

软件工程导论

第1章

软件工程学概述



江南大学-软工导论课程——软件工程导论，江南大学研究生团队录制，严谨复制！详情加q191683746

重难点

1、软件危机

定义，什么是软件，典型表现[7]，产生软件危机的原因，消除软件危机的途径。

2、软件工程

定义，本质特性[7]，软件工程的基本原理[7]，软件工程方法学（传统方法学，面向对象方法学）及其三者定义、两者要点、优缺点。

3、软件生命周期[8]

为什么说分阶段的生命周期模型有助于软件项目管理。

重难点

4、软件过程

定义，软件生命周期模型定义，什么是里程碑？它应该有哪些特征，瀑布模型的步骤、优缺点、适用于、质量保证的观点，快速原型模型的步骤、优缺点、适用于，增量模型的步骤、优缺点、适用于，螺旋模型的步骤、优缺点、适用于，螺旋模型的步骤、优缺点、适用于、螺旋模型和RUP有哪些相似之处？有何差异，为什么说喷泉模型较好地体现了面向对象软件开发过程无缝和迭代的特性，Rational统一过程的最佳实践[6]、RUP软件开发生命周期、步骤、优缺点、适用于、试比较RUP与敏捷过程，敏捷过程与极限编程中敏捷过程的4个简单的价值观、极限编程的有效实践[14]、步骤、优缺点、适用于，微软过程的微软生命周期的5个阶段、步骤、优缺点、适用于、试讨论微软过程与RUP及敏捷过程的关系

主要内容

1.1 软件危机

1.2 软件工程

1.3 软件生命周期

1.4 软件过程

江南大学-软件工程课程——软件工程导论，江大研究生团队录制，严谨复制
! 详情加q191683466

1.1 软件危机

1.1.1 软件危机的介绍

研究生团队录制，严谨复制

在计算机软件的开发和维护过程中所遇到的一系列严重问题。

包含两方面的问题：（1）如何开发软件，以满足社会对软件日益增长的需求；（2）如何更有效地维护数量不断膨胀的已有软件。

软件危机又叫软件萧条、软件困扰。

1.1 软件危机

软件危机的典型表现

- 1、对软件开发成本和进度的估计常常很不准确
- 2、用户对“已完成的”软件系统不满意的现象经常发生
- 3、软件产品的质量往往靠不住。
- 4、软件常常是不可维护的。

1.1 软件危机

软件危机的典型表现

5、软件通常没有适当的文档资料。

6、软件成本在计算机系统总成本中所占的比例逐年上升。

7、软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。

1.1 软件危机

1.1.2 产生软件危机的原因

客观原因：与软件本身特点有关

1 软件不同于硬件，管理和控制软件开发过程相当困难。

2 软件在运行过程中不会因为使用时间过长而被“用坏”。如果运行中发现了错误，很可能是遇到了一个在开发时期引入的在测试阶段没能检测出来的错误。

3 软件不同于一般程序，它的一个显著特点是规模庞大，而且程序复杂性将随着程序规模的增加而呈指数上升。

1.1 软件危机

与软件本身特点有关

4 事实上，对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。

5 目前相当多的软件专业人员对软件开发和维护还有不少糊涂观念。在实践过程中或多或少地采用了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

6 错误的认识和做法主要表现为忽视软件需求分析的重要性，认为软件开发就是写程序并设法使之运行，轻视软件维护等

1.1.2 产生软件危机的原因

1.1 软件危机

严谨复制

主观原因：软件开发与维护的方法不正确有关

1 只重视程序而忽视软件配置其余成分的糊涂观念。

2 软件开发人员在定义时期没有正确全面地理解用户需求，直到测试阶段或软件交付使用后才发现“已完成的”软件不完全符合用户的需要。

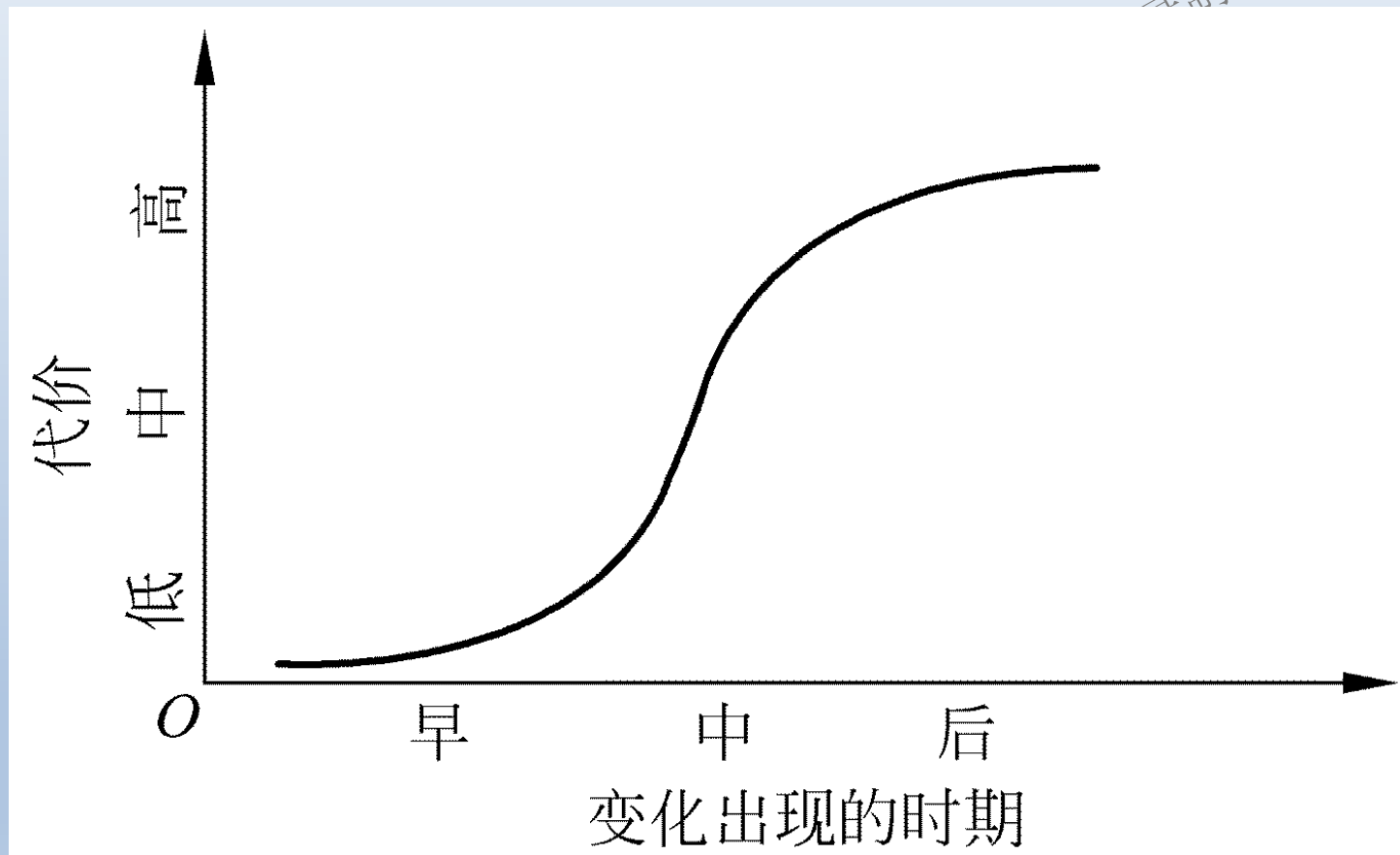
3 严重的问题是在软件开发的不同阶段进行修改需要付出的代价是很不相同的，如下图所示。

1.1 软件危机

在软件开发的不同阶段进行修改需要付出的代价

复制

复制



1.1.2 产生软件危机的原因

1.1 软件危机

1.1.3 消除软件危机的途径

1 首先应该对计算机软件有一个正确的认识。

2 充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是各类人员协同配合，共同完成的工程项目。

3 推广使用在实践中总结出来的开发软件的技术和方法，并且研究探索更好更有效的技术和方法。

4 应该开发和使用更好的软件工具。

1.2 软件工程

平谨复制

定义：软件工程是①把系统的、规范的、可度量的途径应用与软件开发、运行和维护过程，也就是把工程应用于软件；②研究①中提到的途径。

目的：提高软件的可维护性，减少软件维护所需要的工作量，降低软件系统的总成本。

1.2 软件工程

软件具有的本质特性

复制，严谨复制

软件工程关注于大型程序的构造

软件工程的中心课题是控制复杂性

软件经常变化

开发软件的效率非常重要

和谐地合作是开发软件的关键

必须有效地支持它的用户

两种背景的人创造产品这个特性与前两个特性紧密相关

1.2 软件工程

软件工程的基本原理

录制，严谨复制

1、用分阶段的生命周期计划严格管理

2、坚持进行阶段评审

3、实行严格的产品控制

4、采用现代程序设计技术

5、结果应能清楚地审查

6、开发小组的人员应该少而精

7、承认不断改进软件工程实践的必要性

1.2 软件工程

1.2.3 软件工程方法学

禁止录制，严谨复制

通常把在软件生命周期全过程中使用的一整套技术方法的集合，称为方法学，也成为泛型。软件工程方法学包含3个要素：方法，工具和过程。其中，方法是完成软件开发的各项任务的技术方法，回答“怎样做”的问题；工具是为运用方法而提供的自动的或半自动的软件工程支持环境；过程是为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

目前使用最广泛的软件工程方法学，分别是传统方法学和面向对象方法学。

1.2 软件工程

1.2.3 软件工程方法学

课程录制，严谨复制

1、传统方法学

- 传统方法学也称为生命周期方法学或结构化范型。它采用结构化技术(结构化分析、结构化和结构化实现)来完成软件开发的各项任务，并使用适当的软件工具或软件工程环境来支持结构化技术的运用。

2、面向对象方法学

- 与传统方法相反，面向对象方法把数据和行为看成是同等重要的，它是一种以数据为主线，把数据和对数据的操作紧密地结合起来的方法。

1.2 软件工程

严谨复制

传统方法学的要点:

属于传统方法学。传统的软件开发方法大部分采用瀑布模型。这种模型要求每一阶段都以前一阶段写成的文档为基础完成工作每一阶段将要完成时都要求开发人员进行验证和确认。

优点: 把软件生命周期划分成若干个阶段，每个阶段的任务相对独立，而且比较简单，便于不同人员分工协作，从而降低了整个软件开发工程的困难程度。

缺点: 当软件规模庞大，或者对软件的需求是模糊的或会随时间变化而变化的时候，使用传统方法学开发软件往往不成功，维护起来仍然困难。

1.2 软件工程

严谨复制

面向对象范型的要点：

程序中任何元素都是对象，复杂对象由比较简单的对象组合而成；

把所有对象都划分成类，每个类都定义了一组数据和一组操作；

按照父类与子类的关系，把若干个相关类组合成一个层次结构的系统；

对象彼此间只能通过发送消息互相联系；

优点：降低了软件产品的复杂性，提高了软件的可理解性，简化的软件的开发和维护工作，促进了软件重用。

1.3 软件生命周期

研究生团队录制，严谨复制

软件生命周期由软件定义、软件开发和运行维护(也称为软件维护)3个时期组成，每个时期又进一步划分成若干个阶段。

江南大学-软工复试课程
! 详情加q1916837466

1.3 软件生命周期

软件定义时期：包括问题定义（确定要求解决的问题是什么），可行性研究（决定该问题是否存在一个可行的解决办法），需求分析（深入了解用户得需求，在目标系统必须做什么这个问题上和用户取得完全一致的看法）

