

# 第1章. 软件需求工程概述

---

韩锐

北京理工大学 计算机学院

Email: [379068433@qq.com](mailto:379068433@qq.com)  
[hanrui@bit.edu.cn](mailto:hanrui@bit.edu.cn)

# 导论

- 1. 软件的发展史
- 2. 软件工程生命周期和过程模型
- 3. 软件危机
- 4. 需求在软件开发中的重要性
- 5. 软件需求工程概述
- 6. 需求工程的特性
- 7. 需求工程师的知识要求

# 导论

- 1. 软件的发展史
- 2. 软件工程生命周期和过程模型
- 3. 软件危机
- 4. 需求在软件开发中的重要性
- 5. 软件需求工程概述
- 6. 需求工程的特性
- 7. 需求工程师的知识要求

# 1. 什么是软件？

- 软件包括

- 程序
- 相关文档
- 数据

所有都是为了使程序正确的运行

- 更具体的来说，软件系统由以下组成：

- 独立的程序
- 程序的配置文件
- 系统的详细架构文档
- 描述如何使用的用户文档

# 软件产品分类

## ■ 一般产品

- 独立系统
- 在开放市场上卖给所有人的产品，如：
  - 文字处理软件
  - 数据库
  - 画图软件

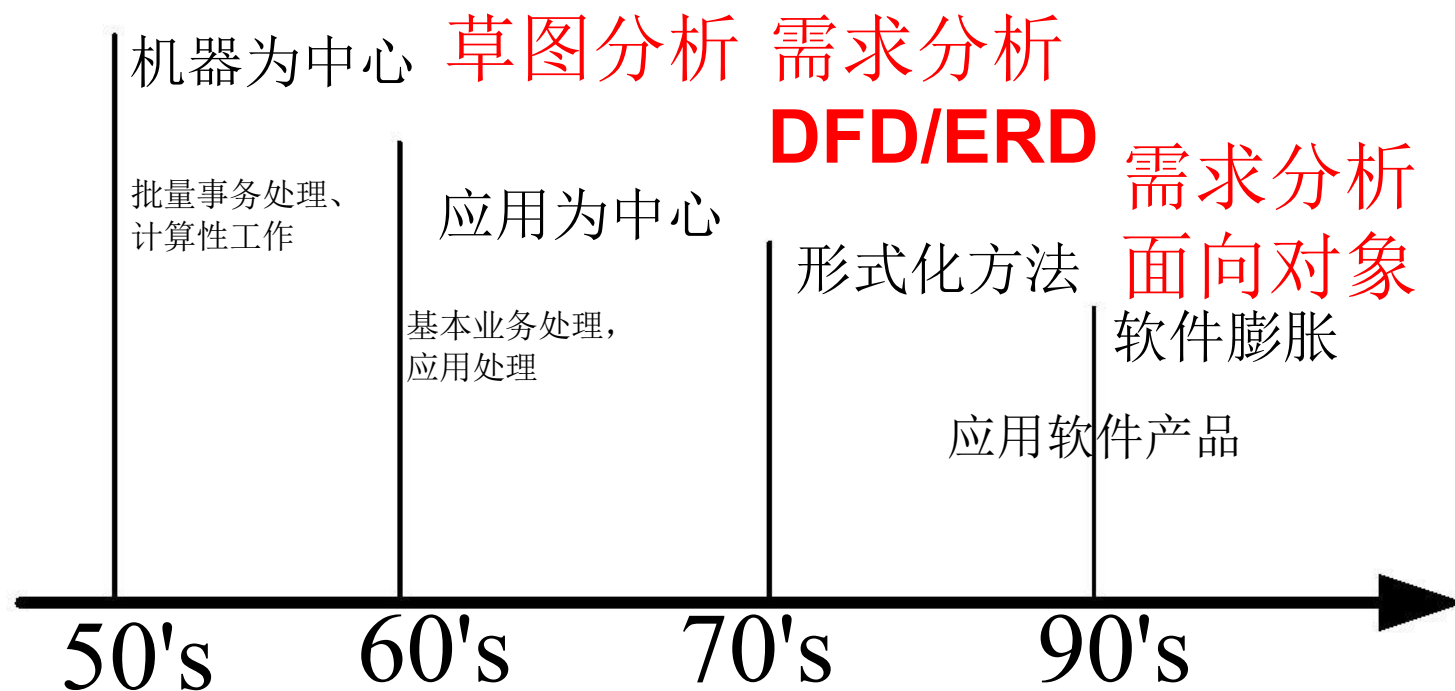
## ■ 定制产品

- 为了单一用户特制的软件产品

# 软件的发展

## ——60年代的发展

无需求处理

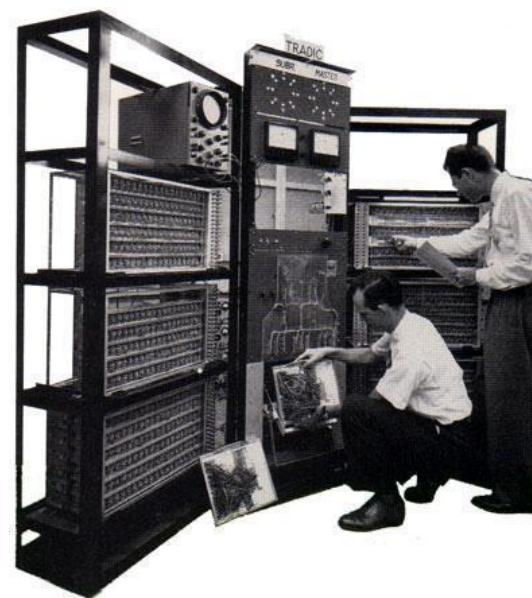
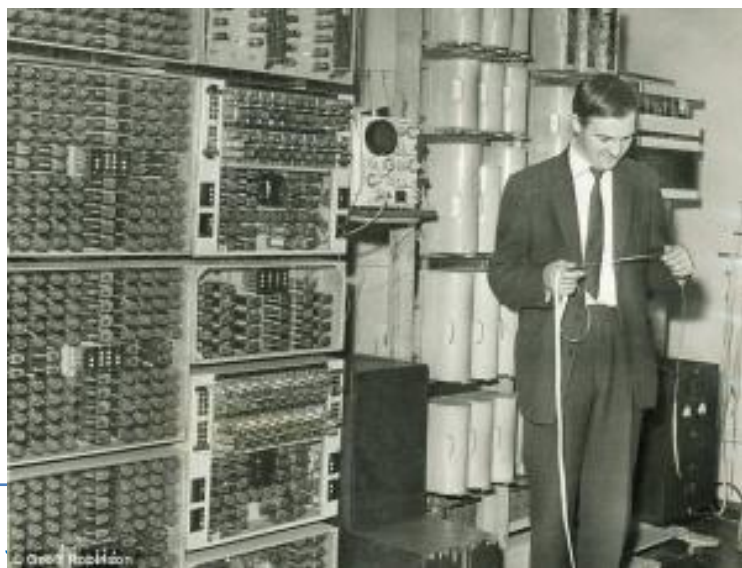


