上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

软件工程原理与实践
SOFTWARE ENGINEERING

软件工程引论

沈备军

饮水思源 · 爱国荣校

大 纲

01-什么是软件

软件的作用、发展、定义和特性
软件危机和问题

02-什么是工程

03-什么是软件工程

软件工程的概念和知识域
软件工程技术
软件建模及方法

@第1章和第3章.教材

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!

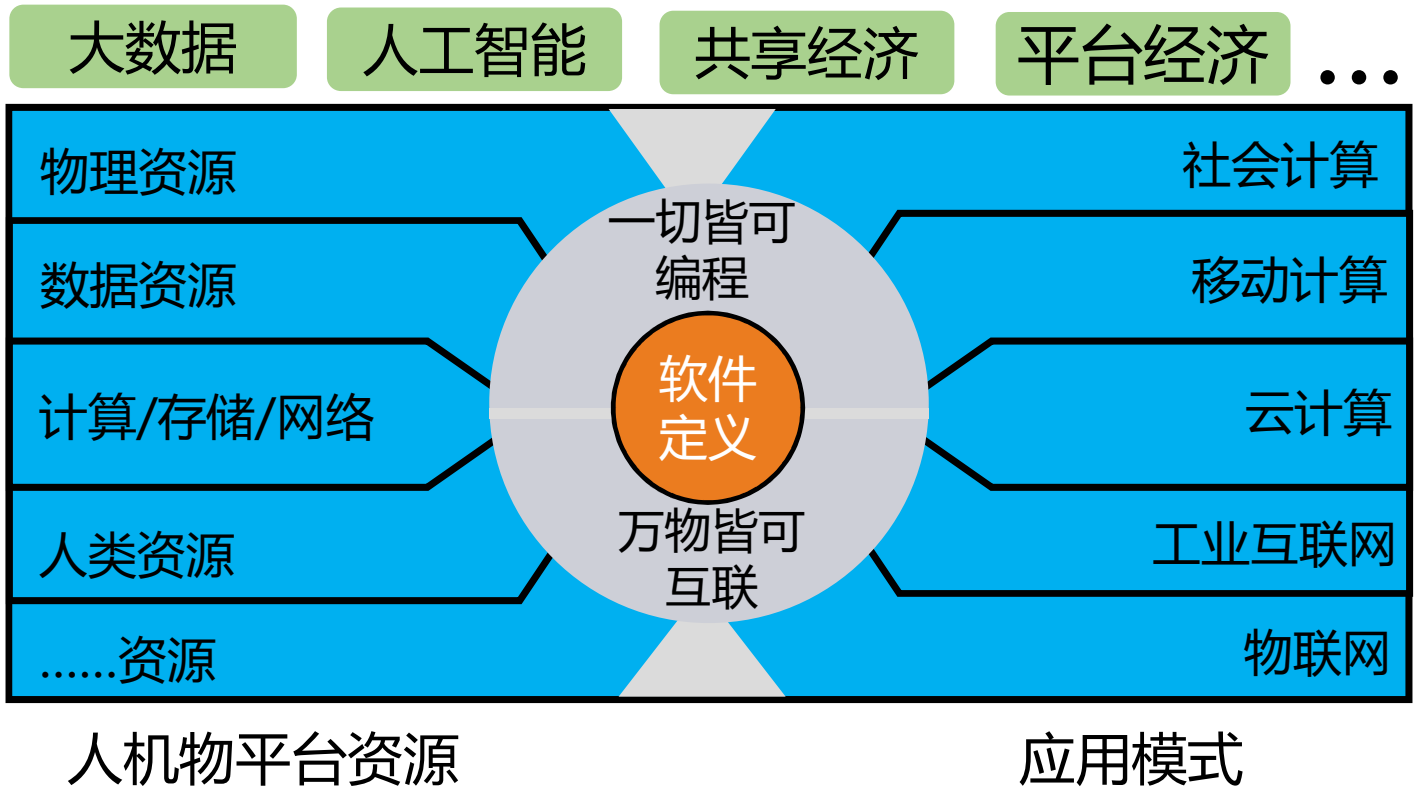
人类文明运行在软件之上！

Bjarne Stroustrup
C++语言发明人

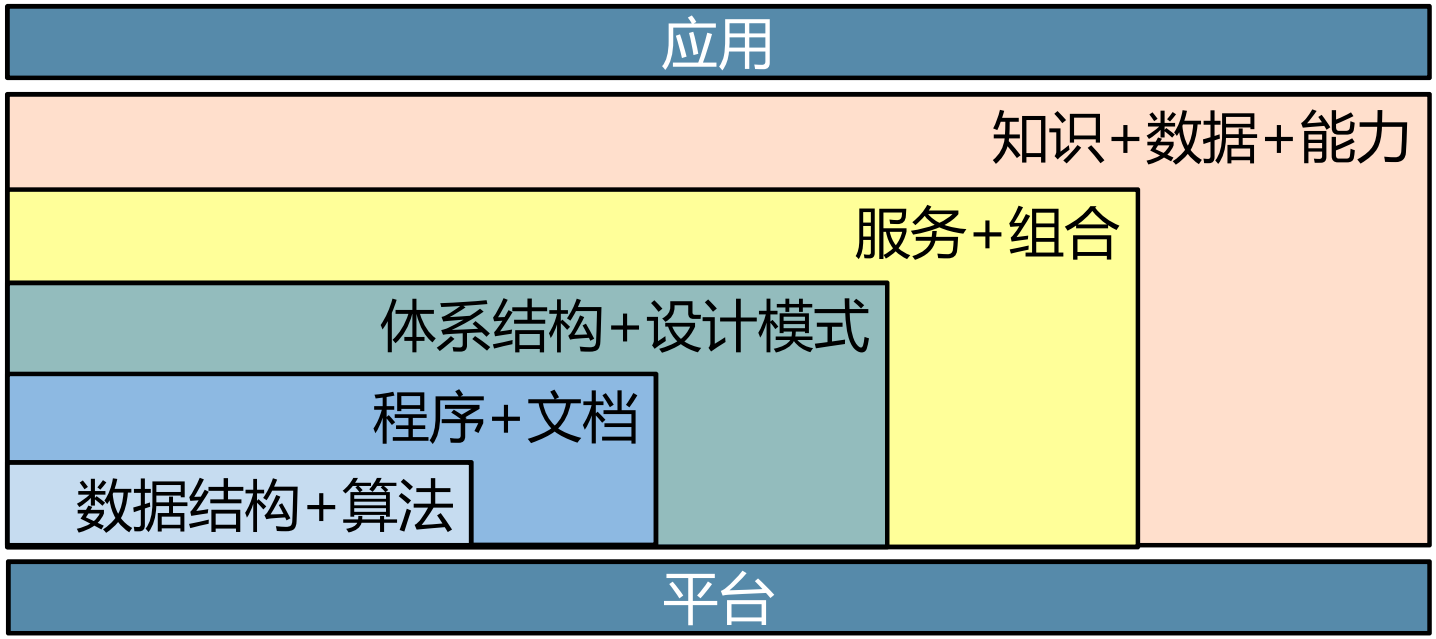
IEEE Computer，January，2012, pp 47-58

Our civilization runs on software

软件成为社会基础设施


大数据
人工智能
共享经济
平台经济
...
物理资源
数据资源
计算/存储/网络
人类资源
.....资源
一切皆可编程
软件定义
万物皆可互联
社会计算
移动计算
云计算
工业互联网
物联网
人机物平台资源
应用模式

什么是软件



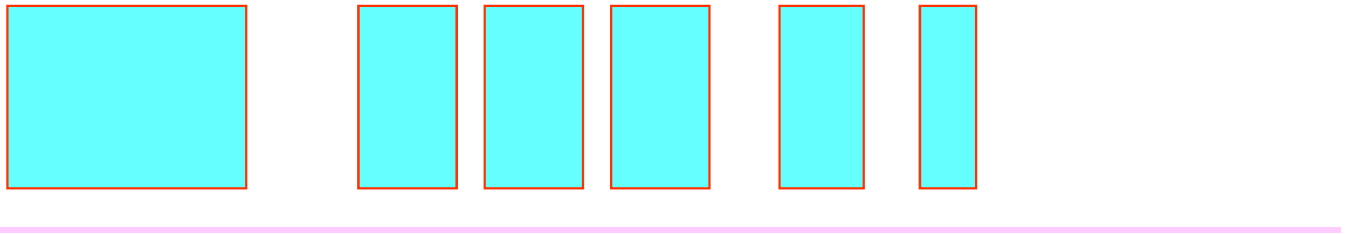
平台和应用是软件发展的外在驱动力

软件特征

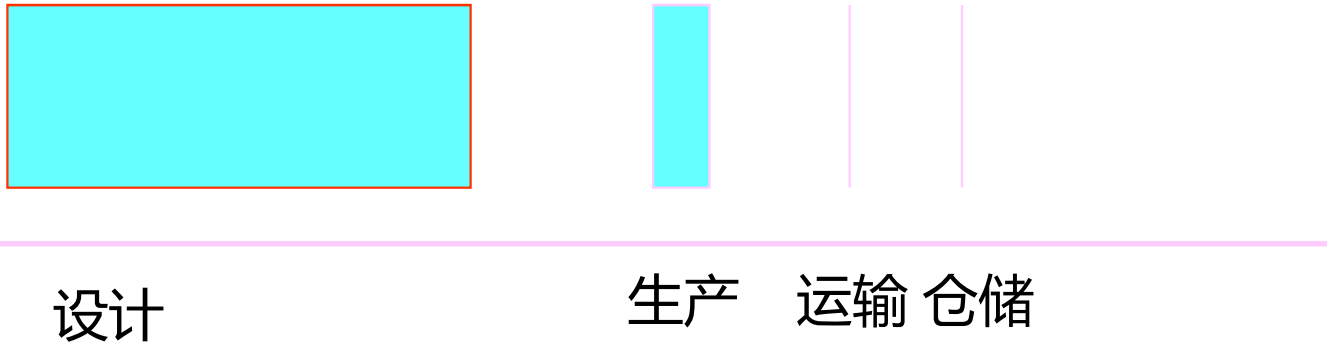
- 软件开发不同于硬件设计
  - 与硬件设计相比，软件更依赖于开发人员的业务素质、智力，以及人员的组织、合作和管理。同时对硬件而言，设计成本往往只占整个产品成本的一小部分，而软件开发占整个产品成本的大部分。
- 软件生产不同于硬件制造
  - 硬件设计完成后就投入批量制造，制造也是一个复杂的过程，其间仍可能引入质量问题；而软件成为产品之后，其制造只是简单的拷贝而已。软件的仓储和运输也非常简单。
- 软件维护不同于硬件维修
  - 硬件在运行初期有较高的故障率，在缺陷修正后的一段时间中，故障率会降到一个较低和稳定的水平上。随着时间的改变，故障率将再次升高，这是因为硬件会受到摩损等损害，达到一定程度后就只能报废。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

