

Software Engineering Review



School of Software,
SYSU

Prof. Yang YU

yuy@mail.sysu.edu.cn

主要内容

1. 理解软件工程
2. 为什么学软工？
3. 软工的知识体系
4. 若干重要概念
5. 软工的演变与趋势

1. 理解软件工程——概念

- 软件工程：(1)将系统化、规范化、可度量的方法应用与软件的开发、运行和维护的过程，即将工程化应用于软件中。(2)对(1)中所述方法的研究。——IEEE[IEE93]

- 软件工程：是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过实践考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，这就是软件工程。

1. 理解软件工程——要素

- 工程化、管理、技术。 融合了多个学科的知识。
- 人：软件工程项目中最重要、也是最活跃的资源因素。组织的原则和模式、角色定义和分工、人员的配备、绩效的评定等等。 1992-1995.9 SEI P-CMM V1.0

1. 理解软件工程——软件开发过程的实质

风险管理（控制意外）

计划与进度管理（控制时间）

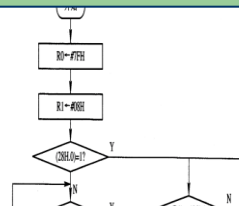
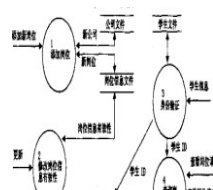
配置管理（控制变化）

质量管理（控制错误）

在规定的
时间、
费用内
做出符
合质量
要求的
软件

第一章 上下文关系图

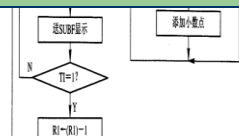
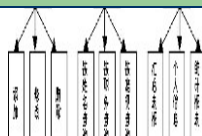
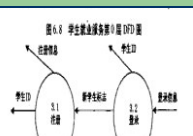
图 1-1 上下文关系图



```
graph TD
    subgraph "图 1-1 上下文关系图"
        direction TB
        subgraph "过程 1"
            direction LR
            I1[输入 1] --> P1[过程 1]
            I2[输入 2] --> P1
            P1 --> O1[输出 1]
            P1 --> O2[输出 2]
        end
        DS1[(数据仓库 1)] --> P1
        P1 --> DF1[数据流 1]
    end
```

技术过程

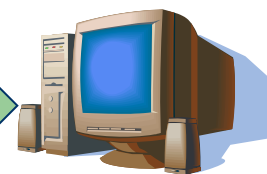
图 1-2 学生选课系统需求图 (UML)



```
graph TD
    subgraph "图 1-2 学生选课系统需求图 (UML)"
        direction TB
        subgraph "过程 1"
            direction LR
            I1[输入 1] --> P1[过程 1]
            I2[输入 2] --> P1
            P1 --> O1[输出 1]
            P1 --> O2[输出 2]
        end
        DS1[(数据仓库 1)] --> P1
        P1 --> DF1[数据流 1]
    end
```

费用管理（控制预算）

人员管理（控制人力）



2. 软件工程的重要性（为什么学？）

2.1. [历史角度]发展的必然——软件危机

- 六十年代以来，随着计算机应用需求的驱动，系统软件和应用软件有很大的发展，如操作系统，编译系统和大型应用软件等。由于软件生产的复杂性和高成本，使大型软件的生产出现了很大的困难，即出现软件危机。
- 软件工程正是为克服软件危机而提出的一种概念，并在实践中不断地探索它的原理，技术和方法。在此过程中，人们研究和借鉴了工程学的某些原理和方法，并形成了一门新的学科——软件工程学。

2.1. [历史角度]发展的必然——软件危机

●传统工程行业对软件业的影响：软件行业是一个年轻的行业，建筑业从我们老祖宗挖山洞到盖摩天大楼积累了丰富的工程管理和技术经验。不同时代，人思考解决的问题大不相同，但人的思维模式和方法却是相对稳定的，经验也是渐进积累的。新兴行业完全应该继承传统行业的所积累的智慧 and 财富。

建筑：规划设计、工程施工、工程监理

软件：顾问咨询、方案提供、软件设计、软件工厂、软件工程监理

●虽然时至今日人们并没有完全克服软件危机，但软件工程却很大的改善了软件业的情况，让人们看到了希望，也说明我们还要继续努力，更深入地认识软件开发管理的规律、深化软件工程的研究和实践。

2.2. [社会角度]团队协作的时代——个人英雄主义时代的终结

●中国的软件英雄：吴晓军（2.13）、史玉柱（巨人汉卡）、鲍岳桥（UCDOS）、周志农（自然码）、朱崇君（CCED）、求伯君（WPS）、刘旭（瑞星）、王江民（KV）、王文京（用友）、张小龙（Foxmail）、丁磊（163邮局）、洪以容（netants）…… 越来越少，影响也越来越小。这些英雄如今不是销声匿迹，就是组成了自己的团队。我们需要英雄，但是，如果只有英雄，那么就只有软件、而没有软件业了。能够支撑起民族软件业的只能是靠工程化思想武装起来的团队、还要有几只超级规模的团队！

