

2023年春季

软件过程管理

黄宏程

重庆邮电大学

通信与信息工程学院



课程的前世今生。。。

1. 软件过程基本概念及分类 4学时

1.1软件过程的基本概念

1.2软件生存周期过程标准

1.3软件支持过程

1.4软件组织过程

1.5几类软件过程间的关系

1.6几类软件过程关系案例

2. 软件过程的管理活动 4学时

2.1需求获取与管理

2.2软件项目规划与监控

2.3软件质量保证

2.4软件配置管理

2.5软件分包管理

2.6软件过程活动案例

3. 软件过程的建模 6学时

3.1软件过程模型的层次

3.2软件过程的基本框架

3.3经典软件过程模型

3.4敏捷开发模型

3.5DevOps

4. 软件过程的定义、评估与改进 4学时

4.1软件过程定义

4.2软件过程评估方法

4.3软件过程改进模型

4.4软件能力成熟度模型

4.5软件过程改进案例

5. 软件度量 4学时

5.1过程度量与产品度量

5.2软件信息模型

5.3定量与定性的过程度量技术

5.4量化的过程管理方法

5.5软件度量与持续改进案例

6. 软件过程工具 4学时

6.1文字与表格工具

6.2软件各领域的基本工具

6.3软件项目管理工具

6.4软件过程管理工具

7. 软件过程实践项目 6学时

7.1软件过程的定义

7.2软件过程度量与评估

1. 过程成绩（50%）

过程成绩由预习作业及堂上测验进行评价。

- （1）预习作业：根据章节知识点内容布置预习作业，评价预习作业完成情况和完成质量；
- （2）堂上互动及实验：根据章节内容设置堂上实验，以及用于评价对知识链的理解掌握情况以及软件过程管理的能力。

2. 项目报告（50%）

由多人（ ≤ 4 ）组成团队完成给定的或自拟的软件过程管理项目，完成项目报告的撰写，根据完成的质量进行评价。

- [1] 朱少民, 左智. 软件过程管理[M]. 北京: 清华大学出版社, 2020.
- [2] 金花, 宁涛. 软件过程管理[M]. 北京: 中国铁道出版社, 2021.
- [3] Watts S. Humphrey. 软件过程管理[M]. 北京: 清华大学出版社, 2003.

1.1 软件生命周期

软件过程是指软件整个生命周期，从需求获取，需求分析，设计，实现，测试，发布和维护一个过程模型。

软件的诞生和生命周期是一个过程，我们总体上称这个过程为软件过程。软件过程是为了开发出软件产品，或者是为了完成软件工程项目而需要完成的有关软件工程的活动，每一项活动又可以分为一系列的工程任务。任何一个软件开发组织，都可以规定自己的软件过程，所有这些过程共同构成了软件过程

过程定义了运用方法的顺序，应该交付的文档资料，为保证软件质量和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。通常，使用生命周期模型简洁地描述软件过程。生命周期模型规定了把生命周期划分为哪些阶段及各个阶段的执行顺序，因此也称为过程模型。

一、软件过程基本概念及分类

软件过程可概括为三类：基本过程类、支持过程类和组织过程类。

- 基本过程类

获取过程、供应过程、开发过程、运作过程，维护过程和管理过程。

- 支持过程类

文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程以及问题解决过程

。

- 组织过程类

基础设施过程、改进过程以及培训过程。

1.2 软件生命周期

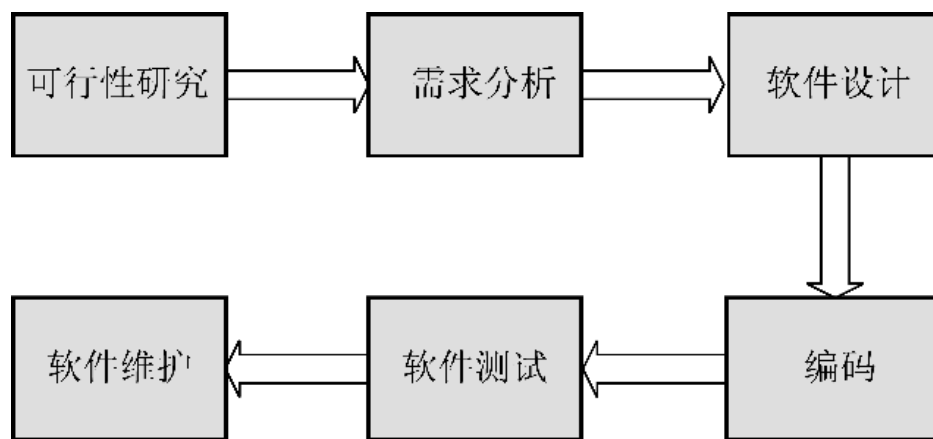
- 2.2.1 软件生命周期的概念

- 软件产品的生命周期是指从设计该产品的构想开始，到软件需求的确定、软件设计、软件实现、产品测试与验收、投入使用以及产品版本的不断更新，到最终该产品被市场淘汰的全过程。
- 软件生命周期这个概念从时间的角度将软件的开发和维护的复杂过程分解为了若干个阶段，每个阶段都完成特定的相对独立的任务。