# 开拓时代(1955-1965)

- 几乎每一两年就有新电脑问世，这使得现有电脑很快就变的过时了。
- 为了使原有软件能在新的电脑上运行，软件人员就需要重写所有旧的软件。
- 程序员的办公桌上没有电脑，如果要编写代码就需要去机房。

# 开拓时代- 2

- 工作如何进行：
  - 通过注册机器时间或让操作人员来操作
  - 将穿孔卡片输入机器的读卡器，等待结果返回打印机
- 这领域太新了，以至于管理不能按计划进行
- 预测一个项目的完工日期几乎是不可能的





# 开拓时代- 3

- 计算机硬件有特定的应用程序。而科学和商业任务又需要不同的机器。
- 由于需要满足经常**重写**旧软件以满足新机器的需求，因此开发了FORTRAN，COBOL和ALGOL等高级语言来简化工作。

# 开拓时代- 4

- 硬件供应商免费赠送系统软件，这是因为没有软件的硬件无法卖出。
- 虽然有一些公司推出了定制软件的服务，但没有软件公司只销售成套的软件。

# 开拓时代- 5

- 复用的概念蓬勃发展。但由于软件是免费的，用户们通常会**放弃复用**。像IBM这样的公司会给特别用户SHARE提供可复用的组件。
- 学术界还没有计算机科学原理。
- **模块化编程**和**数据抽象**已经开始在编程中使用。

# 开拓时代-经典案例

## ■ SEGA-半自动地面环境防空系统

### ■ 背景：

美国在冷战时期为抵御苏联等其他国家的空中袭击所建立的一套自动化追踪、拦截敌军飞行器（尤其是轰炸机）的指挥系统。



# 开拓时代-经典案例

## ■ SEGA的参与者

### ■ 项目主要负责人

- 来自麻省理工学院-林肯实验室的  
John McPherson

### ■ 主要参与公司:

- IBM

- 同时，该项目雇佣了当时美国1200名程序员中的700人，历时13年，耗资约80亿美元



图 54 弗雷斯特

# 开拓时代-经典案例

## ■ SEGA-系统构成

- SEGA的核心是由IBM开发构建并维护的--AN/FSQ-7计算机，它可以自动计算并有效避开攻击的导弹和飞行器，同时让军事指挥官了解空战的全局形势。为了方便使用，SAGE系统最早还采用了人机交互的显示器。



AN/FSQ-7拥有100个系统控制台，包括 OA-1008状态显示器，光枪（塑料盖下方电缆终端），打火机，烟灰缸（光枪左侧

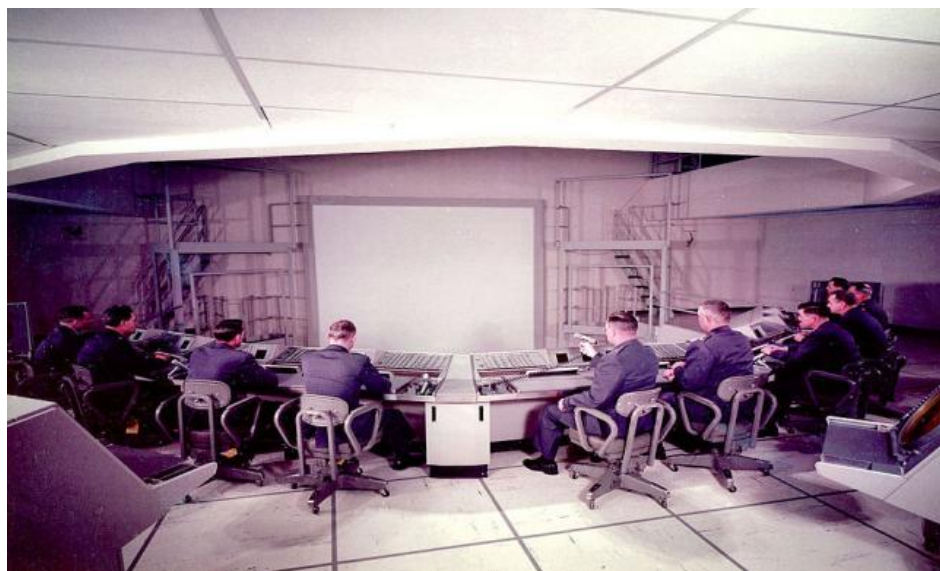


光枪使用

# 开拓时代-经典案例

## ■ SEGA-系统构成

该系统将美国划分为24个雷达区，全国设立三个战斗中心，每个区设立一个指挥中心，每个区分别装有电子雷达和电子计算机24小时连续运行，侦测并计算外来飞机的位置、飞行轨迹和速度，利用这些结果，指挥系统迅速指挥飞机拦截入侵飞机或用高射炮歼灭入侵飞机。



位于锡拉库扎空军基地的SAGE作战中心配有控制台和大型显示器，可进行界别分组指挥。设备安装期间拍摄的档案图片



# 稳定时代(1965-1980)

## ■ 整个工作模式已经**制度化**

- 除了机载计算机等特殊应用外，程序员不再需要额外的工作。
- 为了处理这些工作，**中央计算机中心**周围出现了一个巨大的机构。



# 稳定时代- 2

- IBM 360标志着稳定时代的开始。
- 迄今为止最大的软件项目结束了每年或每两年出现更快，更便宜的计算机的时代。
- 程序员最终可以花时间编写新软件而不是重写旧软件。
- IBM 360还将科学应用程序和商业应用程序整合到一台机器上。它提供了二进制和十进制算术。

