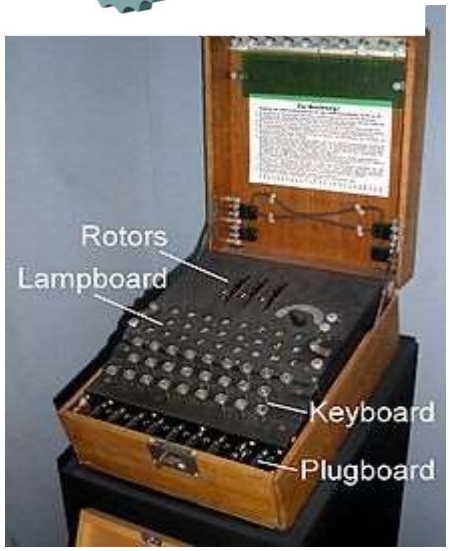
•Finally, on July 12, 1950, Zuse completed and shipped the Z4 computer, which is considered to be the first commercial computer.

The Turing-Welchman bombe



Turing-Welchman bombe

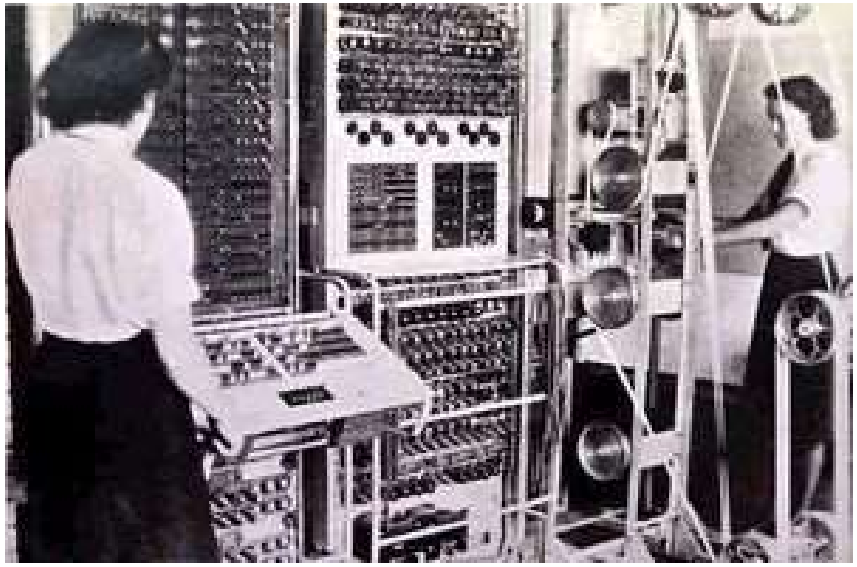




恩尼格玛密码机（德语：Enigma，又译哑谜机）

电子计算机

- 1943年，英国的“巨人”（Colossus）的计算机投入运行，用于译解德国密码，英政府1970年前一直保密。
- 1945年，ENIAC诞生—现代意义上的计算机



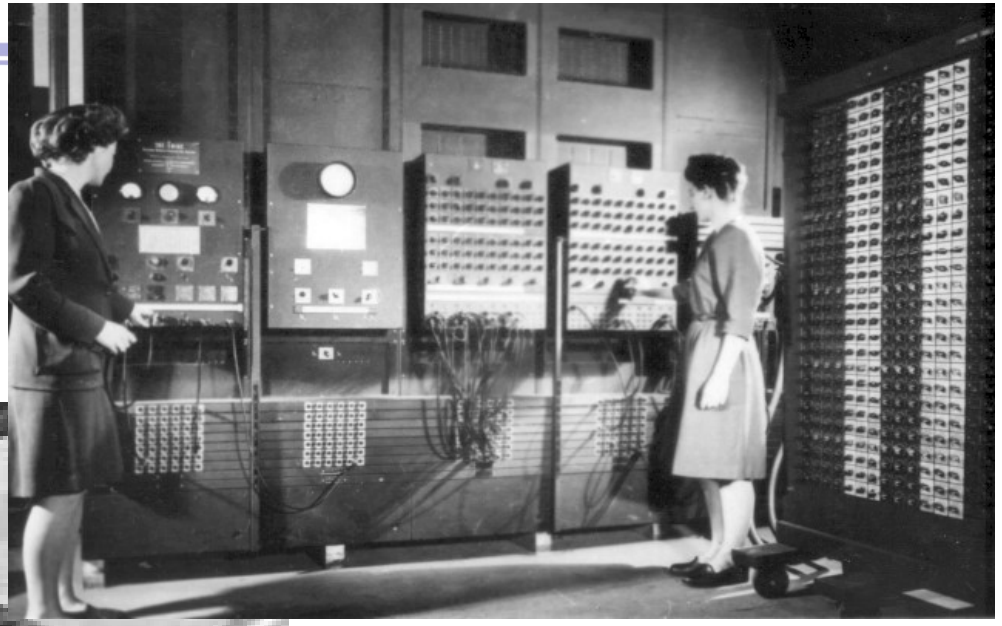
The Colossus machines were early computing devices used by British codebreakers to read encrypted German messages during World War II.
Colossus was an early electronic digital computer.