●在个人英雄主义时代，对一般软件从业者来讲，会两句DBASE，能写个工资程序、打打报表就能生存，那个时代已经一去不复返了。

2.2.[社会角度]团队协作的时代——个人英雄主义时代的终结

●我国的软件开发水平，单从编码来看，我们拥有众多的高水平程序员，绝不逊色于任何国家，但软件开发管理水平与发达国家有很大差距，这也就是我们不能开发出像Office、操作系统之类软件的原因。我们从单个程序员来说，能够编写出一流的代码，漂亮的界面。但是却去不能协调众多程序员去开发一个项目（工程）软件。在这一方面说我们中国程序员一个人是一条龙，十个人是一条虫并不为过。

●例：国内某机场。印度人写的一个模块200多行，我们程序员只用不到50行，但人家的模块同样正确、而且坚固、文档清晰易懂。技术技巧固然重要，但**软件开发中最大的问题已不是技术问题，而是管理问题！**

2.3. [个人角度] 专业人员必备的个人素质

- 1) **基础理论**：训练分析解决问题的能力，深层研究的工具。数学分析训练严密的逻辑思维。普通物理和西方哲学史等教会你很多思维方式，比如：假说——大胆假设、认真求证。
- 2) **专业理论**：数据库理论、网络基础、编译原理等。
- 3) **专业技术**：程序设计语言等。
- 4) **工程技术**：软件工程，本、研教育中唯一涉及工程规范与管理的课程，它象粘合剂，技术经过它的粘合才能形成软件产品！
- 5) **外语**。多门语言多条路。
- 6) **表达与交流能力**：团队内部、与客户通讯的基本素质
- 7) **法律常识与职业道德**：知识产权保护、行业特点的职业道德

2.3. [个人角度] 专业人员必备的个人素质

- 程序员是一个吃青春饭的行当，也是英雄出少年的行当，就象运动员一样。一个程序员，10年以后很少再只写代码。软件工程是你从程序员到高级程序员、系统分析员，或项目经理、部门经理、CTO一步步迈进必备的知识，积累起来的经验也是你超越年轻人坐在更高职位的资本。

- 软件行业分工越来越细、一个团队内也有各种角色。软件工程的知识可以使你结合自己的特点准确定位自己的发展方向。

3. 软工的知识体系——三个层次

围绕质量目标的三个层次：

- 过程：做什么 模型、流程、标准——对应管理要素。 关键过程域KPA。定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变更所需要的管理、里程碑、项目管理、人员组织等。

- 方法：如何做 往往采用某种特殊的语言或图形表达方法及一套质量保证标准——对应技术要素。项目计划与估算、需求分析、总体设计、详细设计、编码、测试及维护的具体做法和基本原则。

- 工具：对方法的支撑，自动或半自动的软件开发支撑环境。CASE。

3. 软工的知识体系——内容索引

基础：软件生存周期和模型

标准：ISO9000-3质量管理和保证标准 ISO/IEC 12207:1995信息技术——
软件生存周期过程 CMM软件能力成熟度模型

生命周期、 KPA/ 技术方法 过程	可行性研究	需求分析	系统设计	编码	测试	维护
技术过程	做什么?如何做? 工具?	???	???	???	???	???
质量管理	???	???	???	???	???	???
项目计划	???	???	???	???	???	???
项目跟踪	???	???	???	???	???	???
人员管理	???	???	???	???	???	???
配置管理	???	???	???	???	???	???
风险管理	???	???	???	???	???	???
.....	???	???	???	???	???	???

小调查:

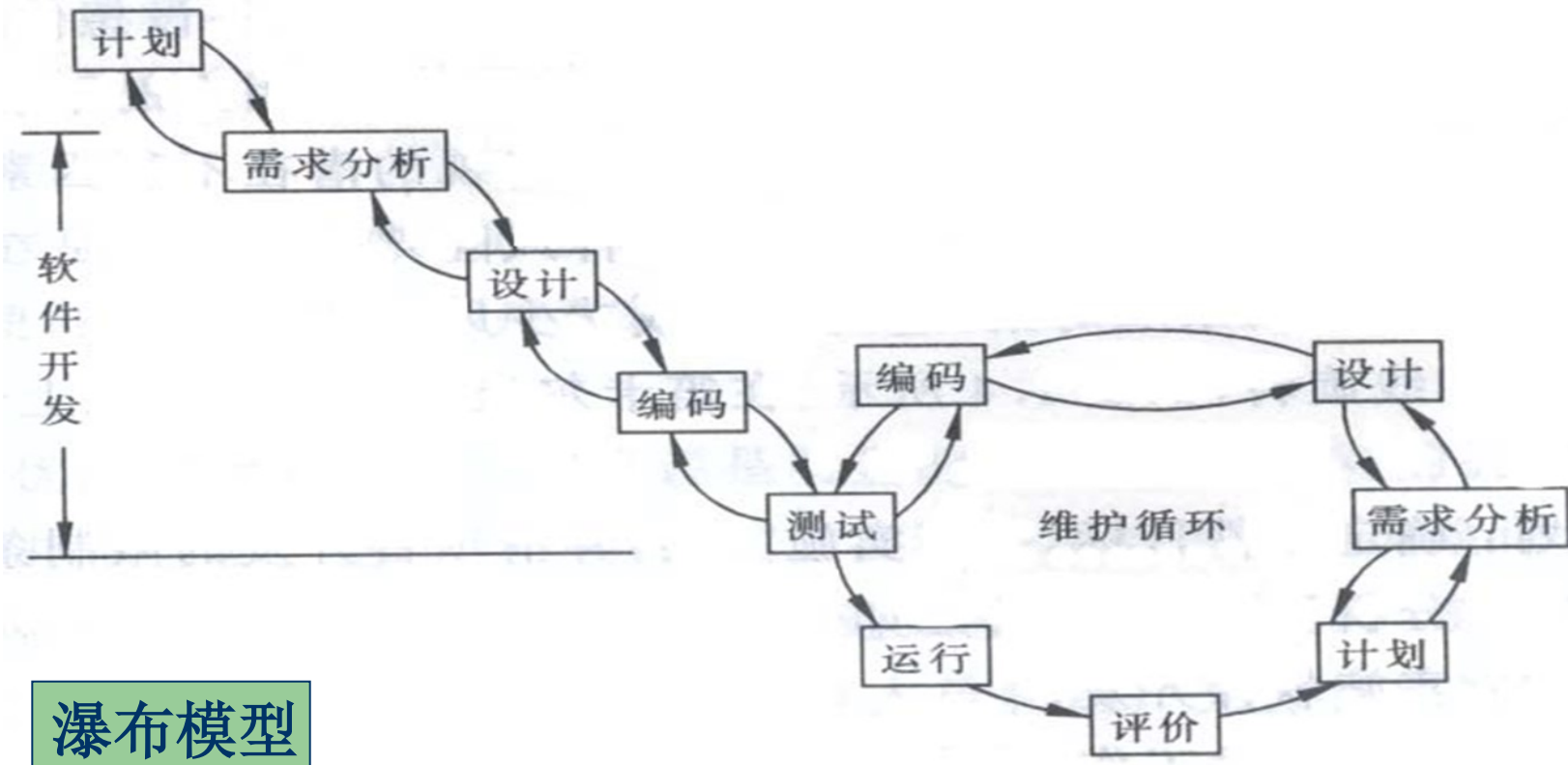
- 假设你和5位同学组队做一**软件产品**，你认为哪些是关键问题（可以选1-3个）：
 1. 项目前期需求模糊
 2. 项目开发阶段，需求不断变化并难以控制
 3. 缺乏有合适技能的人员
 4. 软件模块关系混乱，难以修改维护
 5. 别人的代码难读，不如自己重写
 6. 算法设计问题常影响进度
 7. 编程水平差影响进度
 8. 非功能性需求（实时、易用、高性能）实现超预期
 9. 缺乏同类软件或技术开发经验
 10. 沟通问题，包括难以说服某些人或误解您的设计意图
 11. 到项目提交时，才得到用户的反馈
 12. 推广使用难
 13. 其他（.....）