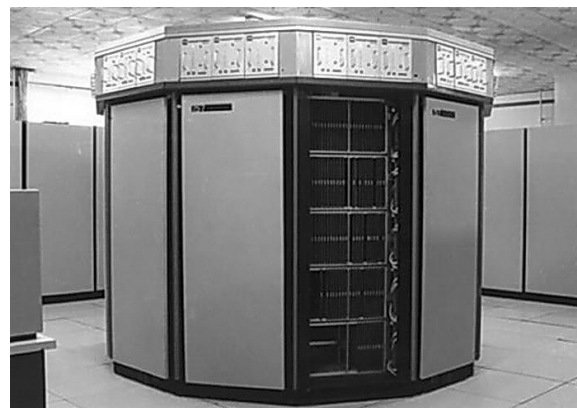
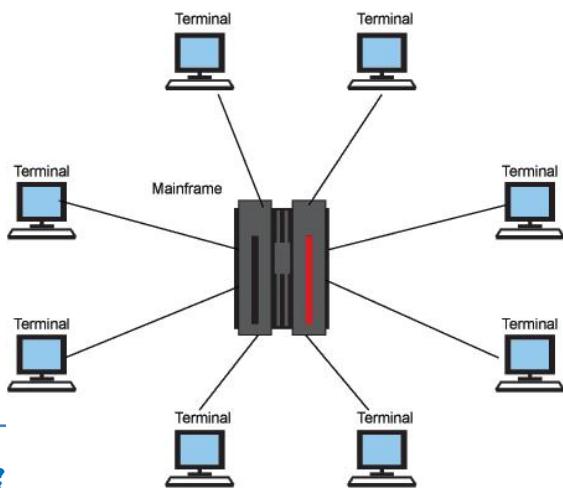


# 稳定时代- 3

- 大量的O / S仍然可以通过购买计算机**免费提供**，它控制着正在运行的程序所需的大部分服务进程。
- 作业控制语言JCL提出了一类全新的问题。程序员必须用一种全新的语言编写程序，告诉计算机和操作系统要做什么。JCL是IBM 360最不受欢迎的功能。
- PL / I失败了，它是由IBM提出，将所有编程语言合并为一个的替代语言。
- 程序员的需求量超过了**供应量**。

# 稳定时代- 4

- 分时使用终端的概念开始出现，在终端上作业可以直接提交到各种类型的队列中，这遭到了传统主义者的一些抵制。
- 随着软件越来越稳定，软件开始成为企业资产，其价值也开始凸显。
- 软件的稳定性导致了60年代后期学术计算学科的出现，但**软件工程学科还不存在**。



# 稳定时代- 5

- 人工智能等许多“热门”学科应运而生。但由于这些新概念无法转化为预测效益，计算领域的可信度开始下降。
- “结构化编程”在这个时代的中期突然出现。
- 制定标准的组织开始控制住竞争的局面。制定标准的供应商可以通过使标准与自己的技术相匹配来获得显著的竞争优势。

# 稳定时代- 6

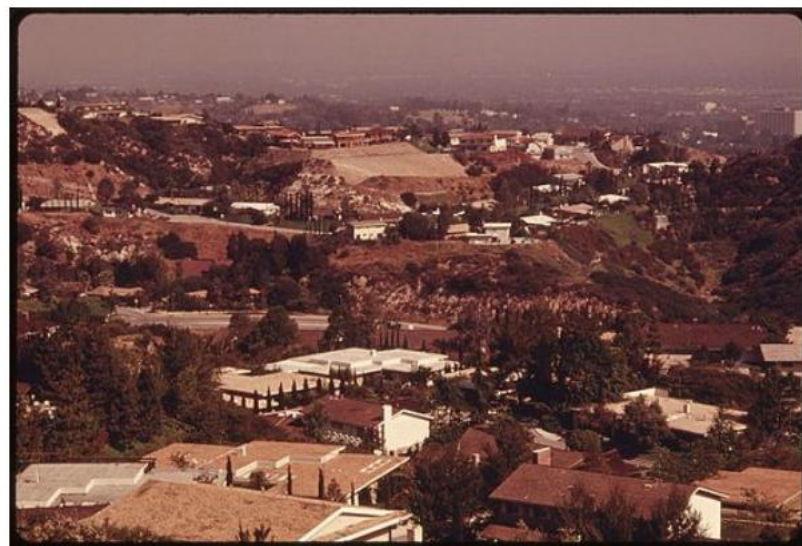
- 尽管硬件供应商试图通过保持低价格来遏制软件行业，但软件供应商却接连不断的出现。
- 大多数定制应用的程序仍需要在内部完成。程序员还是需要去机房，而且他们的办公桌上依然没有电脑。

# Infomatics和它的 MARK IV

- Informatics General Corporation(早期的 Informatics, Inc.公司)
  - 一家美国计算机软件公司，成立于1962年，总部位于加州洛杉矶。



The earlier company logo, as seen in the 1970s



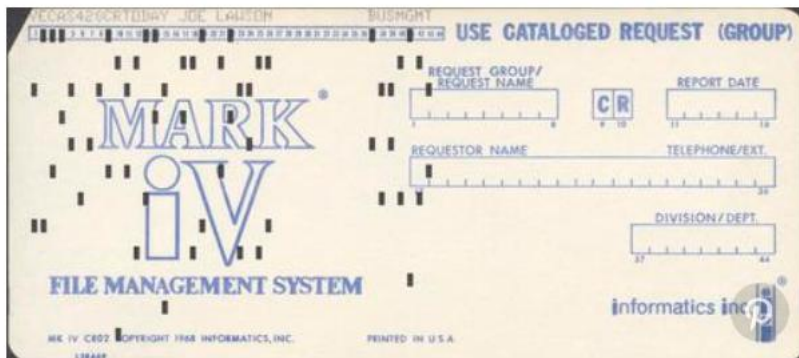
Informatics began in the Woodland Hills area of Los Angeles, California.