电子计算机的大规模生产和应用

- 与Babbage机械计算装置相比
- 无法形成大规模的工业化应用，直到晶体管线路出现以后，用电子线路构造的计算机才进入工业化使用阶段。
- 推动电子计算机装置最有效的技术是“集成电路”----由于集成电路可以集成成千上万个电路器件，降低了能耗，提高了硬件的可靠性，从而使集成电路计算机为主体的计算机得到普遍。

ENIAC计算机

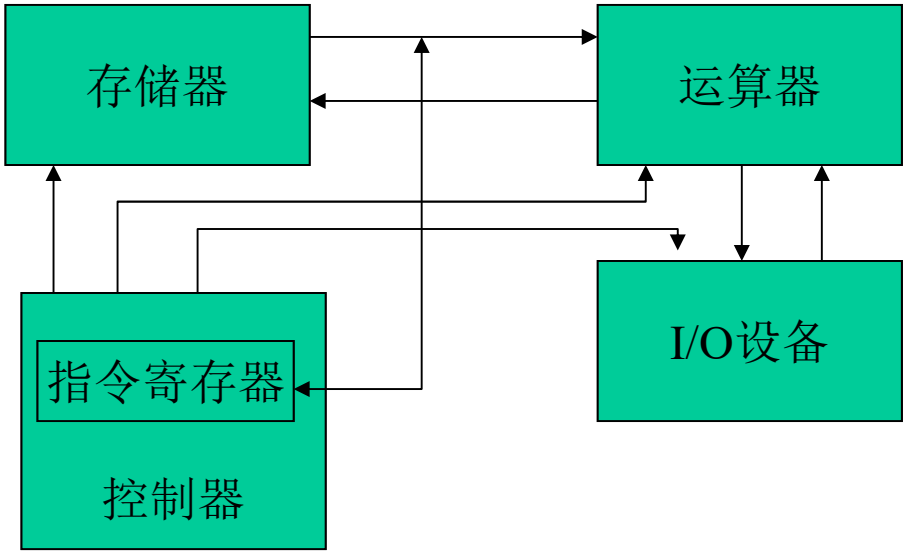
第一台通用ENIAC计算机采用的是真空管实现了高低电平分别表示“0”和“1”，并能够运行起来了。



ENIAC占用的空间很大，耗电也很多。

冯.诺依曼型计算机

- 1946年，美国数学家冯.诺依曼(Von Neumann)和同事完成《电子计算装置逻辑结构设计》的研究报告，给出了由控制器、运算器、存储器和I/O设备组成的存储程序式计算机的组织结构。奠定了现代计算机的基础。



John von Neumann
in the 1940s

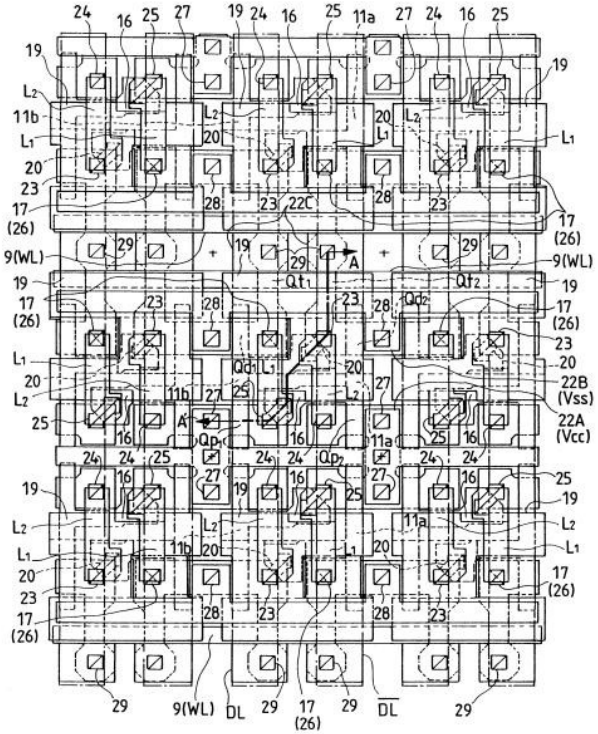
目录

电子计算机的大规模生产和应用

- ENIAC无法形成大规模的工业化应用，直到晶体管线路出现以后，用电子线路构造的计算机才进入工业化使用阶段。
- 推动电子计算机装置最有效的技术是“集成电路”----由于集成电路可以集成成千上万个电路器件，降低了能耗，提高了硬件的可靠性，从而使集成电路计算机为主体的计算机得到普遍。

semiconductor integrated circuit

FIG. 1



摩尔定律

- 1965年Intel公司创始人之一Gordon Moore预测：

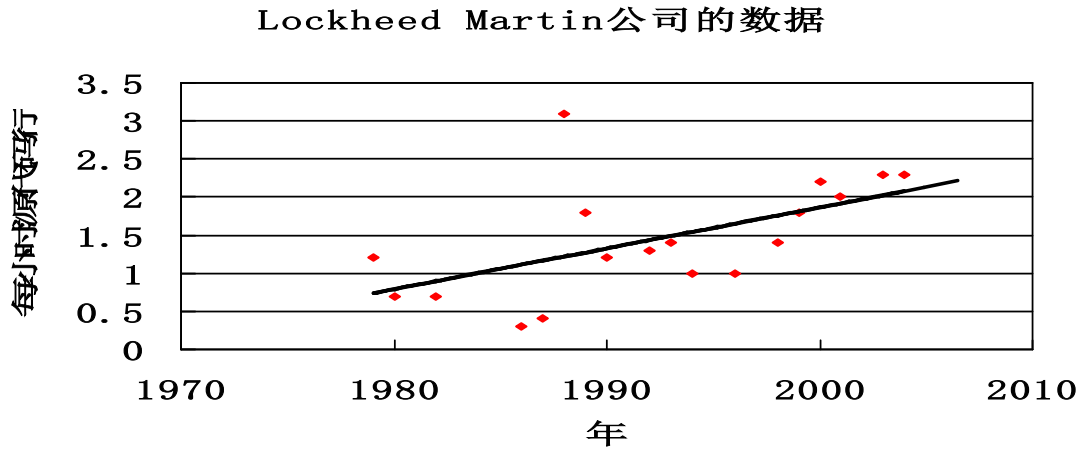
  “集成电路中的集成密集度每两年翻一番”。

- 事实上，1971年，Intel的4004微处理器有2300个晶体管。2004年的Intel® Itanium® 2 processor (9MB cache)有592,000,000个晶体管。