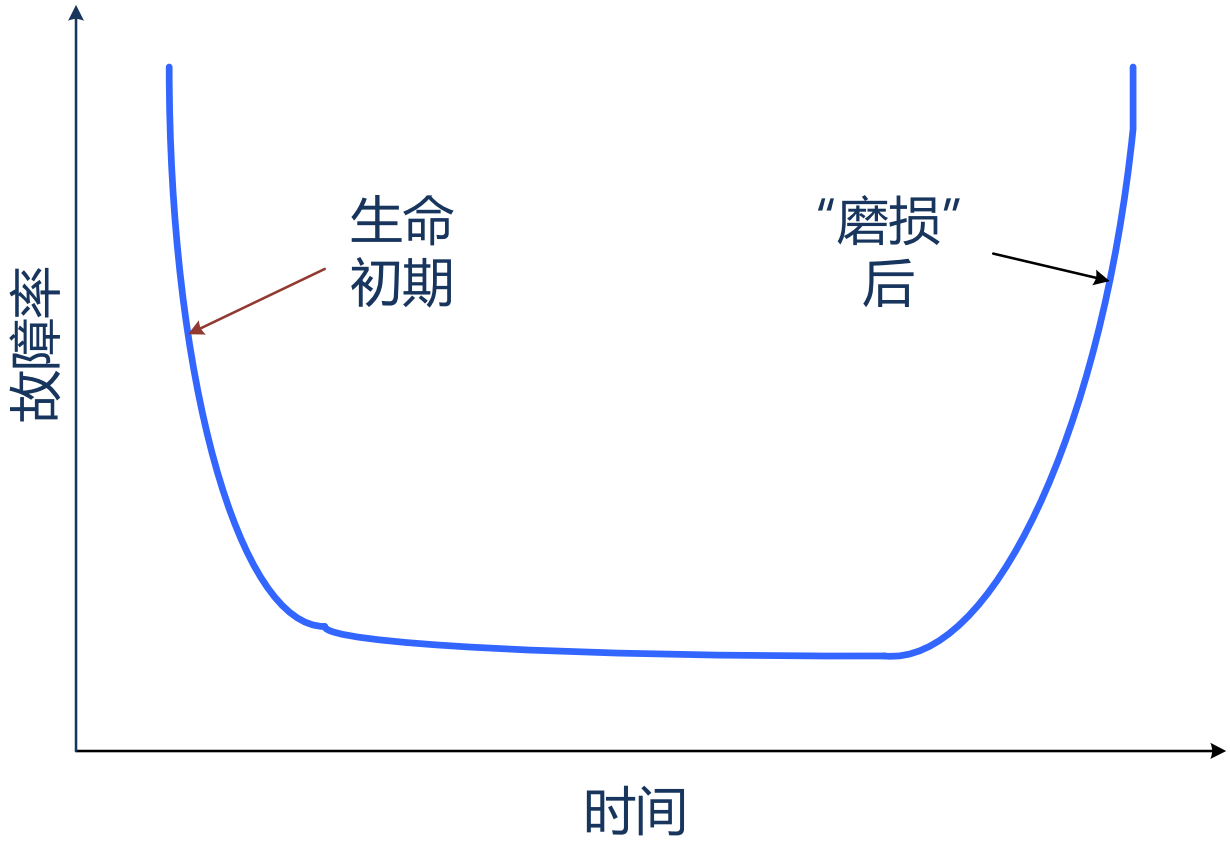
制造业：



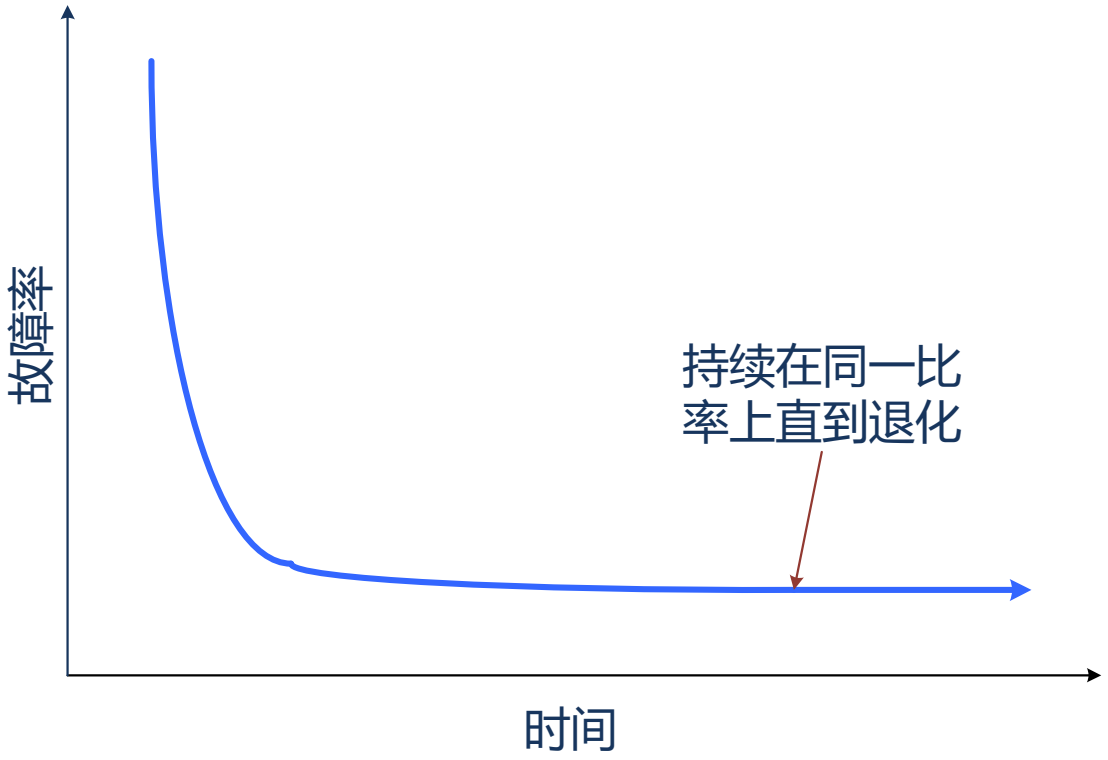
软件业：



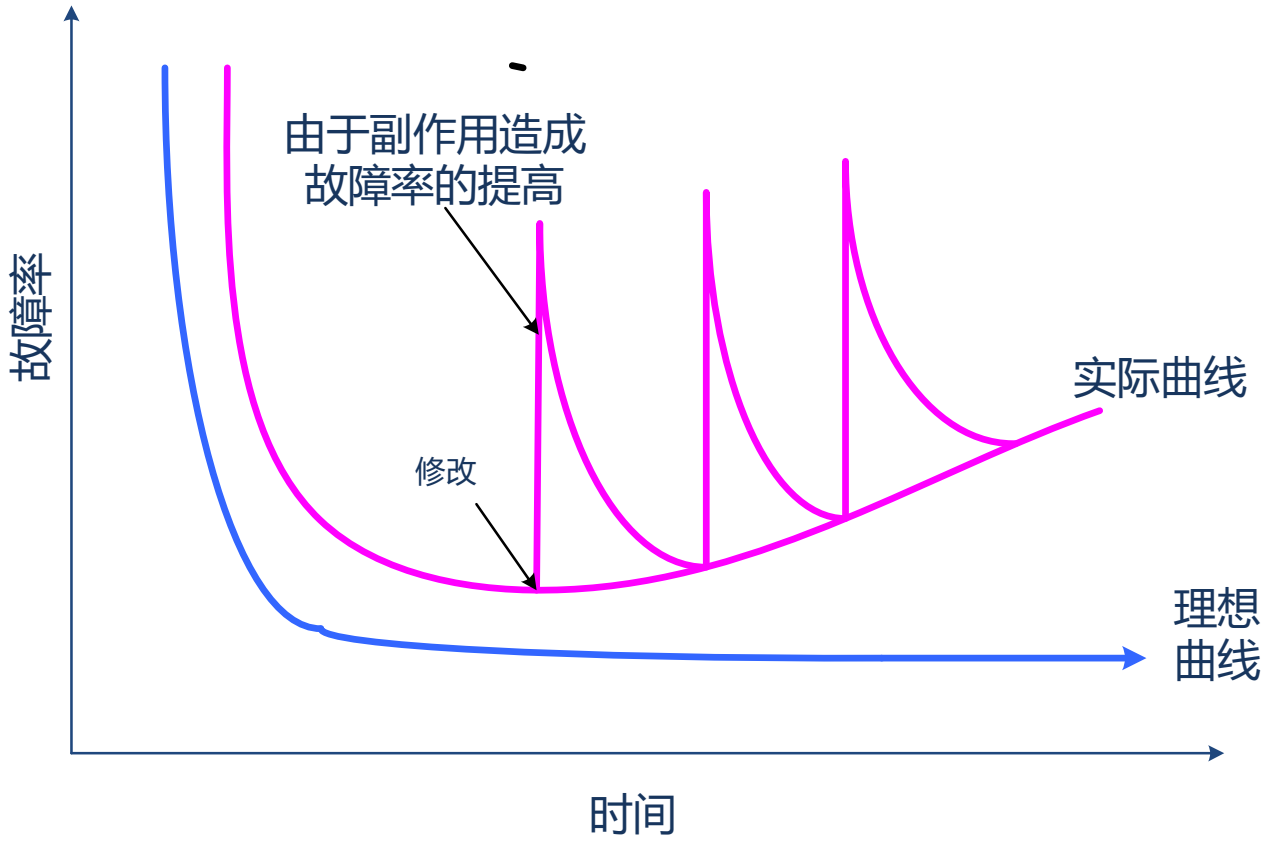
硬件的故障率曲线（浴缸曲线）



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)



软件面临的新挑战

- 软件复杂性的增加
- 软件规模的不断扩大
- 软件环境的变化
- 遗留系统（Legacy System）的集成和复用
- 软件开发的高质量和敏捷性要求
- 分散的开发团队的协同

- 宇宙飞船的软件系统源程序代码多达2000万行，如果一个人一年编写1万行代码，那么需要2000人年的工作量。
- Windows2000 开发团队是微软历史上最大的团队，整个团队5354人，合影时动用了直升飞机。

- 网络应用
- 多种硬件 - 普适计算
- 多操作系统
- 多编程语言
- 高可信、易伸缩
- 层出不穷的新技术

大 纲

01-什么是软件

软件的作用、发展、定义和特性

软件危机和问题

02-什么是工程

03-什么是软件工程

软件工程的概念和知识域

软件工程技术

软件建模及方法

软件危机的主要表现

- 对软件开发成本和进度的估计常常不准确。开发成本超出预算，实际进度比预定计划一再拖延的现象并不罕见。
- 用户对“已完成”系统不满意的现象经常发生。
- 软件产品的质量往往靠不住。Bug一大堆，Patch一个接一个。
- 软件的可维护程度非常之低。
- 软件通常没有适当的文档资料。
- 软件的成本不断提高。
- 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

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

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30年前一样 - - “code-fix”

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。

软件工程师是否需要发执照?

中国的软件产业现状

- 软件作坊 vs. 软件工厂
- 缺少系统化和规范化的管理
- 规模小，大多数在50人以下
- 生产率低
- 质量差

和美国仍存在较大的差距！

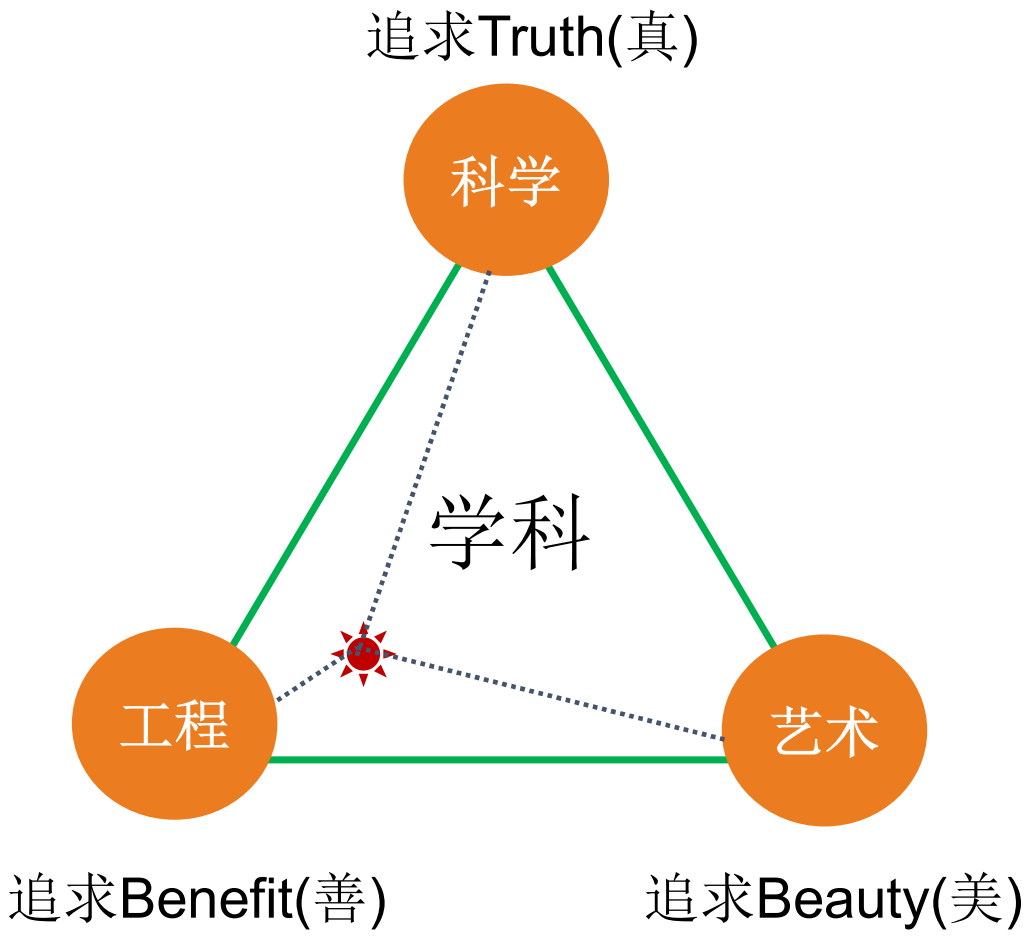
软件危机和问题的根源

- 一方面是与软件本身的特点有关
- 另一方面是由软件开发和维护的方法、过程、管理、规范和工具不正确有关

如何解决软件危机和问题?

软件开发是...?

- 软件开发
  - 是一门艺术?
  - 是一门科学?
  - 是一门工程?



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，具有竞争优势
  - 技术优势
  - 政策优势
  - 资源优势

大 纲

01-什么是软件

软件的作用、发展、定义和特性
软件危机和问题

02-什么是工程

03-什么是软件工程

软件工程的概念和知识域
软件工程技术
软件建模及方法

什么是工程?

- 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
  - 高度的组织管理性
  - 多种学科的综合
  - 高度的实践性

如何做工程？

- 一个工程实践包括以下四个核心步骤：
  - 理解问题。在软件工程领域，就是需求分析；
  - 规划方案。在软件工程领域，就是设计；
  - 实施方案。在软件工程领域，就是编码；
  - 验证结果的准确性。在软件工程领域，就是测试和质量保证。

大 纲

01-什么是软件

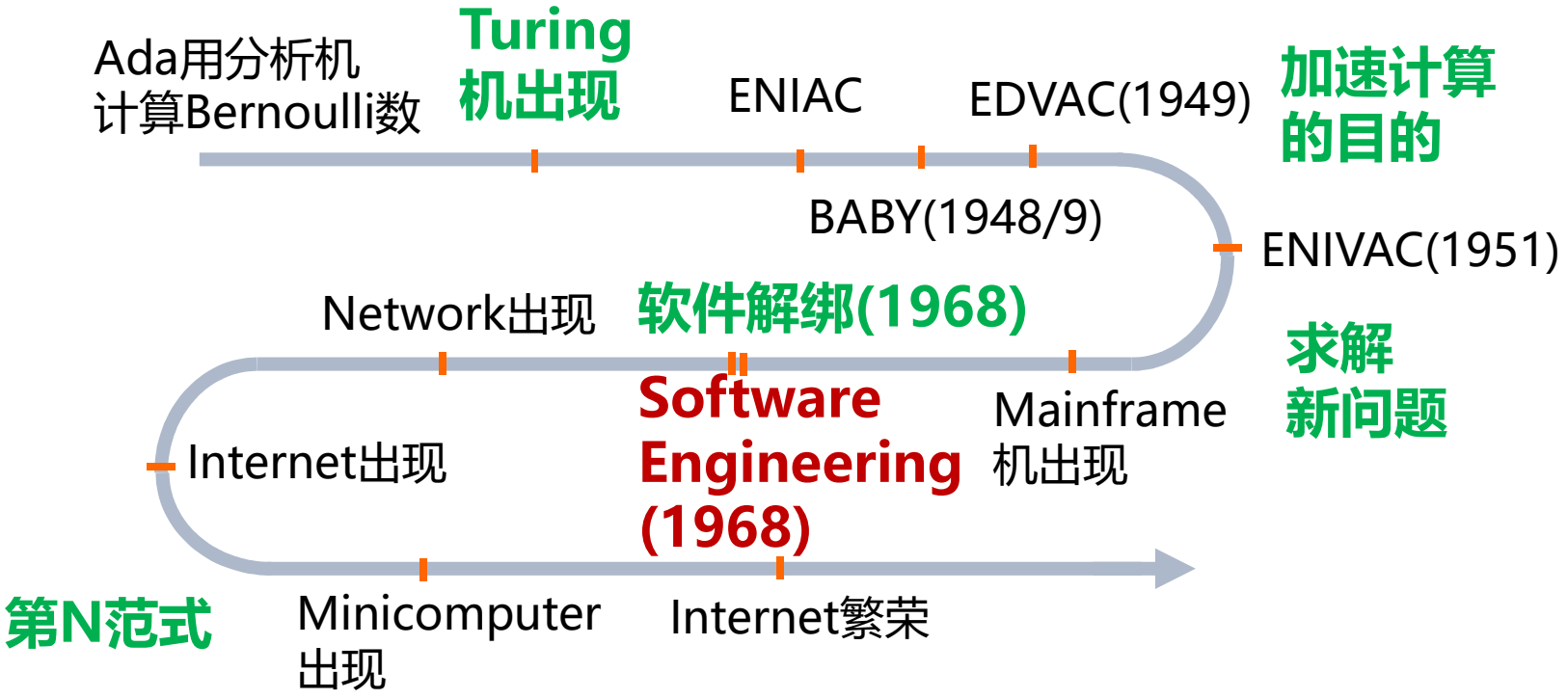
软件的作用、发展、定义和特性
软件危机和问题

02-什么是工程

03-什么是软件工程

软件工程的概念和知识域
软件工程技术
软件建模及方法

软件工程的提出


Ada用分析机
计算Bernoulli数
Turing
机出现
ENIAC
EDVAC(1949)
加速计算
的目的
BABY(1948/9)
ENIVAC(1951)
Network出现
软件解绑(1968)
求解
新问题
Software
Engineering
(1968)
Mainframe
机出现
Internet出现
第N范式
Minicomputer
出现
Internet繁荣