1.2 软件生命周期

- 2.2.2 传统软件生命周期的各个阶段

— 在传统的软件工程中，软件产品的生命周期一般可以划分为6个阶段，如图所示。



传统的软件生命周期

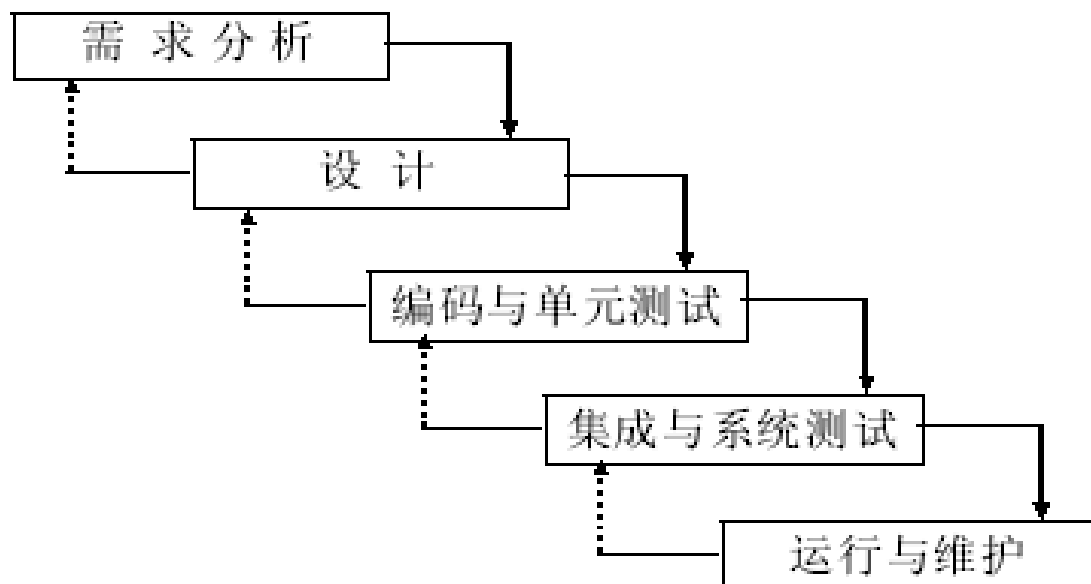
1.3 软件过程模型

在软件工程中，人们通过建立抽象的软件开发模型，把软件生命周期中的各个活动或步骤安排到一个框架中，将软件开发的全过程清晰且直观地表达出来。

常见的软件开发模型有很多种，这里主要介绍瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、基于组件的开发模型、统一软件开发过程模型以及敏捷模型与极限编程。