# Infomatics和它的 MARK IV

## ■ 主要产品--MARK IV

- Infomatics生产了各种各样的软件产品，其中最为显著的是其为IBM 360等大型机生产的**Mark IV文件管理和报告生成产品**，该产品成为当时最畅销的企业打包软件产品。



The Mark IV product became a big success back when keypunch cards were a common input mechanism in computing.



The IBM System/360 mainframe was the platform that Mark IV and many other Informatics software products ran on.

# Infomatics和它的 MARK IV

## ■ MARK IV 创始人

- John Appel Postley（1924年11月29日-2004年8月1日）

- 他是一位美国企业家，被认为是计算机软件行业的创始人之一，是第一批计算机软件产品Mark IV的创始人。

- Mark IV开创了软件产品的概念和商业模型的先河，成为该模型最早的成功范例。



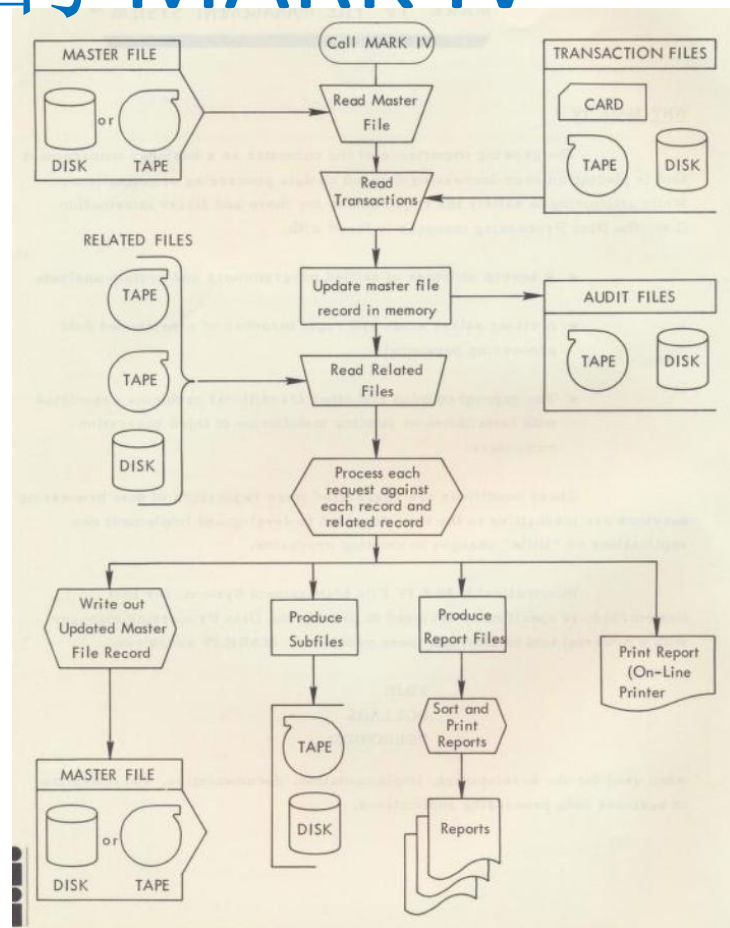
# Infomatics和它的 MARK IV

## ■ MARK IV 文件管理系统

Mark IV是一种批处理的早期第四代编程语言，它将文件管理、维护与报告生成功能结合在一起。

起初的Mark IV是为**非程序员**设计的，为报告请求和数据更新提供了简单的接口。

这些接口包括手工填写几个纸质表单中的一个，然后将它输入一个机器可读的表单，然后通过批处理操作来运行。



MARK IV的系统功能流

# Infomatics和它的 MARK IV

## ■ MARK IV 生成电话簿

MARK IV FILE MANAGEMENT SYSTEM  
BASIC REQUEST

Request Name: TELDIR Use File Named: PERSONEL Dated: TODAY

Requestors Name: I.V. MARK Telephone/Ext.: 371 Division/Dept.: 15

TITLE: 001 TELEPHONE DIRECTORY

RECORD SELECTION

*NONE REQUIRED*

REPORT SPECIFICATION

Field To Be Printed	Sort Sequence
NAME	1
PHONE EXT	
ADDRESS	
HOME PHON	

RESULTS PRODUCED BY MARK IV FROM EXAMPLE BASIC REQUEST

TELEPHONE DIRECTORY

NAME	PHONE EXTENSION	HOME ADDRESS	HOME PHONE
ABRAMOWSKY, T.	345	134 RODDA AVE	781-2345
ALTUCHER, S.	321	1781 WEST BLVD	781-7788
BARSAMIAN, L.	678	1214 HILL ST	345-7829
BAUER, W.F.	478	891 WESTLAKE AVE	347-7865
BECKER, R.M.	471	812 N. ORANGE AVE	378-2121
BENJAMIN, R.	128	121 NATIONAL BLVD	347-8876
BUETTELL, D.E.	341	144 E. GROVE AVE	873-4211
BRADDOCK, F.	321	782 S. BRAND AVE	781-9782
COLLINS, T.W.	217	817 W. ROSCOE ST	871-3131
COOPER, W.R.	117	1218 STANDARD AVE	781-9650
CORYELL, N.T.	228	1814 NATIONAL BLVD	781-9600
COUCH, S.	229	161 E. MAIN ST	347-2011
CROWLEY, D.	382	128 COMMERCE ST	781-2515
CUTLER, W.C.	340	130 AVIATION BLVD	347-1111
DOWKONT, A.J.	821	161 TRADE ST	
HASKELL, E.D.	671	170 ALLEN BLVD	
HECKATHORNE, R.B.	421	1117 MAPLE AVE	
JACOBSON, H.	487	444 IVY ST	
KREINER, P.J.	507	411 LEAGUE	
LIVELY, J.T.	875	1007 HILL	
LOCKWOOD, L.	218	1025 TR	
MARK, I.V.	371	8241 M	
PINETTE, J.	788	784	
POSTLEY, J.A.	220	41	
REDEKOPP, J.	871		
STOKES, M.	733		
STONE, R.D.	225		
SUNDERLAND, R.M.	217		
TAYLOR, T.R.	371		
WAGNER, F.V.			
WILLIAMS, L.M.			
ZANICCHI, J.A.			