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]
- 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]
- 软件工程是将系统的、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。

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

- 本课程是软件学院的最核心课程，区别于计算机科学系。

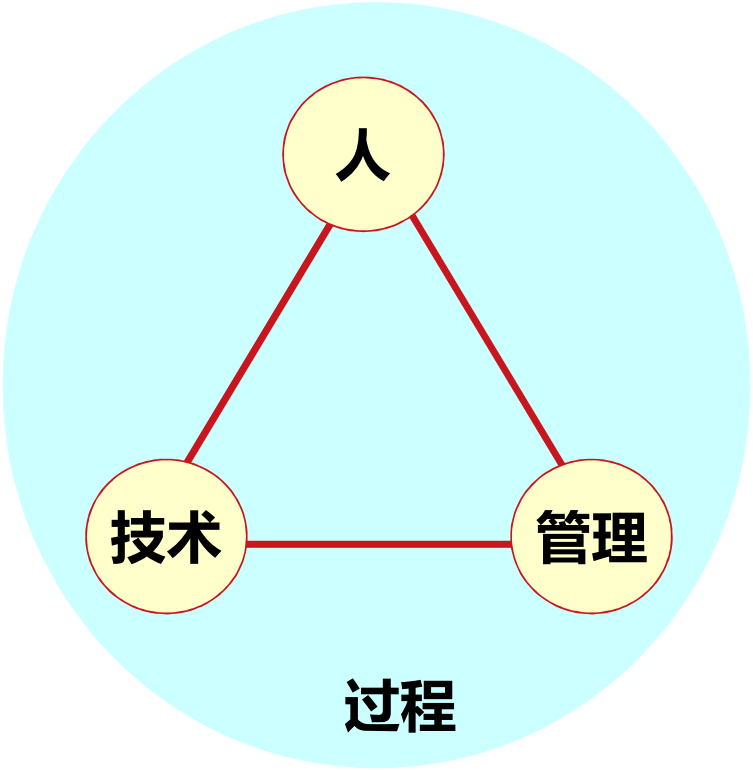
软件工程知识体系SWEBOK

- 软件工程知识体系(Software Engineering Body of Knowledge) 简称SWEBOK
  - 由IEEE国际组织推出
  - 版本：当前版本　V3(2014)
  - 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 V3的15个知识域

- 软件工程实践类知识域
  - 软件需求、软件设计、软件构造
  - 软件测试、软件维护、软件配置管理
  - 软件工程管理、软件过程管理
  - 软件工程方法、软件质量
  - 软件工程职业实践
- 软件工程教育要求类知识域
  - 工程经济基础、计算基础
  - 数学基础、工程基础



软件工程的金三角

大 纲

01-什么是软件

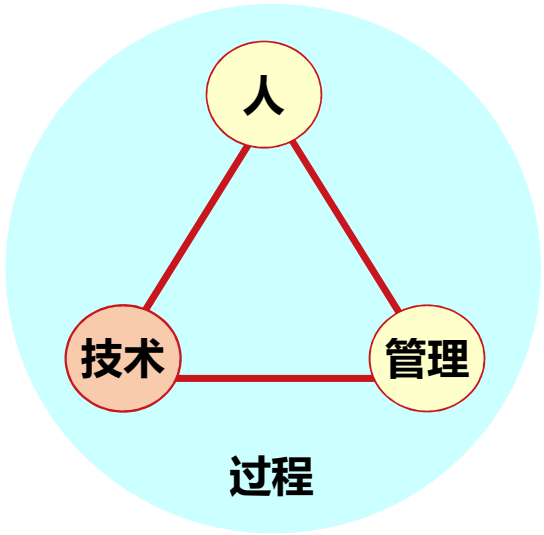
软件的作用、发展、定义和特性
软件危机和问题

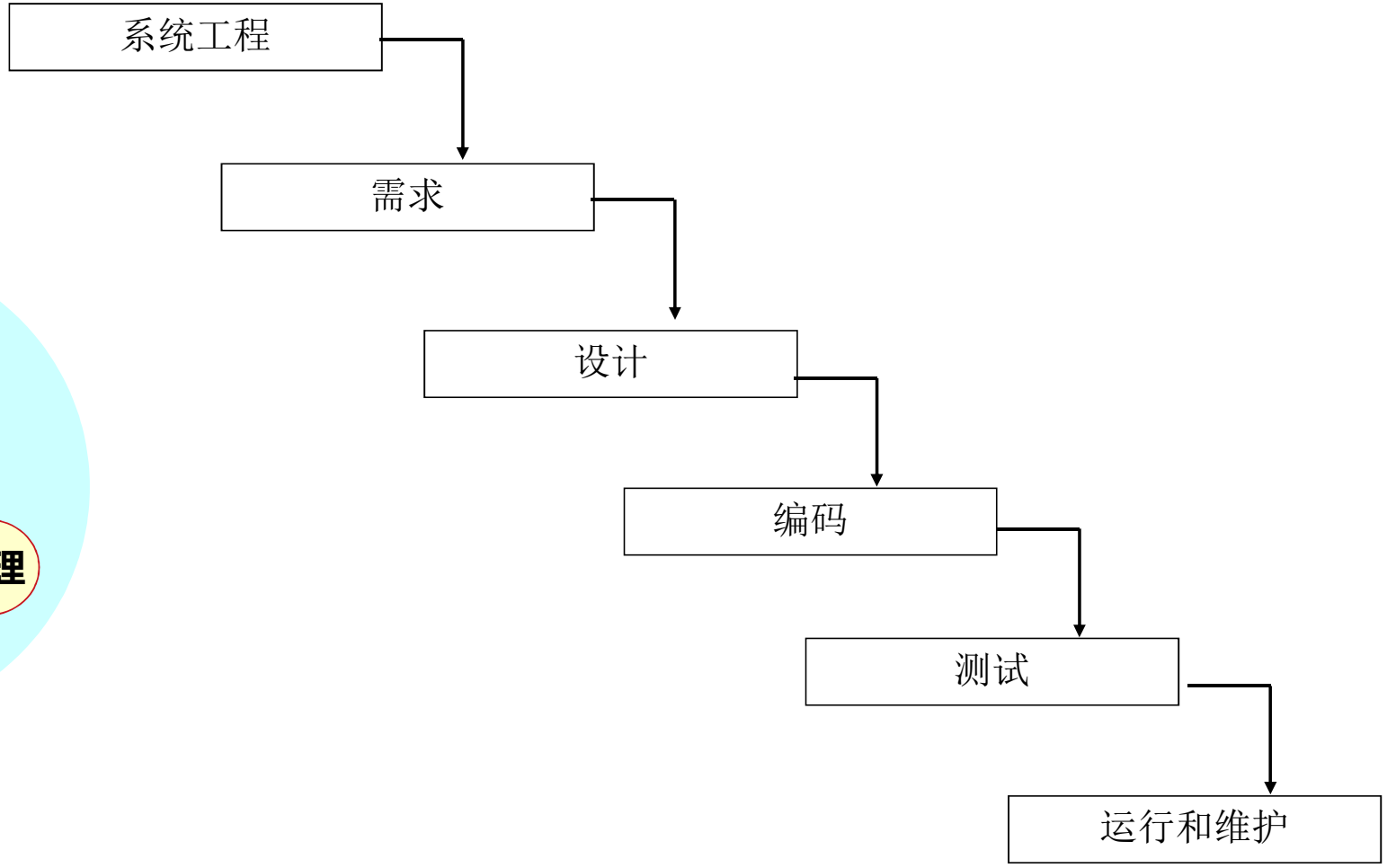
02-什么是工程

03-什么是软件工程

软件工程的概念和知识域
软件工程技术
软件建模及方法

软件工程技术





系统工程 (System Engineering)

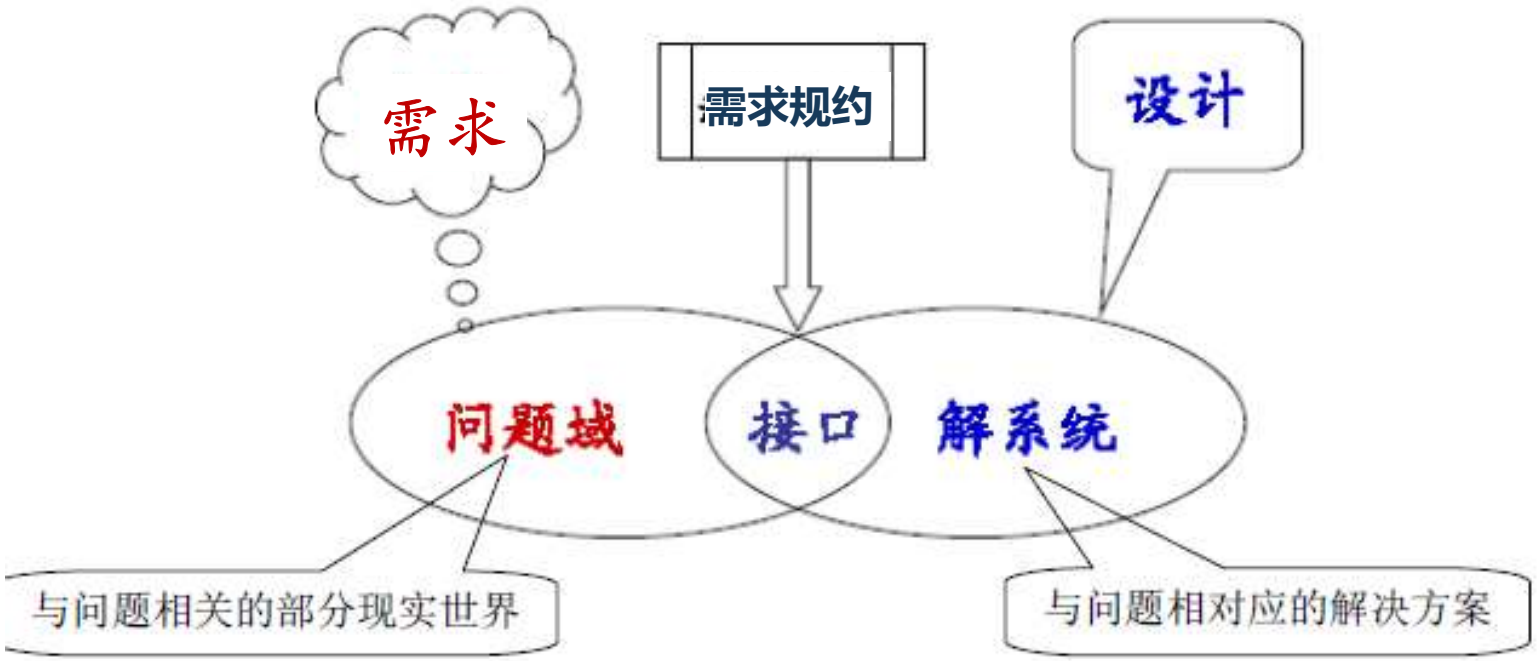
- 在软件开发之前，必须了解该软件所外的外部“系统”。
- 计算机系统包括计算机硬件、软件、人员、数据库、文档、规程等系统元素。
- 系统工程的任务：
  - 系统建模 - - 系统模型
  - 系统仿真
- 系统工程的表现形式
  - 信息系统，关注企业，业务过程工程 - - 业务模型
  - 嵌入式系统，关注产品，产品工程 - - 产品模型
  - 多媒体系统，关注内容，内容工程 - - 剧本

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)

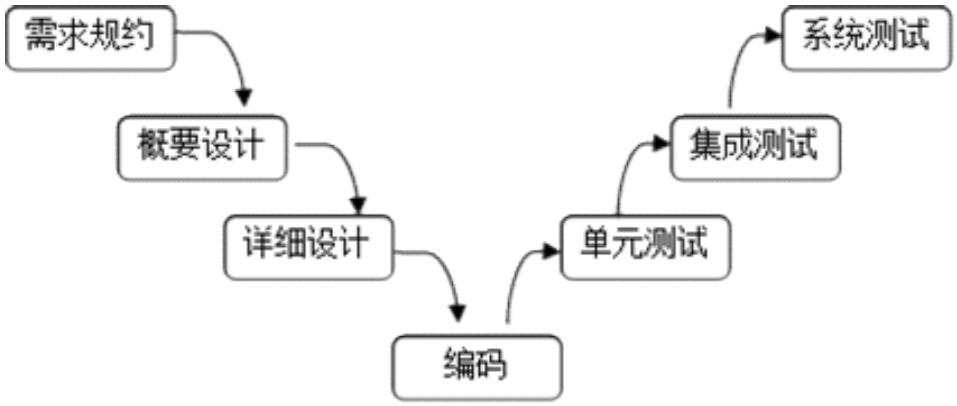
- 任务:依据设计说明书为每个模块编写程序
- 交付:程序
  - 程序符合编程规范，含有必要的注释
  - 不含有语法错误
- 交付这样的程序就是编程阶段完成的里程碑