- 集成电路工业界认为摩尔定律至少还能持续50年。
  - 摩尔预测还可以坚持到2022年！

软件工程人才需求---软硬件生产率对比

- 计算机硬件生产率---Moore定律：
  - 1965年，Intel公司创始人之一Gordon Moore预测：“集成电路中的集成密集度每两年翻一番”
    - 事实上，1971年，Intel的4004微处理器有2300个晶体管。2004年的Intel® Itanium® 2 processor (9MB cache)有592,000,000个晶体管。
    - 集成电路工业界认为摩尔定律至少还能持续50年
- 软件生产率



目录

通过穿孔实现织布花纹的纺织机----程序

能够通过“读”穿孔卡上的信息完成预定的任务的织布机

编程的历史

- 软件是由计算机程序和程序设计的概念发展演化过来的。是程序和程序设计发展到一定规模后并且逐步商品化的过程中形成的。
- 19世纪初，法国人约瑟夫.雅各(Josephe Marie Jaquard)设计的织布机，就能够通过“读”穿孔卡上的信息完成预定的任务。
- 英国诗人拜伦(Byron)的女儿，数学家爱达.奥古斯塔.拉夫拉斯伯爵夫人(Ada Augusta Lovelace) 在帮助巴贝奇研究分析机时，指出分析可以向织布机一样进行编程，并发现进行程序设计和编程的基本要素，被认为是有史以来的第一位程序员，而著名的计算机语言Ada就是以此命名的。

Ada Lovelace

- Augusta Ada King, Countess of Lovelace (December 10, 1815 – November 27, 1852) is mainly known for having written a description of Charles Babbage's early mechanical general-purpose computer, the analytical engine.
  - Ada was the only legitimate child of the poet Lord Byron and his wife, Annabella Milbanke. She was named after Byron's half-sister, Augusta Leigh, by whom he was rumoured to have fathered a child.

• On *December 10, 1980, (Ada's birthday),* the U.S. Defense Department approved the reference manual for its new computer programming language, called "Ada".

• The U.S. Department of Defense Military Standard for *Ada (MIL-STD-1815)* was assigned a number to commemorate the year of her birth.

Ada Lovelace

程序→软件

- 1960年代，随着计算机硬件的批量生产，工业界和学术界认识到了计算机程序的工程和使用价值。
  - 一方面计算机机程序必须随着硬件一起销售，仅仅向客户提供硬件不足于支持计算机的使用，即，计算机程序具有复制价值；
  - 另一方面，计算机程序的开发过程不仅仅是上来就写程序，往往需要花费大量的时间搞清需求，花力气进行算法设计，在编程后还要对程序进行测试，以及向用户提供使用手册和文档，也就是说，计算机程序的是一种由多人合作、经历不同阶段的开发，且具有可复制和重复使用的器或件(ware)。
- 借助于器和件的概念，例如，瓷器(Chinaware)、铁器等概念，人们把计算的电子线路等人眼可见、占据物理空间的器或件称为硬件(hardware)，把计算机的程序和相关数据的集合，这些肉眼不可见的、逻辑器或件称为软件(Software)。

程序→软件

- 软件是一个概念或逻辑实体，由计算机程序、过程和相关的操作文档组成。软件直接依靠硬件或依靠其他软件执行它所实现的程序功能。
- 软件(software)与硬件(hardware)直接对应。与硬件的物理实体相对比，软件是无形的(intangible)，即，“不可触摸到的”。
- 通常可以把软件分为两类：系统软件和应用软件。
  - 系统软件告诉计算机如何工作，
  - 应用软件告诉计算机如何完成用户特定的工作。
- 软件具有其商业价值

目录

程序设计艺术

- 在计算机中运行的程序是计算机指令序列集合，告诉计算机执行特定的任务。
  - 第一种形式是可执行程序(executable)，是在计算机直接执行的指令集合；
  - 第二种是人可读的源代码，源代码可以转换出(例如，经过编译器)可执行的程序。
- 程序设计是一种艺术。
  - 精美的算法和代码是计算科学工作者所追求的主要目标之一。
  - 在1968年之前，科学界认为计算机程序仅仅是一门“科学或艺术”。
  - 计算机程序的“科学和艺术”特征表现在其独创性，以及不需要重复劳动，而一个数学定理的第一次证明具有科学价值，随后的证明只具有学习价值了。

程序的艺术价值

- 计算机程序具备艺术创造性的特征，只有第一次的创造具有价值和成本，其后的复制几乎是无成本的。
- 然而与艺术品不同的是，复制的代码同样具有使用价值。实际上，人们创作计算机程序艺术品的目的是其使用价值，而非欣赏价值。

- 书画的赝品/软件盗版和抄袭
  - 失去艺术价值
- 软件必须创新
  - 用法、算法、界面等

目录

软件工程侧面

- 书画家是个人行为，编程序也是个体劳动
- 但是，微软Windows 8系统有3千多万行代码
  - 一个人或几个人如何做的出来？
  - 洛克希德公司，每个程序员每小时可生产2.5行代码。
  - 3千万行的代码，要多少人？多少年？才能生产出来。
  - 生产出来，能不能用？
- 软件必然是一项人(脑)力密集的劳动，是工程行为

软件工程侧面----软件危机

- 60年代末多项大型软件以失败告终,例如:
  - IBM公司的OS/360,
  - 美国空军的后勤系统(2.17亿美金),
  - Univac联合航空订票系统 (5600万美金)
- 软件出现危机:OS/360负责人Brooks ……像巨兽在泥潭中垂死挣扎,挣扎得越 猛,泥浆就沾得越多, 最后……