# 微时代(1980-现在)

- 计算的成本急剧下降，使得计算无处不在成为可能。现在每个程序员都可以在办公桌上放一台电脑。
- 旧的JCL已经被有良好用户体验的GUI所取代。

IBM首部个人电脑



# 微时代- 2

- 软件领域仍然存在很多问题。自从IBM大型机和第一个Intel芯片问世以来，程序员必须了解的硬件体系结构中的软件部分（如指令集）没有发生太大变化。目前使用最多的编程语言是面世以来15到40年之间的语言。
- 第四代语言从来没有实现“没有程序员的编程”的梦想，其思想基本都局限于从数据库中生成报告。不过，目前对于寻求更多更好的软件开发和研究的呼声越来越高。

# Windows 1.0

## ■ Microsoft的成立



- 微时代最具代表性的软件公司当属微软---  
Microsoft

- 创始人:

- 比尔·盖茨 (Bill Gates)
- 保罗·艾伦 (Paul Allen)



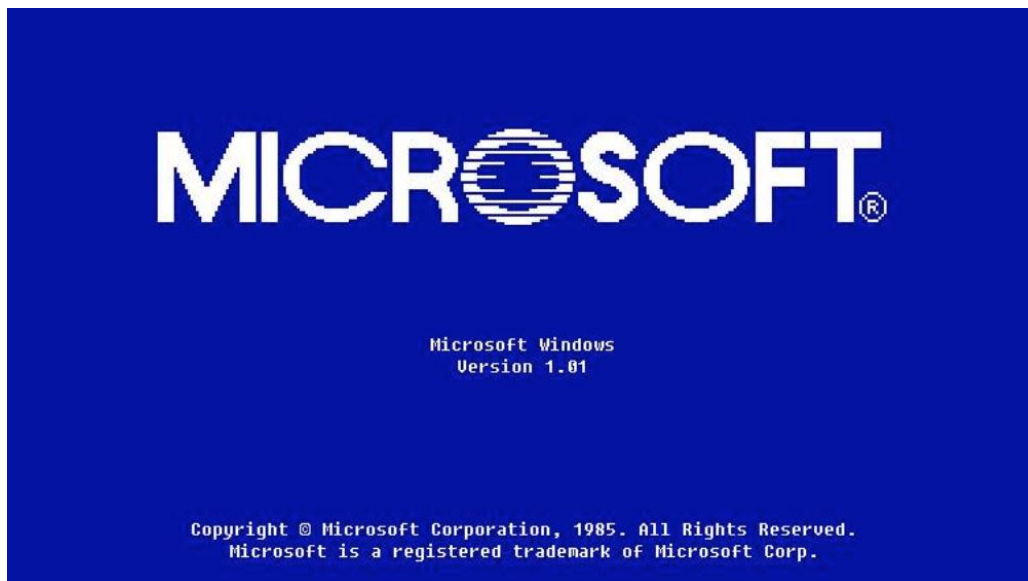
保罗·艾伦 (Paul Allen) 和比尔·盖茨 (Bill Gates) 于1981年10月19日与IBM签订了重要合同<sup>[15]</sup> 228

