

北京理工大学

# 软件工程基础

北京理工大学 计算机学院

胡思康

skhu@163.com

# 软件工程基础

## 大数据中心建设



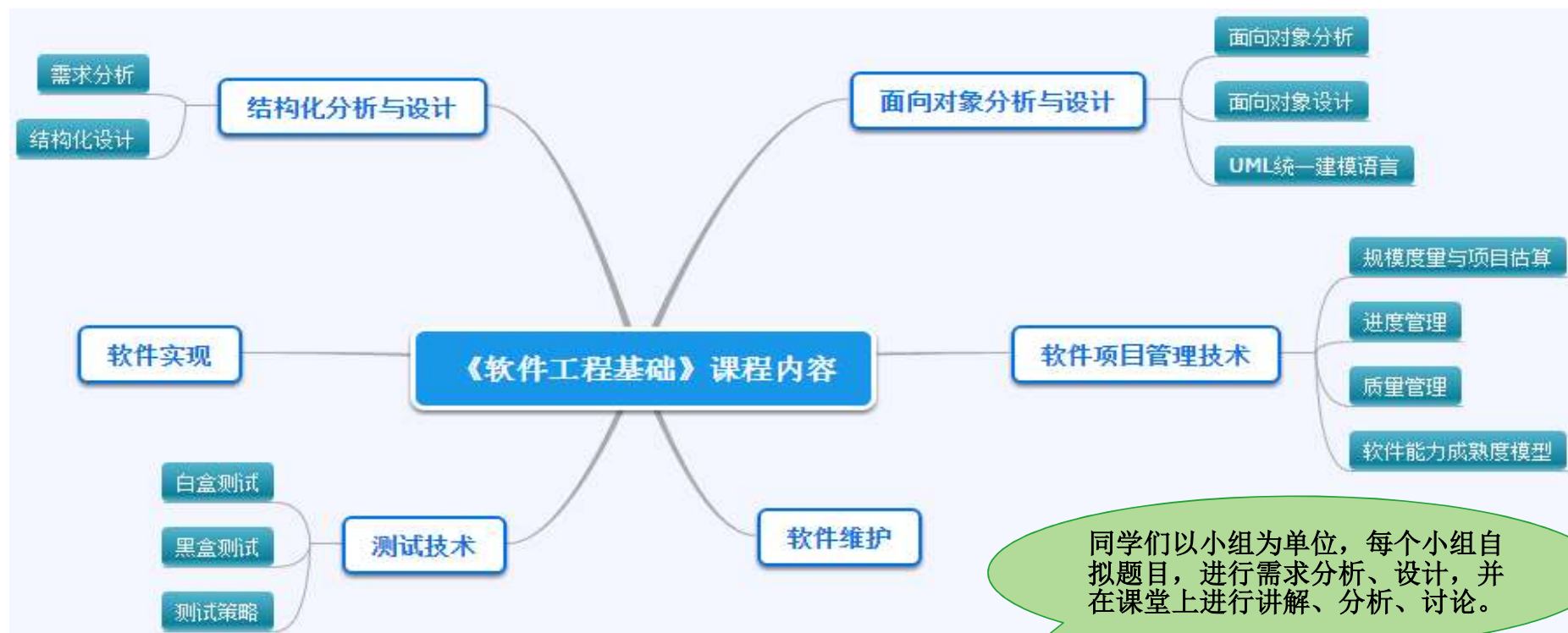
腾讯绿色数据中心位于贵阳市贵安新区，隧洞面积超过3万平方米，是一个特高等级绿色高效灾备数据中心，未来将用于存储腾讯最核心的大数据。

华为云贵阳市贵安新区数据中心——“欧洲小镇”。它将成为华为全球最大的云数据中心，预计将容纳100万台服务器。

此外，苹果（除欧美之外）、阿里、京东、中国人民银行等数据中心也建在贵阳市贵安新区。

# 软件工程基础

课程的主要内容（软件工程知识体系）：



同学们以小组为单位，每个小组自拟题目，进行需求分析、设计，并在课堂上进行讲解、分析、讨论。

课 堂 讨 论

# 第1章 软件工程概述

---

- ❖ 引言
- ❖ 软件工程的发展历程
- ❖ 软件工程的概念
- ❖ 软件与软件工程过程
- ❖ 软件过程模型
- ❖ 软件开发方法

# 引言

## 为什么要讲软件和软件工程（1）？

- ▶ 软件不仅仅是在计算机运行的程序；
- ▶ 任何预先定义好的程序步骤的地方，都有软件的身影。
- ▶ 软件无处不在，为我们的生活创造了无限精彩。在当今的信息时代，世界正在变得更加“智慧”，万事万物间感知化、互联化和智能化的程度不断加深。
- ▶ 软件工程为这一切做出了巨大贡献，随着本学科的发展，其未来的贡献将不可限量。



# 引言

## 为什么要讲软件和软件工程（2）？

### ➤ 2018-2025年中国数字经济核心产业增加值规模

数据来源：亿欧智库



### 软件、软件系统、软件生态已经成为信息时代的焦点

- ▶ 计算机和软件导致了“知识的民主化”：维基百科、百度知道、知乎、微博……
- ▶ “网络共享”是全球知识交换的关键：网络爬虫、微信小程序、文字图片视频……
- ▶ 由计算机控制的信息和知识，将成为权力的焦点：网络暴力、键盘侠、舆情……



# 引言

## 为什么要讲软件和软件工程（3）？

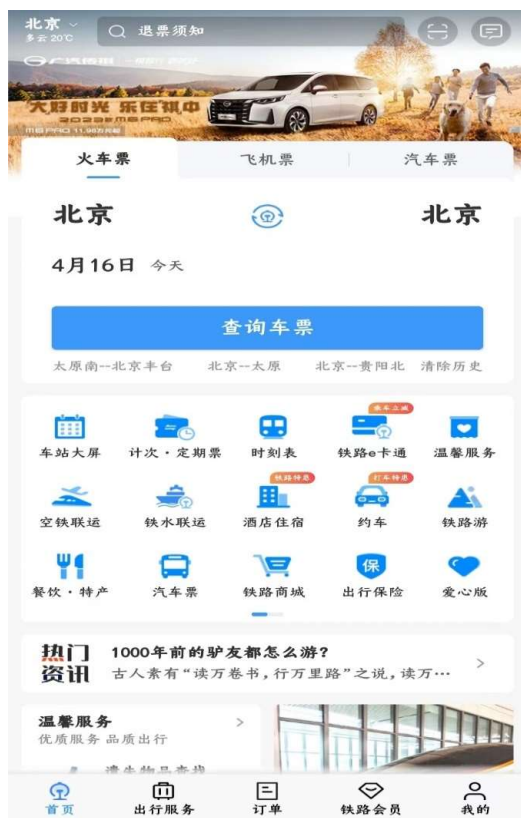
### 软件系统开发

- 将现实问题域中的概念、处理转换为计算机（网络）的概念、以及对处理过程的逻辑映射。从而软件开发的本质可概括为：在不同抽象层对术语和处理逻辑的映射。
- 将围绕软件系统的开发过程，分析如何实现上述映射、以及如何管理这些映射相关的活动，这样才能更好的开发出过程受控、质量受控的软件产品。



# 软件危机

## 软件系统仍面临的问题



12306网络购票系统:

- ▶ 历时两年研发成功，耗资3亿元人民币，于2011年6月12日投入运行。
- ▶ 2012年1月8日春运启动，9日网站点击量超过14亿次，出现网站崩溃、登录缓慢、无法支付、扣钱不出票等严重问题。
- ▶ SQL语句注入错。



# 软件危机

## 软件系统仍面临的问题

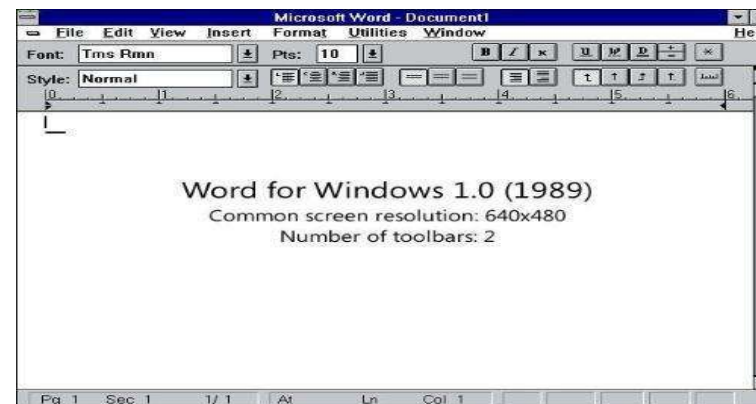


- ❖ Windows 95超过1000万行代码
- ❖ Windows XP 超过4000万行代码
- ❖ Windows Vista 超过5000万行代码，9000余位开发人员，耗资60亿美元。2001年开始研发，2006年底正式发布。之后的公测发现2万余错误。