1.3 软件过程模型

- 1.3.1 瀑布模型



1.3 软件过程模型

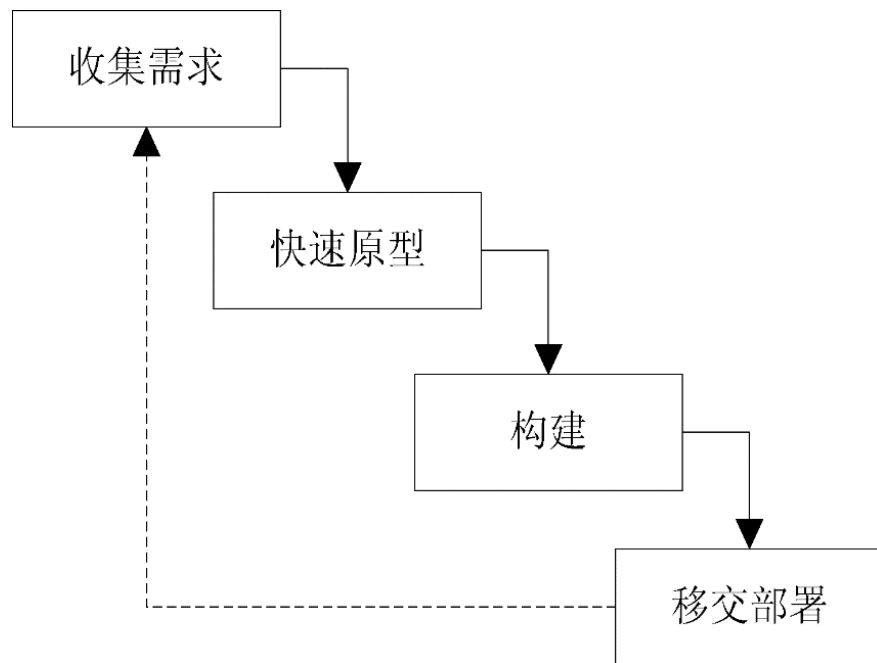
- 1.3.1 瀑布模型

瀑布模型的优点是过程模型简单，执行容易；缺点是无法适应变更。瀑布模型适应于具有以下特征的软件开发项目。

- 在软件开发的过程中，需求不发生或发生很少变化，并且开发人员可以一次性获取到全部需求。否则，由于瀑布模型较差的可回溯性，在后续阶段中需求经常性的变更需要付出高昂的代价。
- 软件开发人员具有丰富的经验，对软件应用领域很熟悉。
- 软件项目的风险较低。瀑布模型不具有完善的风险控制机制

1.3 软件过程模型

- 1.3.2 快速原型模型



1.3 软件过程模型

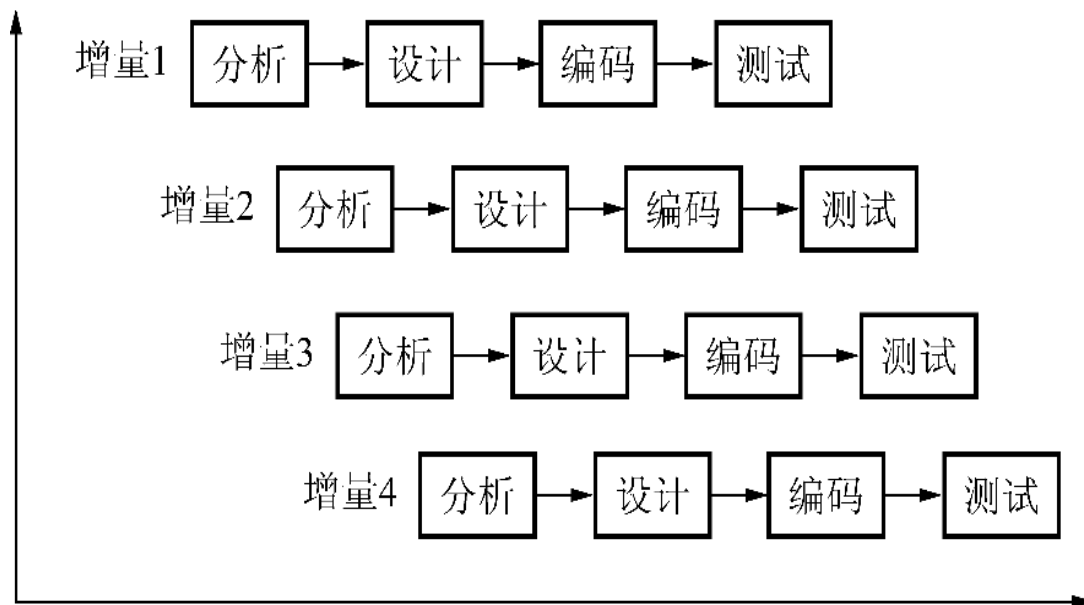
- 1.3.2 快速原型模型

快速原型模型适用于具有以下特征的软件开发项目。

1. 已有产品或产品的原型（样品），只需客户化的工程项目
2. 简单而熟悉的行业或领域
3. 有快速原型开发工具
4. 进行产品移植或升级

1.3 软件过程模型

- 1.3.3 增量模型



1.3 软件过程模型

增量模型的最大特点就是待开发的软件系统模块化和组件化。基于这个特点，增量模型具有以下优点。

- 将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展。
- 以组件为单位进行开发降低了软件开发的風險。一个开发周期内的错误不会影响到整个软件系统。
- 开发顺序灵活。开发人员可以对构件的实现顺序进行优先级排序，先完成需求稳定的核心组件。当组件的优先级发生变化时，还能及时地对实现顺序进行调整。