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)

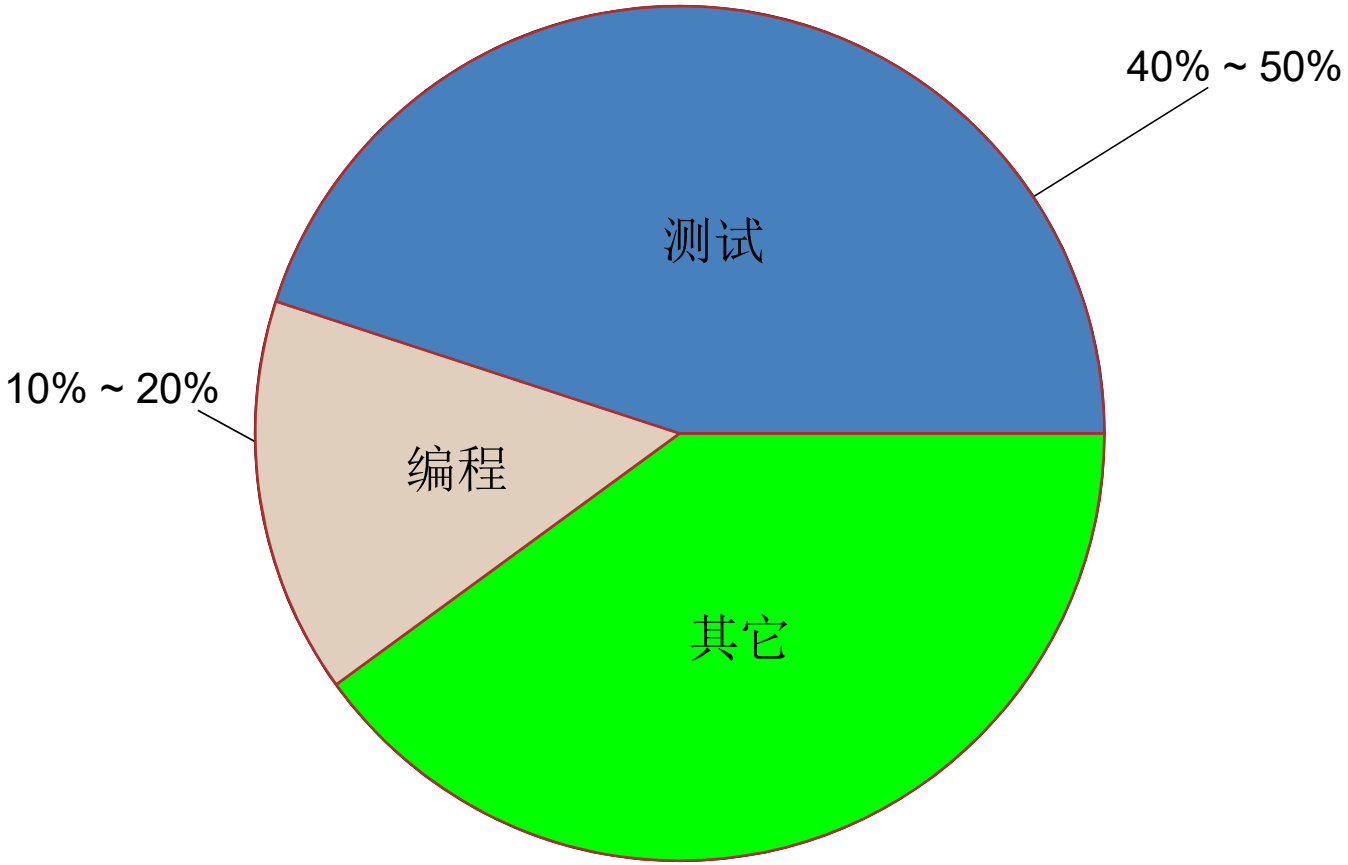
- 任务：发现并排除软件中的缺陷
- 不同层次的测试：
  - 单元测试 (Unit testing)
  - 集成测试 (Integration testing)
  - 系统测试 (System testing)



部署 (Deployment)

- 任务：交付、支持和反馈
- 部署原则：
  - Manage customer expectations for each increment
  - A complete delivery package should be assembled and tested
  - A support regime should be established
  - Instructional materials must be provided to end-users
  - Buggy software should be fixed first, delivered later

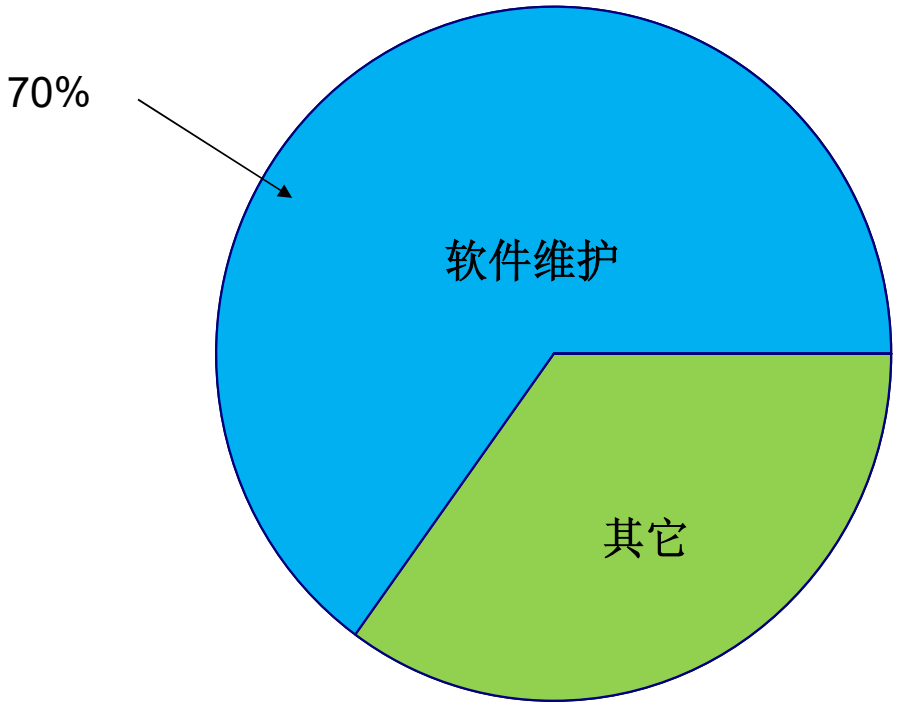
软件开发工作量分配比例



运行和维护 (Operation and Maintenance)

- 所谓维护，是指软件系统交付使用以后，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。

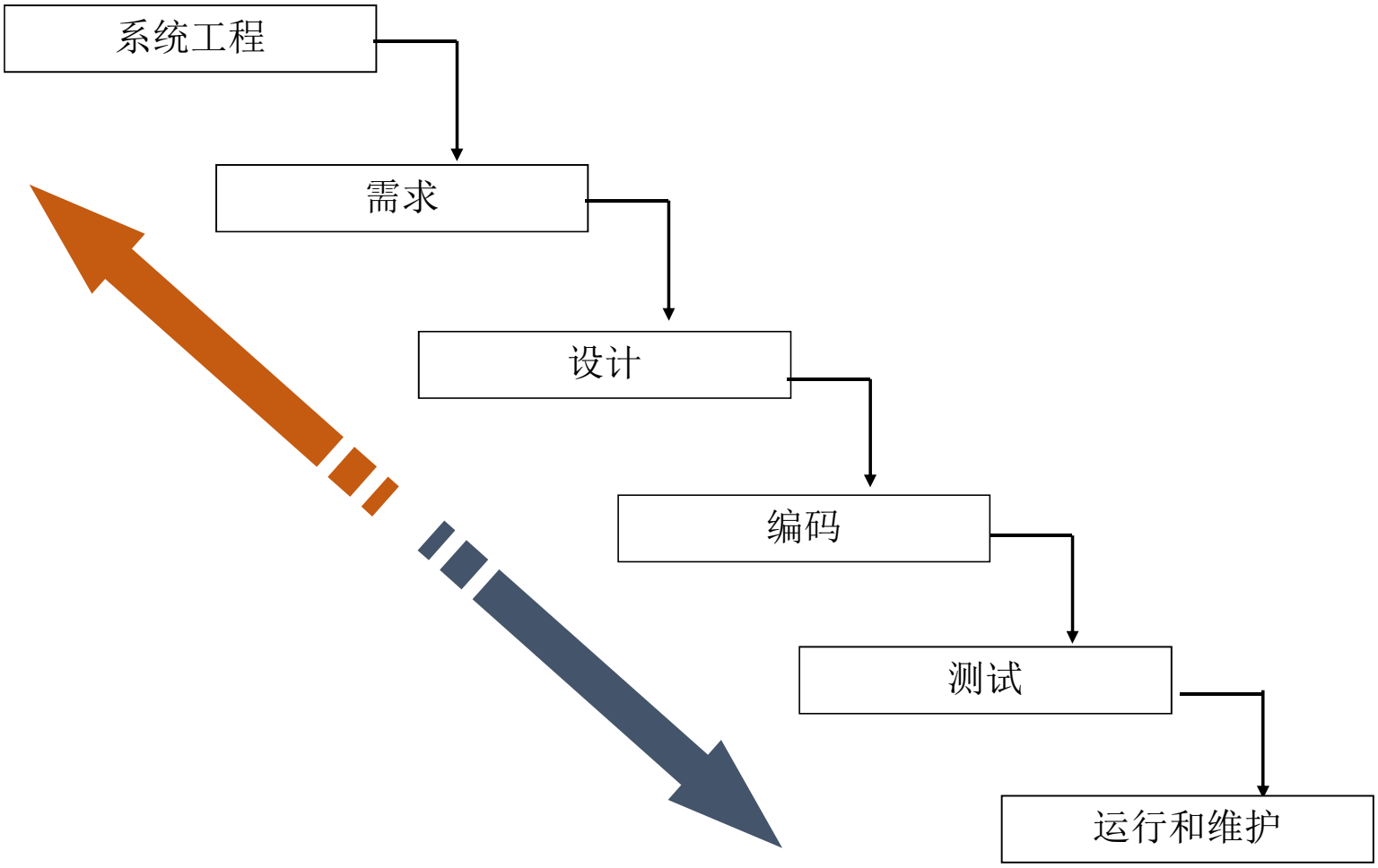
费用分配比例



维护类型

- 纠错性维护（Corrective maintenance）
  - 由于软件中的缺陷引起的修改
- 完善性维护（Perfective maintenance），
  - 根据用户在使用过程中提出的一些建设性意见而进行的维护活动
- 适应性维护（Adaptive maintenance）
  - 为适应环境的变化而修改软件的活动
- 预防性维护（Preventive maintenance）
  - 为了进一步改善软件系统的可维护性和可靠性，并为以后的改进奠定基础