# 软件危机

## 软件的本质特性是什么？

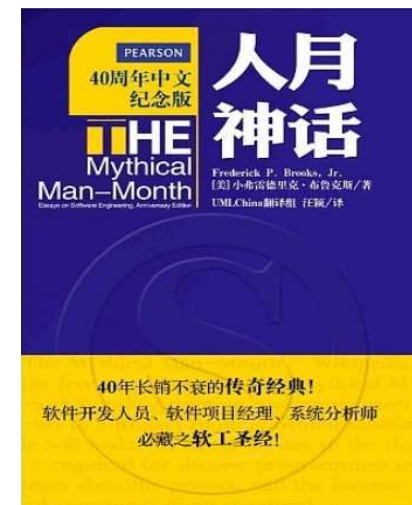
- 微软在**1989.11**发布了**Word 1.0** 版本，约有**25**万行代码，花费了**55**人 / 年，并比预期晚了**4**年交付。
- **1969**年建成投产到今天，位于美国西雅图的波音艾弗雷特工厂，是生产波音**747**、**767**、**777**的地方。生产组装飞机，是一件非常复杂的、巨大的工程，一架波音**747**需要近**600**万个零部件，每天有**15**列轨道车进出工厂运送零件。但每月能产出数架飞机。



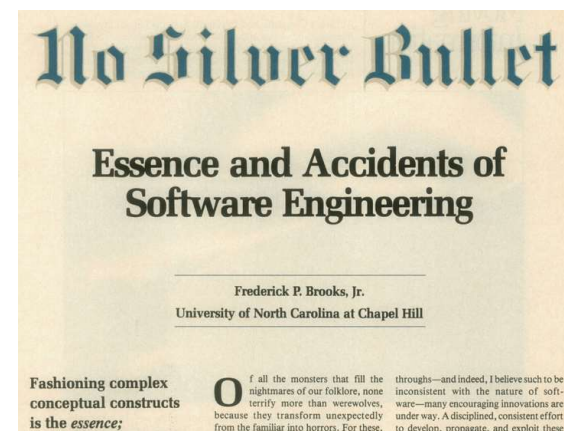
# 软件危机

## 软件的本质特性

Fred Brooks（1999年，图灵奖）是美国北卡罗来纳大学的计算机科学教授，在计算机体系结构、操作系统和软件工程方面作出了里程碑式的贡献。



Fred Brooks在1987年发表了一篇题为《Essence and Accidents of Software Engineering》的论文中提出：软件具有复杂性、一致性、可变性和不可见性等固有的内在特性，这是造成软件开发困难的主要原因。

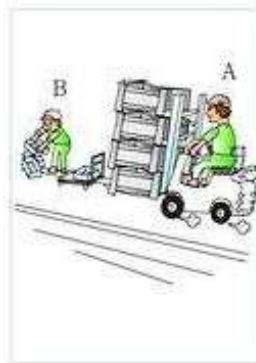
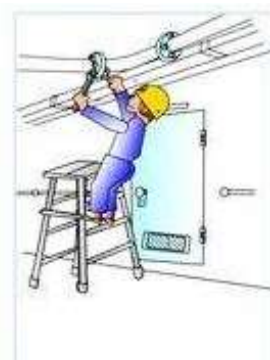


# 软件工程的发展历程

## 软件危机出现的原因

### (1) 从参与人员与软件质量标准上看

- ❖ 对软件开发缺乏正确的理论指导
- ❖ 软件人员与用户缺乏充分的交流
- ❖ 对软件开发过程缺乏整体认识
- ❖ 对软件产品缺乏有效一致的质量评价标准，不符合用户的操作习惯等一系列问题。



# 软件工程的发展历程

## 软件危机出现的原因

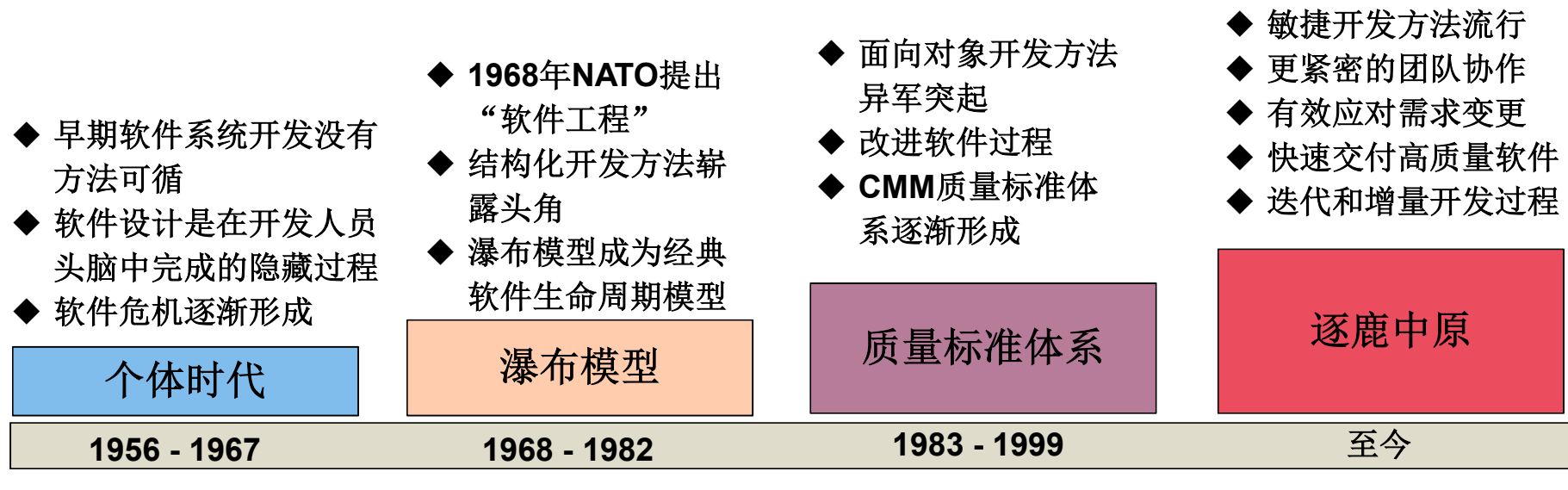
(2) 从开发过程与项目管理上看

- ❖ 软件开发进度和成本难以控制
- ❖ 软件产品难以满足用户的需求
- ❖ 软件质量难以得到保证
- ❖ 软件产品难以进行维护
- ❖ 软件的文档资料难以管理
- ❖ 软件产品的生产率难以得到提高



# 软件工程的产生和发展

## 软件工程的发展阶段





# 软件工程的产生和发展

## 中国软件产业的形成与发展

- 萌芽期（20世纪70年代）：计算机工业管理局成立，中软公司成立；
- 起步期（20世纪80年代）：中关村计算机发展及软件企业的出现；
- 进入期（20世纪90年代）：全国各地软件园的建立与发展；
- 发展期（21世纪00年代）：门户网站及大规模软件企业如新浪、腾讯、百度等；
- 增长期（如今）：阿里巴巴、抖音，以及引出的社交方式、低头族、支付习惯等。

# 软件工程的产生和发展

## 中国软件收入的增速及占GDP比重变化趋势



数据来源：中国通信院

# 软件工程的产生和发展

## 中印软件产业的对比

中国目前从事软件与服务的企业超过一万家，是印度的五倍以上。但中国软件销售仅占世界的1.2%，而印度已成为世界软件业的第二。

中印软件业务收入比较

单位：亿美元 制作：北海居

年份/财年	软件国内收入			软件出口收入			软件业务收入		
	印度	中国	印度=100	印度	中国	印度=100	印度	中国	印度=100
2000	20.57	63.70	309.7	62.04	3.99	6.4	82.61	67.69	81.9
2001	22.80	83.47	366.1	76.53	7.25	9.5	99.33	90.72	91.3
2002	27.68	117.92	426.0	95.24	14.98	15.7	122.92	132.91	108.1
2003	35.38	173.37	490.0	126.82	19.93	15.7	162.20	193.31	119.2
2004	48.36	262.54	542.8	178.38	28.03	15.7	226.74	290.57	128.2
2005	66.84	440.57	659.1	235.08	36.26	15.4	301.92	476.82	157.9
2006	81.70	539.40	660.2	311.36	62.85	20.2	393.06	602.25	153.2
2007	116.82	664.83	569.1	408.54	102.40	25.1	525.36	767.23	146.0
2008	128.50	948.41	738.1	470.86	142.00	30.2	599.36	1090.41	181.9
2009	142.99	1274.52	891.3	499.83	185.00	37.0	642.82	1459.52	227.1
2010	172.68	1739.97	1007.6	589.36	267.35	45.4	762.04	2007.32	263.4
2011	191.40	2572.15	1343.9	694.05	346.19	49.9	885.45	2918.34	329.6
2012	192.28	3533.49	1837.7	756.99	394.24	52.1	949.27	3927.72	413.8
2013	189.74	4469.74	2355.8	871.61	469.14	53.8	1061.35	4938.88	465.3
2014	211.67	5540.89	2617.7	981.24	486.71	49.6	1192.91	6027.60	505.3
2015	214.56	6384.59	2975.6	1069.34	494.87	46.3	1283.90	6879.46	535.8
2016	239.68	6761.95	2821.2	1161.43	499.46	43.0	1401.11	7261.41	518.3
2017	266.88	7613.44	2852.7	1250.00	538.00	43.0	1516.88	8151.44	537.4
2018	280.00	8975.04	3205.4	1360.00	554.50	40.8	1640.00	9529.54	581.1