增量模型的缺点是要要求待开发的软件系统可以被模块化。如果待开发的软件系统很难被模块化，那么将会给增量开发带来很多麻烦。

1.3 软件过程模型

增量模型适用于具有以下特征的软件开发项目。

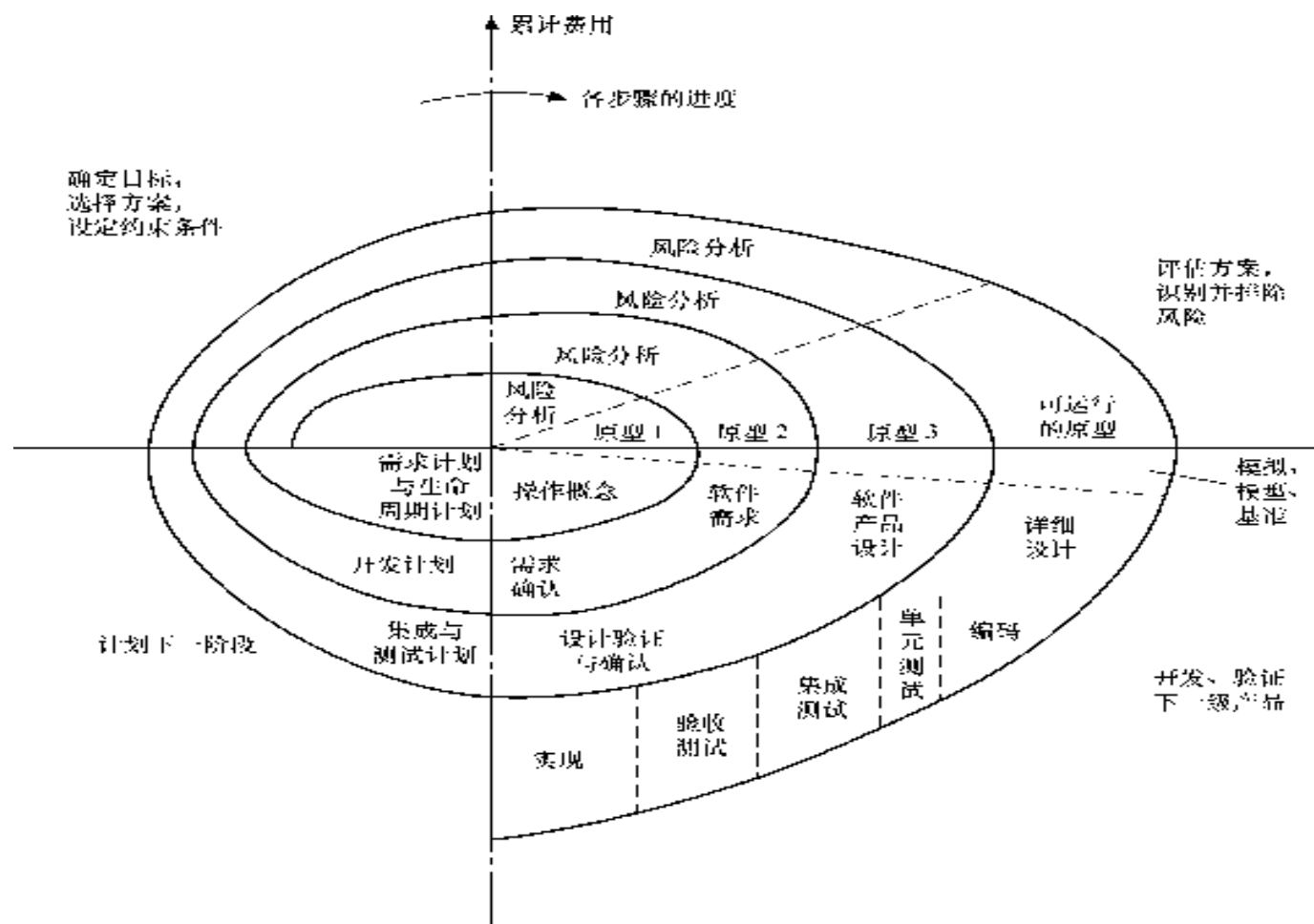
- 软件产品可以分批次地进行交付
- 待开发的软件系统能够被模块化
- 软件开发人员对应用领域不熟悉，难以一次性地进行系统开发
- 项目管理人员把握全局的水平较高

1.3 软件过程模型

- 1.3.4 螺旋模型

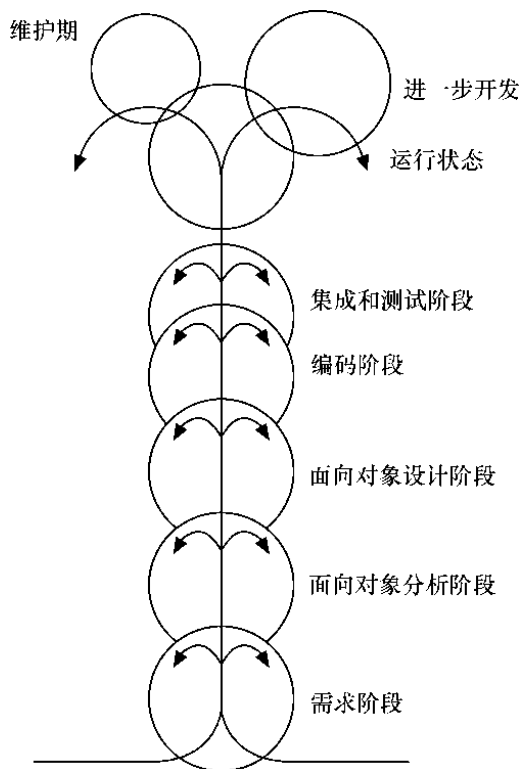
- 螺旋模型是一种用于风险较大的大型软件项目开发的过程模型。该模型将瀑布模型与快速原型模型结合起来，并且加入了这两种模型忽略了的风险分析。它把开发过程分为制定计划、风险分析、实施工程和客户评估4种活动。
- 螺旋模型适应于风险较大的大型软件项目的开发。它的优点是将风险分析扩展到各个阶段中，大幅度降低了软件开发的开发风险。但是这种模型的控制和管理较为复杂，可操作性不强，对项目管理人员的要求较高。