- 盖茨和艾伦于1975年4月4日成立微软公司，盖茨担任首席执行官。

# Windows 1.0

## ■ 第一款软件产品-Windows 1.0

- Windows 1.0是微软于1985年11月20日发布的操作系统。在此系统中微软第一次对个人电脑操作平台进行用户图形界面的尝试。

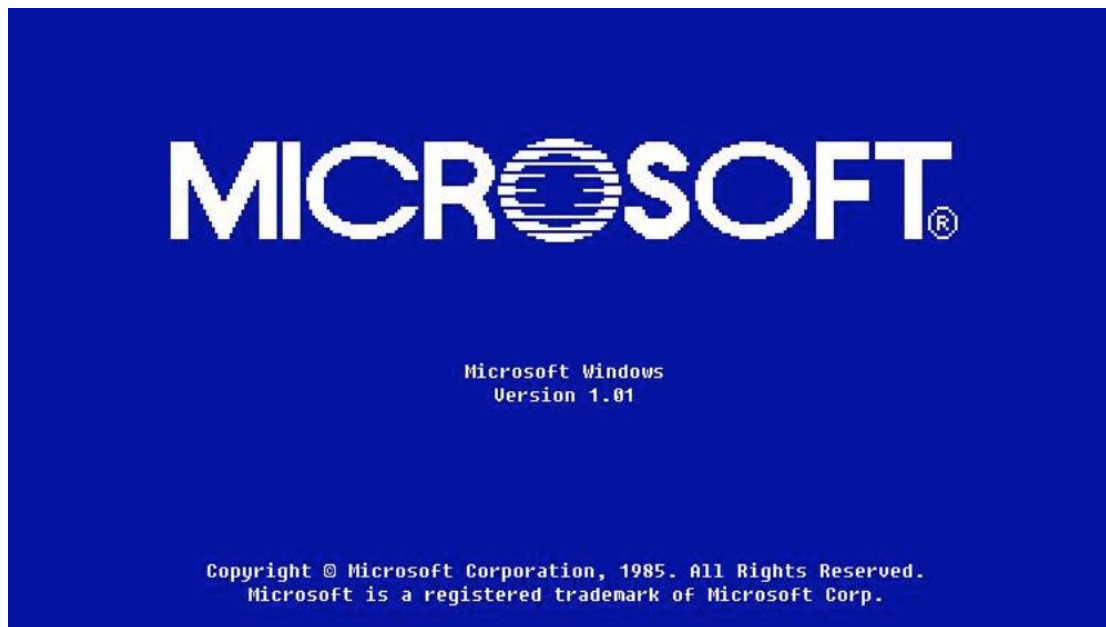




# Windows 1.0

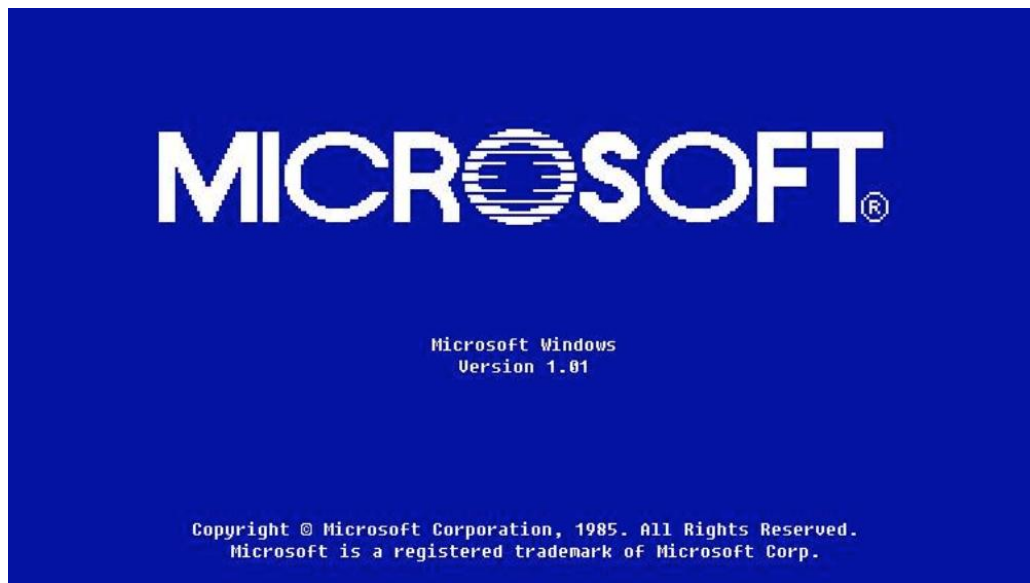
## ■ 第一款软件产品-Windows 1.0

- Windows 1.0基于MS-DOS操作系统，本质上宣告了DOS操作系统的终结。。



# Windows 1.0

- 第一款软件产品-Windows 1.0
  - Windows 1.0基于MS-DOS操作系统，实际上其本身并非操作系统，至多只是基于在当时的纯DOS下运行的应用软件。