4. 主要概念——软件生存周期

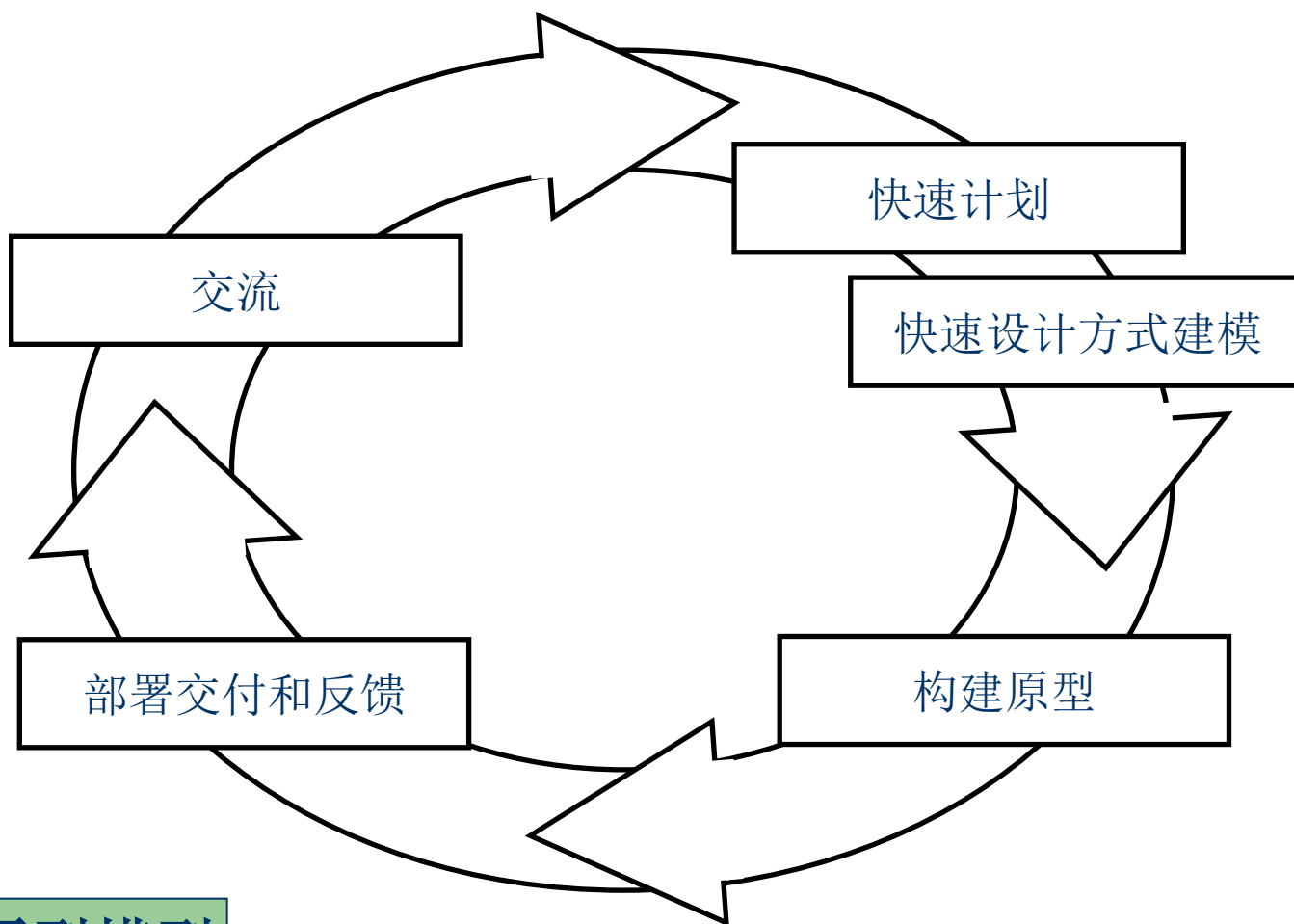
- 从时间角度，把整个周期划分为若干个阶段
- 划分的原则：各阶段的任务彼此间尽可能相对独立，同一个阶段各项任务的性质尽可能相同，从而降低每个阶段任务的复杂性，简化不同阶段之间的联系，有利于软件开发过程的组织管理。受软件规模、性质、种类、开发方法等因素的影响。
- 典型划分GB8567（4个时期7个阶段）：
 - 1) 软件分析时期：问题定义、可行性研究、需求分析
 - 2) 软件设计时期：总体设计、详细设计
 - 3) 编码与测试时期：编码、测试
 - 4) 运行与维护时期