注：印度为财年数据，根据财年平均汇率折算。

数据来源：印度统计年鉴2018、印度通信和信息技术部年报2017-18、2018-19、中国工业和信息化部

# 软件工程的产生和发展

## 软件工程的定义

- ▶ 1993年IEEE提出软件工程概念：将系统化的、规范化的、可度量的工程原则和方法，应用于软件的开发、运行和维护及其上述方法的研究。

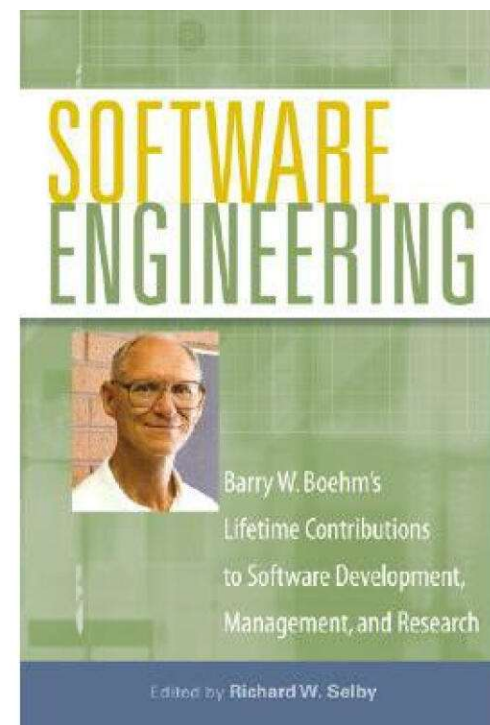
定义中的四个要点：

- ❖ 软件工程方法——系统化、规范化、可度量的工程化方法；
- ❖ 软件工程方法的作用和范围——软件的开发、运行和维护；
- ❖ 软件工程方法的目标——即将工程化的原则和方法应用于软件开发过程；
- ❖ 软件工程方法的趋势——方法的研究；

# 软件工程的观念

## 软件工程的基本原理 (Barry Boehm)

- ✓ 需求与设计能力
  - ✓ 执行过程的成熟度
  - ✓ 项目管理的保证
- 用分阶段的生命周期计划严格管理
  - 坚持进行阶段评审
  - 执行严格的产品质量控制
  - 采用现代化程序设计技术
  - 结果应能清楚地审查
  - 开发人员应少而精
  - 承认不断改进软件工程的必要性