软件工程的发展



大 纲

01-什么是软件

软件的作用、发展、定义和特性
软件危机和问题

02-什么是工程

03-什么是软件工程

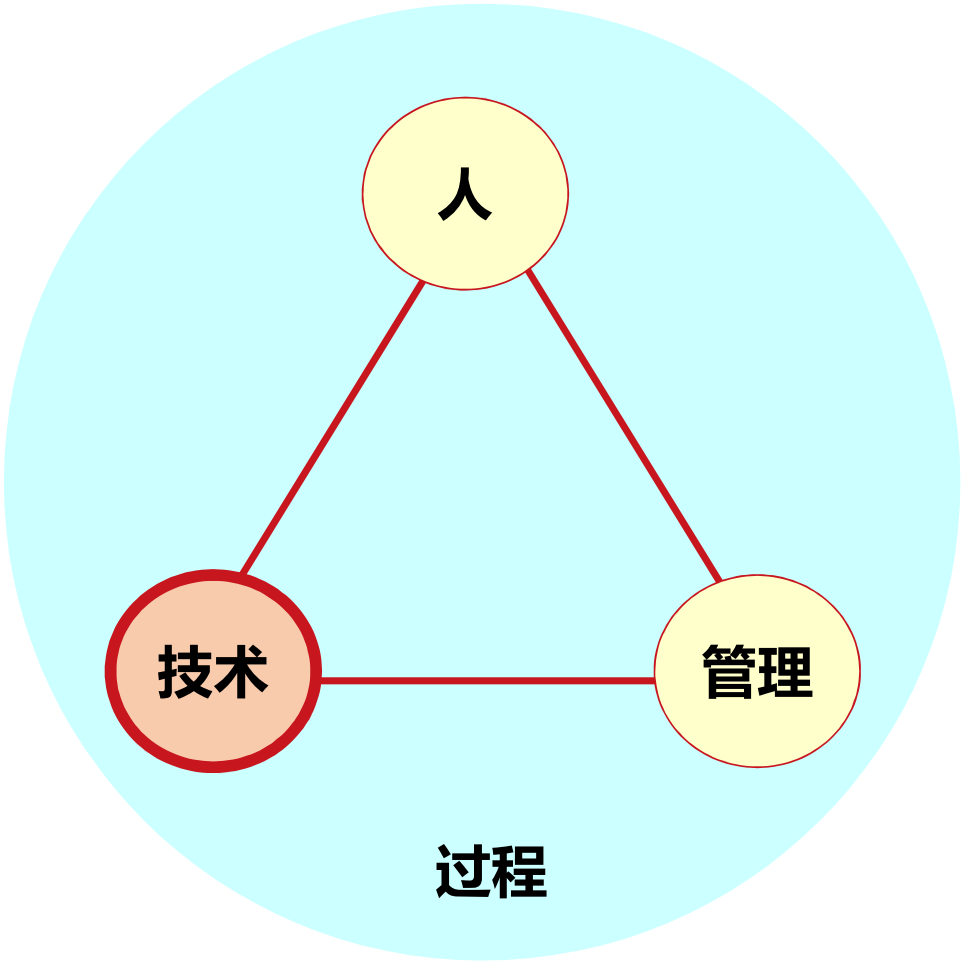
软件工程的概念和知识域
软件工程技术
软件建模及方法

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

如何控制复杂性？

软件建模是软件工程的核心技术



人

完成软件开发的主体

技术

提供了建造软件在技术上需要“如何做”的方法

管理

提供了质量管理、成本管理、时间管理、范围管理等知识和技能

过程

这是将人、技术、管理结合在一起的凝聚力

思考

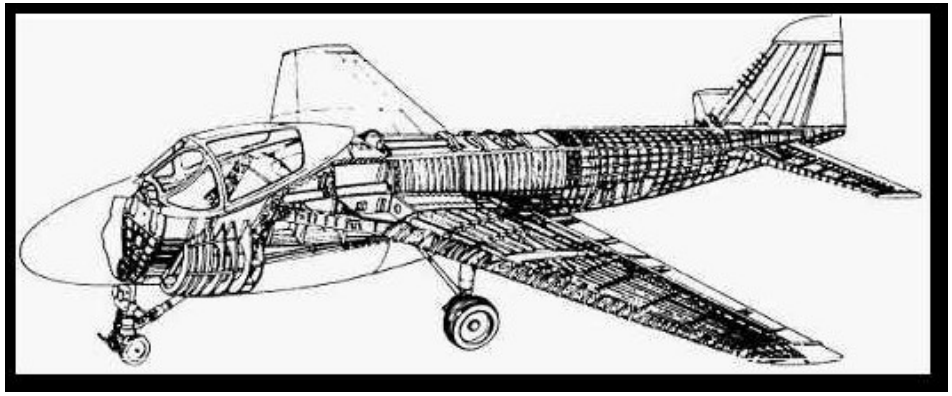
- 什么是模型?
- 为什么要建模?
- 软件模型有哪几种?

波音747

- 超过6百万个零件，仅机尾就有上百个零件
- 能承载上百个旅客
- 能不停地飞行上千公里

波音747

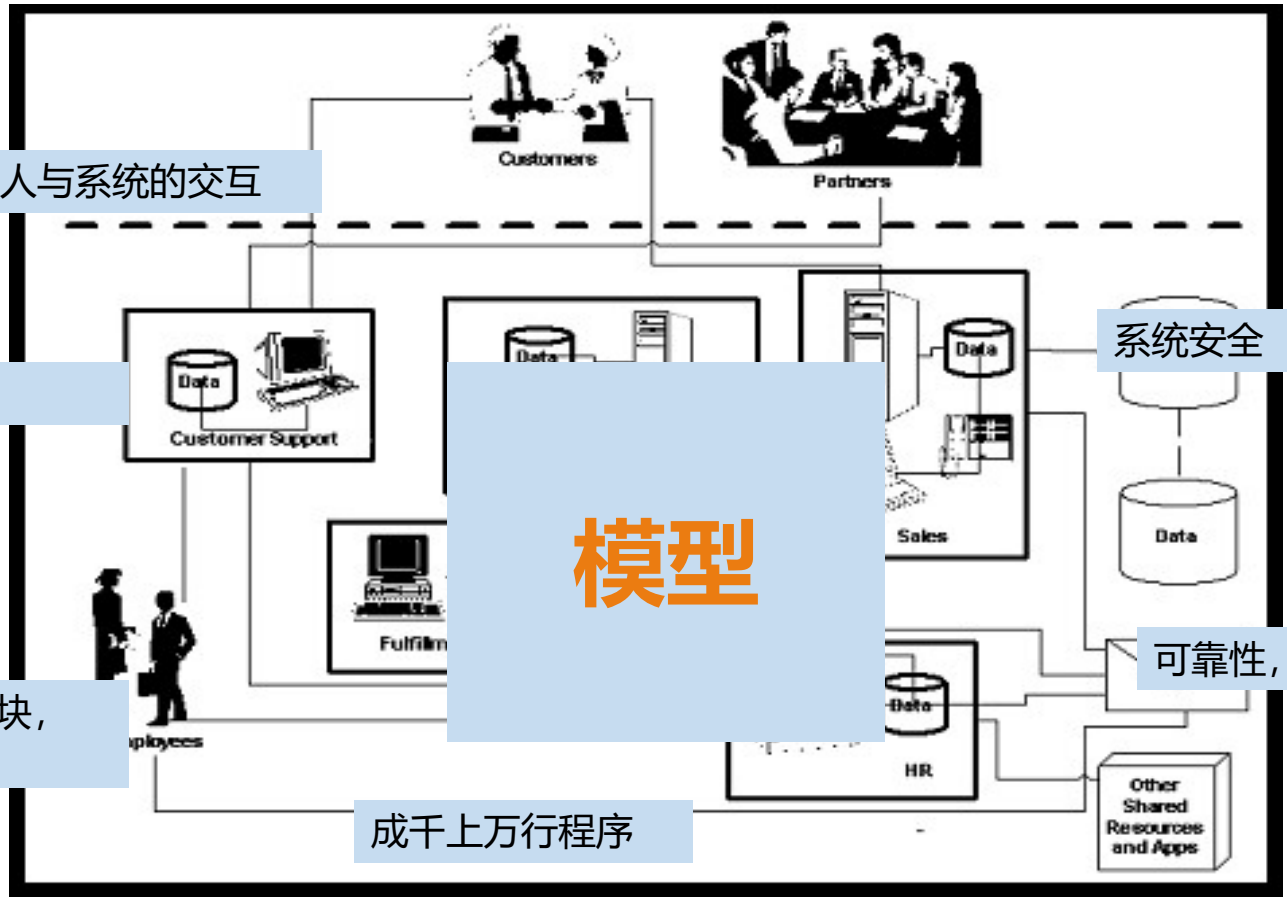
- 当波音在60年代开始生产747的时候，一共画了75，000张工程图



为什么?

因为他们正在生产一个极其复杂的系统，
而且不能出哪怕是一点点的差错

软件开发也同样复杂



人与系统的交互

交易，存储

子系统，模块，分布式

成千上万行程序

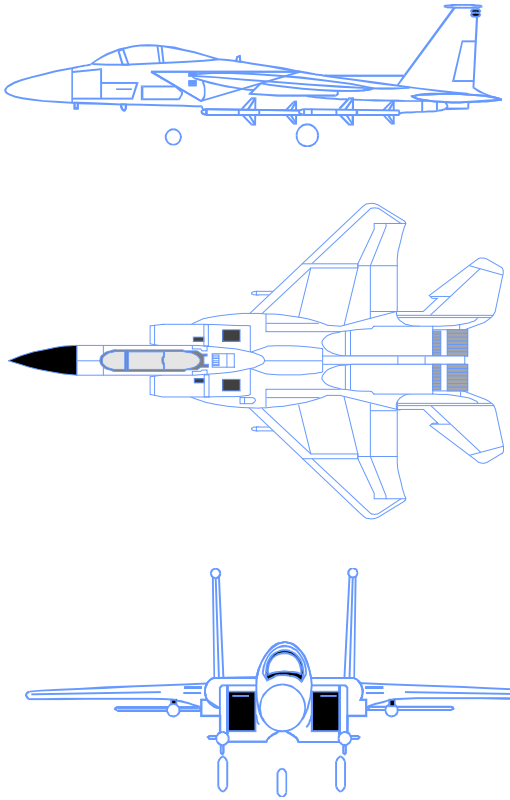
模型

系统安全

可靠性，可扩展性

什么是模型?

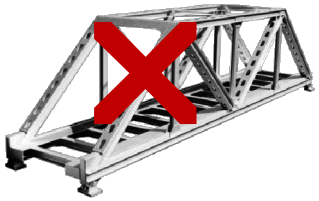
- A model is a simplification of reality.



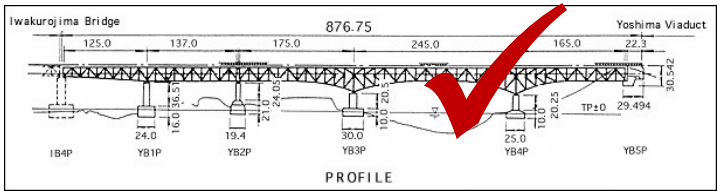
工程师们这样做...

- 他们在建造实际的物体之前...

...首先建立模型



...然后在模型的基础上进行研究和改进



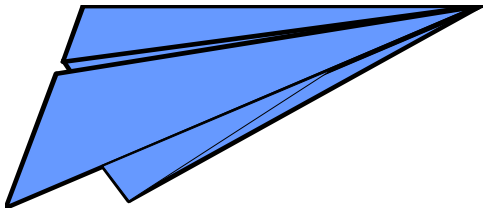
模型的功能

- 在正式启动工程项目之前发现设计中的错误和遗漏之处
  - 通过(形式化的)分析和实验，降低工程的风险
- 研究和比较不同的解决方案
- 用来和项目的所有者进行交流
  - 客户、用户、实现者、测试者、开发文档管理员，等等.
- 促进工程的实现

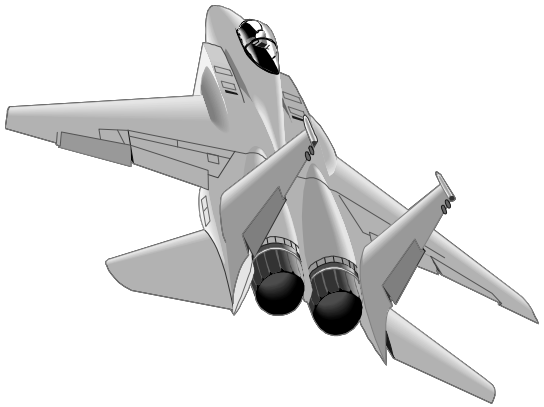
建模的重要性

Less Important

More Important



Paper Airplane



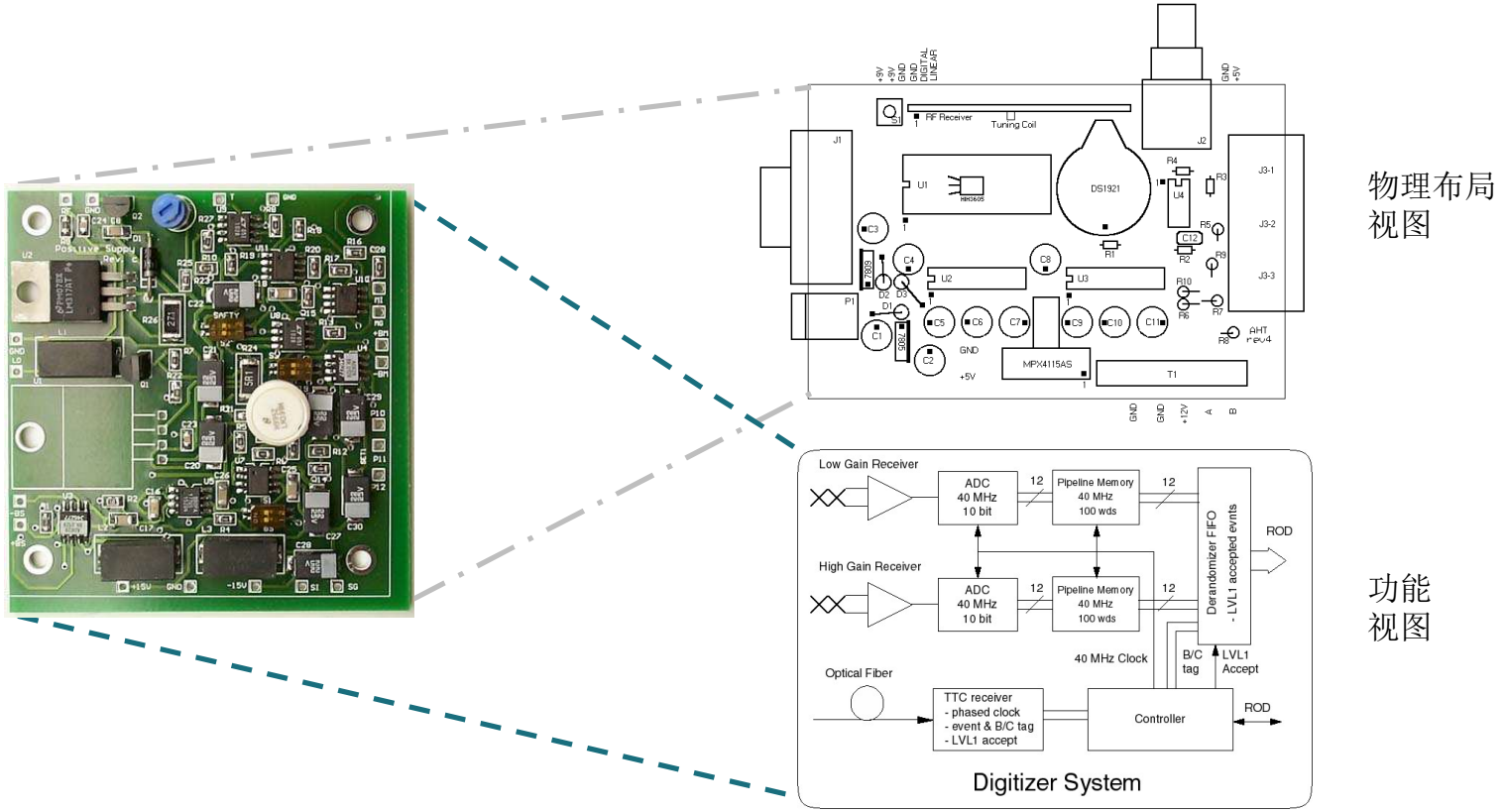
Fighter Jet

有用模型的特征

- 抽象性
  - 突出重点方面，去除无关紧要的细节
- 可理解性
  - 模型的表达方式能被模型的观察者很容易地理解
- 精确性
  - 忠实地表达被建模的系统
- 说明性
  - 能够被用来对被建模系统进行直观地分析，并得出正确的结论
- 经济性
  - 模型的建立和分析比被建模系统更廉价，更经济

作为有用的工程模型，必需具备以上所有特征!