4. 主要概念——软件开发模型

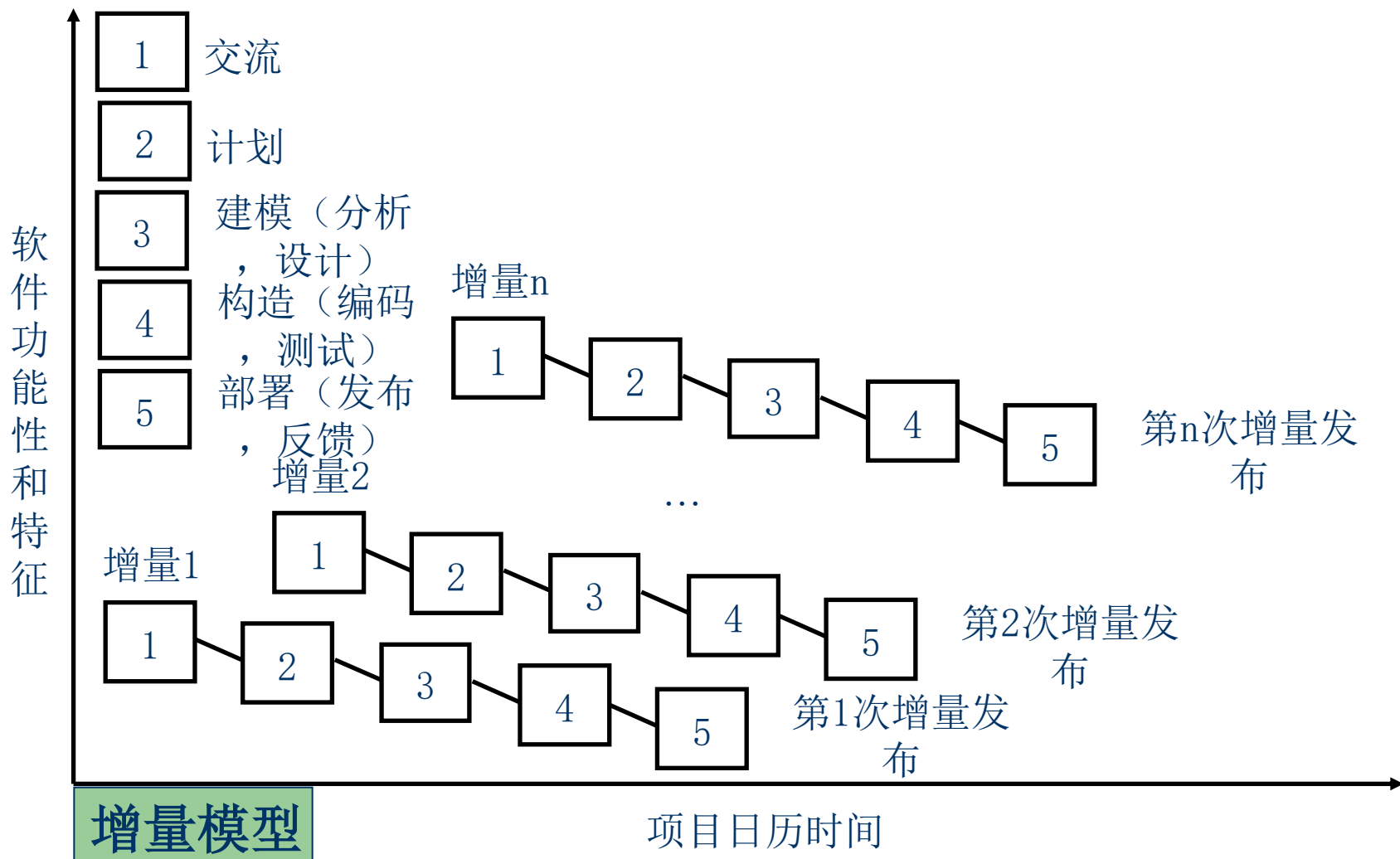
- 软件开发模型描述了软件开发过程中各阶段或活动之间的关系。它确立了软件开发和演绎中各阶段的次序限制及活动的准则。