# 软件与软件过程

❖ **软件**: 是计算机系统中与硬件相互依存的另一部分, 包括 程序、数据 及 相关文档 的完整集合。

三要素



执行要素

**程序**: 按事先设计的功能、性能要求执行的指令(语句)序列;

**数据**: 使程序能正常操纵信息的数据结构;

支撑要素

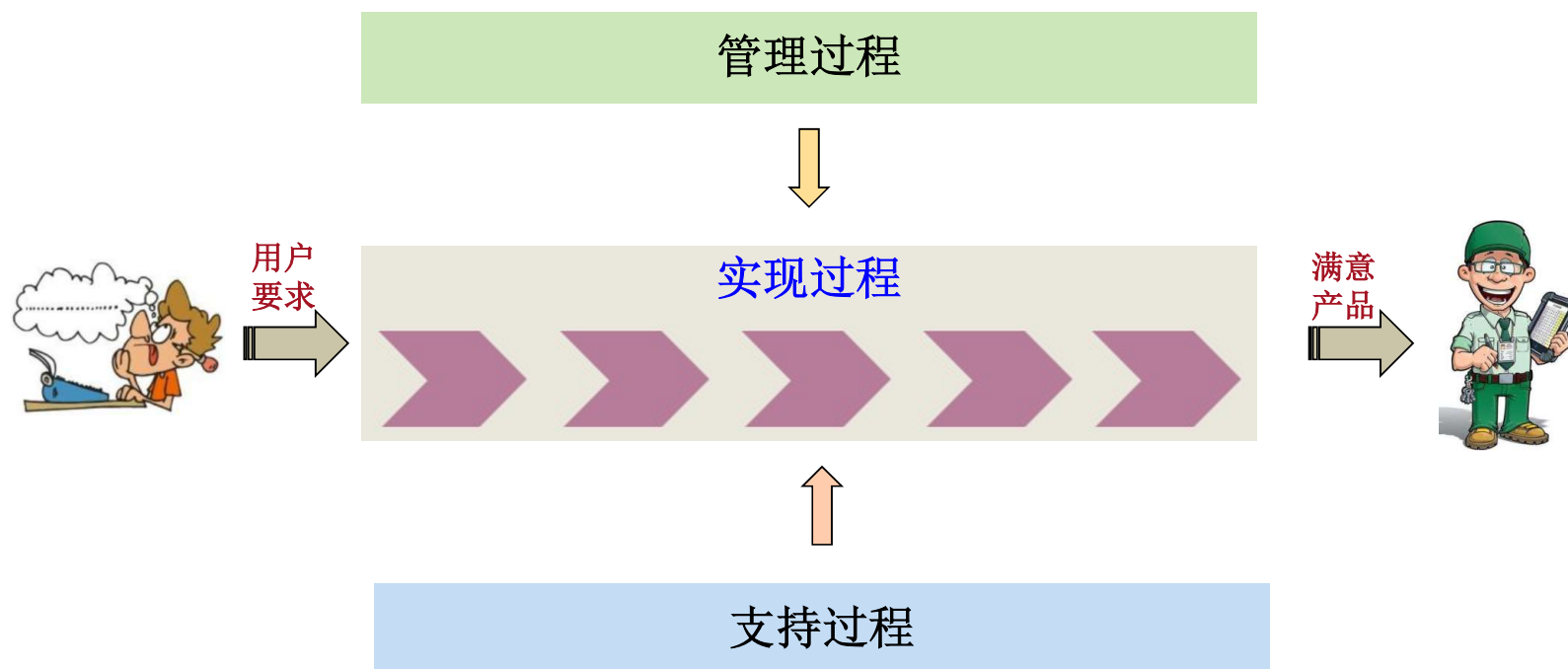


**文档**: 与程序开发、维护和使用有关的图文资料。



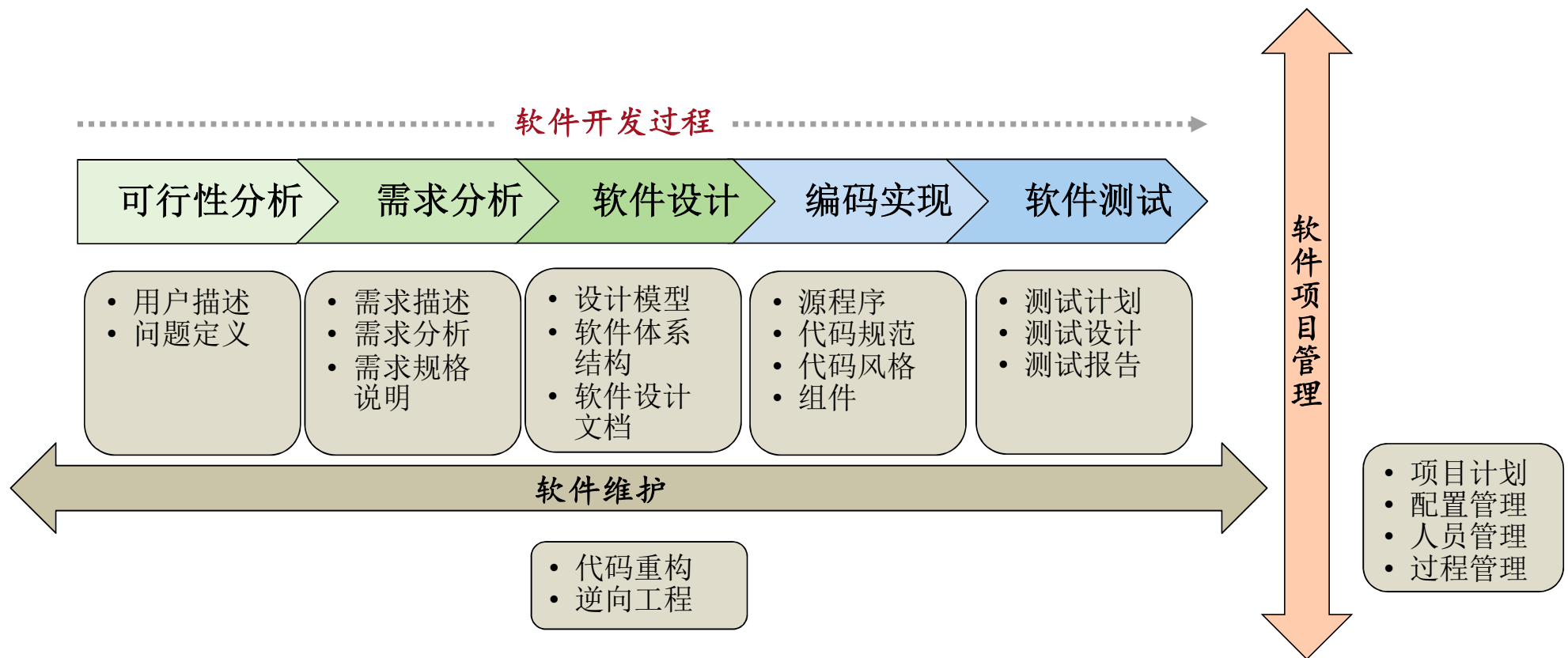
# 软件与软件过程

**过程：**系统地识别和管理组织内部所使用的、不同阶段的步骤、方法和工具，保证更有效的获得期望的结果。



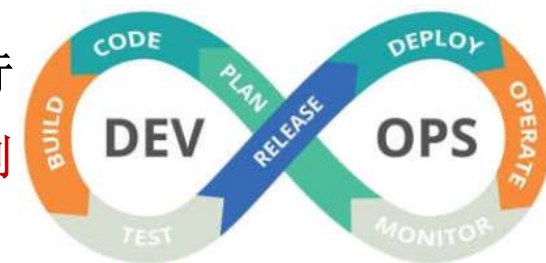
# 软件与软件过程

软件工程的过程模型划分了软件生命周期，把软件开发的全过程分阶段、定任务，按先后顺序依次完成。



# 软件过程模型

软件工程的过程模型针对软件开发活动的一般规律，对软件开发过程进行定量度量的**量化**，为软件工程管理提供**阶段性评价**，为软件开发过程提供**原则和方法**，也被称为软件生命周期模型。



目前典型的软件开发模型有：

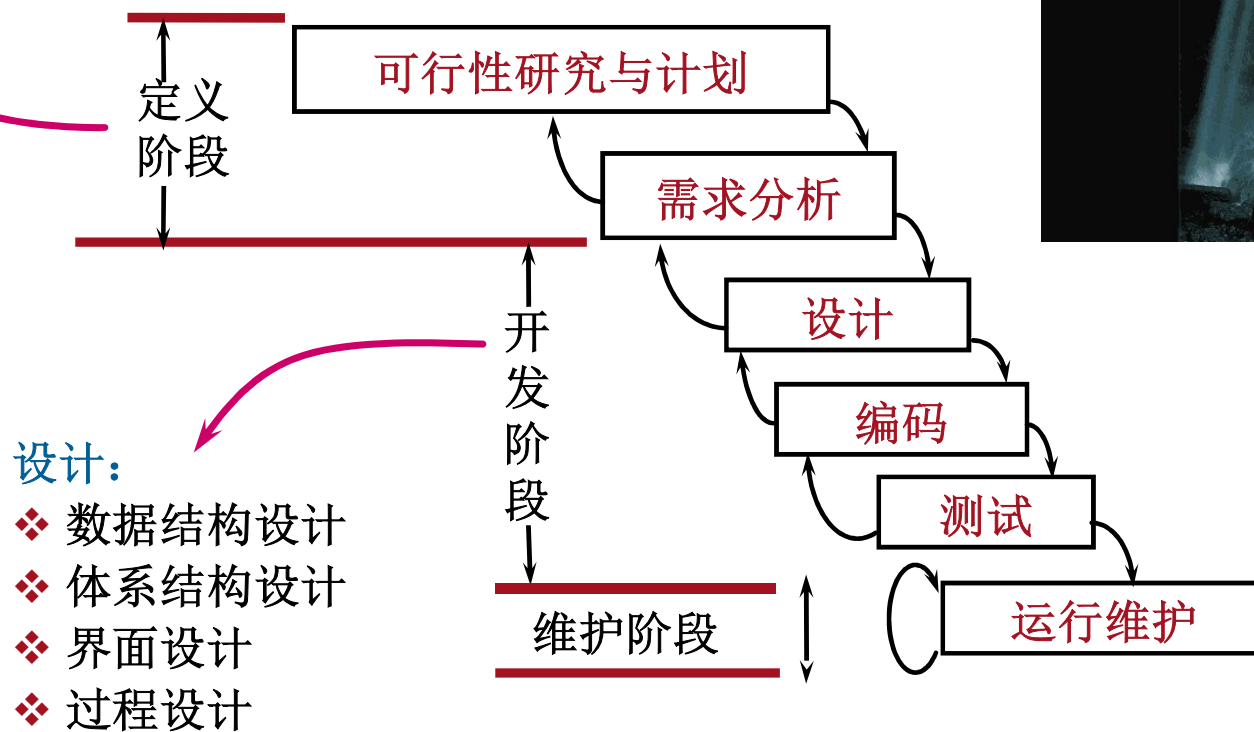
**瀑布模型**、**原型模型**、**增量模型**、**螺旋模型**、**喷泉模型**、**敏捷过程模型**、**渐进交互迭代模型**、以及**组合模型**等。

# 软件过程模型

## 瀑布模型（线形顺序模型）

### 分析：

- ❖ 系统需求
- ❖ 软件需求

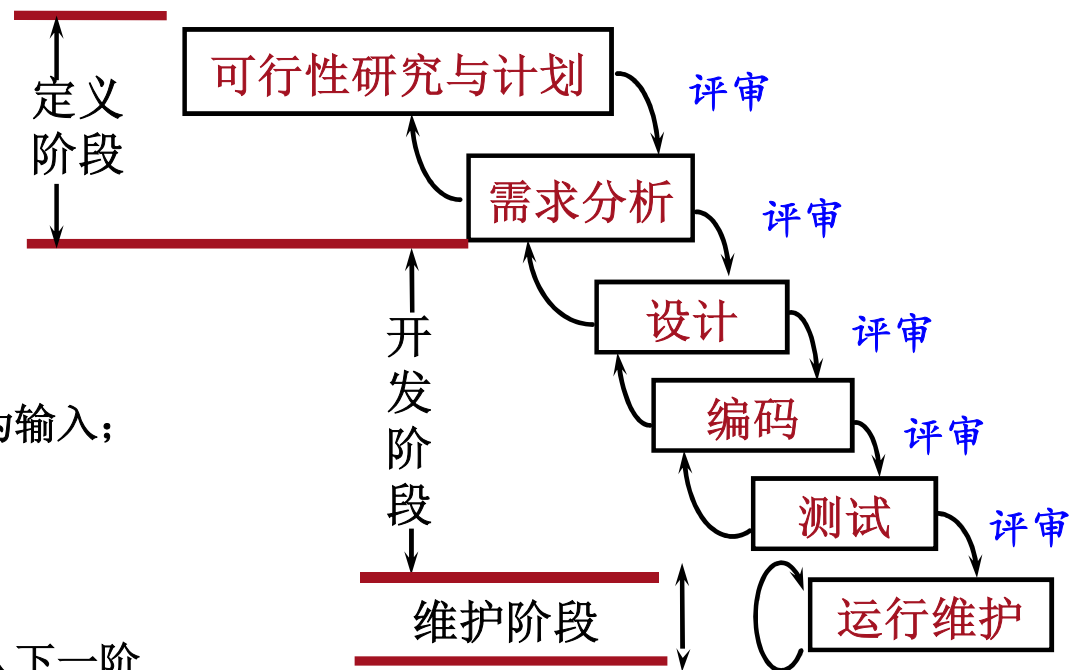


# 软件过程模型

## 瀑布模型（线形顺序模型）

### 软件开发的基本原则：

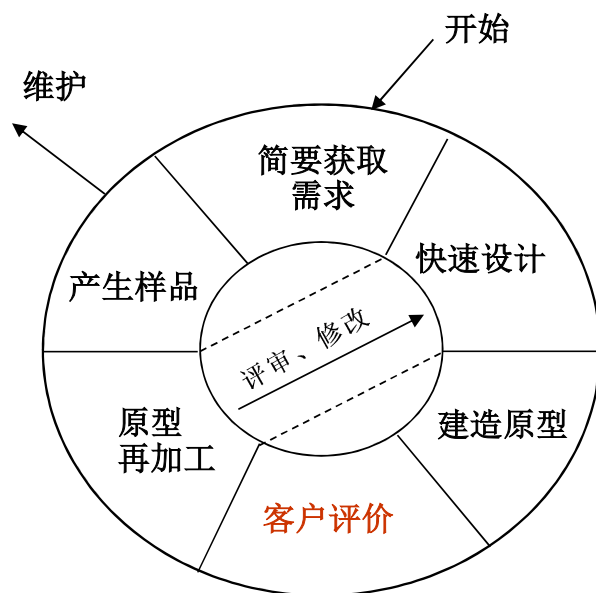
- ❖ 承接上一阶段活动中接收本阶段活动所需的内容作为输入；
- ❖ 利用输入内容进行本阶段活动；
- ❖ 把本阶段的活动成果传送下一阶段；
- ❖ 对本阶段工作进行评审，在评审正确之后，才能进入下一阶段工作。