模型的多个视图



物理布局
视图

功能
视图

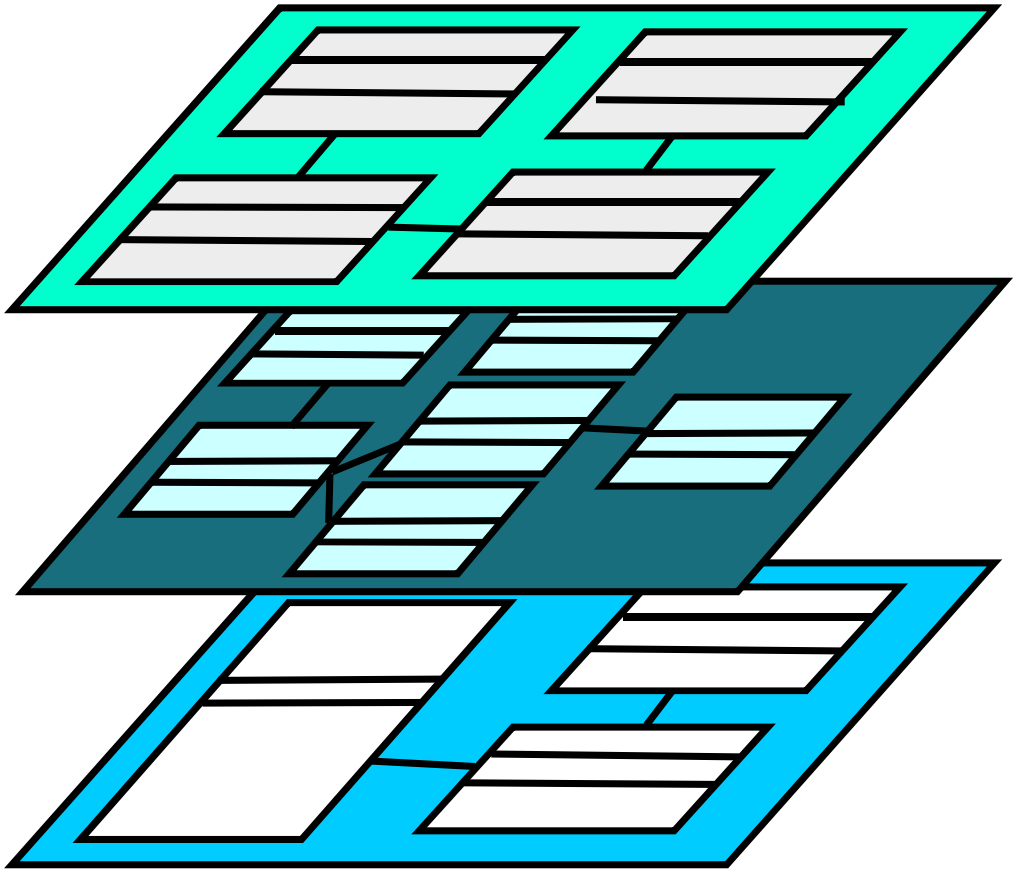
举例：UML 的软件模型视图

- 需求:
  - 用例图 Use Case diagram
- 结构:
  - 本体论: 类图 Class diagram
  - 实例: 对象图 Object diagram
- 行为
  - 状态图 Statechart diagram
  - 活动图 Activity diagram
  - 交互：顺序图和协作图　Sequence diagram & Collaboration diagram
- 实现:
  - 构件图 Component diagram
  - 部署图 Deployment diagram

这些视图相互集成

构成一个完整的模型

软件的三种模型



Computation-Independent Model (CIM)

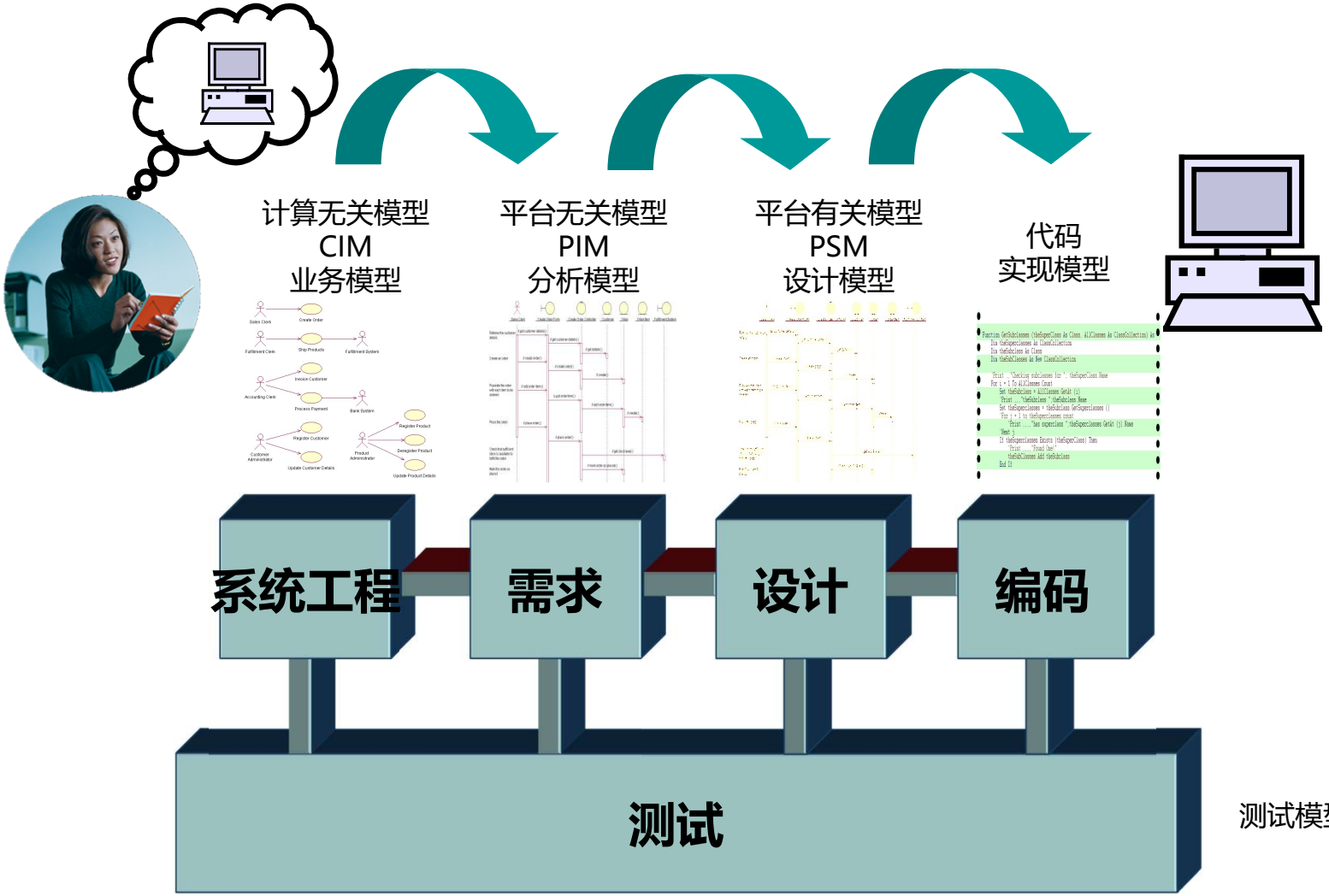
Platform-Independent Model (PIM)

Platform-Specific Model (PSM)

More Abstract

More Specific

模型间的转换


计算无关模型
CIM
业务模型
平台无关模型
PIM
分析模型
平台有关模型
PSM
设计模型
代码
实现模型
系统工程
需求
设计
编码
测试

测试模型

软件建模的方法

- 形式化方法 (Formal Method)
- 结构化方法 (Structured Method)
- 面向对象方法 (Object Oriented Method)
- 基于构件的软件开发方法 (Component Based Software Development)
- 面向服务方法 (Service Oriented Method)
- 模型驱动的开发方法 (Model-Driven Development)
- 敏捷建模方法 (Agile Modeling Method)
- ……

形式化方法

- 形式化方法是基于数学的技术开发软件，如集合论、模糊逻辑、函数。
- 形式化方法的好处：
  - 无二义性
  - 一致性
  - 正确性
  - 完整性

举例

--------AddBlock----------------

△BlockHandler

Ablocks？：　P BLOCKS

Ablocks？⊆ used

BlockQueue'=BlockQueue⌢〈Ablocks？〉

used'=used∧

　free'=free

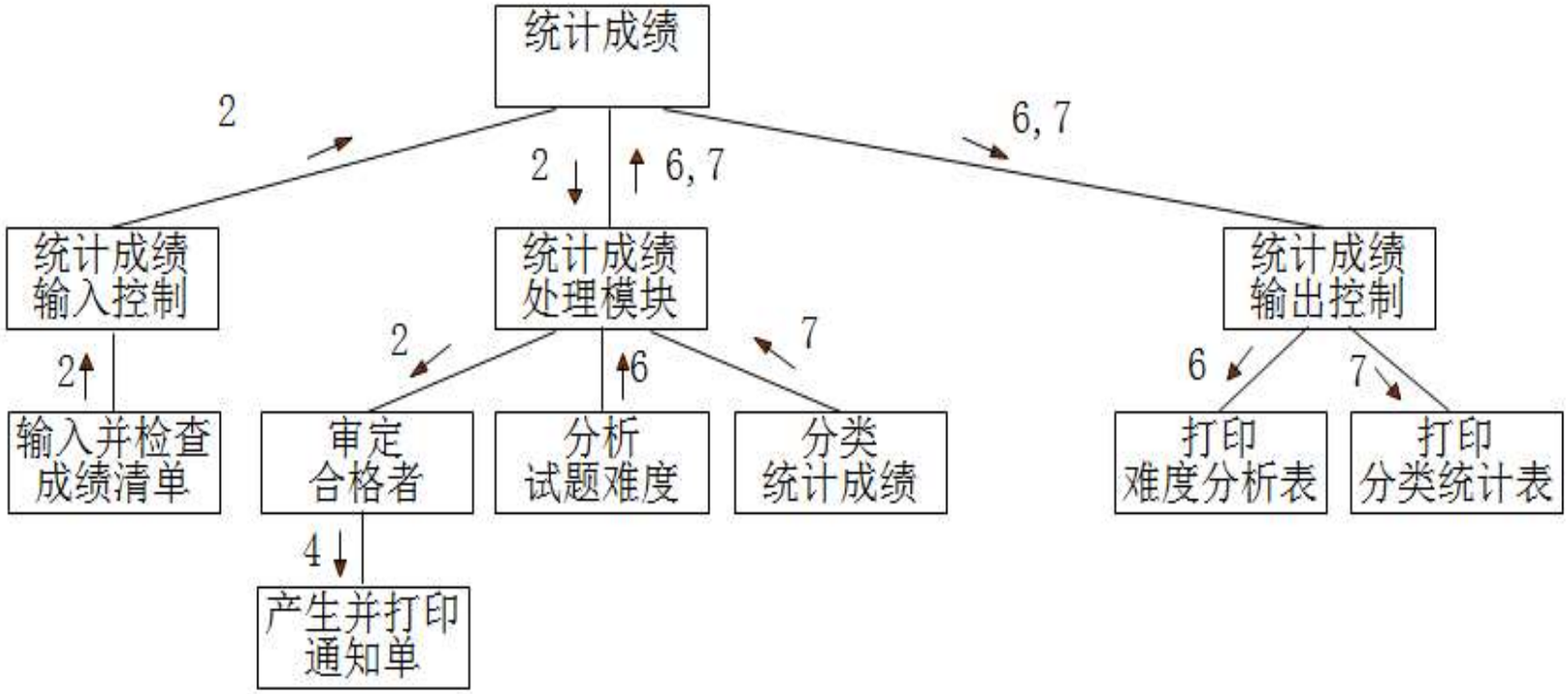
加一个块集合到队列的尾部, 采用Z语言

形式化方法的不足

- 形式化规约主要关注于功能和数据，而问题的时序、控制和行为等方面却更难于表示。此外，有些问题元素(如，人机界面)最好用图形技术来刻划。
- 使用形式化方法来建立规约比其他方法更难于学习，并且对某些软件实践者来说它代表了一种重要的“文化冲击”。
- 难以支持大的复杂系统。

尚未成为主流的开发方法，实践和应用较少

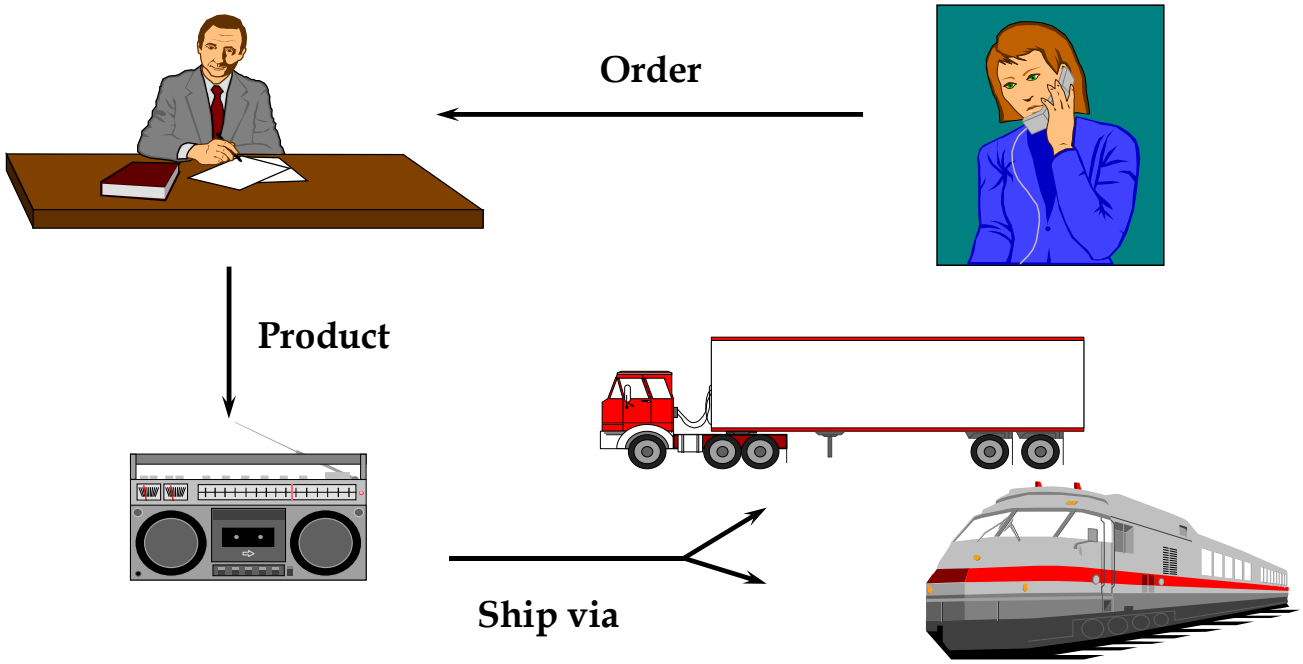
结构化设计的系统示例


统计成绩
2
2
6, 7
6, 7
统计成绩
输入控制
统计成绩
处理模块
统计成绩
输出控制
2
2
6
7
6
7
输入并检查
成绩清单
审定
合格者
分析
试题难度
分类
统计成绩
打印
难度分析表
打印
分类统计表
4
产生并打印
通知单