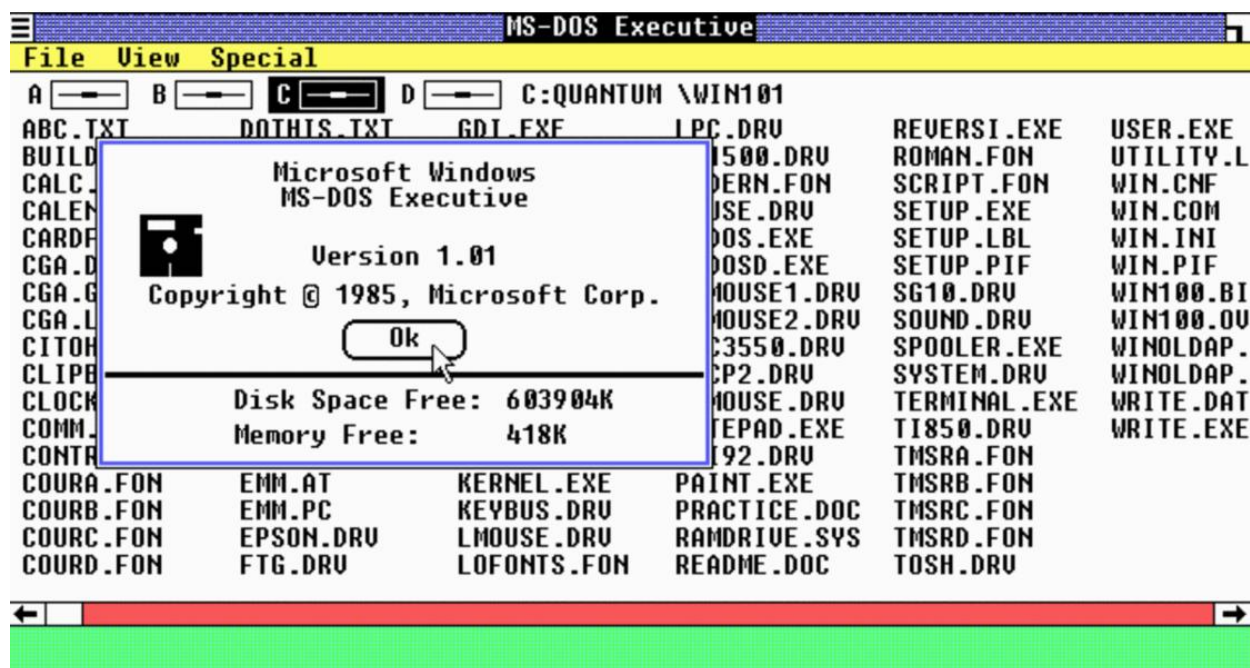




# Windows 1.0

## ■ 第一款软件产品-Windows 1.0

- Windows 1.0中鼠标作用得到特别的重视，用户可以通过点击鼠标完成大部分的操作。



# Windows 1.0

## ■ Windows 1.0 的系统组件

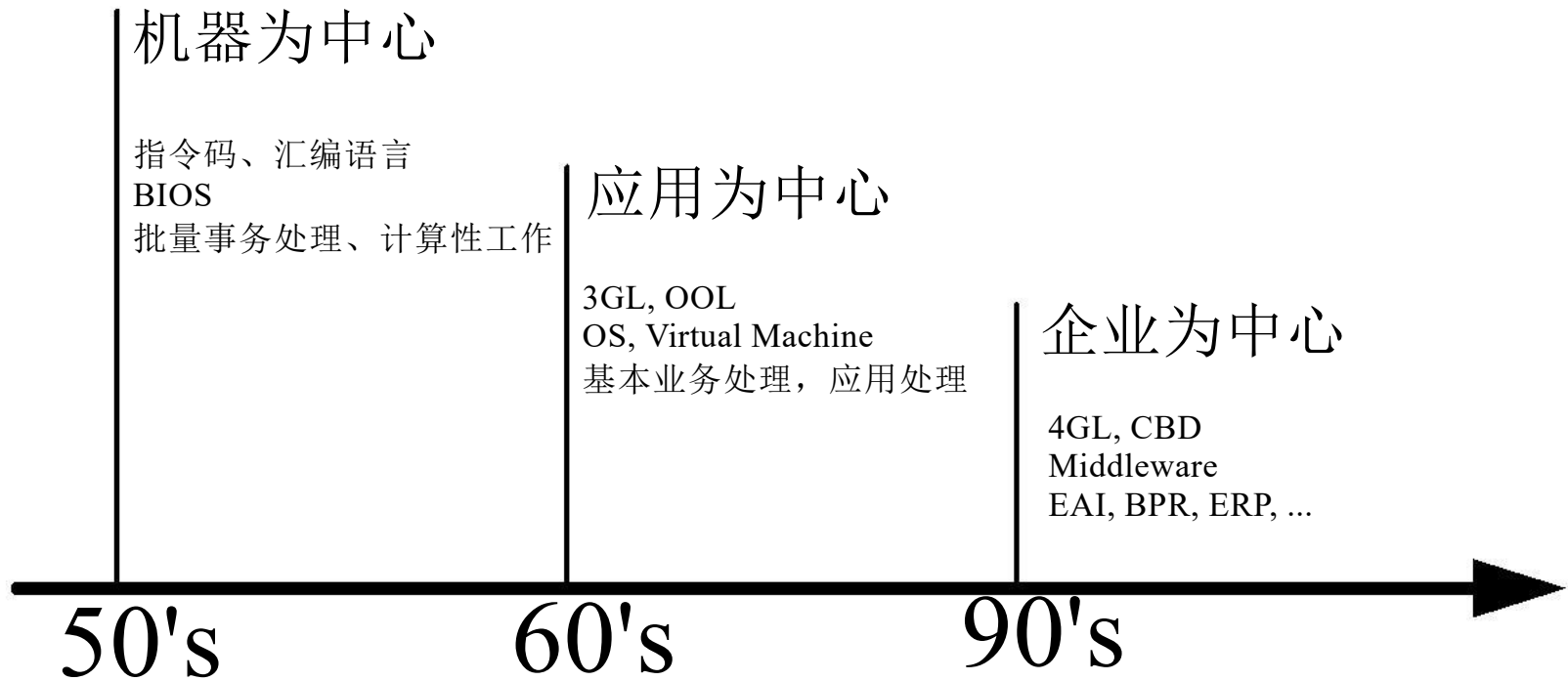
- MS-DOS Executive（程序管理器）
- Calendar（日历）
- Cardfile（名片）
- Notepad（记事本）
- Terminal（终端）
- Calculator（计算器）
- Clock（时钟）
- Reversi（黑白棋游戏）

# Windows 1.0

- Windows 1.0 的系统组件
  - Control Panel（控制面板）
  - Clipboard（剪贴板）
  - RAMDrive（RAM驱动）
  - Windows Write（写字板）
  - Windows Paint（画图）
  - Print Spooler（打印后台处理程序）
  - PIF(Program Information File) Editor（PIF编辑器）

# 软件发展史

## ——60-90年代的发展



# 今天：AI&大数据应用

## 重要里程碑事件



Google Translate服务上线。目前，该服务已支持103种语言

2006.4



Watson在《危险边缘》中击败人类选手

2011.2



Rothschild利用大数据预测奥斯卡奖项，正确率达到88%

2014.3



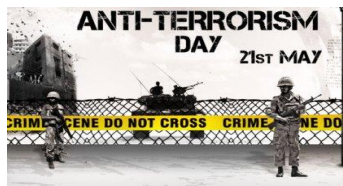
Nate Silver利用多种形式的民意调查数据，成功预测特朗普当选美国总统

2016.11



Google利用搜索引擎查询数据侦测流感

2009.2



Palanir通过分析财务、DNA样本等数据，助美军方发现并击毙本拉登

2011.5



Zions银行通过利用大数据处理框架，实现快速实时诈骗检测

2013.12



AlphaGo 4比1战胜李世石

2016.3

# 大数据的主要特征

欧洲电子对撞机1  
年数据量-CD存储  
约20公里

- 不仅来源于我们的主动获取，更来源于信息系统的自然产生

**Volume**

海量性

数据规模指数级增长，2020年前全球数据量大约**2年翻1倍**(2000年800TB，2012年2.7ZB，2020年预计44ZB)