软件开发时期：设计（概要设计，详细设计），编写程序，测试。

即包括总体设计，详细设计，编码和单元测试，综合测试。
前两个阶段称为系统设计，后两个阶段称为系统实现

维护时期的主要任务：使软件持久地满足用户的需要。

1.3 软件生命周期

软件生命周期每个阶段的基本任务

1. 问题
定义

2. 可行
性研究

3. 需求
分析

4. 总体
设计

5. 详细
设计

6. 编码
和单元
测试

7. 综合
测试

8. 软件
维护

软件定义
时期：
1,2,3

软件开发
时期：
4,5,6,7

软件维护
时期： 8

1.4 软件过程

复制

软件过程是为了获得高质量软件所需完成的一系列任务的框架，它规定完成各项任务的工作步骤；且定义了运用方法的顺序、应该交付的文档资料、为保证软件质量协调变化所需要采取的管理措施，以及标志软件开发各个阶段任务完成的里程碑；软件过程是软件工程方法学的3个重要组成部分之一。

软件生命周期模型的定义：

是跨越整个生存期的系统开发、运作和维护所实施的全部过程、活动和任务的结构框架。

典型的软件过程模型：

瀑布模型，快速原型模型，增量模型，螺旋模型，喷泉模型，Rational统一过程模型，敏捷过程与极限编程，微软过程。

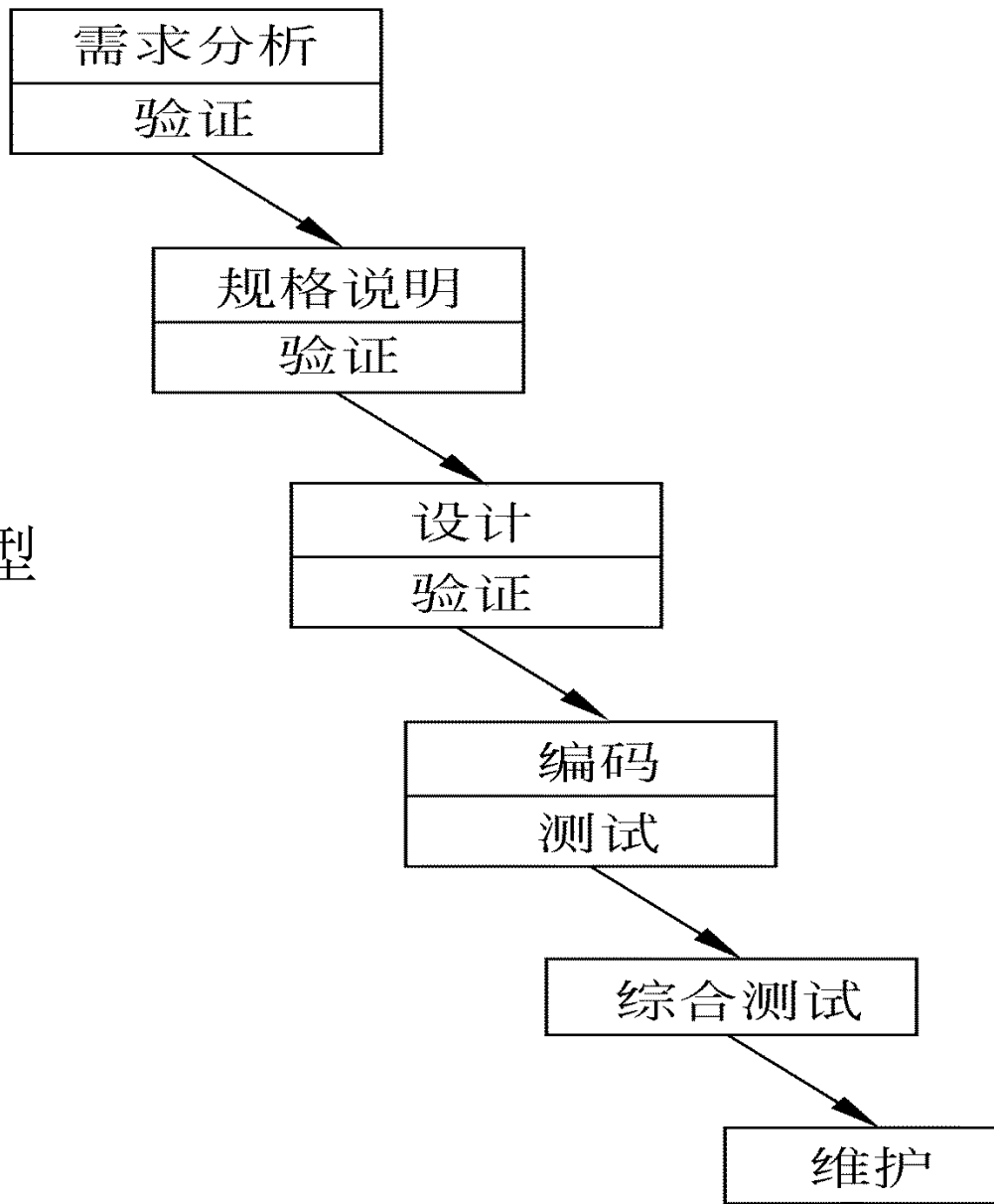
1.4 软件过程

1.4.1 瀑布模型

瀑布模型一直是唯一被广泛采用的生命周期模型，现在它仍然是软件工程中应用得最广泛的过程模型。如下图所示为传统的瀑布模型

如图1.2所示为传统的瀑布模型。

图1.2 传统的瀑布模型



江南大学-软工复试课程—
! 详情加q1916837466

1.4 软件过程

1.4.1 瀑布模型

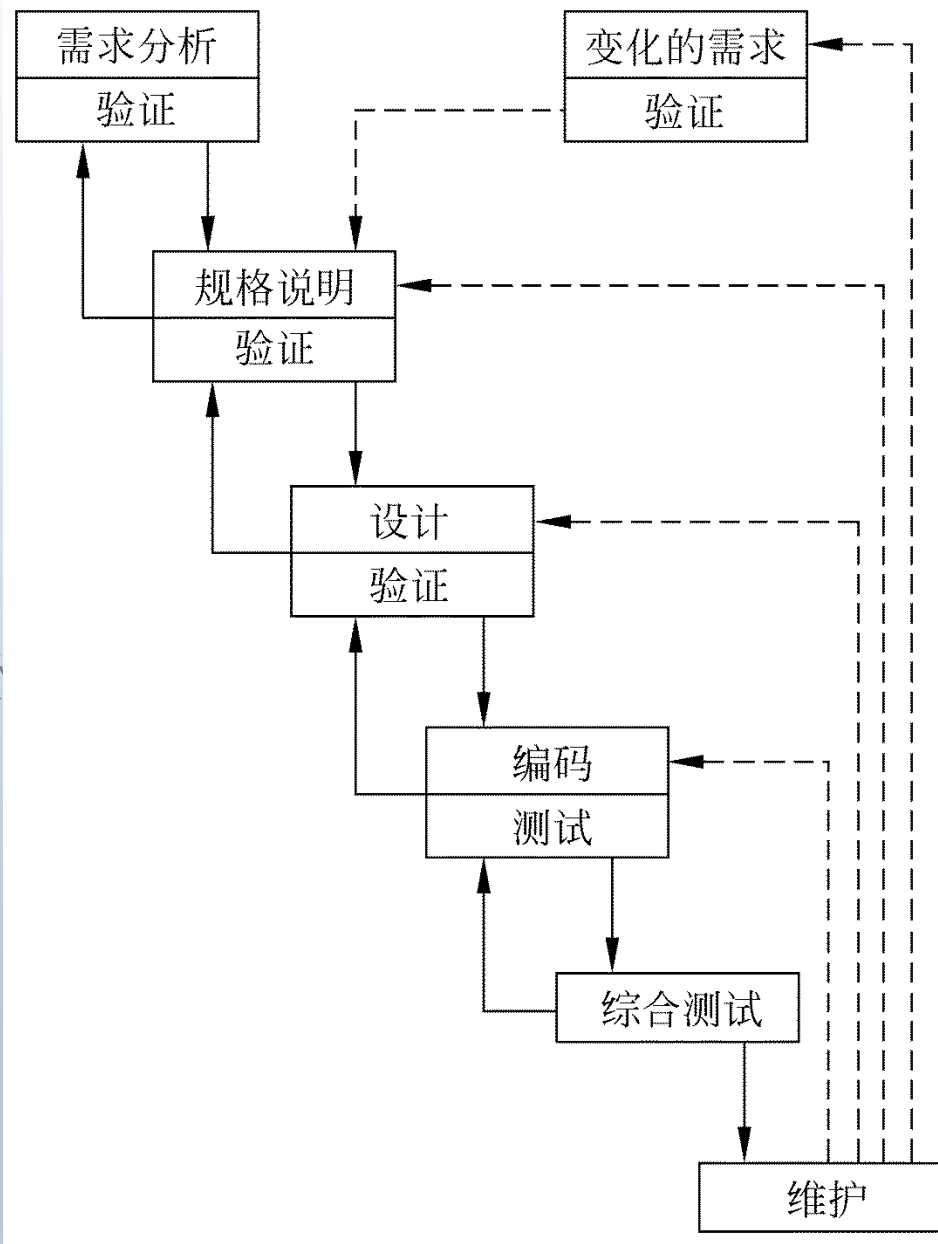
江大研究生团队录制，严谨复制

传统的瀑布模型过于理想化了，事实上，人在工作过程中不可能不犯错误。实际的瀑布模型是带“反馈环”的，如系统图1.3所示。

江南大学-软工
! 详情加q19168374

1、图中实线箭头表示开发过程，虚线箭头表示维护过程。

2、实际的瀑布模型当在后面阶段发现前面阶段的错误时，需要沿图中左侧的反馈线返回前面的阶段，修正前面阶段的产品之后再回来继续完成后面阶段的任务。



1.4 软件过程

1.4.1 瀑布模型

工大研究生团队录制，严谨复制

适用范围：需求明确，小规模软件开发

传统软件工程方法学的软件过程基本上可以用瀑布模型来描述

优点：强迫开发人员采用规范的技术方法；严格地规定了每个阶段必须提交的文档；每个阶段结束前必须正式进行严格的技术审查和管理复查。

缺点：开发人员和用户之间缺乏有效的沟通，很可能导致最终开发出来的软件产品不能真正满足用户的需求。

案例

某个老师（T）想要考察一个同学（S）的学习情况和技术水平，于是交给该学生一个任务。

T：我有一个朋友想要一个图象浏览软件，能够查看多种格式的图象，包括BMP、TIFF、JPG、PNG，并且能够支持一般的放大、缩小、漫游。你能做这样一个软件吗？

S：就是类似ACDSEE这样的软件吗？

T：差不多，不过不需要那么强大的功能，我这个朋友计算机是外行，最好能做的比较方便，傻瓜型的，例如象ACDSEE自动翻页这种功能还是要的。

S：我以前学过BMP和JPG的图象格式解析，我想没有问题

T：好的，给你30天时间，下周你再来一趟，跟我讲一下你的工作进度。

这位同学非常明白老师的意图，回去后想了一下，并列出了一个清单

工作清单

一 功能：

1. 读取、显示、另存四种格式图片（**BMP、TIFF、JPG、PNG**）
2. 放大、缩小、漫游
3. 列出当前目录下所有四种格式图片文件名
4. **PAGEUP（PAGEDOWN）** 自动调出当前目录上一张（下一张）图片