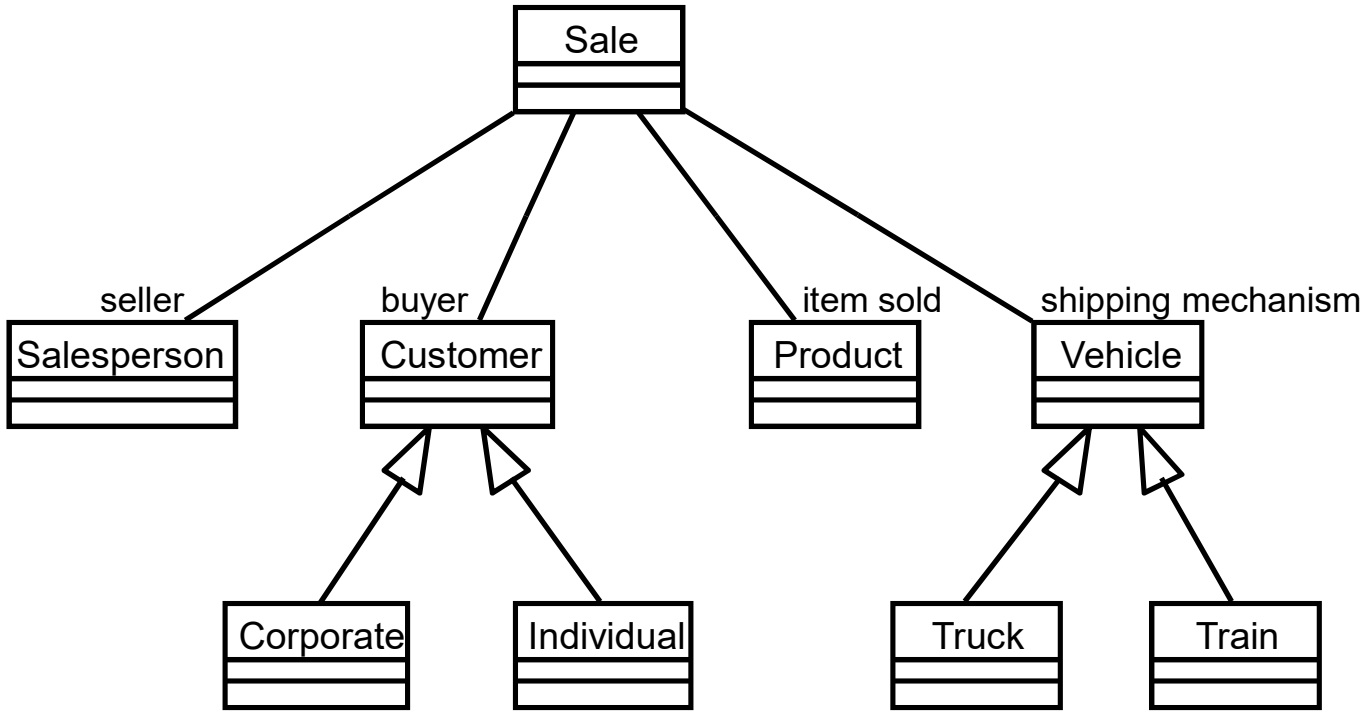
结构化方法

- 核心: 数据和处理
- 手段:自顶向下，逐步求精、模块化
- 常用建模工具:
  - 需求建模:
    - DFD(数据流图) 、 DD(数据字典)、ERD(实体关系图)、STD(状态图)
  - 设计建模:
    - 结构图（SC）
    - 流程图、 N-S图、PAD图、伪代码

面向对象方法示例：销售订单

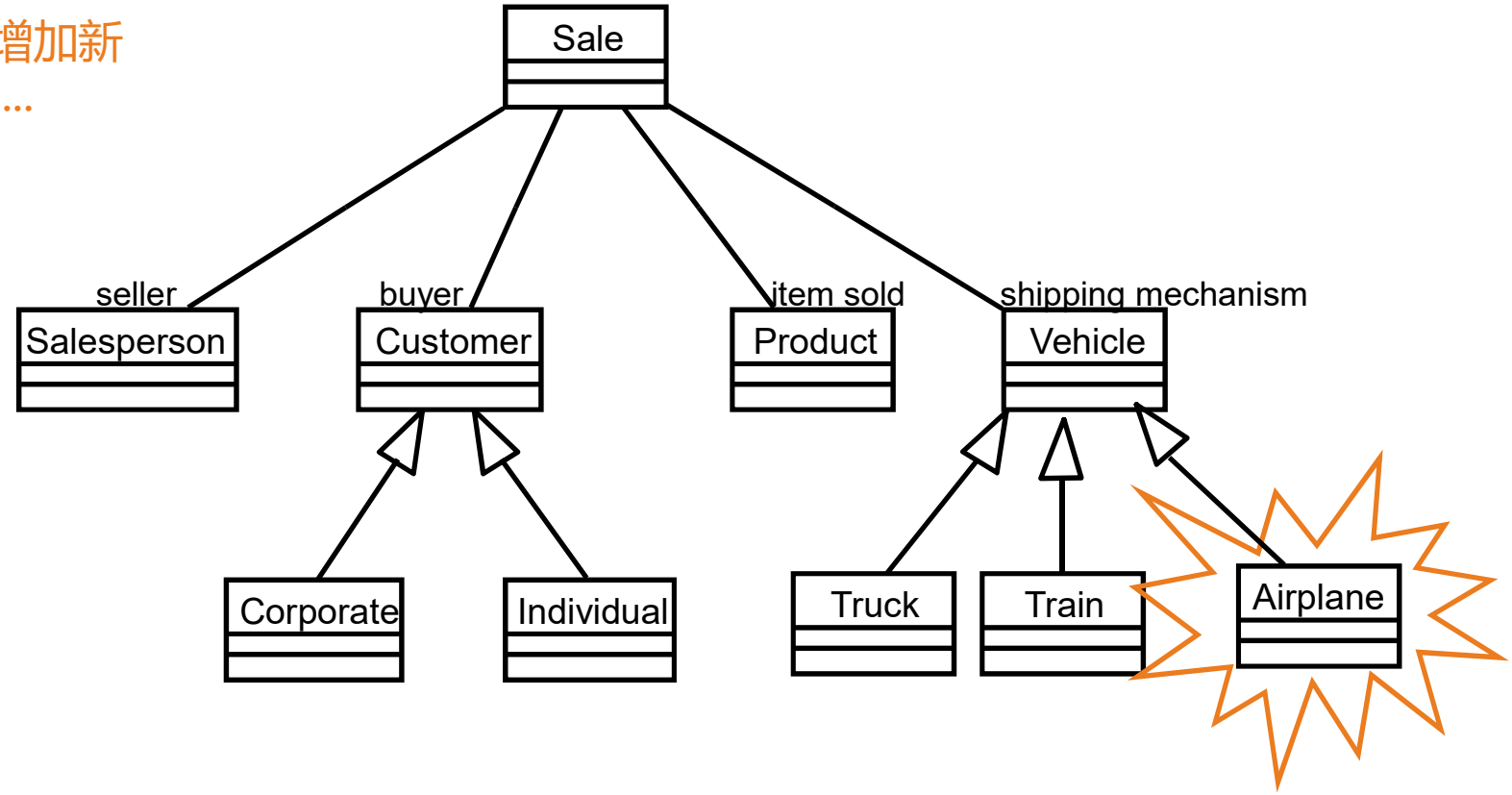


销售订单的类图


Sale
seller
buyer
item sold
shipping mechanism
Salesperson
Customer
Product
Vehicle
Corporate
Individual
Truck
Train

需求变更的影响

假定你需要增加新的运输工具 ...



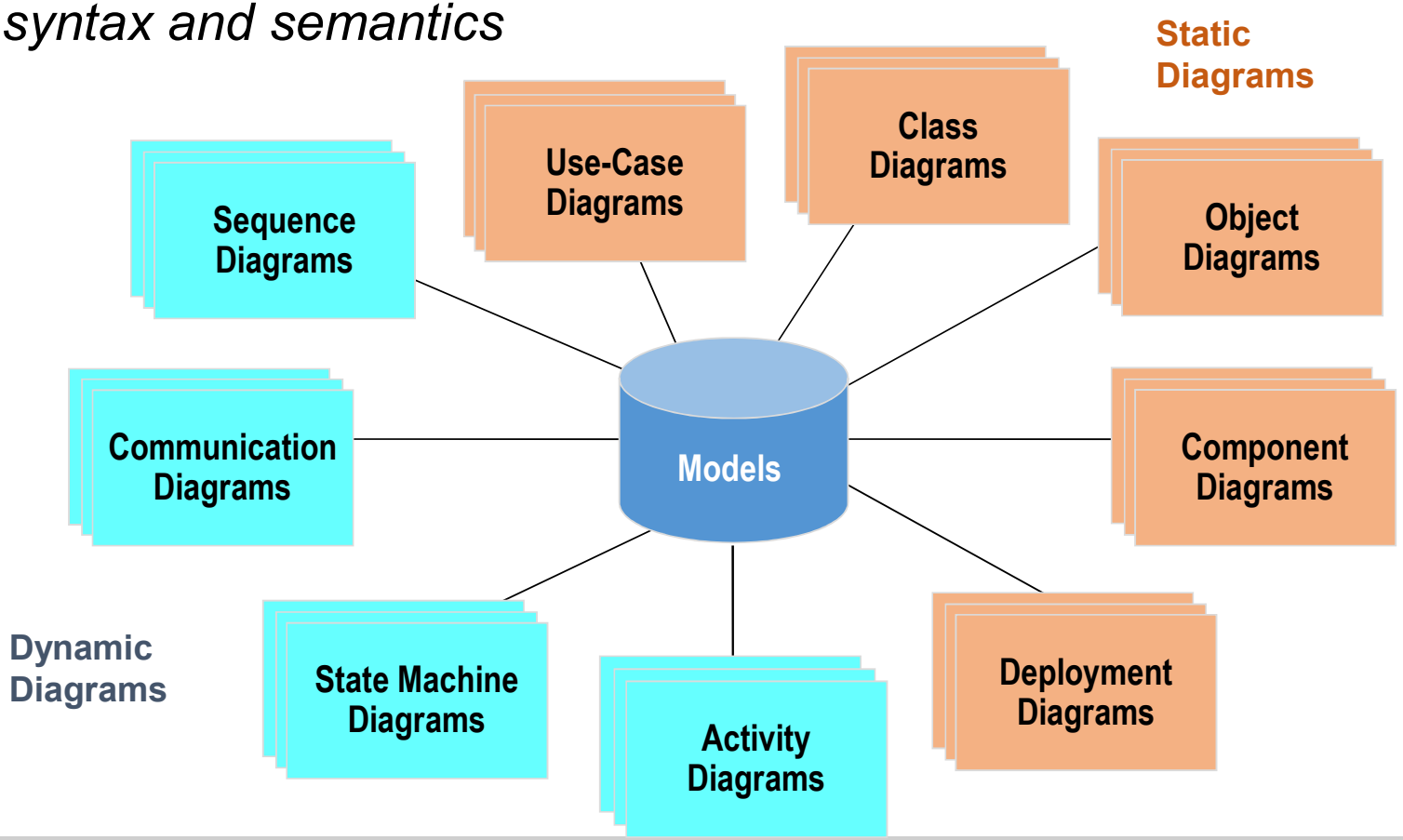
增加一个新的子类

面向对象方法

- 九十年代以来的主流开发方法
  - 符合人们对客观世界的认识规律
  - 开发的系统结构易于理解、易于维护
  - 继承机制有力支持软件复用
- 常见的面向对象方法
  - Booch method 1994
  - Coad and Yourdon method 1991
  - Rambaugh method -- OMT 1991
  - Jacobson method – OOSE 1992
  - Wirfs-Brock method 1990
  - ……
  - 国际标准统一建模语言 UML 1997