# 软件过程模型

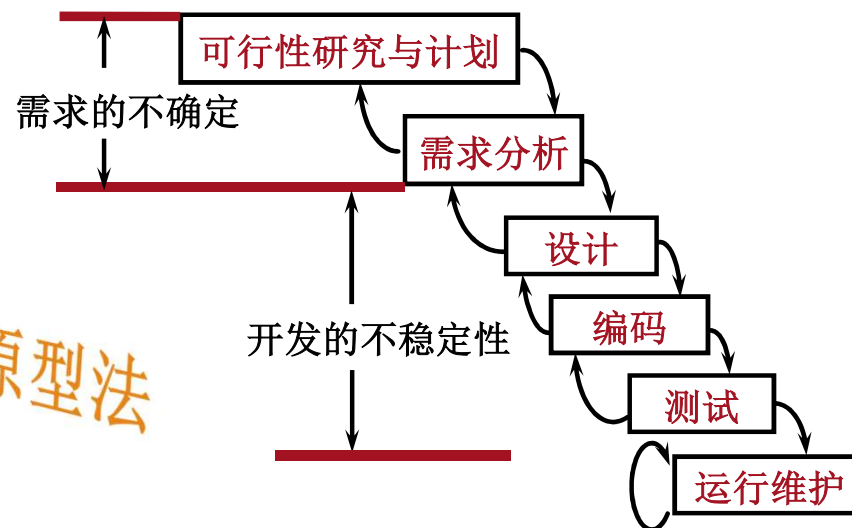
## 原型模型（演化模型）

原型是软件开发过程中一个用于实验的、测试的、或早期能运行的简单系统。由于在软件开发的需求分析阶段，难以确定用户需求，因而软件人员根据用户初步的、不明确的需求快速开发出系统原型。



原型法

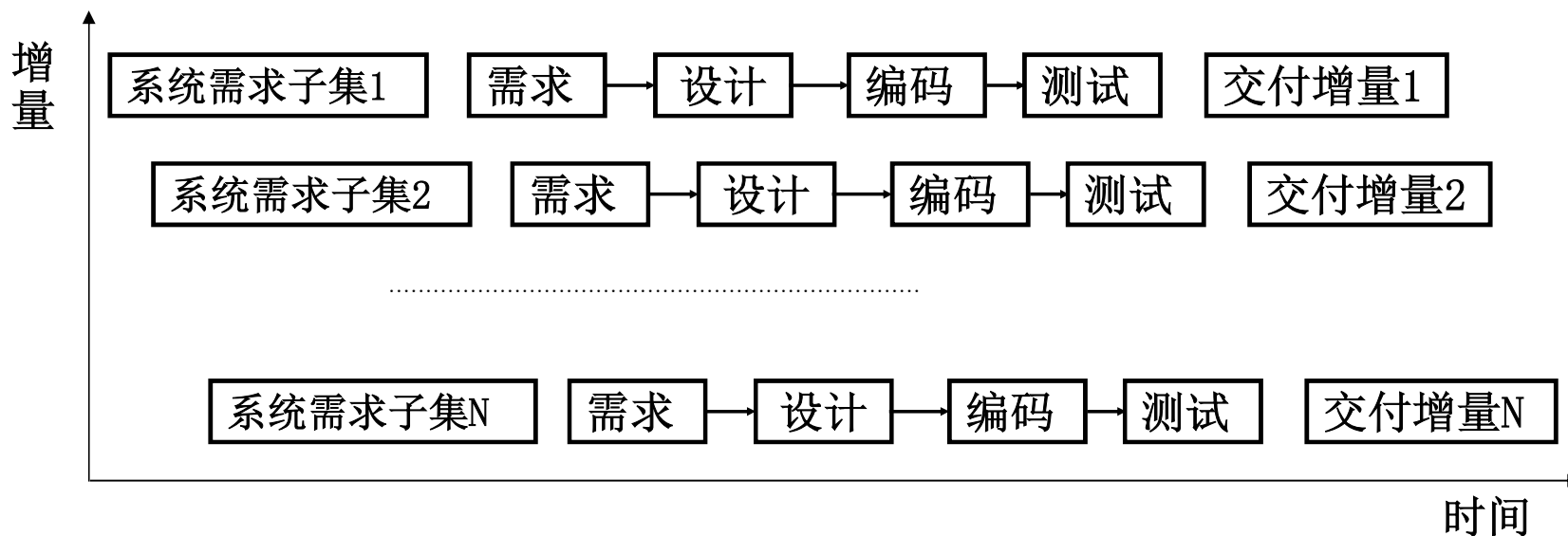
原型法





# 软件过程模型

## 增量模型 (1)



增量模型是对软件项目中功能以一系列增量的方式来开发，也被称为渐增式开发模型。**增量模型是一种非整体开发模型**，对于系统整体需求，增量模型先将需求分解为若干部分，每个部分都按照瀑布模型方法进行开发。

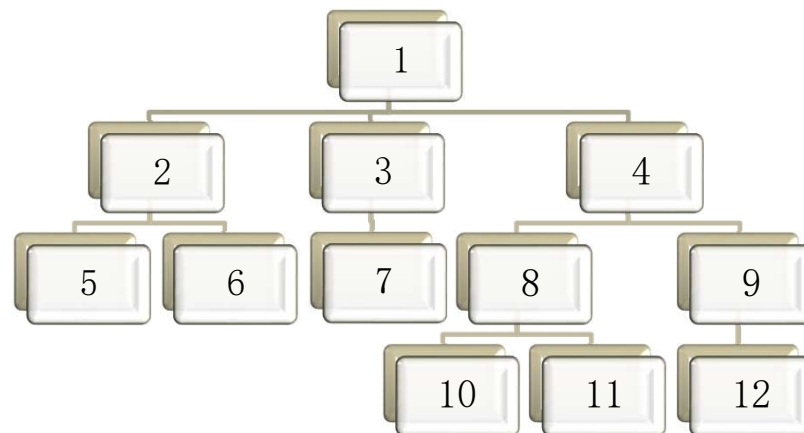
# 软件过程模型

## 增量模型 (2)

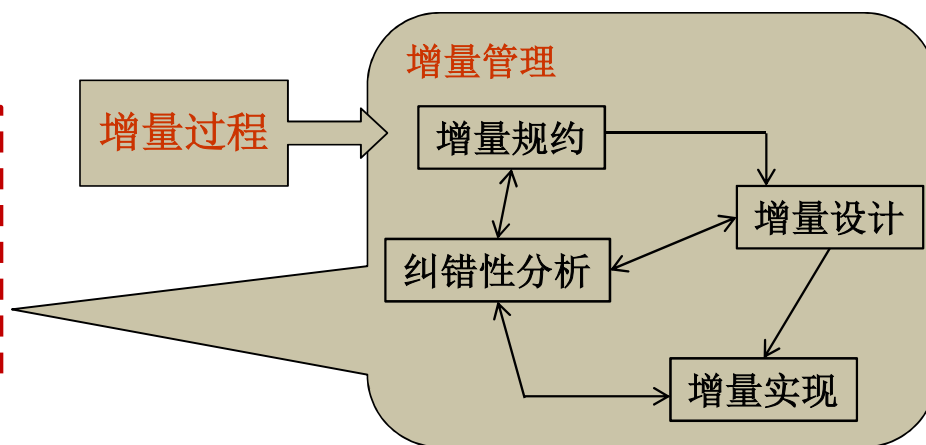
增量1: {1, 2, 5, 6}

增量2: {3, 7, 4, 9, 12}

增量3: {8, 10, 11}

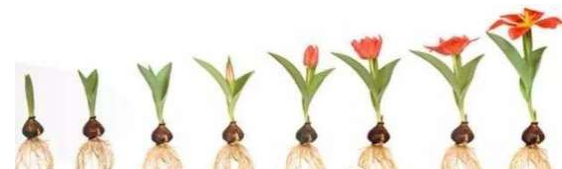


增量模型有一个假设，即需求可以分阶段开发，成为一个增量产品，每一增量可以分别开发，增量间是松散的关联关系。



# 软件过程模型

## 迭代模型——iOS的迭代过程



- 2007.1, iOS1, 触摸移动, 打电话、发短信
- 2008.7, iOS2, 支持中文, 手写输入法
- 2009.6, iOS3, 彩信, 蓝牙, 云控制
- 2010.6, iOS4, 多任务, 多文件夹
- 2011.9, iOS5, 云端功能特性, siri
- 2012.9, iOS6, 基于云的设备同步
- 2013.9, iOS7, 控制中心与通知中心
- 2014.9, iOS8, 支持第三方输入法, 智能家居功能
- 2015.9, iOS9, iPad分屏多任务, 密码扩展到6位
- 2016.6, iOS10, 增加家庭家庭应用HomeKit, Siri相连
- 2017.9, iOS11, 主要是界面的扁平化设计
- 2018.10, iOS12, 主要是新增双SIM卡
- 2019.9, iOS13, 主要是性能提升, 特别是系统越来越慢的问题
- 2020.9, iOS14, 最大变化在于主页UI上新增屏幕小组件
- 2021.6, iOS15, 支持“人声增强模式”, 屏蔽背景噪音
- 2022.11, iOS16, 主要是开放追踪Apple Pay订单
- 2023.6, iOS17, 主要是修复电池电量异常及Siri语言朗读
- 2024.6, iOS18, 即将推出, 预计新增AI (ChatGPT) 功能