二 其它说明：

1. 界面尽量简介，容易操作
2. 不要图片预览和打印

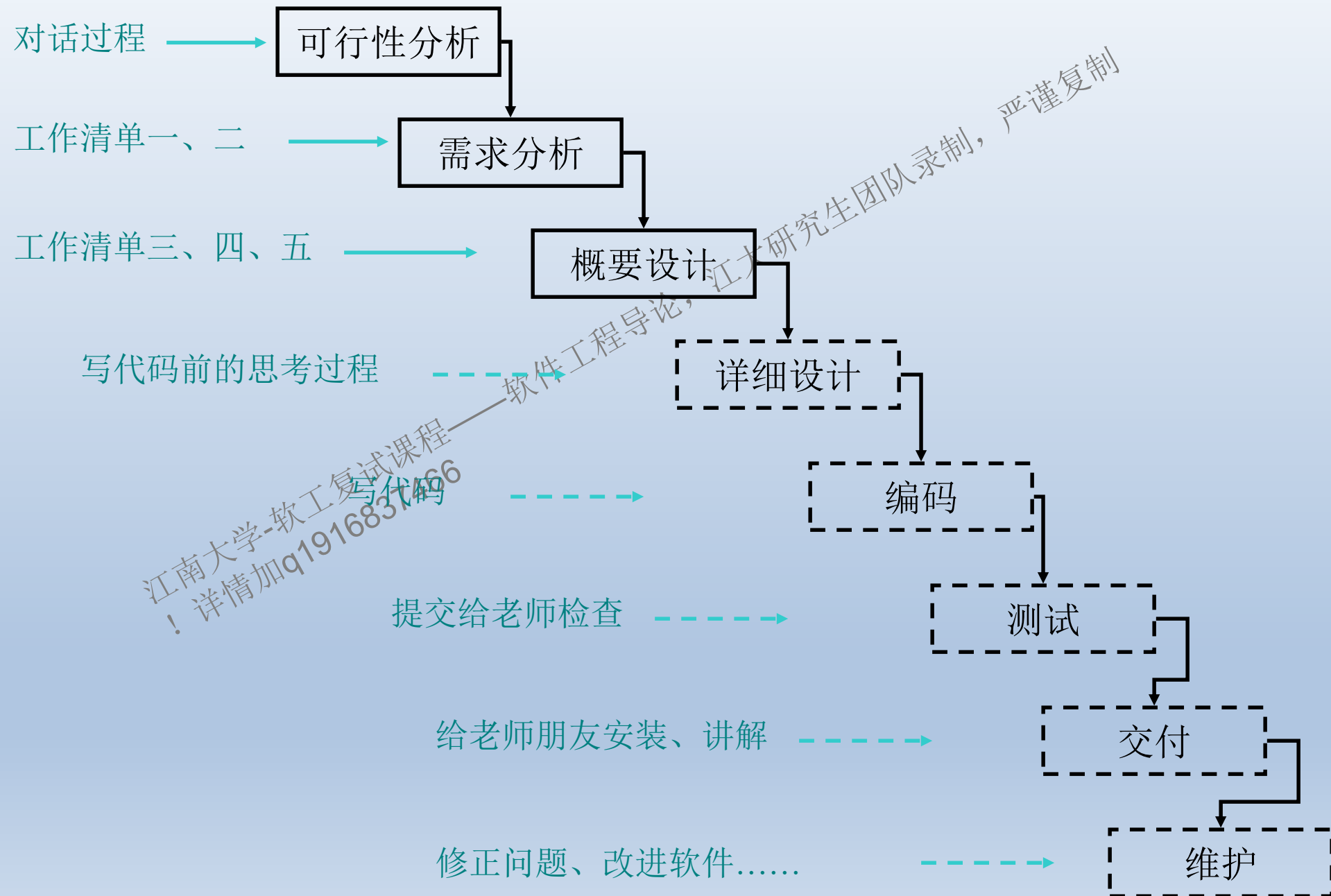
三 开发工具：**VC 6**

四 开发环境：普通PC机；**Window2000/xp**

五 工作量：

- 1.研究一下四种图片的格式
- 2.设计一个解析器类，解析这四种格式
- 3.设计一个文档类，实现读取、另存和目录浏览功能
- 4.设计一个视图类，实现显示、缩放、漫游功能

软件过程的8个一般阶段



可能情况1

一切顺利，学生S按期交付了软件，经过一两周的试用、修改、完善后，三方都比较满意，该软件在老师的朋友那里成为一个得心应手的工具。



Waterfall Model(瀑布模型)，又称为线性顺序模型

江南大学-软工复试课程——
! 详情加q1916837466

软件工程导论，

严谨复制

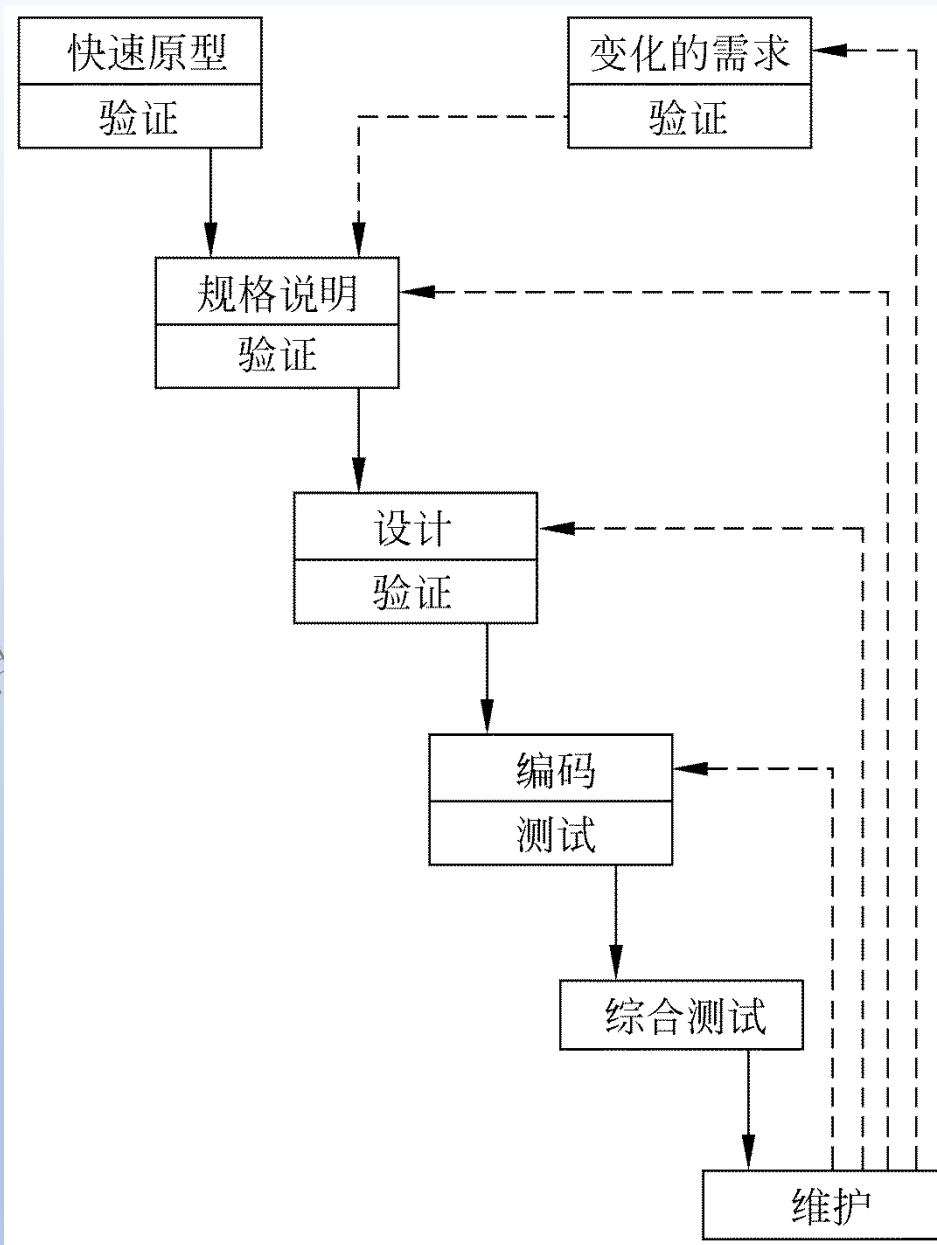
1.4 软件过程

1.4.2. 快速原型模型

概念:

快速原型是快速建立起来的可以在计算机上运行的程序，它所能完成的功能往往是最终产品能完成的功能的一个子集。如下图1.4所示：

图1.4 中实线箭头
表示开发过程
虚线箭头表示维护
过程



1.4.2. 快速原型模型

江南大学-软工复试课程——软件
! 详情加q1916837466

1.4 软件过程

1.4.2. 快速原型模型

适用范围：需求不明确的

所谓快速原型模型是快速建立起来的、可在计算机上运行得程序，它所能完成的功能往往是最终的软件产品所能完成功能的子集。原型是软件开发人员与用户沟通的强有力工具，因此有助于所开发出的软件产品满足用户的真实需求。

优点：使用这种软件过程开发出的软件产品通常能满足用户的真实需求；软件产品的开发过程基本上是线性顺序过程。

缺点：这样开发出来的系统，往往可能因为折中而采取不合适的操作系统或程序设计语言。

1.4.2. 快速原型模型

团队录制，严谨复制

软件过程模型案例

可能情况2

一周后，学生去见老师，并提交了工作清单，他发现老师的这位朋友（C）和老师在一起。

S: 这是工作清单，我已经研究清楚了四种文件的格式，可以写代码了。

T: 很好，不过我这位朋友有一些新想法，你不妨听听。

C: 你好。我新买了一个扫描仪，你的程序可不可以直接扫描图片进来。

S: 你可以自己扫描呀，买扫描仪的时候一般都会送正版软件的。

C: 是的，可是我一直不太会用，你知道我计算机水平不高，学一些新东西很累，也没有时间，如果你能直接链接扫描仪，我只要学会你的软件就行了，我愿意多支付一些费用.....，还有，我想建一个图片库，你知道，我工作时需要上百个图片，经常找不到，最好还带模糊查询。

软件过程模型案例

可能情况2（续）

S:!!!!

C:还有一些，现在一时想不起来，我想起来的话会再跟你联系，时间上可以长一些。

S:.....!!!! !!!!! !!!!!