UML模型

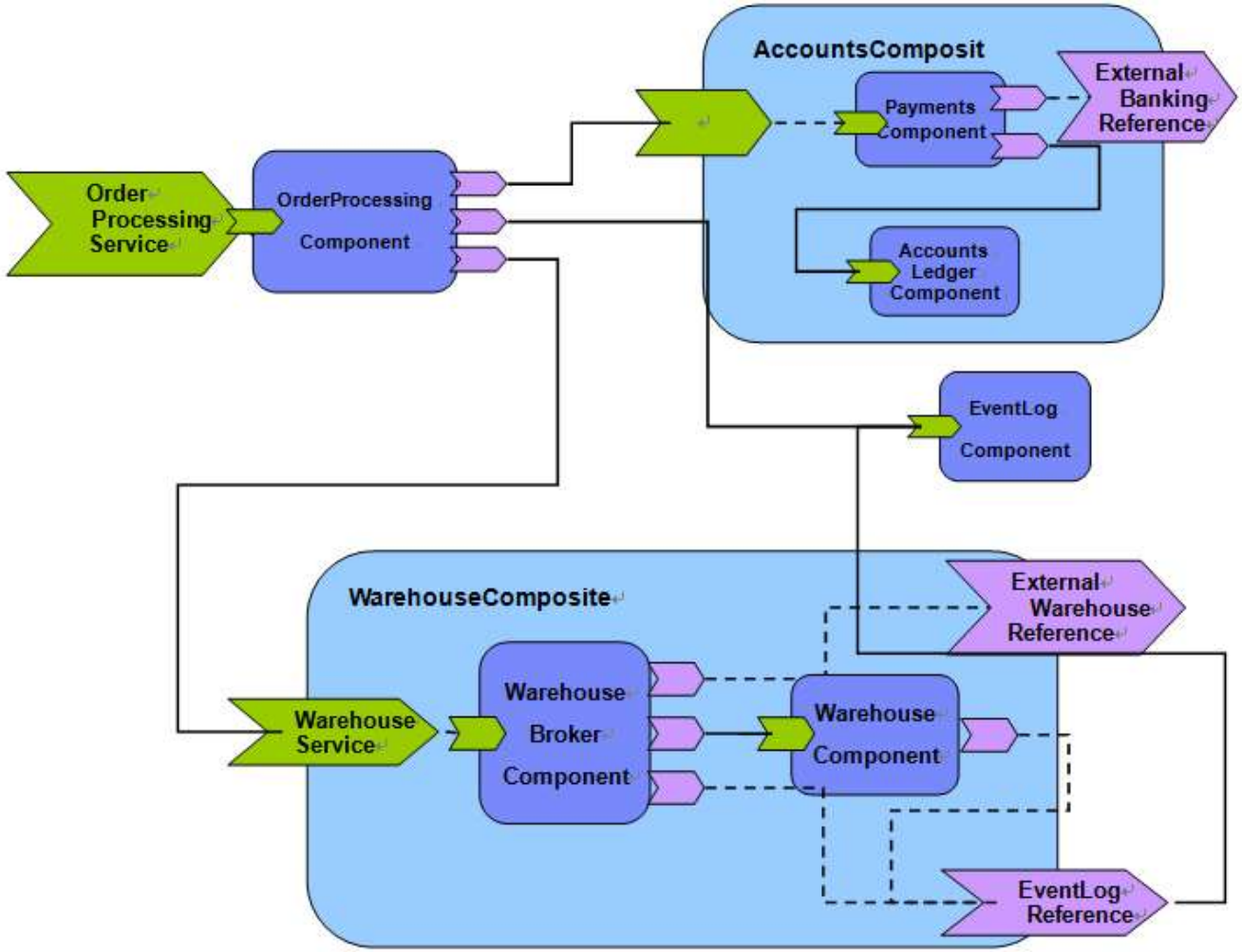
- *Multiple views*
- *Precise syntax and semantics*



UML建模工具

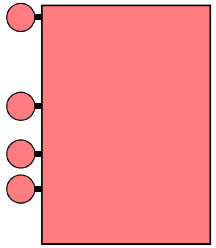
- IBM RSA 和Rational Rose
- Sybase PowerDesigner
- Sparx System EA
- Borland Together
- ArgoUML 开源
- StarUML 开源
- …

基于构件的软件系统示例

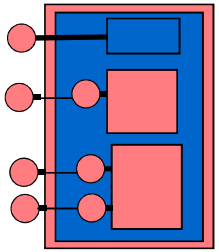


构件

- 构件是软件复用的重要手段，是核心和基础
- 构件由构件规约与构件实现两部分组成

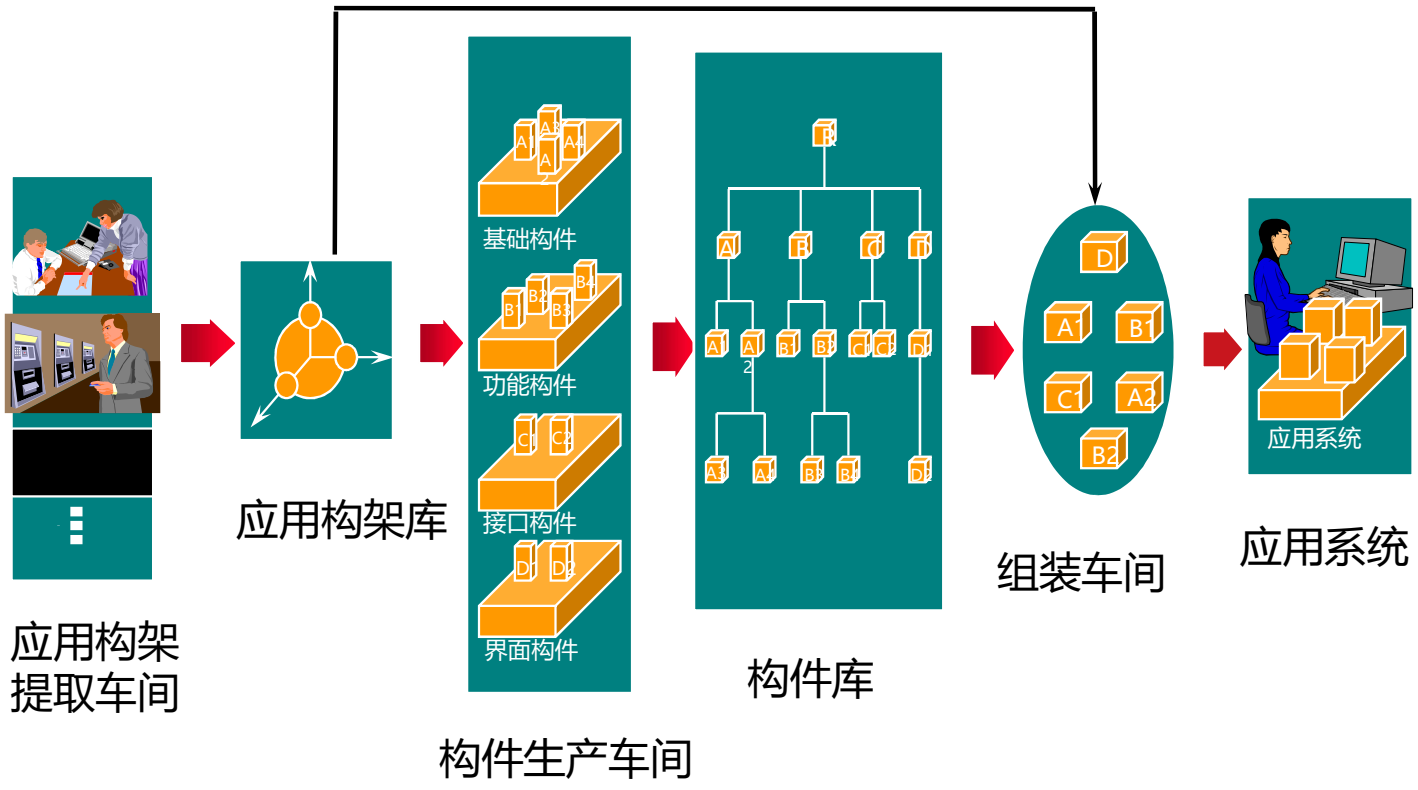


Component interface



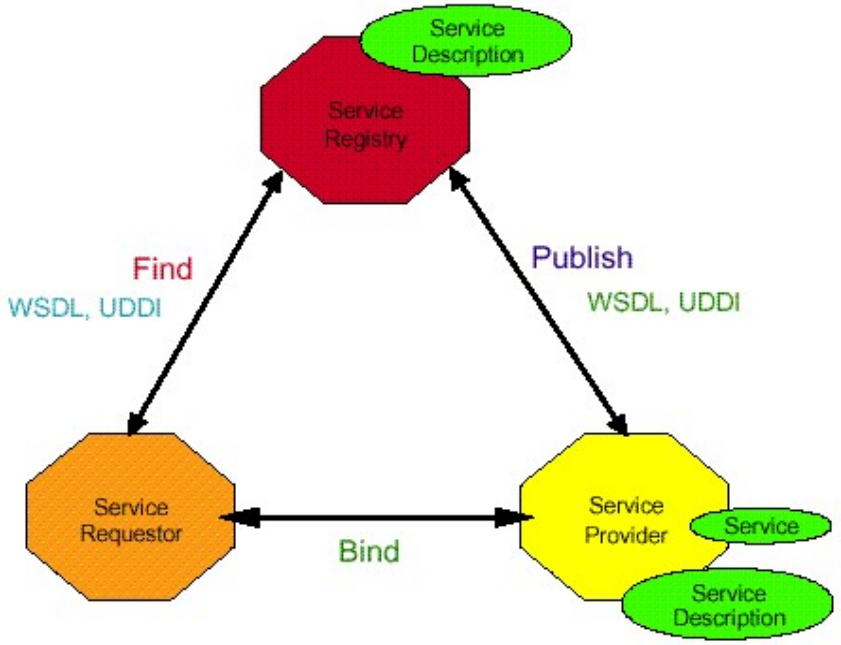
Component implementation

基于构件的开发



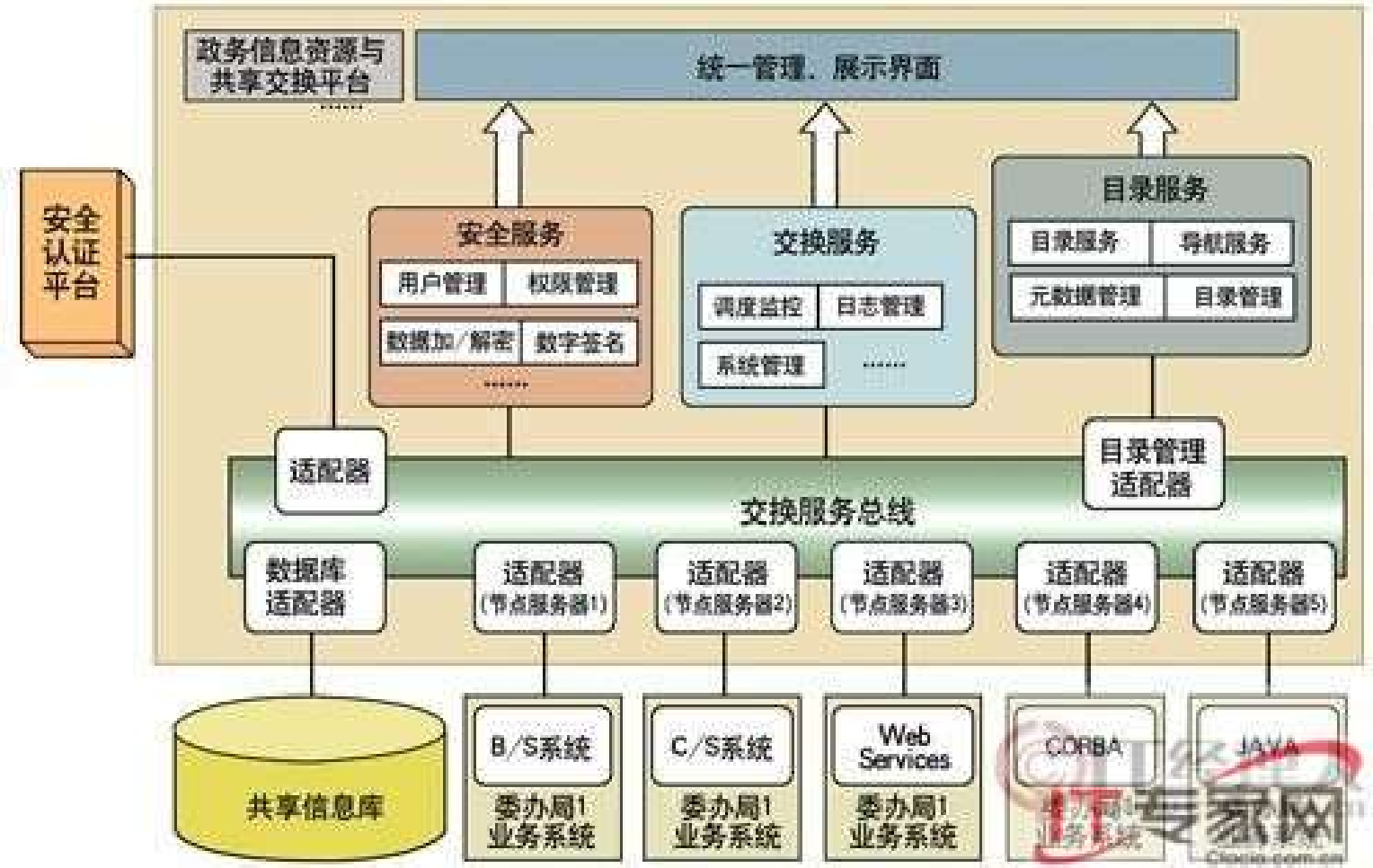
标准规范　　与　　质量保证

面向服务的方法



- 服务的抽象性（基于接口的编程）
- 服务的自治性（实现分布式应用）
- 服务间的松耦合式绑定，基于标准化消息进行通信
- 服务的自描述性（支持动态发现与延迟绑定）
- 服务的粗粒度（支持基于业务逻辑的积木式装配）

面向服务的系统示例


政务信息资源与
共享交换平台
统一管理、展示界面
安全
认证
平台
安全服务
用户管理
权限管理
数据加/解密
数字签名
交换服务
调度监控
日志管理
系统管理
目录服务
目录服务
导航服务
元数据管理
目录管理
适配器
目录管理
适配器
交换服务总线
数据库
适配器
适配器
(节点服务器1)
适配器
(节点服务器2)
适配器
(节点服务器3)
适配器
(节点服务器4)
适配器
(节点服务器5)
共享信息库
B/S系统
委办局1
业务系统
C/S系统
委办局1
业务系统
Web
Services
委办局1
业务系统
CORBA
JAVA