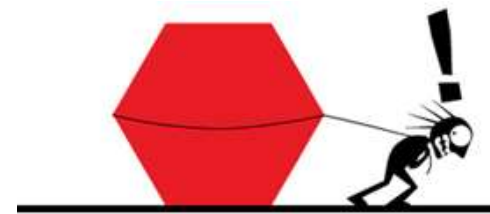
# 软件过程模型

## 敏捷过程——起源发展

在上世纪九十年代，各种**轻量级**软件开发方法被纷纷提出：

- ◆ 1991年: RAD (Rapid Application Development)
- ◆ 1994年: UP (Unified Process) 和 DSDM (Dynamic Systems Development Method).
- ◆ 1995年: Scrum
- ◆ 1996年: Crystal Clear & XP (eXtreme Programming)
- ◆ 1997年: FDD (Feature-Driven Development)

**敏捷开发 (Agile Development)** 一词来源于，2001年17位软件开发者齐聚美国犹他州的一次敏捷方法发起者和实践者的聚会，并在这之后成立了敏捷联盟。



传统、大型软件开发模型

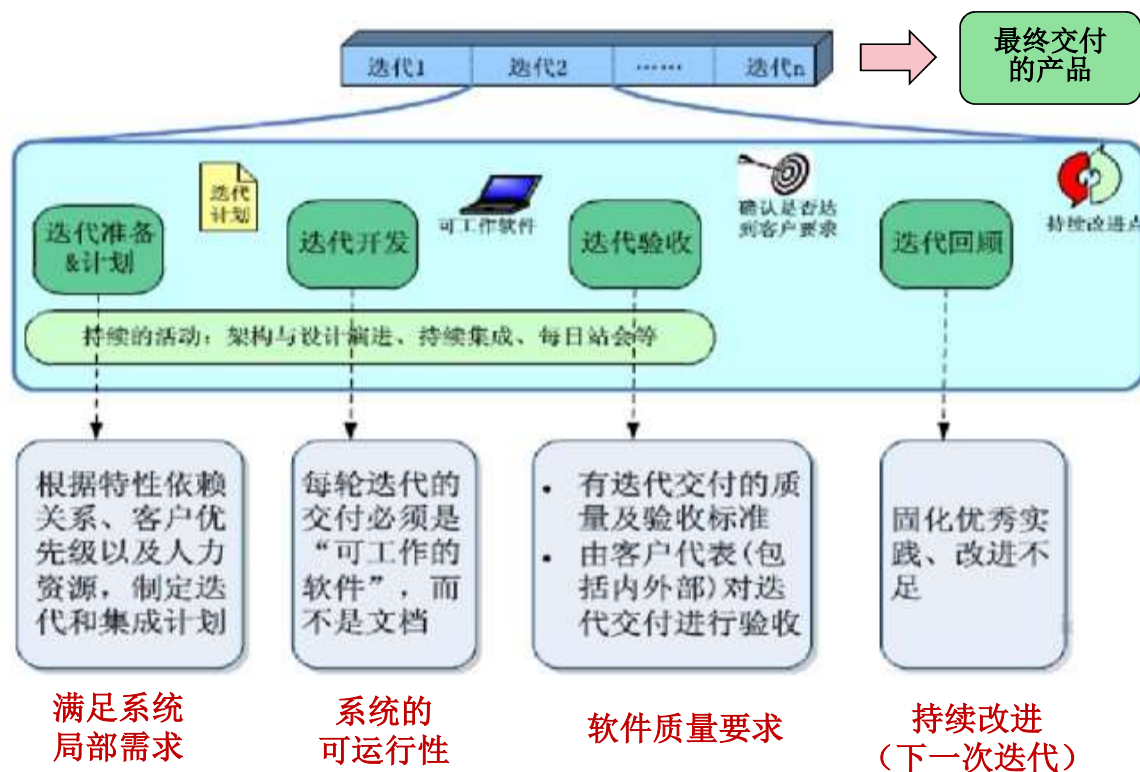


敏捷开发模型

# 软件过程模型

## 敏捷过程模型

敏捷过程模型最核心理念：**迭代**和**增量**。



价值体现：

❖ **沟通：** 个体和交互

胜过 过程和工具

❖ **简单：** 可以工作的软件

胜过 面面俱到的文档

❖ **反馈：** 客户合作

胜过 合同谈判

❖ **勇气：** 响应变化

胜过 遵循计划

# 软件过程模型

## 敏捷过程工作方式



不再是封闭、独立的个人空间，而是开放、互动的环境氛围。



# 软件过程模型

---

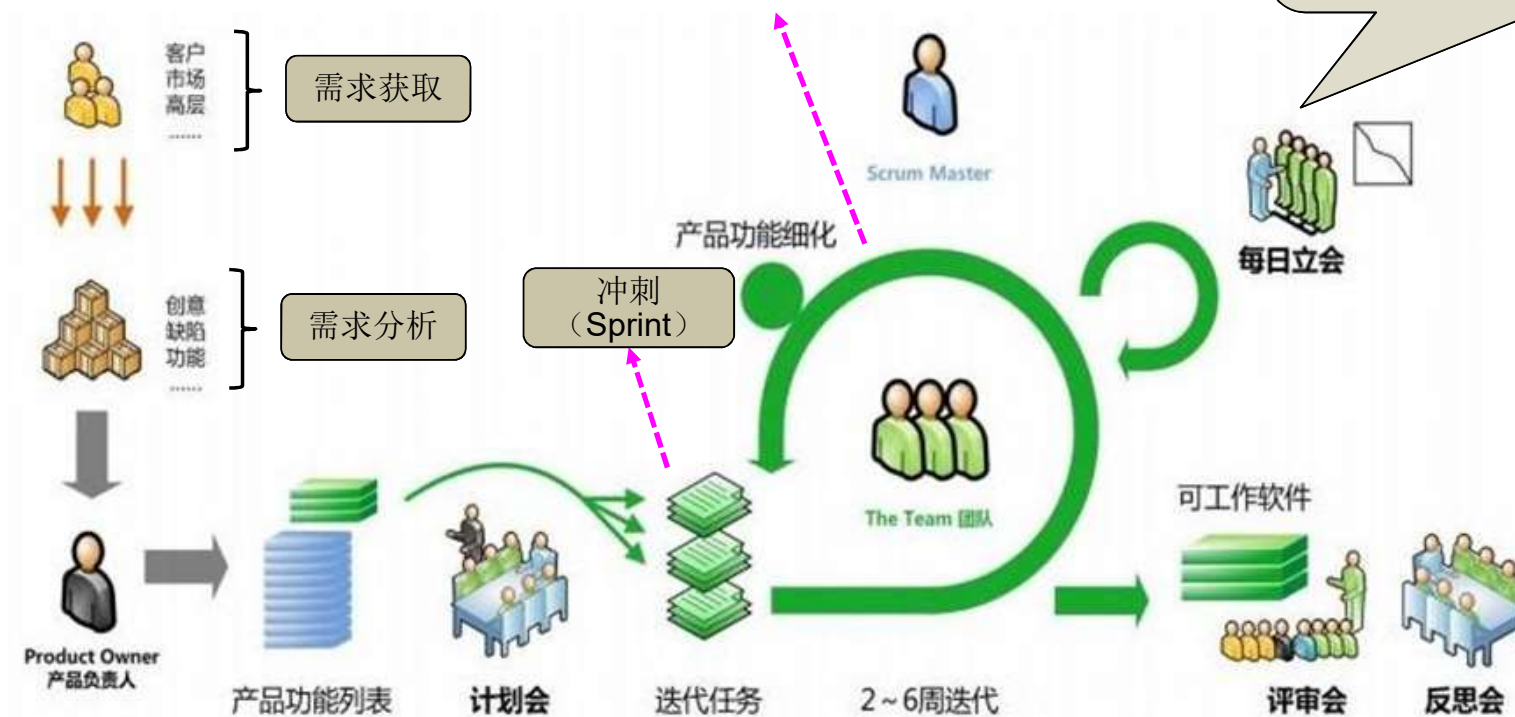
## 敏捷过程方法

- ◆ Scrum
- ◆ 极限编程（**XP: eXtreme Programming**）
- ◆ 测试驱动（**TDD: Test-Driven Development**）