T:要不这样吧，你先做一个样子出来给C看看，一边做，一边改。

C:这样最好，看见一个基本样子我就知道我想要什么了事情就这样定下来了，S愤怒的撕掉了自己的工作清单.....，回去后S花1天时间用DELPHI做了个样子，

只能读BMP和JPG文件，做了些菜单和工具栏，用ACCESS建了一个图片库。就这个“假”的程序，S和C讨论了一天，S又修改了几次，又讨论了几次，一周后，这个“假”的程序表面看起来和真的一模一样。

软件过程

可能情况2

于是S打算用VC重写这个程序，但是他很快发现继续用DELPHI写更方便，因为至少界面不用重做了，于是.....，两个月后，这个事情终于结束了。

S顺利的完成了他的毕业设计《JPG压缩优化算法设计》，C一直使用这个软件管理他的图片，并庆幸花了这么少的钱得到了这么有用的东西，而T，则正在考虑如何为他下一批学生分派任务。



原型模型

江南大学-软工复试课程
! 详情加q1916837466

(原型模型) Prototyping Model

听取客户需求

构建系统 反复修改

客户
测试驱动

江南大学-软工复
! 详情加q19168

—软件工程导论, 14

团队录制, 严谨复制

1.4 软件过程

1.4.2. 快速原型模型

团队录制，严谨复制

理解：快速原型模型的第一步是快速建立一个能反映用户主要需求的原型系统，让用户在计算机上试用它，通过实践来了解目标系统的概貌。通常，用户试用原型系统之后会提出许多修改意见，开发人员按照用户的意见快速地修改原型系统，然后再次请用户试用……一旦用户认为这个原型系统确实能做他们所需要的工作，开发人员便可据此书写规格说明文档，根据这份文档开发出的软件便可以满足用户的真实需求。

1.4 软件过程

1.4.2. 快速原型模型

团队录制，严谨复制

当快速原型的某个部分是利用软件工具由计算机自动生成的时候，可以把这部分用到最终的软件产品中。例如，用户界面通常是快速原型的一个关键部分，当使用屏幕生成程序和报表生成程序自动生成用户界面时，实际上可以把得到的用户界面用在最终的软件产品中。

江南大学-软
! 详情加q19166

1.4 软件过程

1.4.3. 增量模型

研究生团队录制，严谨复制

概念：

增量模型也称为渐增模型。使用增量模型开发软件时，把软件产品作为一系列的增量构件来设计、编码、集成和测试。每个构件由多个相互作用的模块构成，并且能够完成特定的功能。使用增量模型时，第一个增量构件往往实现软件的基本需求，提供最核心的功能。

增量模型如下图1.5所示：

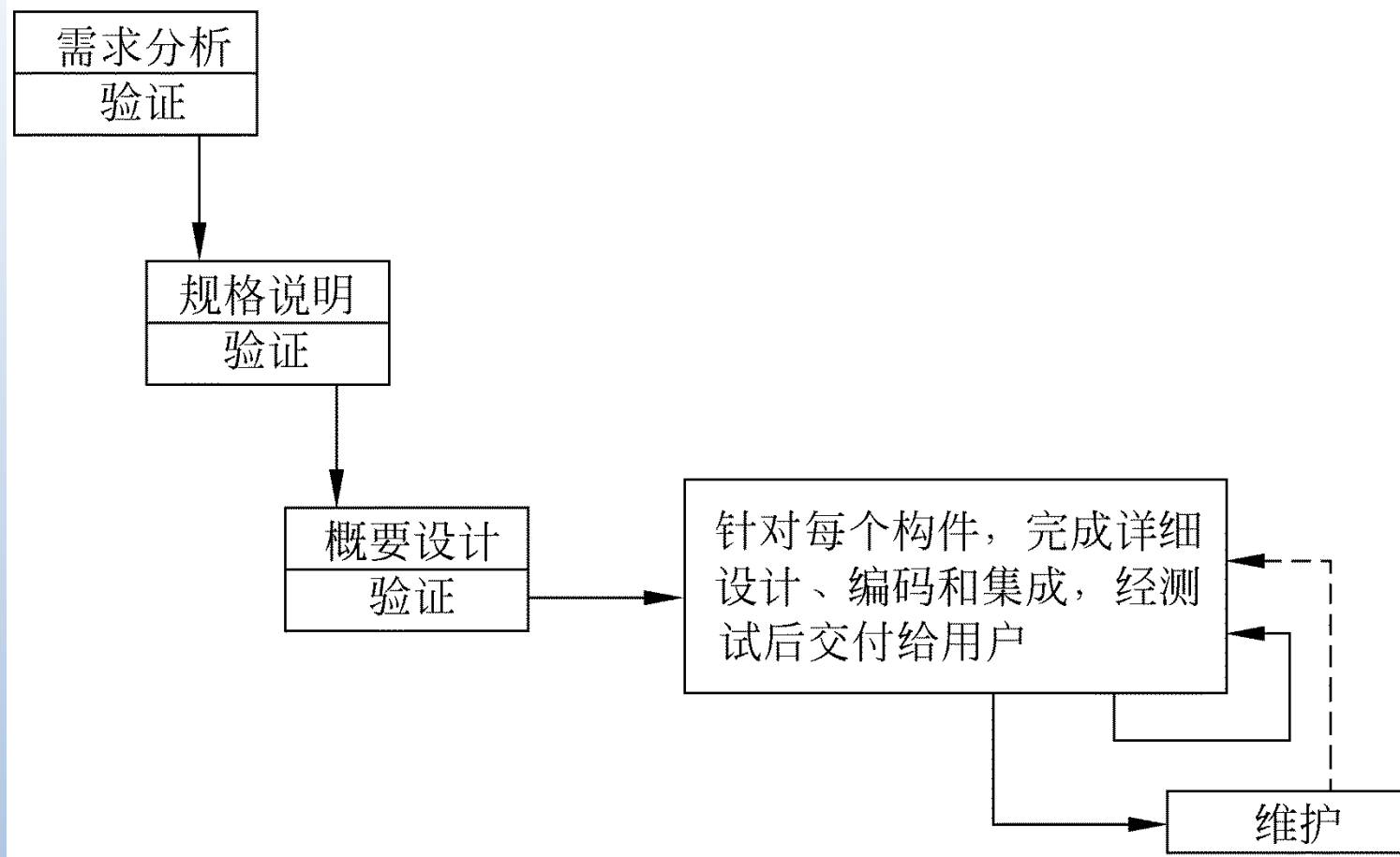


图1.5 增量模型

1.4 软件过程

1.4.3. 增量模型

录制，严谨复制

适用范围：进行已有产品升级或新版本开发；对完成期限严格要求的产品；对所开发的领域比较熟悉而且已有原型系统。

增量模型也称为渐增模型，使用增量模型开发软件时，把软件产品作为一系列增量构件来设计、编码、集成和测试。每个构件由若干个相互协作的模块构成，并且能够完成相对独立的功能。

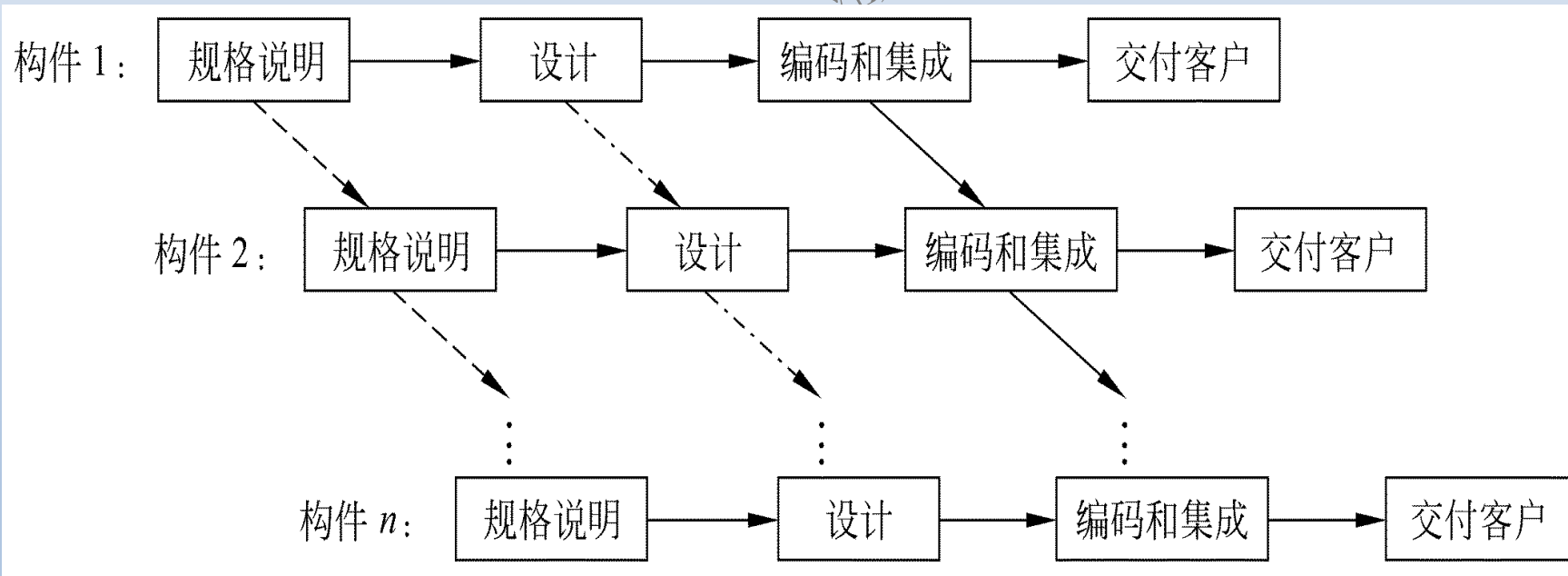
优点：能在较短时间内向用户提交可完成部分工作的产品；逐步增加产品功能，从而使用户有较充裕的时间学习和适应新产品，减少一个全新的软件给用户所带来的冲击。

缺点：集成困难，软件结构开放不好设计。

1.4 软件过程

1.4.3. 增量模型

风险更大的增量模型：



1.4 软件过程

1.4.3. 增量模型

上图描绘了一种风险更大的增量模型：一旦确定了用户需求之后，就着手拟定第一个构件的规格说明文档，完成后规格说明组将转向第二个构件的规格说明，与此同时，设计组开始设计第一个构件……用这种方式开发软件，不同的构件将并行地构建，因此有可能加快工程进度，但是，使用这种方法将冒构建无法集成到一起的风险，除非密切地监控整个开发过程，否则整个工程可能毁于一旦。

江南大学-软工复试辅导班
! 详情加q1916837466

江南大学-软工复试辅导班团队录制，严谨复制

1.4 软件过程

1.4.3. 增量模型

团队录制，严谨复制

理解：使用增量模型时，第一个增量构件往往实现软件的基本需求，提供最核心的功能。例如，使用增量模型开发字处理软件时，第1个增量构件提供基本的文件管理，编辑和文档生成功能；第2个增量构件提供更完善的编辑和文档生成功能；第3个增量构件实现拼写和语法检查功能；第4个增量构件完成高级的页面排版功能。