首次软件工程会议

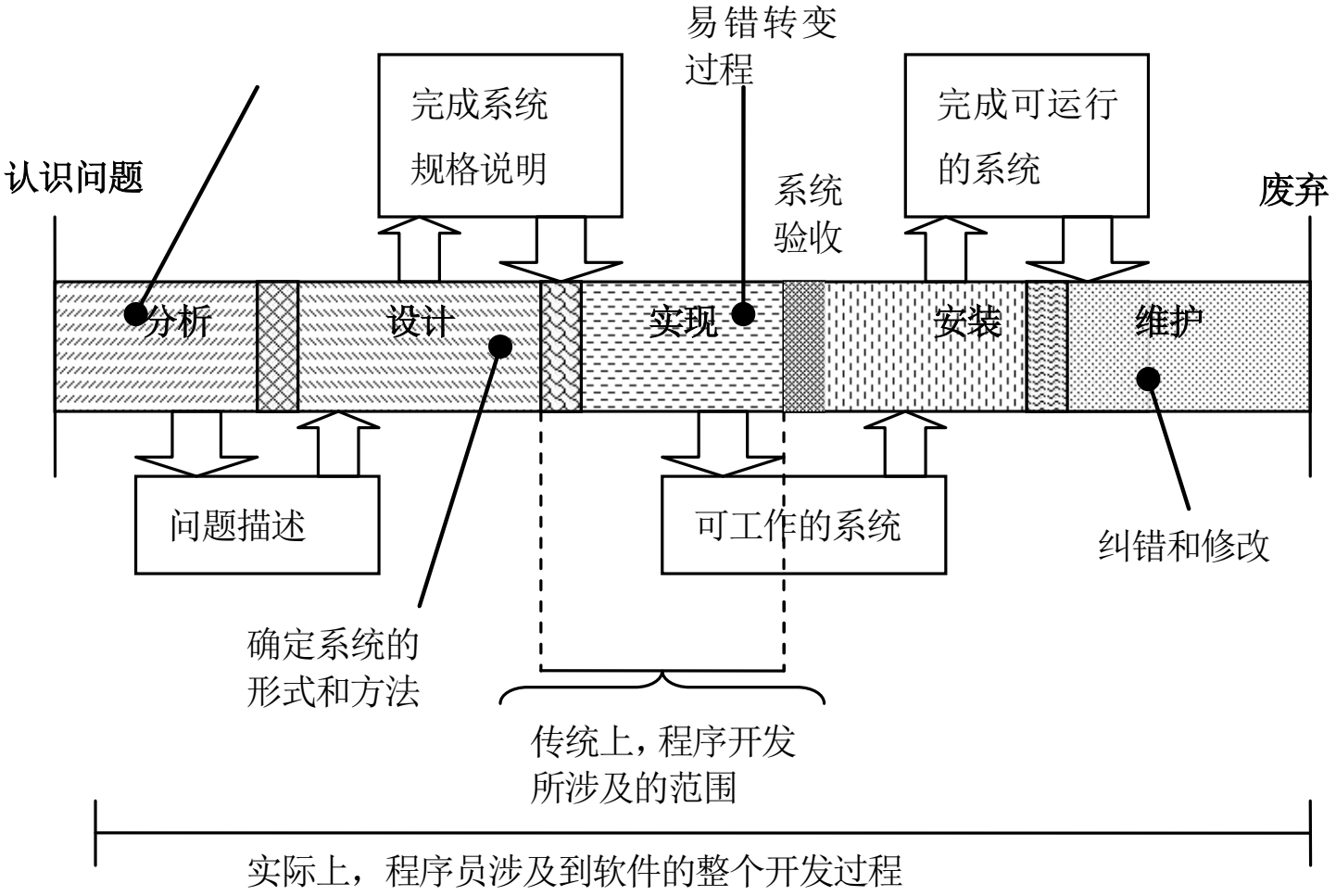
- 1968年NATO赞助的软件工程会在德国召开，与会学者和工业界的代表形成了一个会议总结报告，分别出从：
  - 1）软件工程与社会
  - 2）软件设计
  - 3）软件生产
  - 4）软件服务，以及
  - 5）特别专题等方面讨论了软件工程。
- 在特别专题中提出了：a)软件面临的问题和可能解决方法，b）教育问题，c)软件价格问题。
- 这次会议标志着从“计算机程序艺术”到“软件工程”观念上转变。

Born of Software Engineering

- In the belief that software design, implementation, and maintenance could be put on same footing as traditional engineering disciplines, a NATO study group in 1967 coined the term *software engineering*.
- The claim that building software is similar to other engineering tasks was endorsed by the 1968 NATO Software Engineering Conference held in Garmisch, Germany.

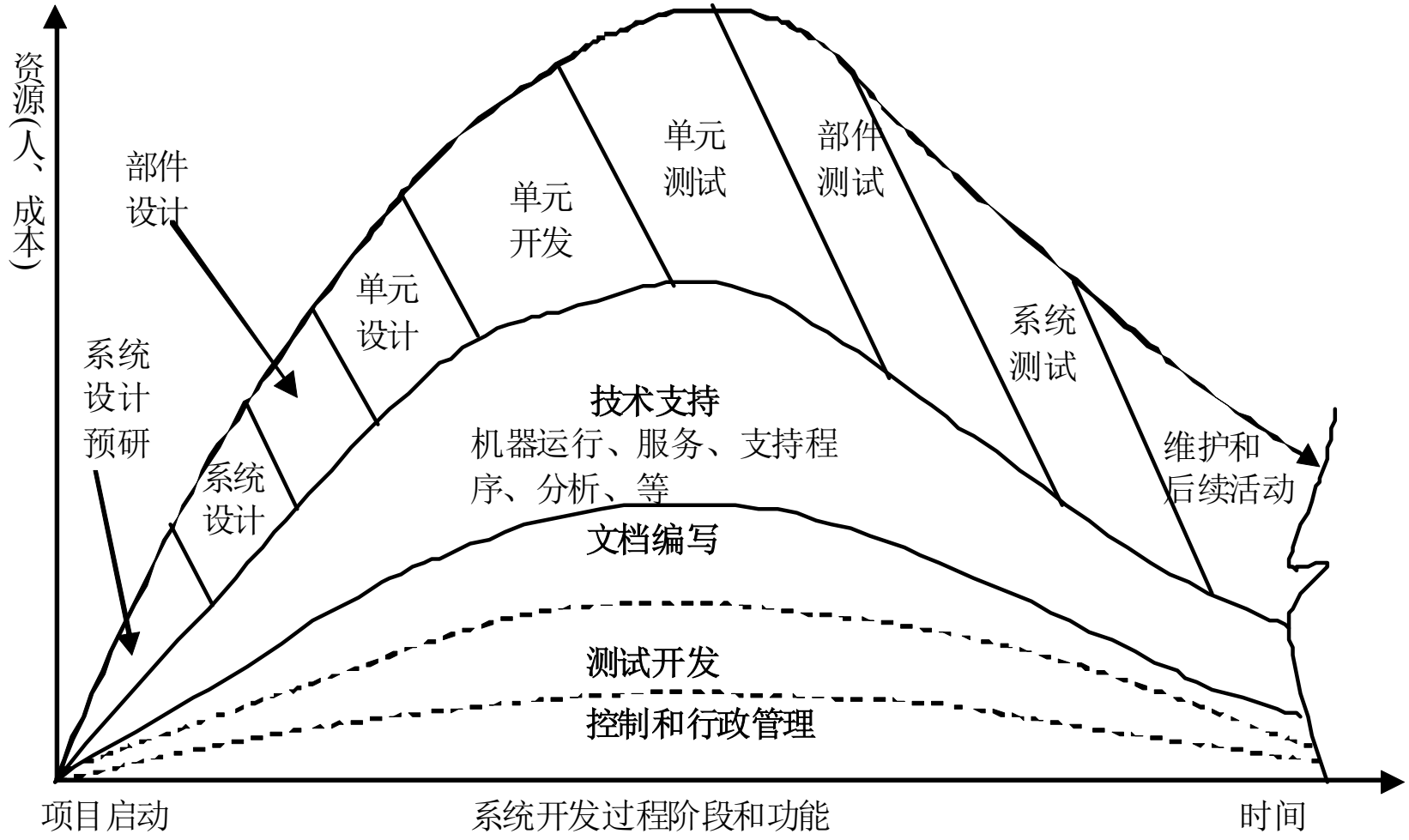
[Naur,Randell, and Buxton 1976]