# 软件过程模型

## 敏捷过程方法(1)——Scrum: 注重过程管理实践

“冲刺”中，不接受需求变更。



- “冲刺环”涉及三个主要问题:
1. 昨天完成的工作
  2. 工作中有无问题 (同步、技术、进展)
  3. 今天的任务

# 软件过程模型

## 敏捷过程方法(2)——极限编程（XP: eXtreme Programming）：偏重编程实践活动

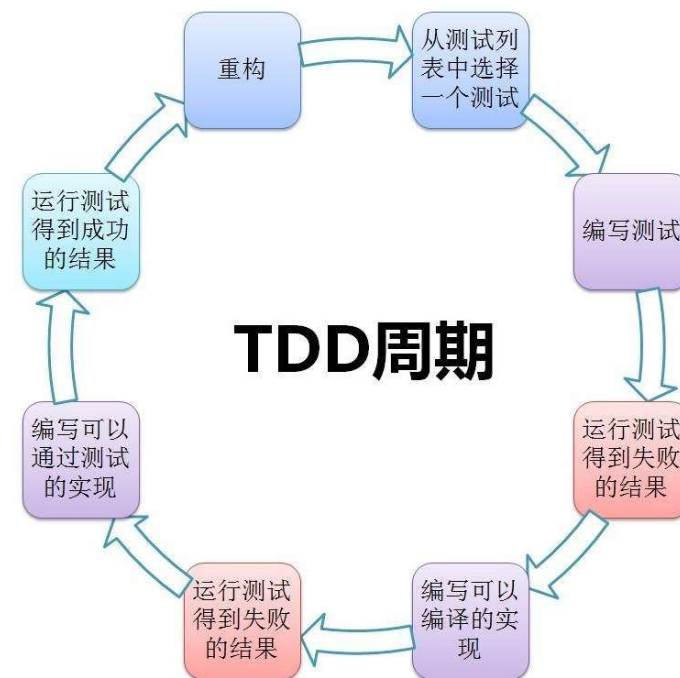
它是一个轻量级的、灵活的、迭代的敏捷开发过程，有**12**个核心实践。



# 软件过程模型

## 敏捷过程方法（3）——测试驱动（TDD: Test-Driven Development）：测试优先实践

- ❖ 先编写测试代码，再编写程序
- ❖ 程序员不断编写单元测试，在这些测试能够准确无误运行的前提下，系统开发才能再继续进行。



# 软件开发模型

## 过程模型的选择与软件“构建”(build)的不同类型

- **Build To Learn:** 开发软件，构建系统的目的是做进一步的实验，试图发现客观规律或某个实验方法的优点与缺点。这些项目常是科研论文的基础工作。
- **Build To Show:** 为了突出地展现某个技术的作用，开发一些演示为目的的软件，这些项目很吸引眼球，经常获得新闻报道，但是功能未必全面或实用。
- **Build To Serve:** 为了服务一定范围的目标用户而构建的工具等，有时以公开的SK形式发布，让别的研发人员使用。
- **Build To Win:** 以在市场上赢得用户为目标而构建的软件。这也是种种科学发现，技术突破最好的试金石。所有以盈利为目的的公司和团队都在为此努力。

*Microsoft China*  
*Principal Software Development Manager*  
*Xin Zou (邹欣)*

# 软件开发方法

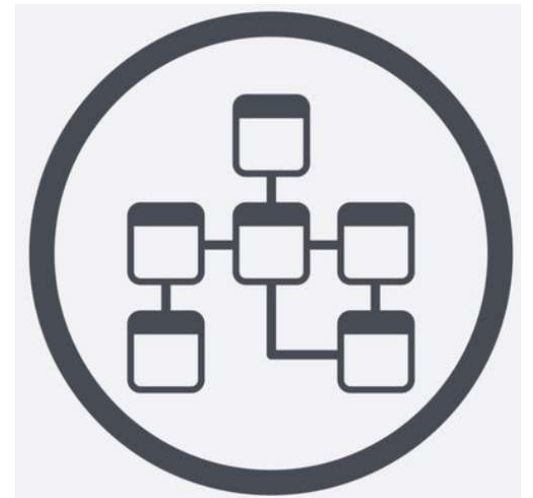
## 软件开发方法（1）——结构化开发

1978年由**E. Yourdon**和**L. L. Constantine**提出，也称为面向功能的软件开发方法或面向数据流的软件开发方法。

- 用结构化分析（Structure Analysis, SA）对软件进行需求分析
- 用结构化设计（Structure Design, SD）方法进行系统设计
- 用结构化编程（Structure Programming, SP）实现软件。

结构化方法提出的开发过程步骤明确，SA、SD和SP三个阶段彼此衔接，前后照应。

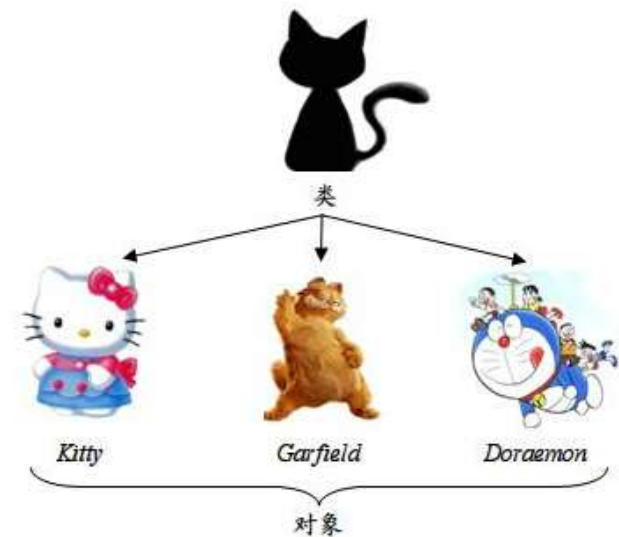
结构化设计 = 数据结构 + 算法



# 软件开发方法

## 软件开发方法（2）——面向对象开发

面向对象思想符合人们对客观世界的认识和描述。客观世界的实体是人们认识世界的基础，面向对象开发方法的基础是对象，而对象就是对客观事务（问题空间）的直接描述。



面向对象 = 对象 + 类 + 继承 + 消息





# DevOps是什么

## DevOps的概念

维基百科, <http://zh.wikipedia.org/wiki/DevOps>

DevOps (Development和Operations的组合词) 是一种重视“软件开发人员 (Dev)” 和“IT运维技术人员 (Ops)” 之间沟通合作的文化、运动或惯例。透过自动化“软件交付”和“架构变更”的流程, 来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。

## DevOps专家的观点

• DevOps打破开发与运维之间的**孤岛**, 积极协作。广义上指整个组织级的协作优化, 包括IT、HR、财务、和公司供应商。约束理论告诉我们, 需要优化整体而非单个“孤岛”。整体是与客户问题相关的业务, 从精益角度来说, 是整个价值链。

Patrick Debois, DevOps名词的发明和运动的推动者, DevOps Days创办人之一。

• DevOps通常指的是**新兴的专业化运动**, 提倡开发和IT运维之间的高度协同, 从而在完成高频率部署的同时, 提高生产环境的可靠性、稳定性、弹性和安全性”。DevOps聚焦实现更快的上市时间, 减少IT浪费, 提高组织效率。

Gene Kim, DevOps领导者, 《The Phoenix Project: A Novel About IT, DevOps》、《The Visible Ops Handbook》作者。