江南大学
! 详情加Q

1.4 软件过程

1.4.4 螺旋模型

概念：

螺旋模型的基本思想是，使用原型及其他方法来尽量降低风险。理解这种模型的一个简便方法，是把它看作在每个阶段之前都增加了风险分析过程的快速原型模型。

螺旋模型如下图所示：

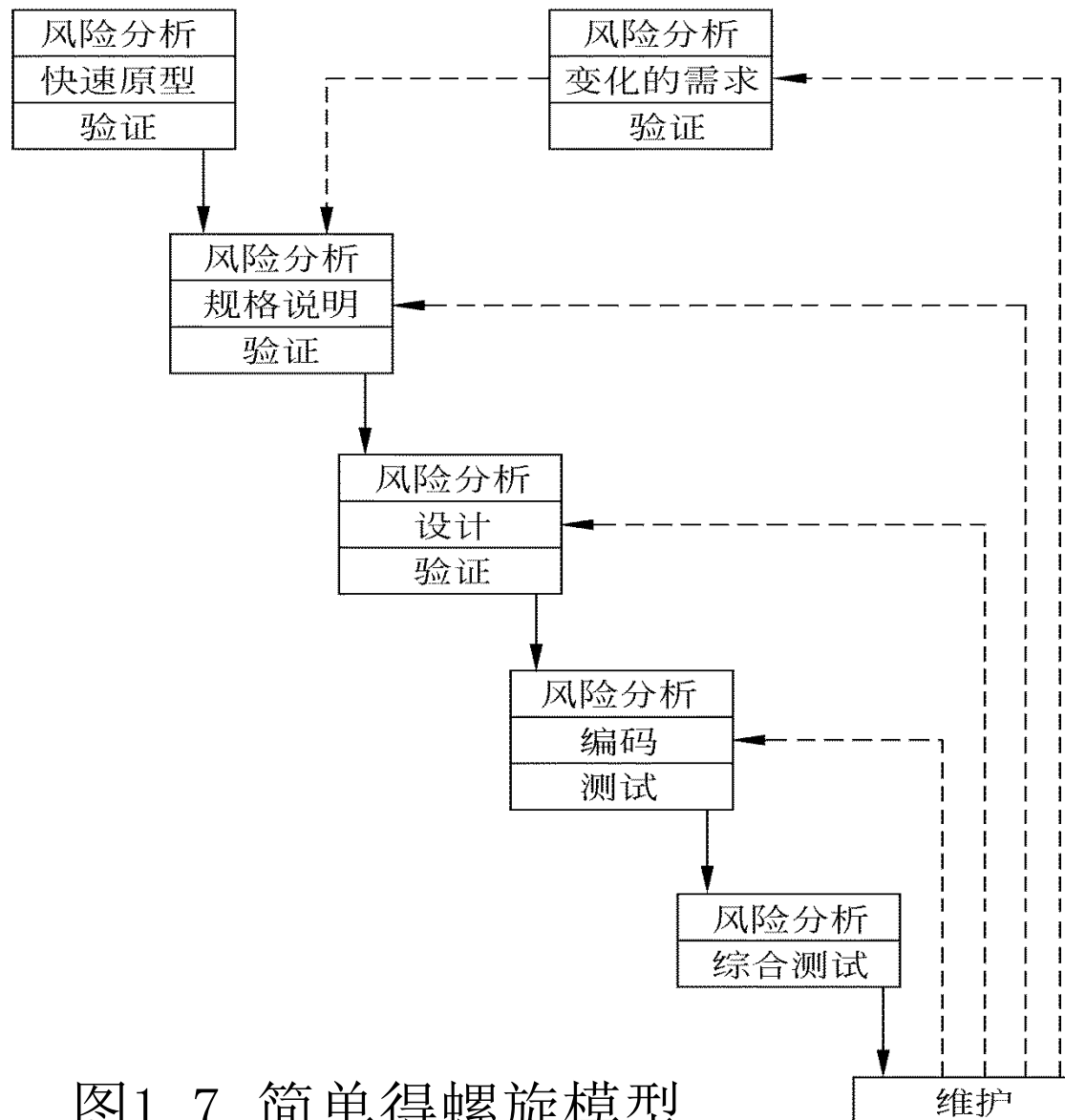


图1.7 简单得螺旋模型

1.4.4 螺旋模型

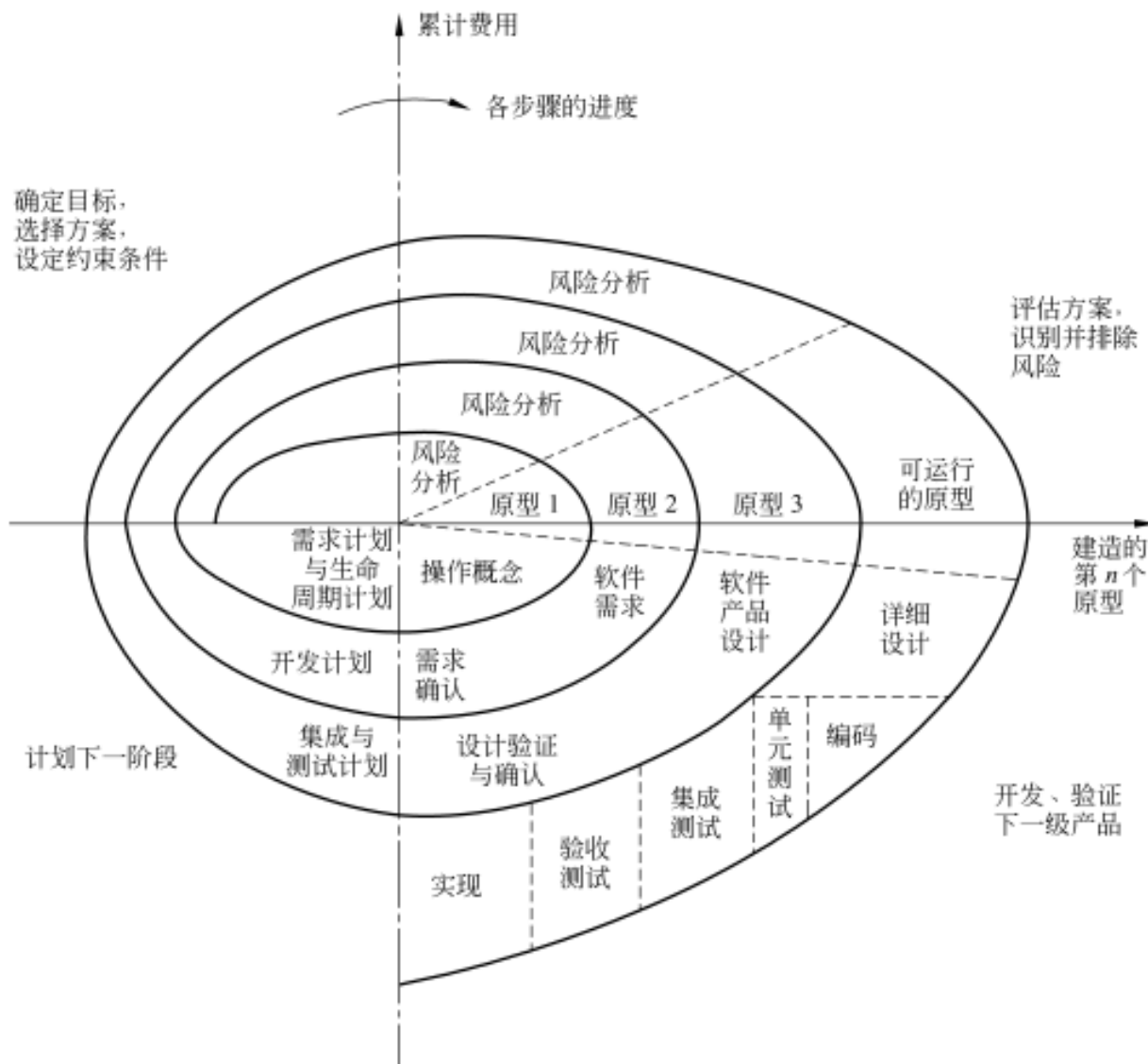


图1.8完整的螺旋模型

1.4.4 螺旋模型

1.4 软件过程

1.4.4 螺旋模型

复制，严谨复制

适用范围： 内部开发的大型软件项目

把它看做在每个阶段之前都增加了风险分析过程的快速原型模型，是一种风险驱动的迭代式开发过程。具体为不断重复生命周期模型，并在每个生命周期结构后，向用户提交一个可运行的软件产品版本。

优点： 有利于已有软件得重用；有助于把软件质量作为软件开发的一个重要目标；减少了过多测试或测试不足所带来的风险；软件维护与软件开发没有本质区别。

缺点： 需要相当的风险分析评估的专门技术，且成功依赖于这种技术。很明显一个大的没有被发现的风险问题，将会导致问题的发生，可能导致烟花的方向失去控制。这种模型相对比较新，应用不广泛，其功效需要进一步验证。

1.4 软件过程

1.4.5. 喷泉模型

团队录制，严谨复制

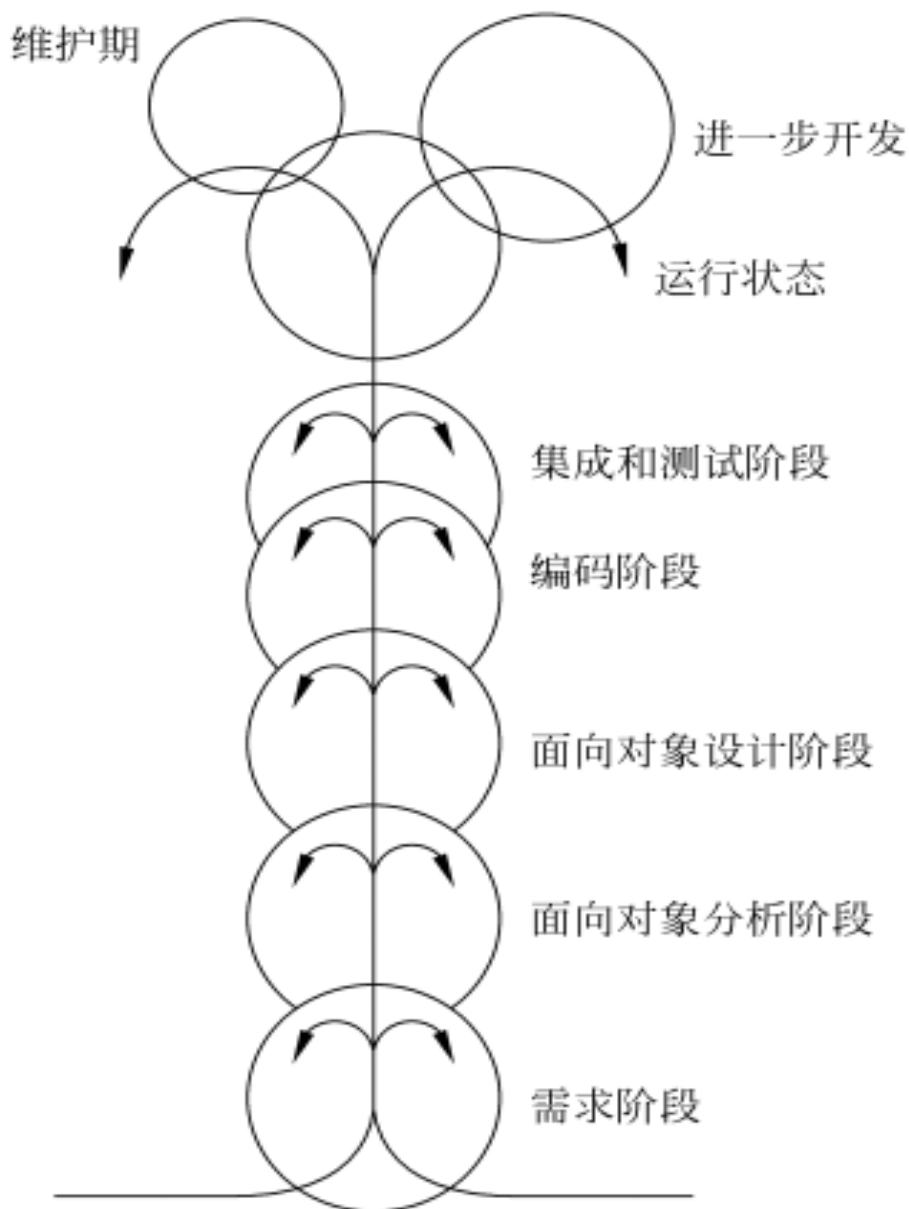
概念：

“喷泉”这个词体现了面向对象软件开发过程迭代和无缝的特性。迭代是软件开发过程中普遍存在的一种内在属性。用面向对象方法学开发软件时，工作重点应该放在生命周期中的分析阶段。

喷泉模型图如下图1.9所示：

图中代表不同阶段的圆圈相互重叠，这明确表示两个活动之间存在交迭；

图中在一个阶段内的向下箭头代表该阶段内的迭代（或求精）。图中较小的圆圈代表维护，圆圈较小象征着采用了面向对象范型之后维护时间缩短了。



1.4.5. 喷泉模型

1.4 软件过程

1.4.5. 喷泉模型

复制，严谨复制

喷泉模型是典型的面向对象的软件过程模型之一。

为什么说喷泉模型较好的体现了面向对象软件开发过程无缝和迭代的特性？

由于各阶段都使用统一的概念和表示符号，因此，整个开发过程都是吻合一致的，或者说是“无缝”连接的，这自然就很容易实现各个开发步骤的多次反复迭代，达到认识的逐步深化。而喷泉模型则很好的提现了面向对象软件开发过程迭代和无缝的特性。