全面的程序开发(软件系统建造)过程

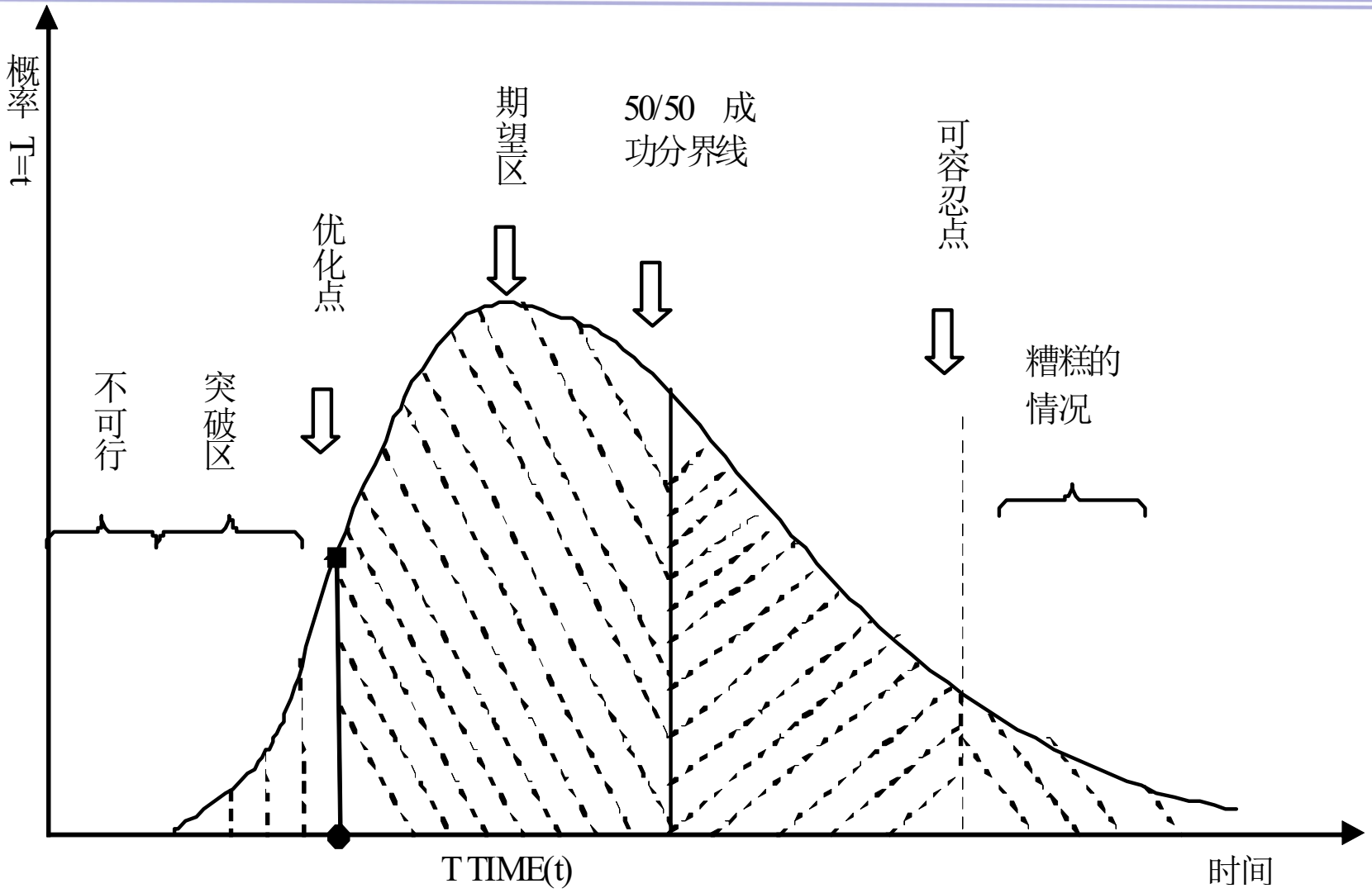


1968年---Slige “程序开发与软件开发”