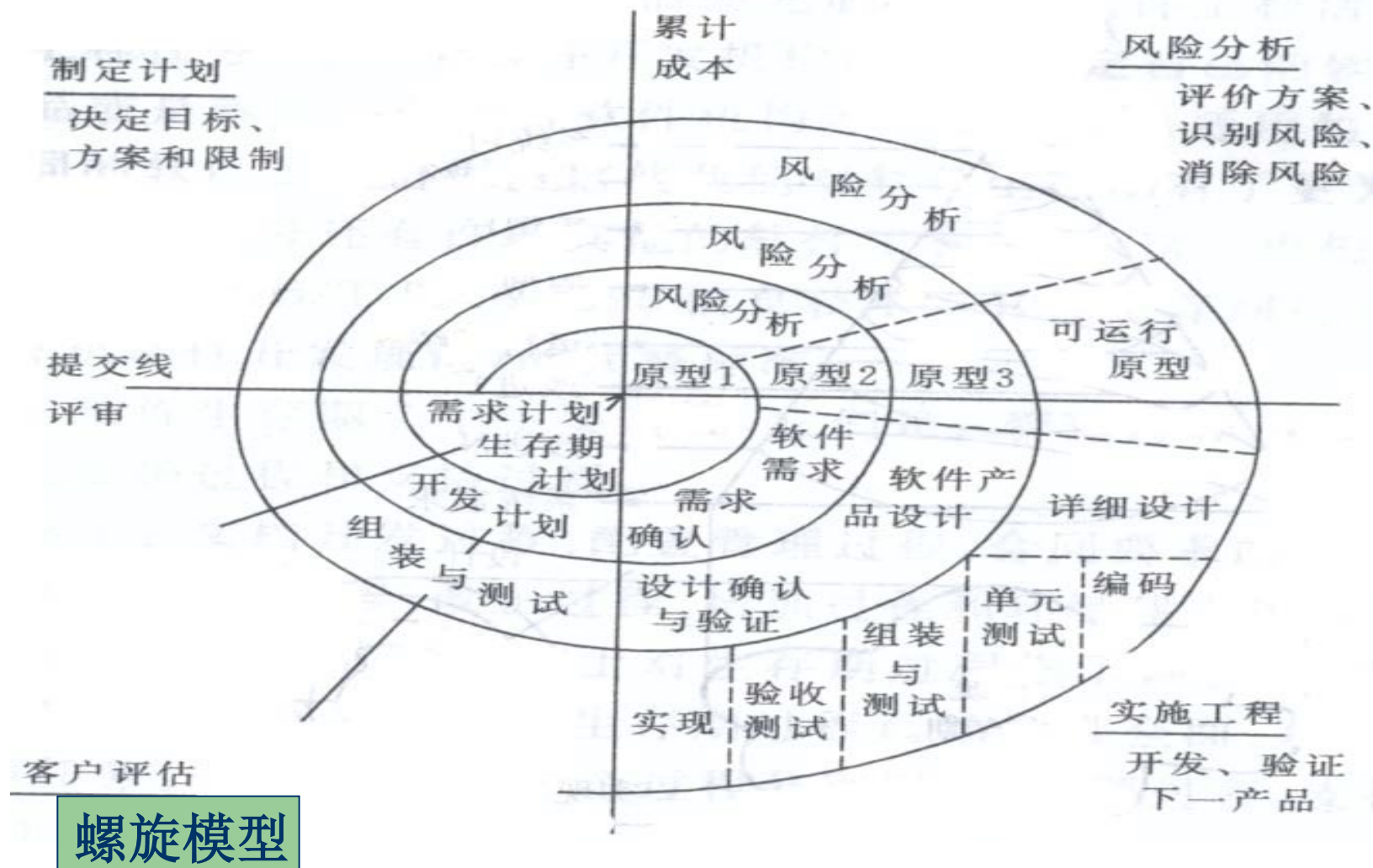
4. 主要概念——软件开发模型



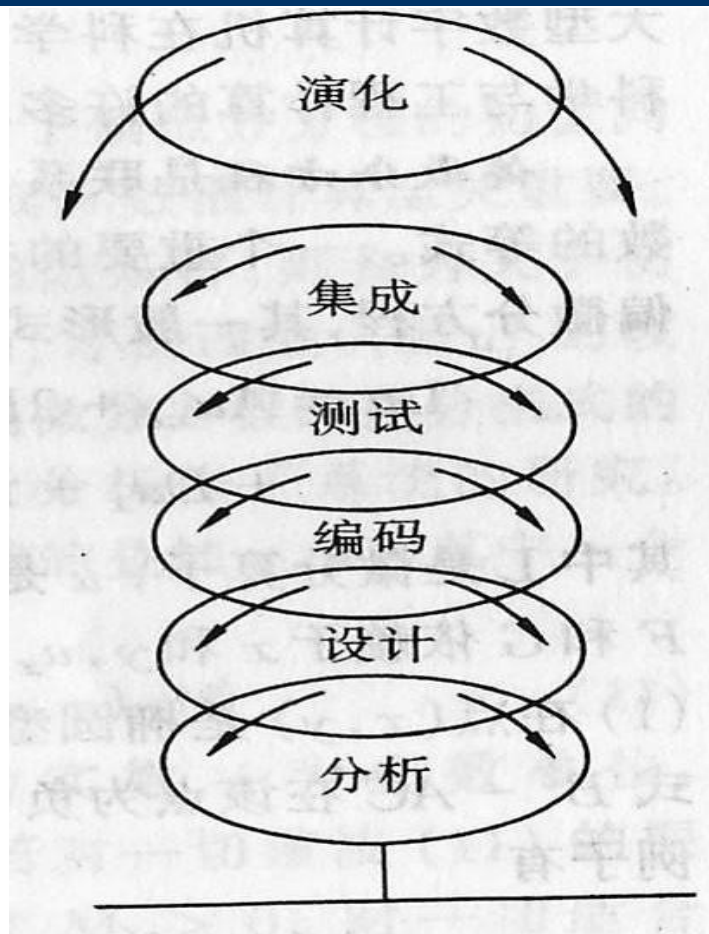
4. 主要概念——软件开发模型



4. 主要概念——软件开发模型



4. 主要概念——软件开发模型

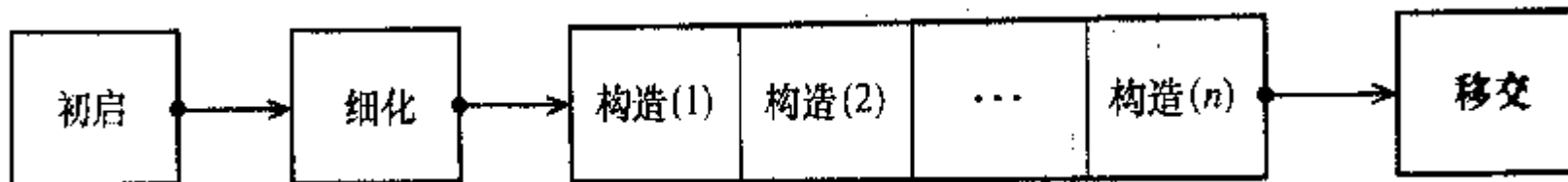


喷泉模型

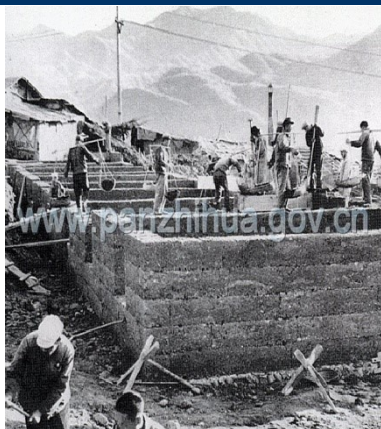
4. 主要概念——软件开发模型

- RUP四个阶段:

- 初启：确定目标、范围
- 细化：初步需求分析、初步高层设计、部分详细设计、部分原型构造
 - ❖ 用到的UML语言机制：用例及用例图、类图、活动图、包图、交互图等
 - ❖ 细化结束的条件：主要需求通过用例及用例图描述；重要风险已标示；能够精确估算实现每一用例的时间
- 构造：通过迭代完成对所有用例的软件实现。
 - ❖ 迭代计划及其原则：业务价值大、风险高的用例优先
 - ❖ 迭代过程：针对用例的分析、设计、编码、测试、集成
 - ❖ 用到的UML语言机制：.....
- 移交



4. 演变与趋势——道法自然



60年代干打垒
面向语句



70年代混合结构
面向功能模块



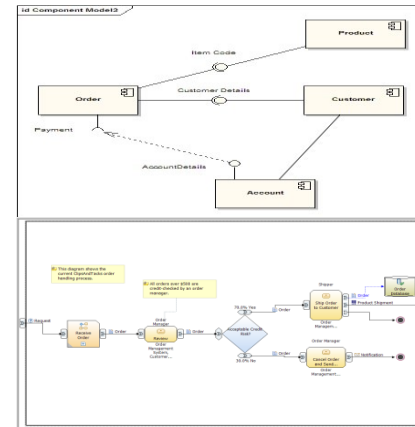
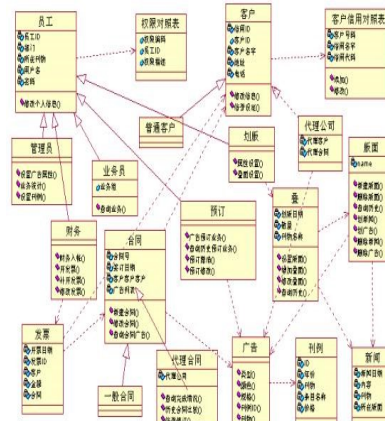
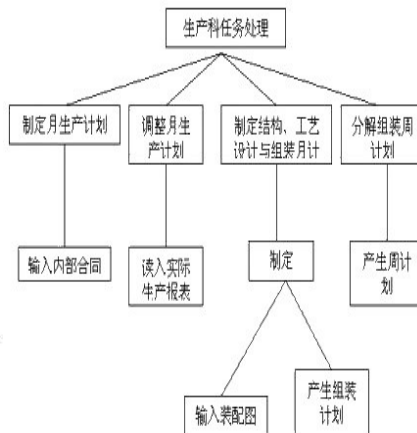
80年代框架结构
面向对象