1.4 软件过程

1.4.6. Rational统一过程

研究生团队录制，严谨复制

概念：

Rational统一过程（Rational Unified Process,**RUP**）是由Rational软件公司推出的一种完整而且完美的软件过程。

RUP总结了经过多年商业化验证的**6**条最有效的软件开发经验，这些经验被称为“最佳实践”。最佳实践包括：迭代式开发；管理需求；使用基于构件的体系结构；可视化建模；验证软件质量；控制软件变更。

1.4 软件过程

1.4.6. Rational统一过程

团队录制，严谨复制

适用范围： 大型的需求不断变化的复杂软件系统项目

优点： 提高了团队生产力，在迭代的开发过程、需求管理、基于组建的体系结构、可视化软件建模、验证软件质量及控制软件变更等方面、针对所有关键的开发活动为每个开发成员提供了必要的准则、模板和工具指导，并确保全体成员共享相同的知识基础。它建立了简洁和清晰的过程结构，为开发过程提供较大的通用性。

缺点： RUP只是一个开发过程，并没有涵盖软件过程的全部内容，例如它缺少关于软件运行和支持等方面的内容，此外，他没有支持多项目的开发结构，这在一定程度上降低了在开发组织内大范围实现重用的可能性。

1.4 软件过程

1.4.6. Rational统一过程

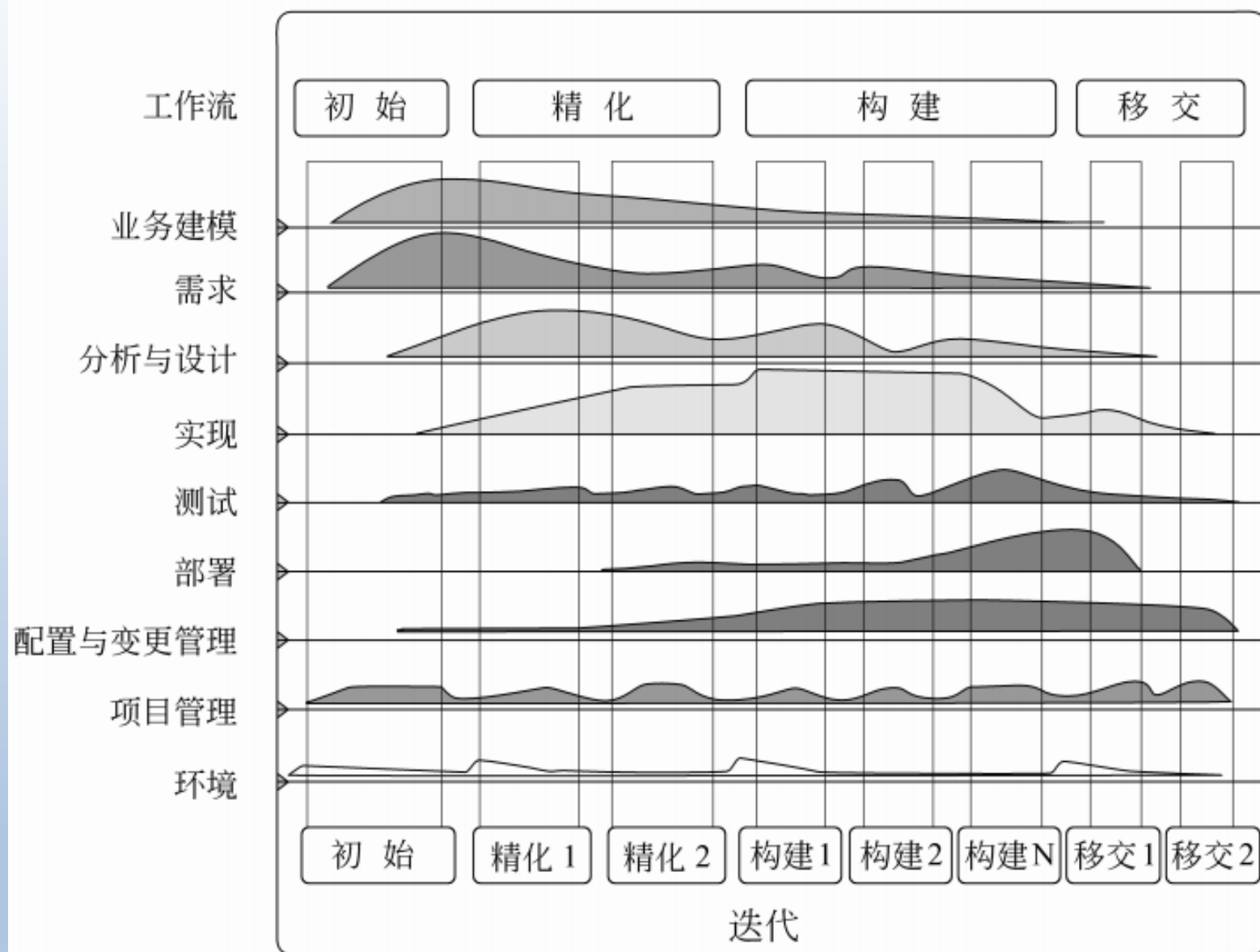
江大研究生团队录制，严谨复制

RUP软件开发生命周期

RUP软件开发生命周期是一个二维的生命周期模型，如下图所示。图中纵轴代表核心工作流，横轴代表时间。9个核心工作流和4个工作阶段。9个核心工作流中前6个为核心过程工作流程，后3个为核心支持工作流程。

9个核心工作流：业务建模，需求，分析与设计，实现，测试，部署，配置与变更管理，项目管理，环境。

阶段



1.4.6. Rational统一过程

1.4 软件过程

1.4.6. Rational统一过程

江大研究生团队录制，严谨复制

工作阶段

RUP把软件生命周期划分成4个连续的阶段。每个阶段都有明确的目标，并且定义了用来评估是否达到这些目标的里程碑。每个阶段的目标通过一次或多次迭代来完成。

下面简述4个阶段的工作目标。

！详情

1.4 软件过程

1.4.6. Rational统一过程

初始阶段：建立业务模型，定义最终产品视图，并且确定项目的范围。

精化阶段：设计并确定系统的体系结构，制定项目计划，确定资源需求。

构建阶段：开发出所有构件和应用程序，把它们集成为客户需要的产品，并且详尽地测试所有功能。

移交阶段：把开发出的产品提交给用户使用。

1.4 软件过程

1.4.7. 敏捷过程与极限编程

江大研究生团队录制，严谨复制

敏捷过程

适用范围：商业竞争环境下对小型项目提出的有限资源和有限开发时间的约束/开发可用资源及开发时间都有较苛刻约束得小型项目

敏捷软件开发宣言的4个简单的价值观：个体和交互胜过过程和工具；可以工作的软件胜过面面俱到的文档；客户合作胜过合同谈判；响应变化胜过遵循计划。

1.4 软件过程

1.4.7. 敏捷过程与极限编程

研究生团队录制，严谨复制

极限编程XP，极限编程是敏捷过程中最富盛名的一个。

适用范围：需求模糊且经常改变的场合。

特点：对变化和不确定性的更快速、更敏捷的反应；在快速的同时仍然能够保持可持续的开发速度。

有效实践：客户作为开发团队的成员；使用用户素材；短交付周期；验收测试；结对编程；测试驱动开发；集体所有；持续集成；可持续的开发速度；开放的工作空间；及时调整计划；简单的设计；重构；使用隐喻。

1.4 软件过程

1.4.7. 敏捷过程与极限编程

极限编程的整体开发过程

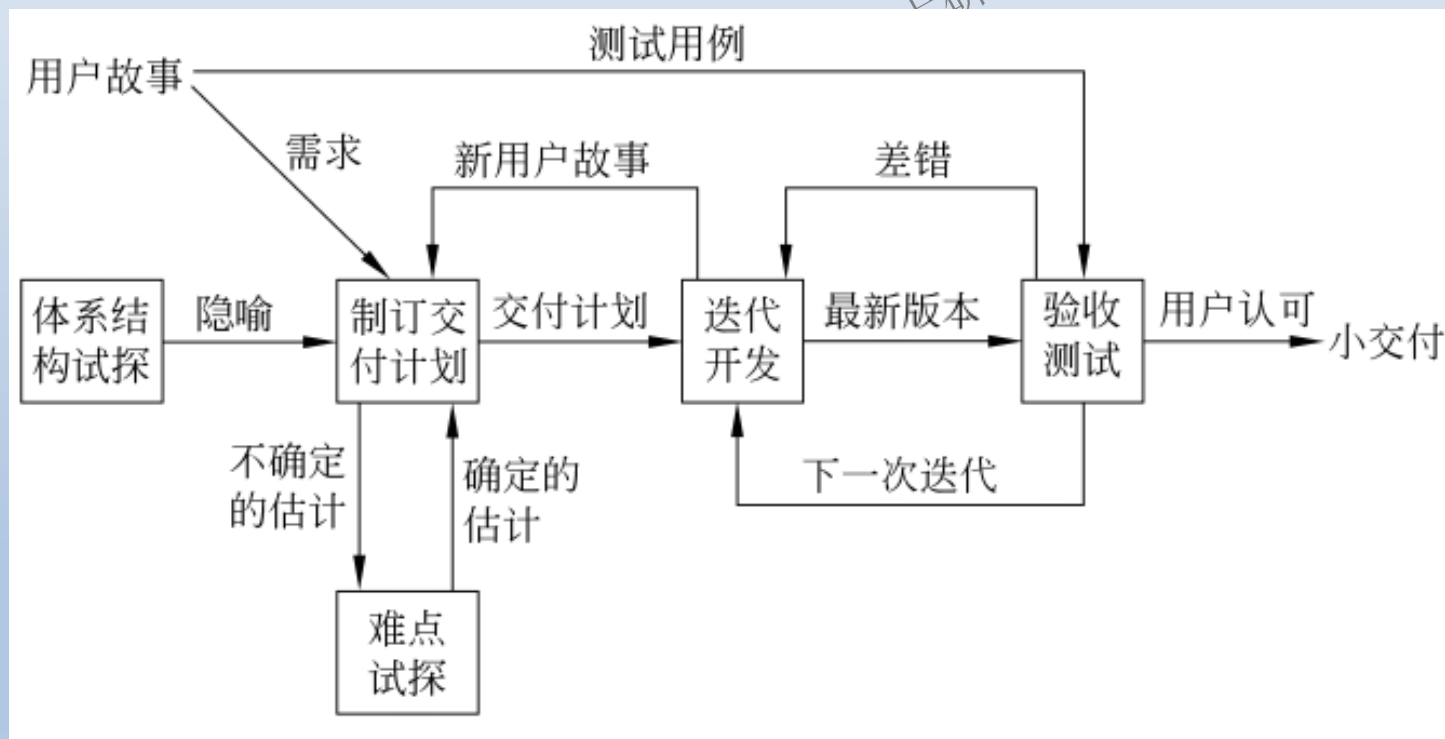


图1.11

1.4 软件过程

1.4.7.敏捷过程与极限编程

极限编程的迭代过程

江大研究生团队录制，严谨复制

图1.11描述了极限编程的整体开发过程。首先，项目组针对客户代表提出的“用户故事”进行讨论，提出隐喻，在此项活动中可能需要对体系结构进行“试探”。然后，项目组在隐喻和用户故事的基础上，根据客户设定的优先级制订交付计划。接下来开始多个迭代过程（通常每个迭代历时1~3周），在迭代期内产生的新用户故事不在本次迭代内解决，以保证本次开发过程不受干扰。开发出的新版本软件通过验收测试之后交付用户使用。