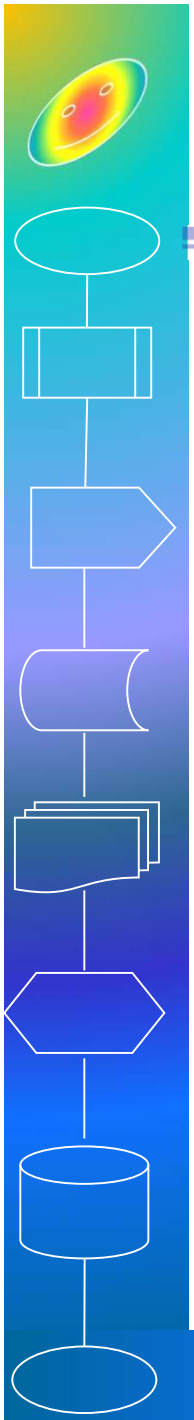
一个软件项目的实际过程



可能的软件项目统计曲线



目录



- 手工计算到机械化计算机器
- 图灵理论计算机
- 电子实现的自动计算机器
- 集成电路
- 从程序到软件
- 软件艺术
- 软件危机与工程侧面
- 软件产业
- 计算科学与软件工程
- 软件工程历程

软件产业化

- 1960年中期IBM主导了计算机产业的几乎全部份额。1969年美国司法部启动了对IBM的垄断诉讼。
- 该诉讼声称IBM违反“谢尔曼法”---垄断或企图垄断通用电子数字计算机系统市场，特别是主要为企业设计的计算机。案件拖到1982年美国司法部终于结束诉讼。
- 虽然是无果而终，但是这场诉讼却影响了整个软件产业。反垄断导致IBM公司决定把软件和硬件分离出来单独定价，结束了IBM在1969年前的“捆绑式”的软件、硬件销售和服务。那时，客户不需要支付软件或服务价格，但却需要支付非常高的硬件价格，而软件按源代码的形式提供。

软件商品交换----许可证制度

- 软件许可证是一个法律协议，规定了专有的或无偿使用许可形式，也是软件生产者和软件用户之间的合同备忘录。
  - 用户可能是任何法律实体或“最终用户”，在这种情况下，软件许可证，常称为最终用户许可协议（EULA--End User License Agreement）指定生产者授予给用户的软件时间和权限。

计算机软件产业的历史

- 计算机软件产业开始于20世纪50年代，随着计算机在商业、国防、教育等领域的使用的迅速增加，导致对程序设计人员需求的增长。出现一部分具有计算机程序设计经验的人分离出来专门从事程序设计工作，并创立自己的程序设计服务公司，根据用户的订单提供相应的程序设计服务。
- 如1955年，Elmer Kubie和John W.sheldon创建的计算机使用公司（CUC）。
- 1959年创立的应用数据研究（ADR）公司。
- 1968年Martin Goetz获得世界上第一个软件专利；
- 1969年春，就IBM垄断软件产业提出诉讼，促使IBM在1969年6月30日宣布结束一些软件和硬件的捆绑销售，为软件产品单独定价。
  - 在这一时期成立的软件公司有美国计算机公司（CCA）、Information Builder、Oracle公司等。

目录

Chemical engineering(化工) v.s Chemistry(化学)

- 案例1：将煤炭转变汽油。
  - 二战期间，德国人就已经完成了实验，从化学家的观点看，他们已经成功了。
  - 从化学工业的角度看，这些实验都是不成功的，因为没有经济和批量生产的价值。
  - 工程师必须探讨批量转变的实验和生产途径，从而将成本降低到具有实际的经济意义。

实验室可行 ≠ 工业化生产可行

理论上可行 ≠ 工程上可行

Chemical engineering(化工) v.s Chemistry(化学)

案例2：侯德榜制碱

- 1862年，比利时人索尔维（Ernest Solvay 1838—1922）发明了以食盐、氨、二氧化碳为原料制取碳酸钠的“索尔维制碱法”（又称氨碱法）。此后，英、法、德、美等国相继建立了大规模生产纯碱的工厂，并组织了索尔维公会，对会员以外的国家实行技术封锁。
- 制碱的主要原料是食盐，也就是氯化钠，而四川的盐都是井盐，要用竹筒从很深很深的井底一桶桶吊出来。由于浓度稀，还要经过浓缩才能成为原料，这样食盐成本就高了。
- 另外，索尔维制碱法的致命缺点是食盐利用率不高，也就是说有30%的食盐要白白地浪费掉，这样成本就更高了，所以侯德榜决定不用索尔维制碱法，而另辟新路。
- 1920年，侯德榜先生毅然回国任职。他全身心地投入制碱工艺和设备的改进上，1924年8月,塘沽碱厂正式投产。1926年，中国生产的“红三角”牌纯碱在美国费城的万国博览会上获得金质奖章。产品不但畅销国内，而且远销日本和东南亚。

Software Engineering v.s Computer science

- An sight into the relationship between SE and CS can be obtained by comparing and contrasting the relationship between chemical engineering and chemistry.
  - 算法 ≠ 程序
  - 要把算法转换为可执行程序，实现自动计算
  - 要把用户的输入和期望看到结果，以易于使用的方式表现出来，让机器替代人的工作

Software engineering

- IEEE standard 610.12-1990
  - SE:
    - (1) The application of a systematic,disciplined,quantifiable approach to the development,operation,and maintenance of software; that is, the, the application of engineering to software.
    - (2)The study of approaches as in (1)
- Fritz Banuer
  - SE is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

Software Engineering

- Software engineering (SE) is the profession of people who create and maintain software applications by *applying technologies and practices from computer science, project management, engineering, application domains and other fields*.