人类所获得数据中，**90%为过去2年内所产生**

**Variety**

多样性

从文本数据为主发展到以视频、图片等非结构化数据为主

**Velocity**

时效性

用1台最快的商用计算机仅把1PB数据读入内存就要约**6天时间**

用**4000台**计算机对1PB数据排序就需约**6小时**

**Veracity**

真实性

充斥着大量伪数据、无效数据、过时数据

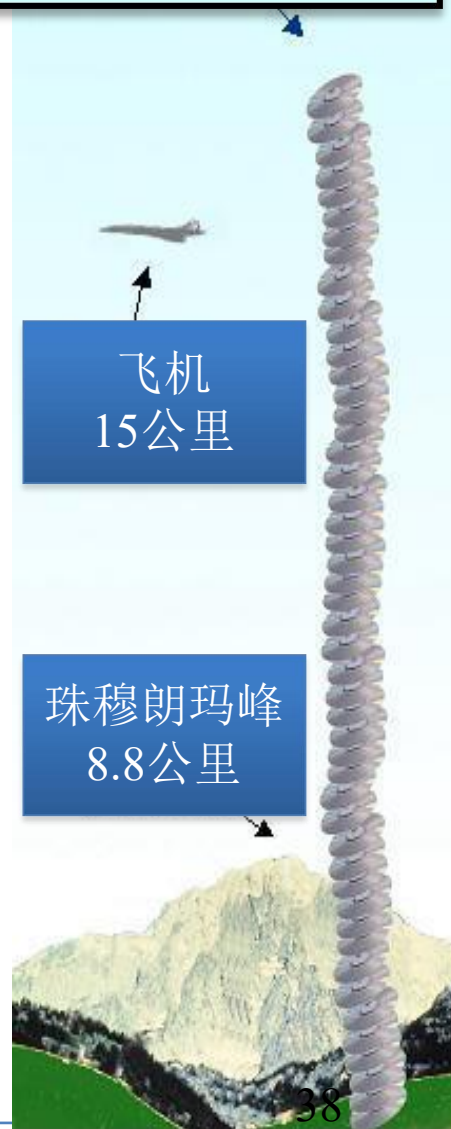
**Value**

潜在价值

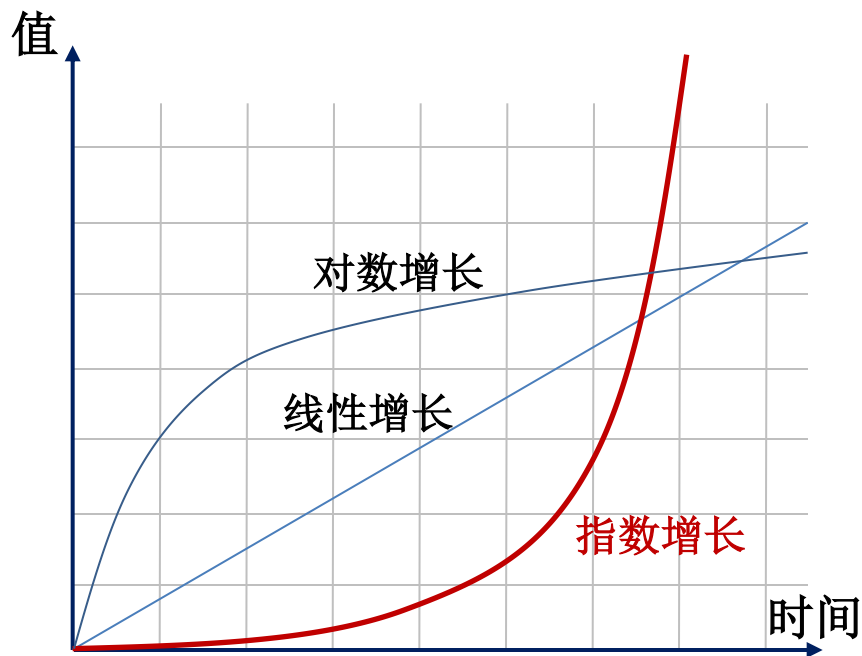
大数据价值密度低

飞机  
15公里

珠穆朗玛峰  
8.8公里

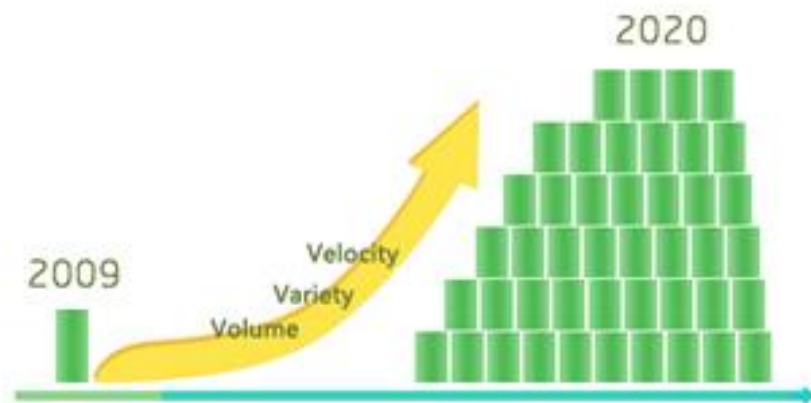


# 数据的指数增长模式



## 全球数据总量统计（IDC）

- 2003年全球产生数据仅5百万TB
- 2009年全球产生数据约8亿TB
- 2012年全球产生数据约27亿TB
- 2020年，预计440亿TB





# 大数据软件栈

## 2. NoSQL systems for Cloud OLTP

### 1. RDBMSs for OLTP (online service)



### 3. Data warehouse for OLAP



典型大数据软件栈



DataMPI



### 5. Systems for offline analytics



S4 distributed stream computing platform

Storm cloudera IMPALA

### 4. Systems for realtime (interactive) analytics





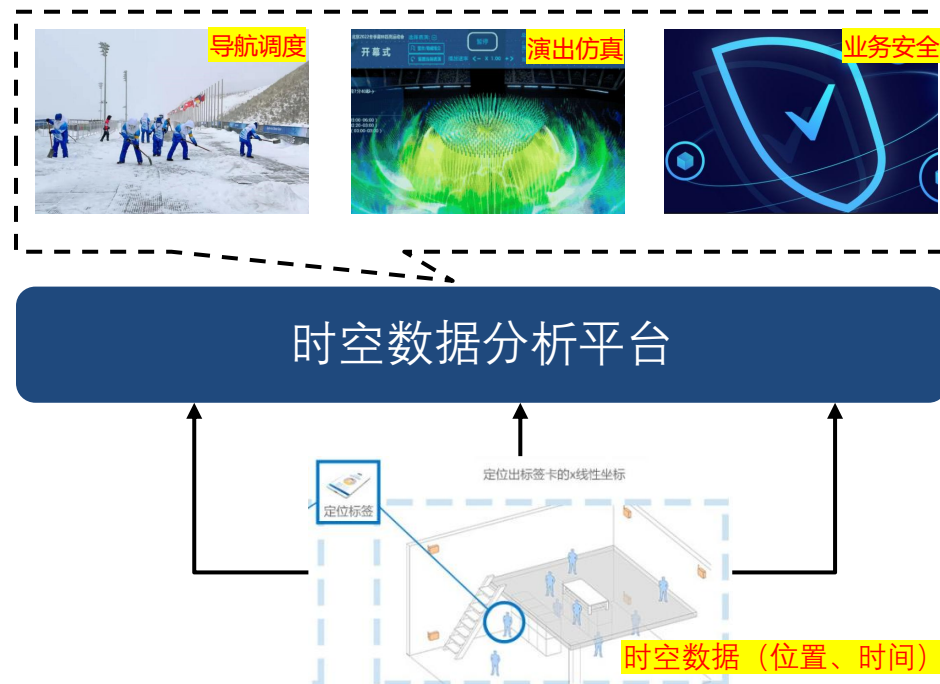
# 今天：大数据&AI软件



# 案例1-时空数据分析平台



首钢冬奥园区



# 案例2-人工智能科研与教学平台