## DevOps领先企业的观点

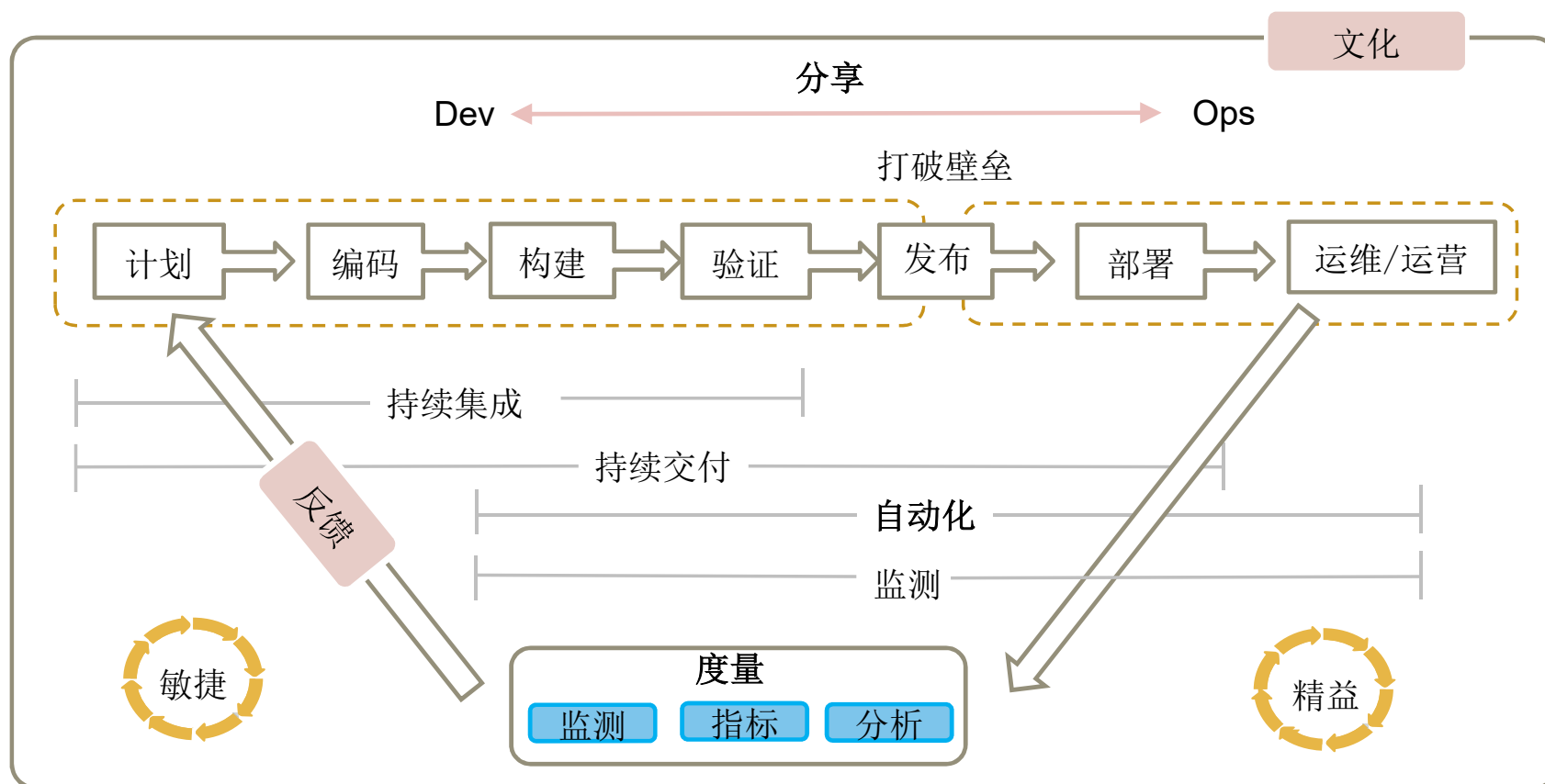
### 亚马逊

- DevOps侧重于改善协作, 沟通, 整合软件开发与IT运维。是一个总称, 用来描述一种**理念、文化变革和模式转变**。
- DevOps帮助亚马逊从软件开发**敏捷走向业务和运维的敏捷**。
- 开发阶段, DevOps重点放在代码构建, 代码覆盖, 单元测试, 打包和部署; 运维阶段, 在基础设施上聚焦环境分配, 配置, 编排和部署。

### 微软

- DevOps不仅是一项技术或工具集, 也是**观念和文化的转变**。它结合人、流程、适当的工具, 使应用程序生命周期更快, 更可预测。
- 我们应将DevOps视为**过程而非目的**, 应该选定适合范围的项目, 增量式实施, 通过这些项目展示成功, 学习, 并演进。

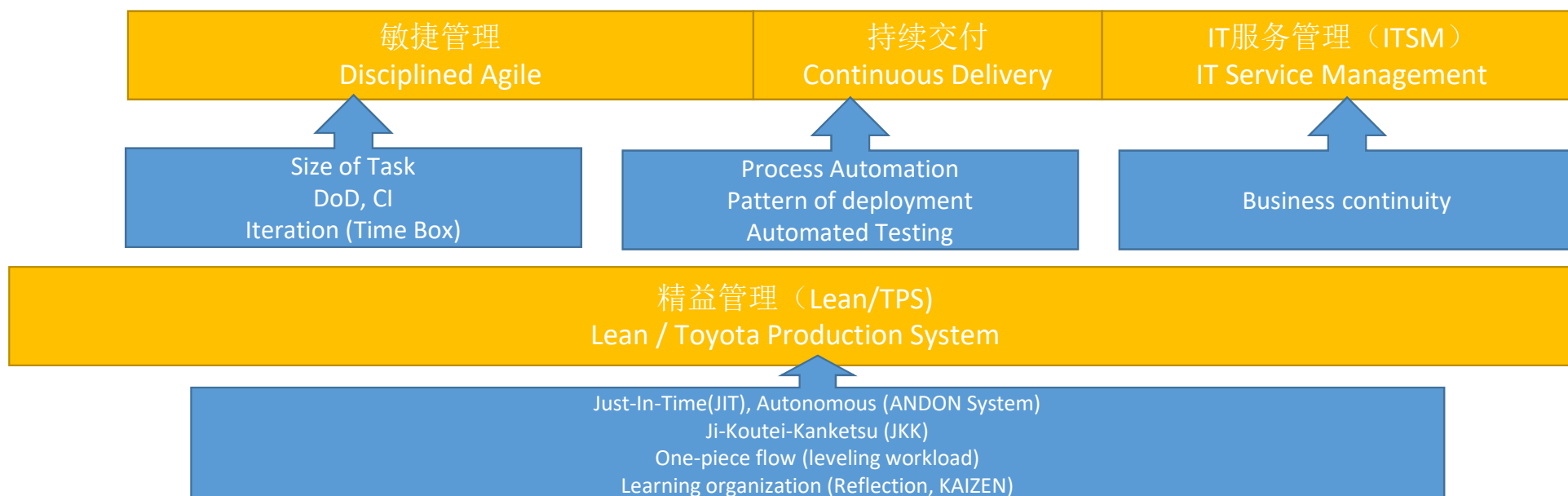
# DevOps生命周期过程



# 敏捷和DevOps关系

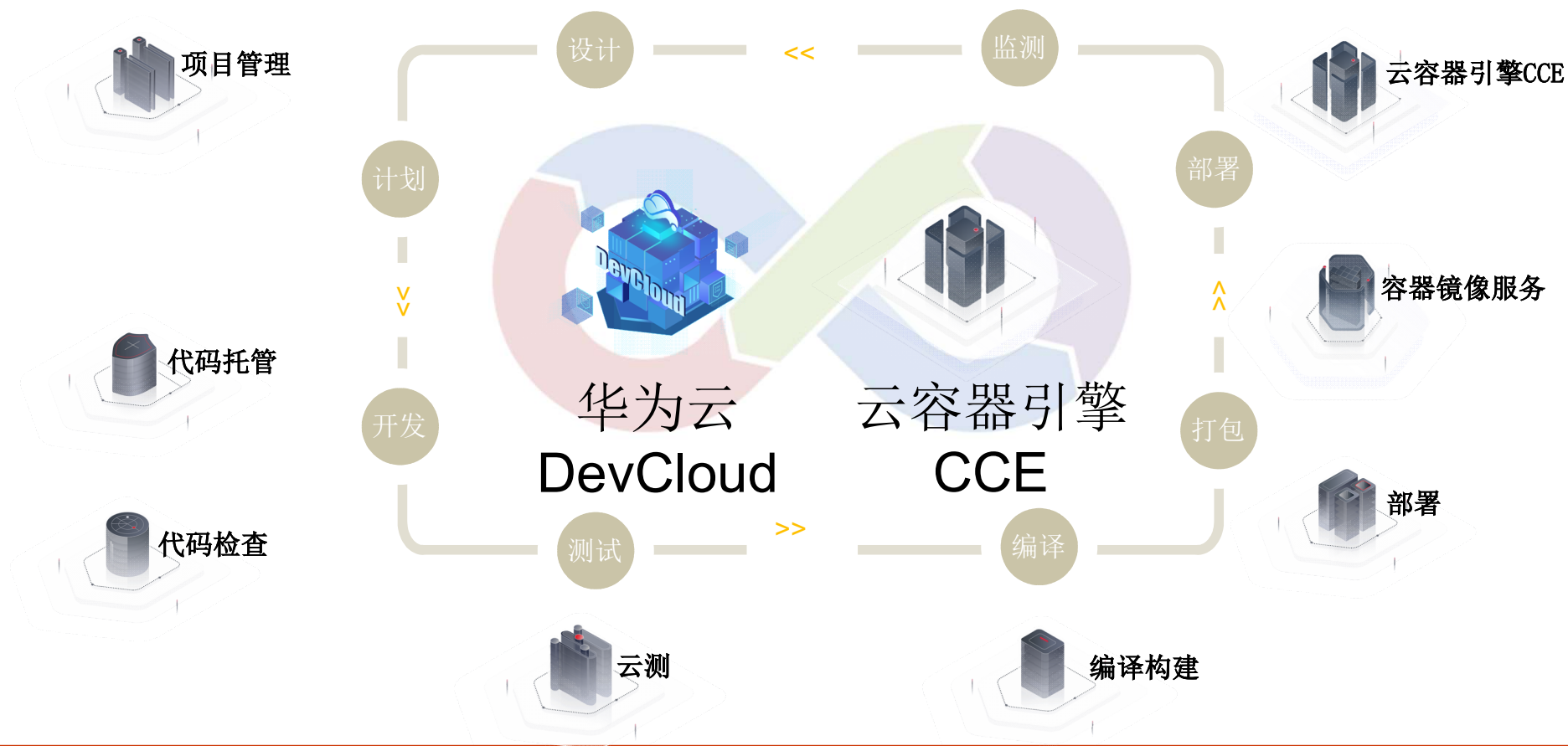
- EXIN DevOps Master白皮书由EXIN（国际信息科学考试学会）推出；DevOps知识体系如下图，
- 包括如下构成部分：敏捷管理、持续交付、IT服务管理（ITSM）、精益管理（Lean/TPS）。

周期终止  
EOL



# 基于华为云DevCloud和云容器引擎的DevOps流水线

<https://edu.hicomputing.huawei.com>



# 第一章 软件工程概述 小结

---

- ❖ 软件工程的发展历程
- ❖ 软件工程的概念
- ❖ 软件与软件工过程
- ❖ 软件过程模型
- ❖ 软件开发方法