- Software engineering deals with matters of cost and reliability, like traditional engineering disciplines. Some software applications contain millions of lines of code that are expected to perform properly in the face of changing conditions, making them comparable in complexity to the most complex modern machines. For example,
  - a modern airliner has several million physical parts (and the space shuttle about ten million parts), while the software for such an airliner can run to 4 million lines of code.

http://en.wikipedia.org/wiki/Software_engineering

目录

- 手工计算到机械化计算机器
- 图灵理论计算机
- 电子实现的自动计算机器
- 集成电路
- 从程序到软件
- 软件艺术
- 软件危机与工程侧面
- 软件产业
- 计算科学与软件工程
- 软件工程历程

软件工程历程

- 1950年代是产生软件工程理论的年代，软件开发人员向硬件工程师学习，产生了一些很好的实践方法，例如，桌面检查、好友互查、手工代码执行等。人们提出了如何通过向硬件工程师的学习，开展软件开发工作。
- 到了1960年代，软件是一个技能(crafting)的时代。由于软件代码及其容易修改，导致了“build and fix”的方法。软件开始成为人员密集型的劳动。
- “黑客文化”在美国的主要大学里迅速发展，产生了“牛仔式”的夜以继日的程序员，帮助满足项目的进度要求。
- 1970年代是形式化和瀑布过程，Dijistra的文章“考虑goto语句的有害性”引发了人们构造结构化的程序运动。从而导致“形式化方法”----通过数学证明或程序演算，集中关注程序的正确性，以及“首席程序员领导下的自顶向下的结构化编程”。

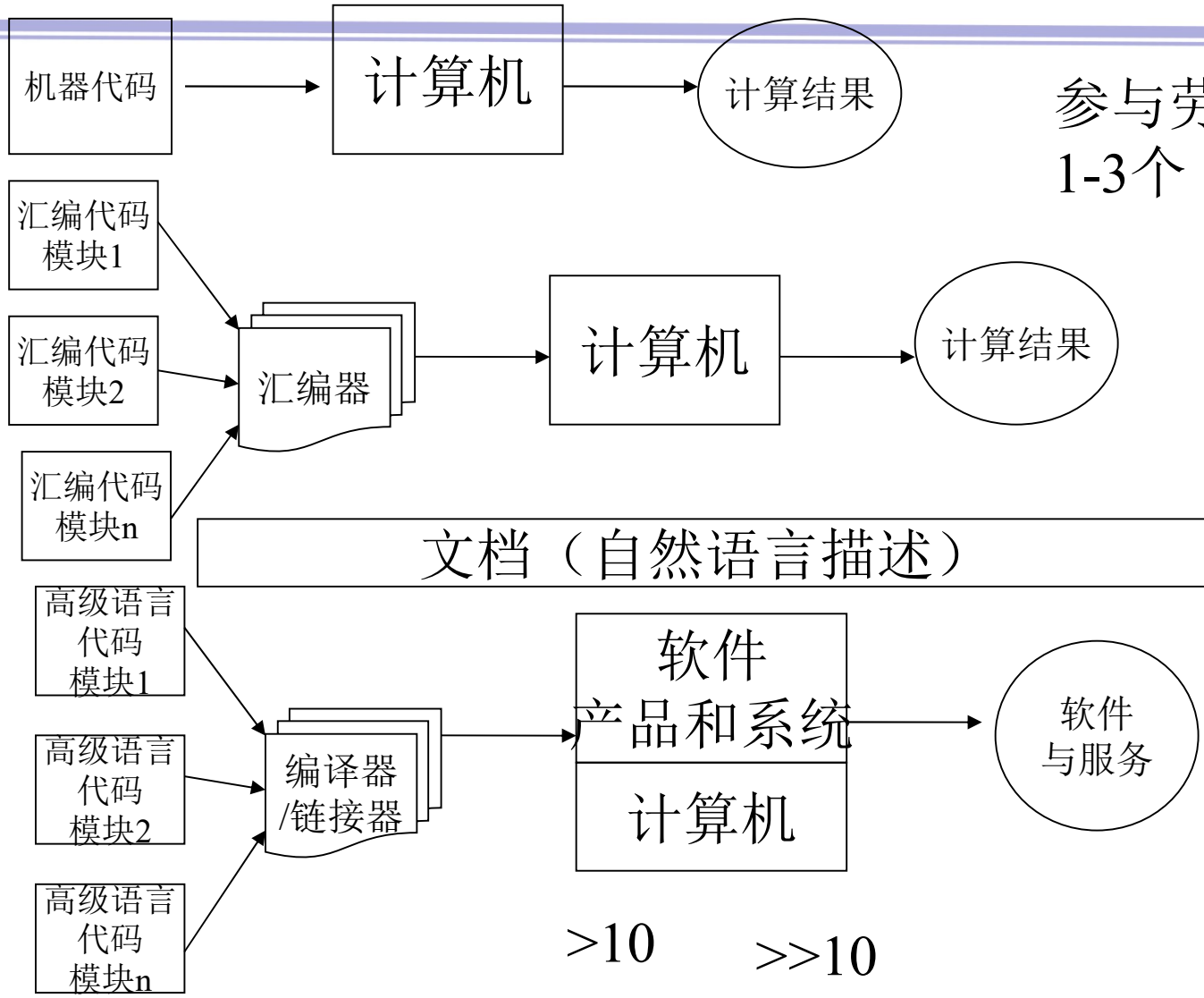
软件工程历程

- 1980年代强调的是软件生产效率和过程可测量性。从美国国防部门制定的软件开发标准(如DoD-STD-2167和MIL-STD-1521B)等开发过程的建立，到卡内基梅隆大学的软件工程研究所建立的提高承包商能力的能力成熟度模型(CMM)模型，以及ISO9000系列的质量体系建立，形成了以过程为中心的软件工程的运动。
- 1990年代是并发和顺序过程。面向对象的方法，UML、Web和Internet，基于开源码的开发、可用性和人机交互等技术推动了人们减少进入市场的时间，提出了从需求、设计、编码等的并行工程。
- 2000年代强调的是敏捷和增值方法。基于网络的信息共享与合作，以及IT公司的并购等，导致了敏捷方法的发展和“基于挣值的软件工程”。同时，将软件的关键性和可信性提到了重要的日程。另一方面，对货架上的商用软件、开源码和遗留系统的改造成为又一个软件开发的主题，导致了围绕商务服务的大量项目的发展。基于货架上的软件的开发，和面向各个行业的软件要求又进一步推动了模型驱动(Model-Driven)的开发。