1.3 软件过程模型



1.3 软件过程模型

• 1.3.5 喷泉模型



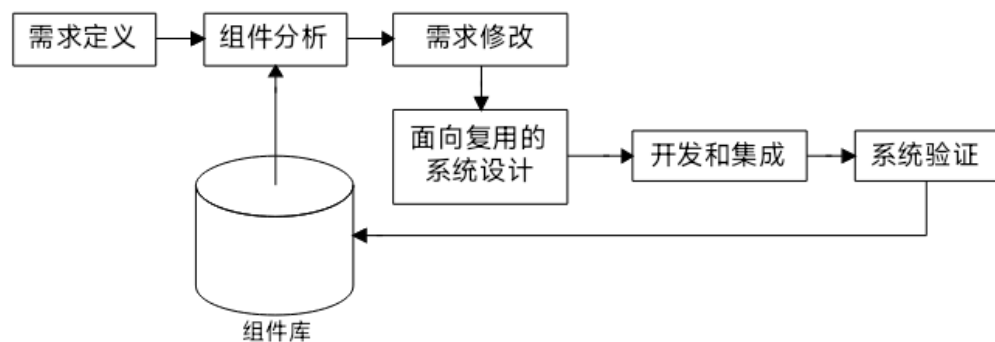
喷泉模型是一种过程模型，同时也支持面向对象开发。在面向对象的方法中，分析模型和设计模型采用相同的符号标示体系，各阶段之间没有明显的界限，而且常常重复、迭代地进行。

“喷泉”一词体现了面向对象方法的迭代和无间隙性。迭代是指各阶段需要多次重复，例如，分析和设计阶段常常需要多次、重复进行，以更好的实现需求。无间隙性是指各个阶段之间没有明显的界限，并常常在时间上互相交叉，并行进行。