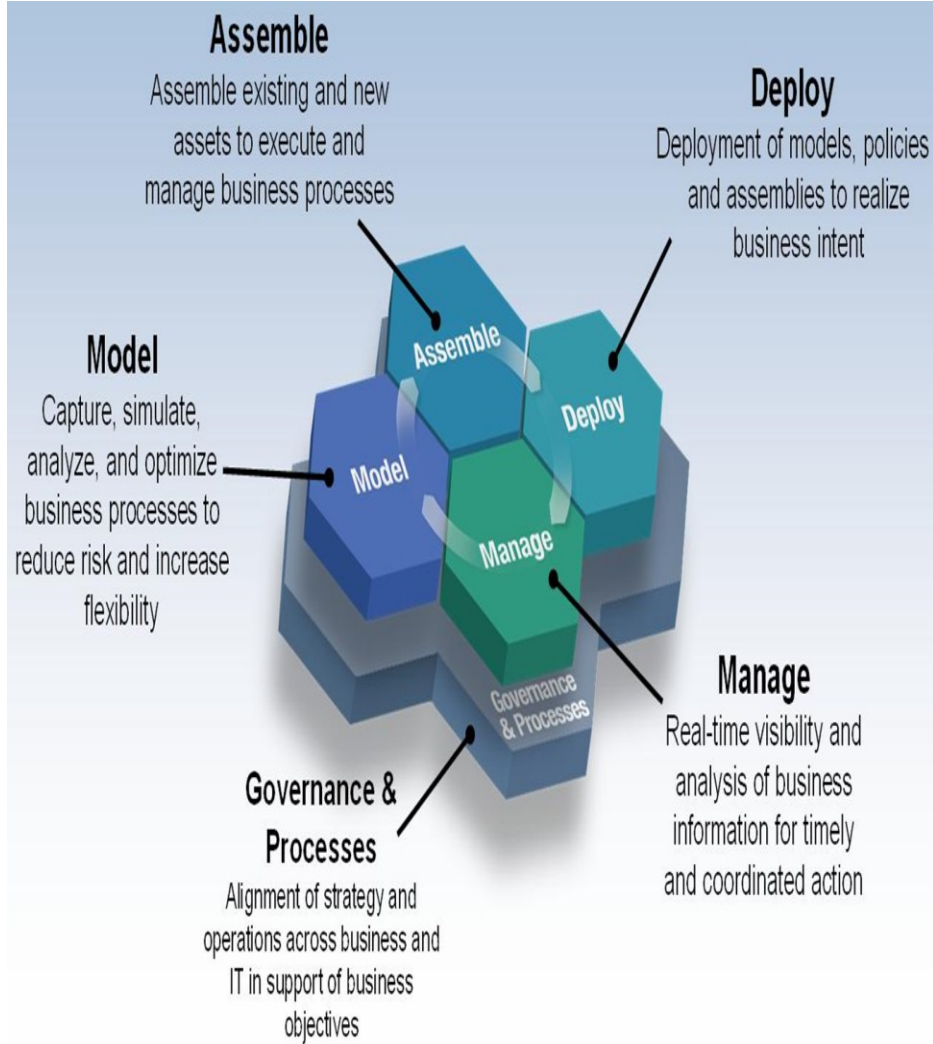
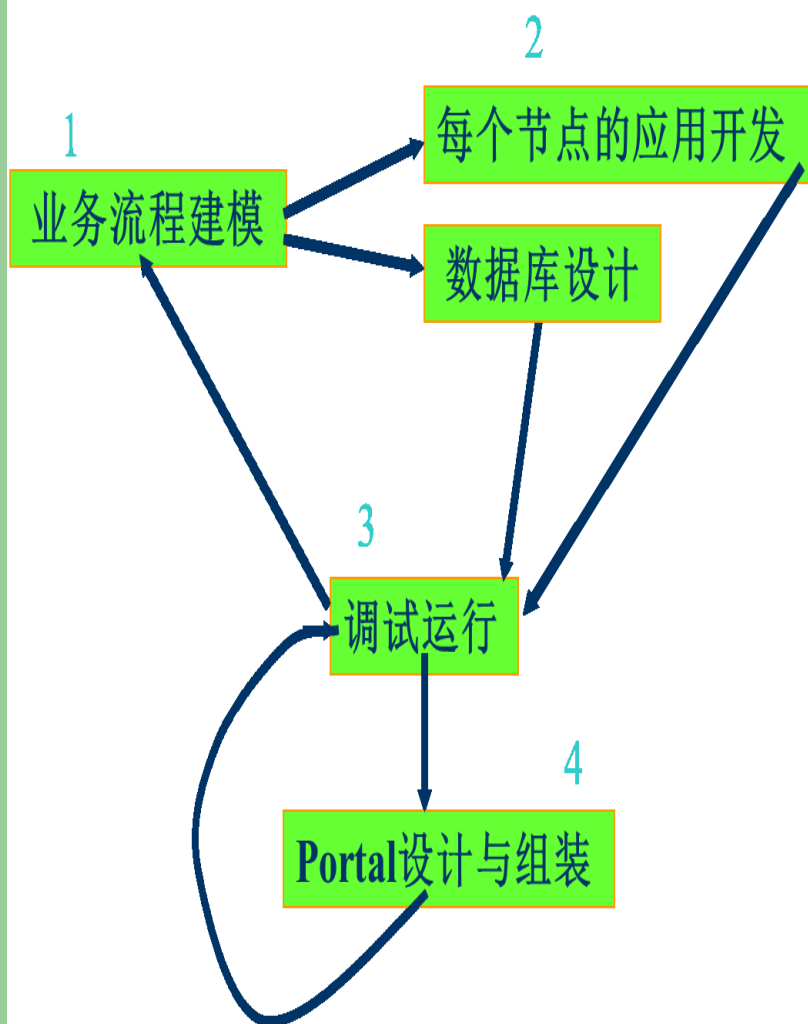
90年代钢结构
面向组件、业务

```

ORAL, 3BH
CALLDISP
MOVAL, BL
ANDAL, 0FH
ORAL, 3BH
CALLDISP
DECCX
JCXZDOME
MOVAL, ' ',
CALLDISP
JMPNEXT
DOME, NOVAH, 4CH;终止当前进程,返回调用进程
INT21H
DISP, NOVAH, 14;写字符到当前光标位置,光标前进一步
INT10H
RET
CSEGENDS
ENDSTART
    
```



4. 演变与趋势——新方法的萌芽



课程要点

- IEEE 93, 软件工程的定义
- 围绕软件质量，软件工程知识体系的三个层次：
过程、方法、工具
- 过程模型与软件生命周期
 - 瀑布模型
 - 迭代开发