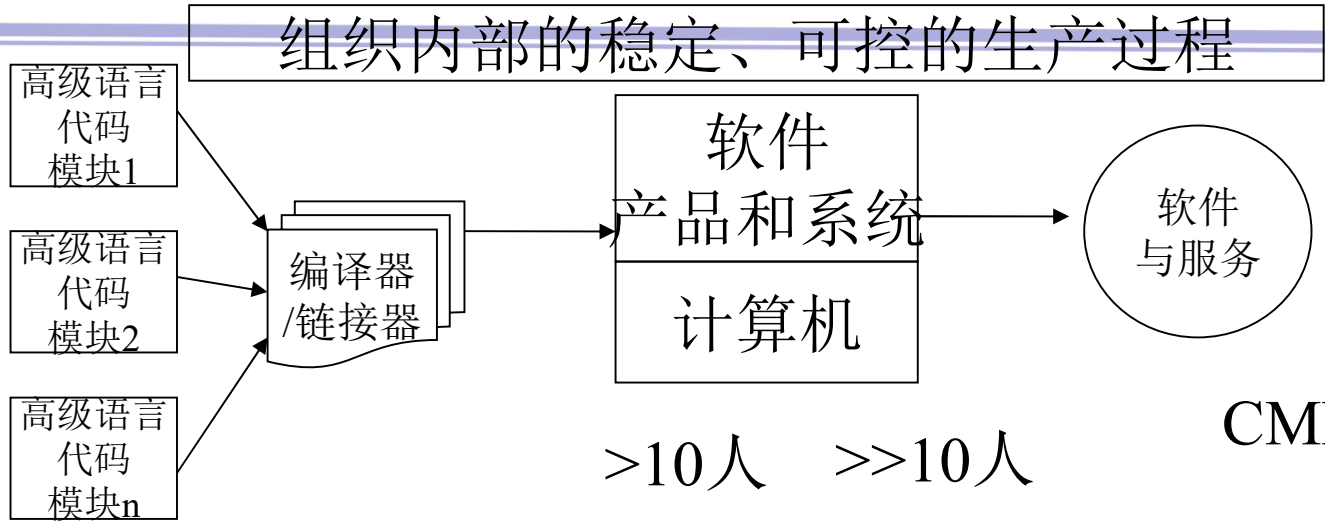
软件工程历程

- Barry Boehm进一步认为2010年到2020年，软件工程的重点是全球化和多系统的系统(System on Systems)。互联网和低费用、高带宽的通信为企业提供了网络经济的机遇。差异的薪酬为全球的劳务输出提供了交流的机会。这就需要建立多时区的、快速开发的方法，并对软件工程管理的可见性和控制力、通信交流、价值共享和信任等提出了挑战。
- 实际上，欧美于2004年提出了软件密级系统的概念，2006年提出软件“超大规模系统(Ultra-Large-Scale Systems)”。
- 软件巨复杂系统是计算富裕的一种体现。富裕的计算与传感器网络、自适应材料等密切结合，给软件工程在如何说明这些的配置和行为，产生应用，验证和确认系统的能力、性能和可信性，并把他们集成到一个“超大规模系统”上的系统。
- 超大规模系统软件工程比Barry Boehm的预测提前到来。这种挑战又进一步加剧了对“大规模的、国际化的”软件工程化生产的人才、组织、管理、技术等的要求。

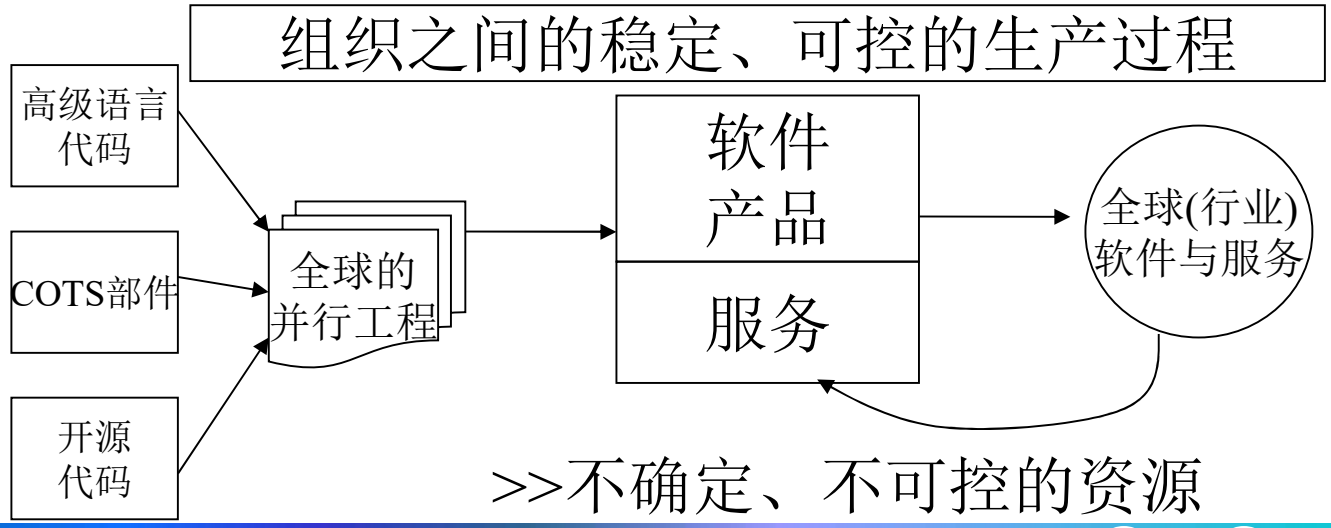
From computing to program, to software



软件生产方式转变---从团队到群体合作



CMMI能力提高



>>不确定、不可控的资源