喷泉模型主要用于面向对象的软件项目，软件的某个部分通常被重复多次，相关对象在每次迭代中随之加入渐进的软件成分。

1.3 软件过程模型

• 1.3.6 基于组件的开发模型

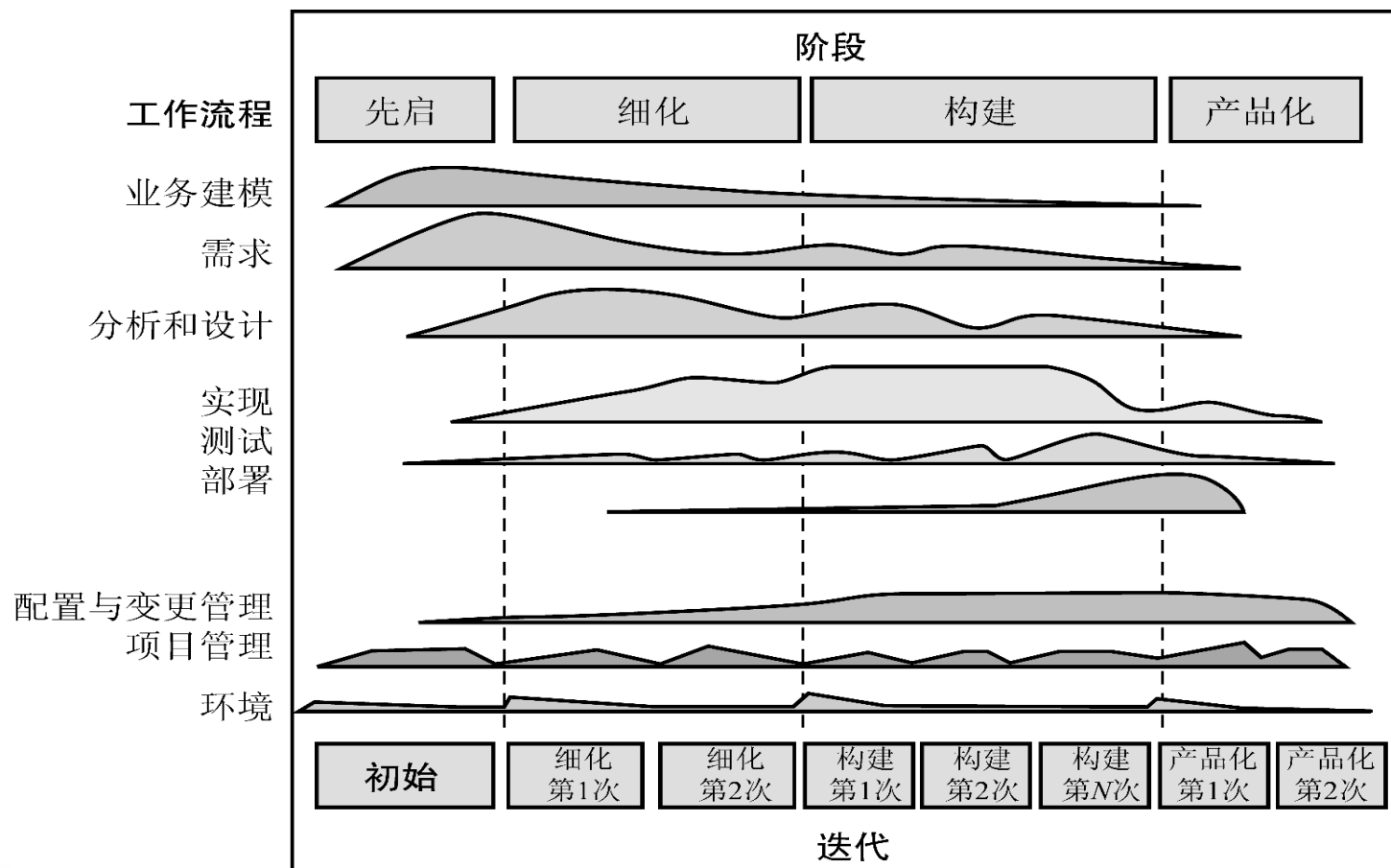


基于组件的开发模型使用现有的组件以及系统框架进行产品开发。在确定需求之后，开发人员开始从现有的组件库中筛选合适的组件，并对组件功能进行分析。在对组件分析之后，开发人员可能适当修改需求来适应现有组件，也可能修改组件或寻找新的组件。组件筛选完成之后，开发人员需要根据需求设计或使用现有的成熟开发框架复用这些组件，一些无法利用现有组件的地方，则需要进行单独的开发，新开发的组件在经历时间考验之后也会加入到组件库中。最后将所有组件集成在一起，进行系统测试。