1.4 软件过程

1.4.7.敏捷过程与极限编程

隐喻。

在极限编程中有一个很有意思的实践，就是隐喻。他是指的系统架构、设计和开发过程中将一些重要、关键、复杂的概念，用隐喻的方式表达出来，方便他人能迅速的理解，双方迅速达成一致。这也是为什么是极限的原因。

隐喻的特征是暗示，暗示的好那隐喻很容易达到沟通的作用，隐喻的不好很容易把人搞晕掉。要隐喻的好，前提条件是双方对隐喻的事物是有一致理解的，即你我都知道对方知道的事实。

在敏捷过程中常见的隐喻都有哪些呢？列举。

1.4 软件过程

1.4.7.敏捷过程与极限编程

如：湖水与石头

当一个湖中有很多水，水面很高时，湖中的石块都被水所覆盖，此时即使有很大的暗礁，人们也看不到。但是当水量减少，水面降低时，一些大石块就暴露出来了。接下来随着湖水的进一步减少，中等石块和小石块也逐步被人们发现。

这告诉我们一个什么道理呢？想想软件开发的过程吧，如果采取大批量的做法，一次性提交很多功能，就好比拥有很多水量的湖，你看不到其中隐含的问题，甚至一些很严重的问题，都隐藏在这里面不容易被发现。如果换种做法，采取小批量的交付模式，每次只提交一小部分功能，这会发生什么呢？这就好比湖水减少了，一些隐藏的“石块”立刻就会暴露出来，这样平时遇到的各种问题，都不会被累积成为一个大包袱，能够被及时发现和解决。

1.4 软件过程

1.4.7.敏捷过程与极限编程

如：猪和鸡的故事

一天，一头猪和一只鸡在路上散步。鸡看了一下猪说：“嗨，我们合伙开一家餐馆怎么样？”。猪回头看了一下鸡说：“好主意，那你准备给餐馆卖什么呢？”。鸡想了想说：“餐馆卖火腿和鸡蛋怎么样？”。猪说：“不开了，我全身投入(火腿是一次性资源)，而你(鸡蛋是可再生的)只是参与而已！”

它展示了在Scrum中的两组角色：**猪和鸡**。猪被认为是Scrum团队中的核心成员，在一个团队中产品的负责人和Scrum主管和开发团队就是“猪”角色。鸡不是Scrum的一部分，但必须要考虑他们，用户、客户或提供商、经理等扮演着“鸡”角色。需要说明的是在Scrum团队中不会有一个同时成为“猪”角色和“鸡”角色。

1.4 软件过程

1.4.7.敏捷过程与极限编程

如：堵车的高速路

一条高速路上，在没有任何车的情况下，一辆车法定最快的速度是120KM/小时，但是这时候高速路的使用效率是最低的。加两辆车，还不错，继续最高速度行驶。但是随着车辆的增多，到了一定程度，司机们就开始减速，不敢以最快的速度往前冲。车辆继续增多，速度继续下降，直到最后，车都在高速上了，高速路同时承载了最多的车辆，但是大家都堵在高速路上打麻将。

为什么会这样呢？每一辆车在我们的研发过程中相当于一个在制品，当我们只有少量在制品的时候，团队能集中精力应对这些在制品，能保证在最短的时间完成工作，相当于车辆在最高的速度在行驶，如果只有一辆车，我们说这种是单件流。每一辆车新开上高速，就相当于我们启动了一个在制品。每一个在制品都会让团队精力分散，启动的在制品越来越多，而交付的越来越少，最终所有在制品都卡在研发过程中，没有能交付的产品。那么怎么办呢？现实生活中如果流量过大，交警一般都会限流，只允许一部分车上高速。在敏捷里面，我们同样要限流，限制同时能启动的任务数，以终为始，真正完成一个任务后才能启动下一个任务。

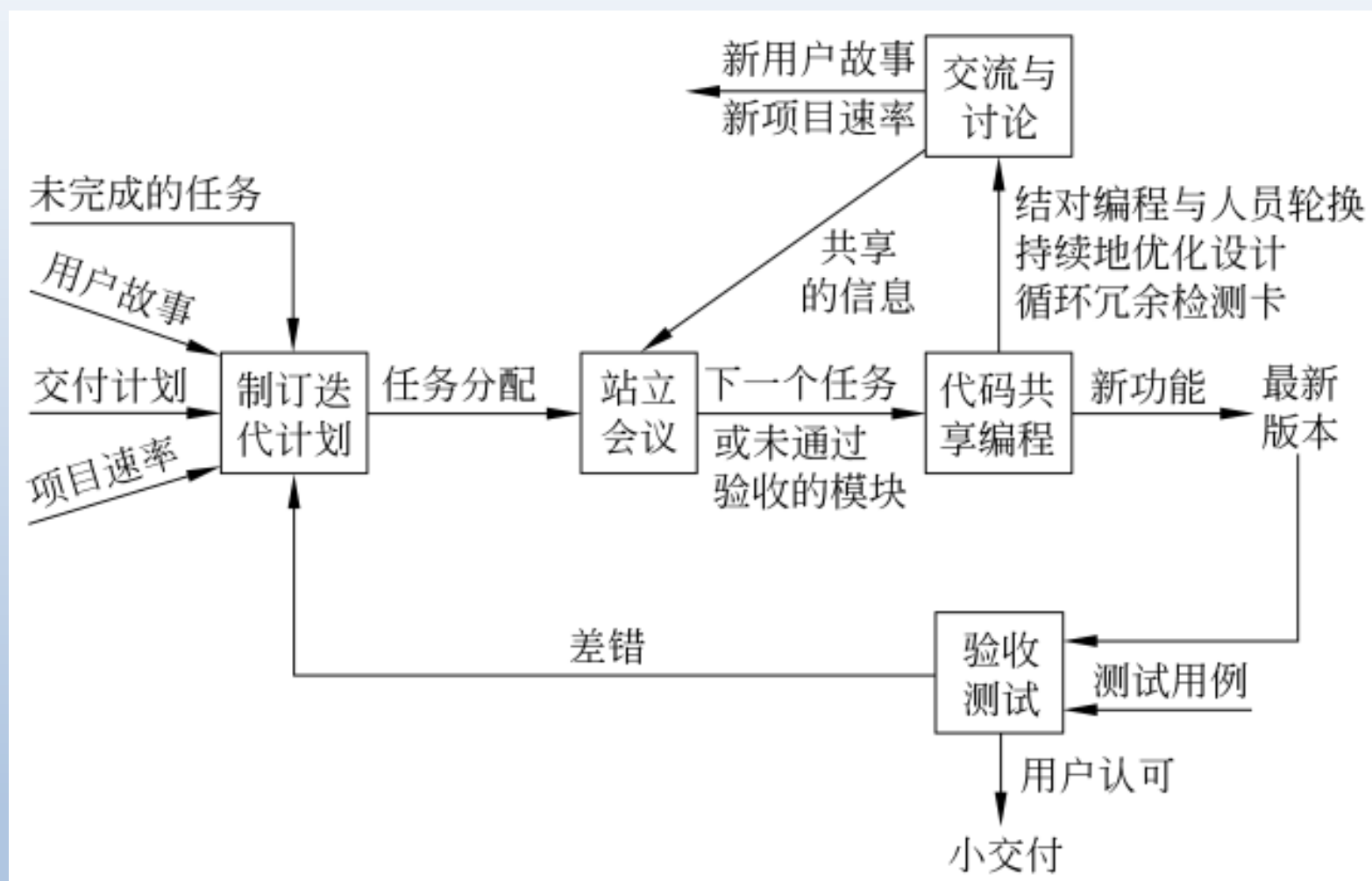


图1. 12

1. 4. 7. 敏捷过程与极限编程

1.4 软件过程

1.4.7.敏捷过程与极限编程

极限编程的迭代过程

江大研究生团队录制，严谨复制

图1.12描述了极限编程的迭代开发过程。项目组根据交付计划和“项目速率”（即实际开发时间和估计时间的比值），选择需要优先完成的用户故事或待消除的差错，将其分解成可在1~2天内完成的任务，制定出本次迭代计划，然后通过每天举行一次“站立会议”（与会人员站着开会以缩短会议时间，提高工作效率），解决遇到的问题，调整迭代计划，会后进行代码共享式的开发工作。所开发出的新功能必须100%通过单元测试，并且立即进行集成，得到的新的可运行版本由客户代表进行验收测试。开发人员与客户代表交流此次代码共享式编程的情况，讨论所发现的问题，提出新的用户故事，算出新的项目速率，并把相关的信息提交给站立会议。