基于组件的开发模型充分的体现了软件复用的思想，降低了开发成本和风险，并加快了产品开发

1.3 软件过程模型

• 1.3.7 统一软件开发过程模型



1.3 软件过程模型

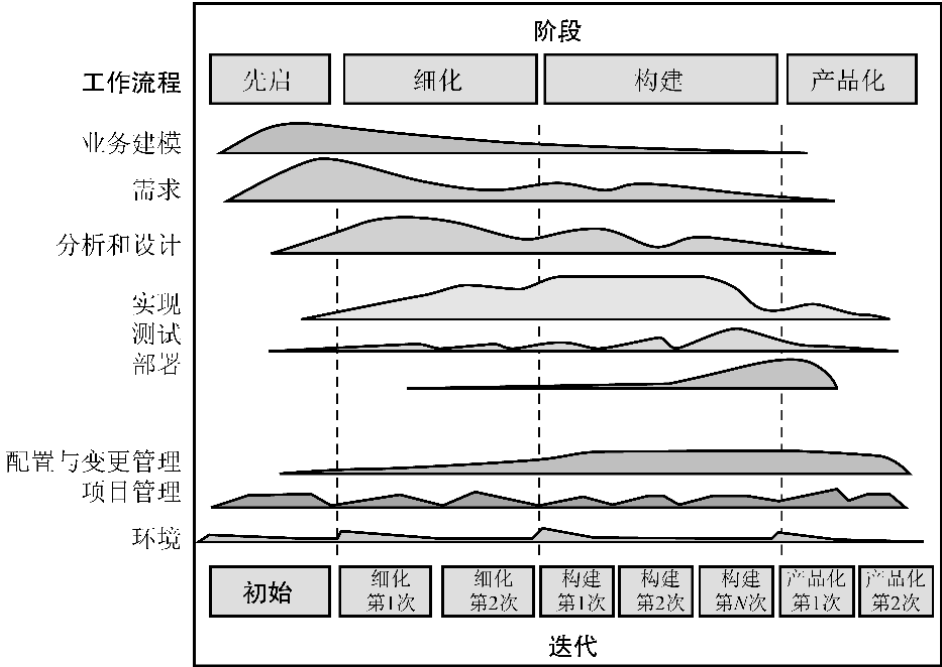


图1 统一软件开发过程模型

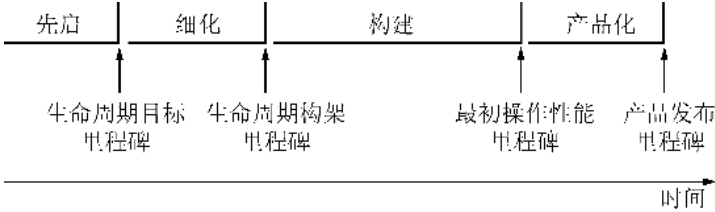


图2 阶段与里程碑的关系

图1中的纵轴以工作的内容为组织方式，表现了软件开发的工作流程。工作流程可以分为核心工作流程和核心支持工作流程。

图1中的横轴以时间为组织方式，表现了软件开发的4个阶段：先启、细化、构建和产品化，每个阶段中都可能包含若干次迭代。这4个阶段按照顺序依次进行，每个阶段结束时都有一个主要里程碑。阶段与里程碑的关系如图2所示。

1.3 软件过程模型

- 统一软件开发过程模型是基于迭代思想的软件开发模型。采用迭代的软件工程思想可以多次执行各个工作流程，有利于更好地理解需求、设计出合理的系统架构，并最终交付一系列渐趋完善的成果。可以说，迭代是一次完整地经过所有工作流程的过程
- 基于统一软件开发过程模型所构造的软件系统，是由软件构件建造而成的。这些软件构件定义了明确的接口，相互连接成整个系统。在构造软件系统时，RUP采用架构优先的策略。软件架构概念包含了系统中最重要静态结构和动态特征，架构体现了系统的总体设计。架构优先开发的原则是RUP开发过程中至关重要的主题。
- 统一软件开发过程模型适用的范围极为广泛，但是对开发人员的素质要求较高。

1.3 软件过程模型

- 1.3.8 敏捷过程与极限编程

1. 敏捷过程概述

- 随着计算机技术的迅猛发展和全球化进程的加快，软件需求常常发生变化，强烈的市场竞争要求更快速的开发软件，同时软件也能够以更快的速度更新。传统的方法在开发时效上时常面临挑战，因此，强调快捷、小文档、轻量级的敏捷开发方法开始流行。敏捷方法是一种轻量级的软件工程方法，相对于传统的软件工程方法，它更强调软件开发过程中各种变化的必然性，通过团队成员之间充分的交流与沟通以及合理的机制来有效地响应变化。

1.3 软件过程模型

敏捷开发开始于“敏捷软件开发宣言”。在2001年2月，17位软件开发方法学家在美国犹他州召开了长达两天的会议，制订并签署了“敏捷软件开发宣言”，该宣言给出了4个价值观。

- (1)个体与交互高于过程和工具
- (2)可运行软件高于详尽的文档
- (3)与客户协作高于合同（契约）谈判
- (4)对变更及时响应高于遵循计划

“敏捷联盟”为了帮助希望使用敏捷方法来进行软件开发的人们定义了12条原则。

1.3 软件过程模型

1. 极限编程

敏捷模型包括多种实践方法，比如

- 极限编程（eXtreme Programming, XP）
- 自适应软件开发（Adaptive Software Development, ASD）
- 动态系统开发方法（Dynamic System Development Method, DSDM）、Scrum
- Crystal
- 特征驱动开发（Feature Driven Development, FDD）等

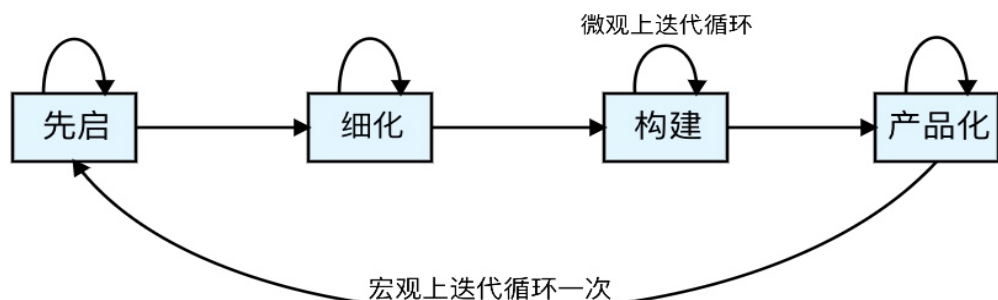
下面介绍极限编程的相关内容

1.3 软件过程模型

- 极限编程是一种实践性较强的规范化的软件开发方法，它强调用户需求和团队工作。利用极限编程方法进行软件开发实践的工程师，即使在开发周期的末期，也可以很快地响应用户需求。在团队工作中，项目经理、用户以及开发人员都有责任为提高软件产品的质量而努力。XP特别适用于软件需求模糊且容易改变、开发团队人数少于10人、开发地点集中（比如一个办公室）的场合。
- 极限编程包含了一组相互作用和相互影响的规则和实践。在项目计划阶段，需要建立合理和简洁的用户故事。在设计系统的体系架构时，可以采用CRC（Class, Responsibility, Collaboration）卡促使团队成员共同努力。代码的质量在极限编程项目中非常重要。为了保证代码的质量，可以采用结对编程以及在编码之前构造测试用例等措施。在测试方面，开发人员有责任向用户证明代码的正确性，而不是由用户来查找代码的缺陷。合理的测试用例及较高的测试覆盖率是极限编程项目测试所追求的目标。

1.3 软件过程模型

• 1.3.9 几种模型之间的关系



1. 瀑布模型与RUP模型之间的关系

在宏观上，瀑布模型是静态模型，RUP模型是动态模型。RUP模型的每一次迭代，实际上都需要执行一次瀑布模型，都要经历先启、细化、构建、产品化这4个阶段，完成瀑布模型的整个过程。

在微观上，瀑布模型与RUP模型都是动态模型。瀑布模型与RUP模型在每一个开发阶段（先启、细化、构建、产品化）的内部，都需要有一个小小的迭代过程，只有进行这样的迭代，开发阶段才能做得更好。

瀑布模型中有RUP模型，反过来，RUP模型中也有瀑布模型。

1.3 软件过程模型

1. 瀑布模型与增量模型之间的关系

- 增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，一个模块接着一个模块地进行开发，直到开发完所有的模块。
- 在开发每个模块时，通常都是采用瀑布模型，从分析、设计、编码和测试这几个阶段进行开发。所以，增量模型中有瀑布模型，即宏观上是增量模型，微观上是瀑布模型。
- 增量模型也体现了迭代思想，每增加一个模块，就进行一次迭代，执行一次瀑布模型，所以，增量模型本质上是迭代的。

1.3 软件过程模型

1. 瀑布模型与快速原型模型之间的关系

- 快速原型的基本思想是快速建立一个能反映用户主要需求的原型系统，在此基础上之后的每一次迭代，都可能会用到瀑布模型。
- 快速原型模型中不但包含了迭代模型的思想，而且包含了瀑布模型的思想。

1.3 软件过程模型

2. 瀑布模型与螺旋模型之间的关系

- 螺旋模型是瀑布模型和快速原型模型的结合，快速原型模型是原型模型的简化，原型模型又是迭代模型和瀑布模型的组合，这些模型之间是相互依存的、彼此有关的。
- 螺旋模型每一次顺时针方向旋转，相当于顺时针方向迭代一次，都是走完一次瀑布模型，这就是瀑布模型与螺旋模型之间的关系。实际上，瀑布模型与喷泉模型也有关系。

1.3 软件过程模型

• 1.3.10 选择软件过程模型

各种软件过程模型反映了软件生命周期表现形式的多样性。在生命周期的不同阶段也可采用不同的软件过程模型。在具体的软件开发过程中，可以选择某种软件过程模型，按照某种开发方法，使用相应的工具进行软件开发。

在选择软件过程模型时需要考虑以下几点。

1. 符合软件自身的特性，如规模、成本和复杂性等
2. 满足软件开发进度的要求
3. 对软件开发的进行进行预防和控制
4. 具有计算机辅助工具的支持
5. 与用户和软件开发人员的知识和技能相匹配
6. 有利于软件开发的和管理和控制

1.3 软件过程模型

一般来说，结构化方法和面向数据结构方法可采用瀑布模型或增量模型进行软件开发；而面向对象方法可采用快速原型模型、喷泉模型或RUP模型进行软件开发。

在实际的软件开发过程中，选择软件过程模型并非是一成不变的，有时还需要针对具体的目标要求进行裁剪、修改等，从而构成完全适合开发目标要求的软件过程模型。

现实中的软件系统有各种各样，软件开发方式也千差万别。对同一个问题，不同的开发组织可能选择不同的开发模型（过程模型）去解决，开发出的软件系统也不可能完全一样，但是其基本目标都是一致的，即应该满足用户的基本功能需求，否则，再好的软件系统也是没有意义的。

◎ 关于软件过程管理的理解

根据堂上的内容及课下查阅资料，基于当下认知，完成1000字左右的关于软件过程管理的理解，以及希望了解的内容。

◎ 分组、策划题目

完成项目小组的分组以及初步确定小组的项目内容；

≤4人/小组，策划一个小型的项目。