1.4 软件过程

1.4.8. 微软过程

研究生团队录制，严谨复制

适用范围：商业环境下具有有限资源和有限开发时间约束的项目的软件过程模式。

优点：综合了RUP和敏捷过程的许多优点，是对众多成功项目的开发经验的正确总结

缺点：对方法、工具和产品等方面的论述不如RUP和敏捷过程全面，人们对它的某些准则本身也有不同意见。

1.4 软件过程

1.4.8. 微软过程

微软软件生命周期：如下图1.13所示

- 规划阶段
- 设计阶段
- 开发阶段
- 稳定阶段
- 发布阶段

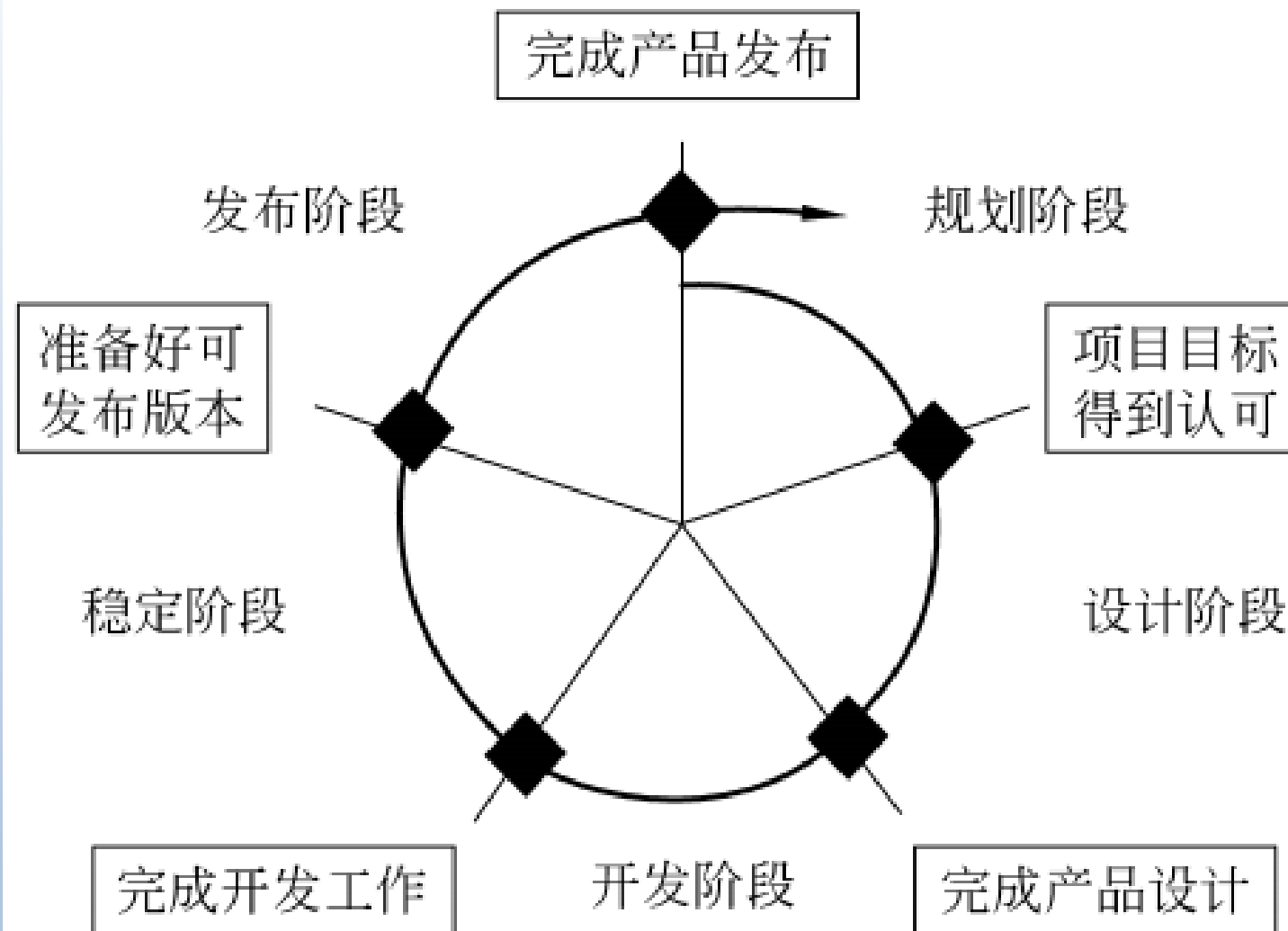


图1.13

1.4.8. 微软过程

1.4 软件过程

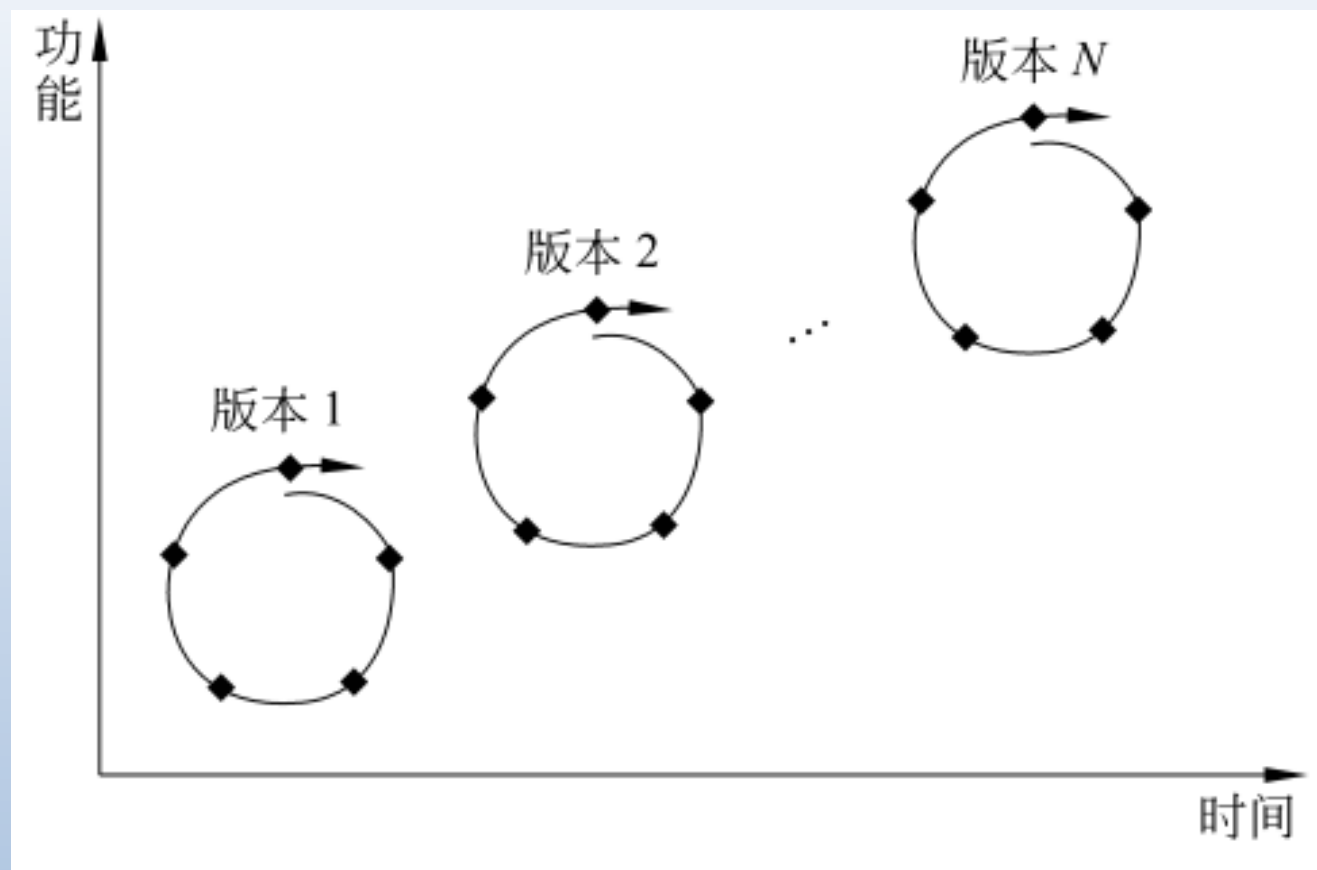
1.4.8. 微软过程

江大研究生团队录制，严谨复制

微软过程模型：

微软过程的每一个生命周期发布一个递进的软件版本，各个生命周期持续、快速地迭代循环。如下图1.14所示

江南大学
！详情加



严谨复制

图1.14

1.4.8. 微软过程

例

根据历史数据可以进行如下的假设。

对计算机存储容量的需求大致按下面公式描述的趋势逐年增加：

$$M=4080e^{0.28(Y-1960)}$$

存储器的价格按下面公式描述的趋势逐年下降：

$$P_1=0.3 \times 0.72^{Y-1974} \text{ (美分/位)}$$

如果计算机字长为 16 位，则存储器价格下降的趋势为：

$$P_2=0.048 \times 0.72^{Y-1974} \text{ (美元/字)}$$

在上列公式中 Y 代表年份，M 是存储容量（字数），P1 和 P2 代表价格。

基于上述假设可以比较计算机硬件和软件成本的变化趋势。要求计算：

（1）在 1985 年对计算机存储容量的需求估计是多少？如果字长为 16 位，这个存储器的价格是多少？

（2）假设在 1985 年一名程序员每天可开发出 10 条指令，程序员的平均工资是每月 4000 美元。如果一条指令为一个字长，计算使存储器装满程序所需要的成本。

（3）假设在 1995 年存储器字长为 32 位，一名程序员每天可开发出 30 条指令，程序员的月平均工资为 6000 美元，重复（1）（2）题。

答：(1) 在 1985 年对计算机存储容量的需求，估计是：

$$M=4080e^{0.28(1985-1960)}=4080e^7=4474263 \text{ (字)}$$

如果字长为 16 位，则这个存储器的价格是：

$$P=0.048*0.72^{(1985-1974)}*4474263=5789 \text{ (美元)}$$

- (2) 如果一条指令的长度为一个字，则使存储器装满程序共需 4474263 条指令。
如果每月有 20 个工作日，则每人每月可开发 200 条指令。需要的工作量是：

$$4474263/200=22371 \text{ (人月)}$$

程序员的月平均工资是 4000 美元，开发出 4474263 条指令的成本是：

$$22371*4000=8948400 \text{ (美元)}$$

- (3) 在 1995 年对存储容量的需求估计为：

$$M=4080e^{0.28(1995-1960)}=4080e^{9.8}=73577679 \text{ (字)}$$

如果字长为 32 位，则这个存储器的价格是：

$$P=0.003*32*0.72^{(1995-1974)}*73577679=7127 \text{ (美元)}$$

如果一条指令的长度为一个字，则使存储器装满程序共需 73577679 条指令。
如果每月有 20 个工作日，则每人每月可开发 600 条指令。需要的工作量是：

$$73577679/600=122629 \text{ (人月)}$$

开发出成本是：

$$122629*6000=735776790 \text{ (美元)}$$

例2

为什么说分阶段的生命周期模型有助于软件项目管理？

答：软件是计算机系统的逻辑部件而不是物理部件，其固有的特点是缺乏可见性，因此，管理和控制软件开发过程相当困难。分阶段的生命周期模型提高了软件项目的可见性。管理者可以把各个阶段任务的完成作为里程碑来对软件开发过程进行管理。把阶段划分得更细就能够跟密切地监控软件项目的进展情况。

江南大学-软工复试课程
! 详情加q1916837466

例3

什么是里程碑？它应该有哪些特征？

答：里程碑是用来说明项目进展情况的事件。通常把一个开发活动的结束或一项开发任务的完成定义为一个里程碑。里程碑必须与软件开发工作的进展情况密切相关，而且里程碑的完成必须非常明显（=里程碑应该有很高的可见性）。

江南大学-软工复试课程——软件工程导论，严谨复制
! 详情加q1916837466

例4

螺旋模型与RUP有哪些相似之处？有何差异？

答：

- 相似之处：都是重复一系列组成系统生命周期的循环，在每次生命周期结束时向用户交付软件产品的一个可运行的版本，每个生命周期由若干次迭代组成，每次迭代都需要风险分析，每次迭代结束时都交付产品的一个增量模型。
- 明显差异：螺旋模型没有规定每次迭代过程结束时所交付的增量原型的具体要求，也未指名不同次迭代过程在经历笛卡儿坐标系中4个象限时所进行的4个方面活动的内容与重点有何不同。**RUP是把产品的整个生命周期划分为4个阶段，并且明确给出了对每个阶段内得若干次迭代过程完成后所交付增量的具体要求，即定义了标志每个阶段结束得主要里程碑。此外，RUP还详细描述了不同阶段的不同迭代过程在经历9个核心工作流程时活动内容得重点和强度有何不同，并且提供了对每次迭代过程中不同核心工作流程活动的并行化支持。RUP的二维迭代生命周期结构对迭代开发方式的体现比螺旋模型更深刻、具体、详细和全面，用于指导需求不明确、不稳定的项目开发，具有更强的可操作性。**

例5

试比较RUP和敏捷过程。

答：敏捷过程是一个一维的迭代过程，过程中每个生命周期交付软件产品的一个可运行的版本，各个生命周期持续地循环；敏捷过程衡量项目进度的首要标准是可以工作的软件。

RUP是一个二维的迭代过程，整个过程由生命周期的若干次循环组成；每个生命周期明确地划分为初始、精化、构建和移交4个阶段，每个阶段由一次或多次迭代完成，每次迭代可能经历9个核心工作流程中的若干个，RUP明确规定了不同阶段的不同迭代过程在经历9个核心工作流程时，工作内容的重点和强度；RUP衡量项目进度的首要标准是各个阶段的主要里程碑。

敏捷过程能适用于.....，RUP提供的是理想开发环境下软件过程的一种完整而且完美的模式，作为软件过程模式来说，敏捷过程远不如RUP全面和完整。

例6

试讨论微软过程与RUP及敏捷过程的关系。

答：相对于RUP，可以把微软过程看做是它的一个精简配置版本。整个微软过程由若干个生命周期得持续递进循环组成，**每个生命周期**划分为5个阶段。微软过程生命周期阶段与RUP生命周期阶段对应关系：**RUP的初始阶段——微软的规划阶段，精化阶段——设计工作，构建阶段——开发和稳定工作，移交阶段——发布工作。**微软的每个阶段精简为由一次迭代完成，每次迭代所完成的工作相当于经历RUP的若干个核心工作流程。

相对于敏捷过程，把微软过程看做是它的一个扩充版本，微软过程补充规定了其每个生命周期内各个阶段的具体工作流程。与敏捷过程类似，微软过程得适用范围也是具有有限资源和有限开发时间约束的项目。