统一软件过程

主讲：华俊昌

目录

CONTENT

01

Part one

软件工程概念

什么是软件?

软件的概念

- 软件的定义软件由计算机程序、数据及文档组成。
- 软件与硬件、数据库、人、过程等共同构成计算机系统。
- 软件按功能分类：应用软件、系统软件、支撑软件。
- 软件的发展经历了三个阶段：程序设计阶段、程序系统阶段、软件工程阶段。
- 软件工程概念的出现源自软件危机。
- 软件危机的主要特征
  - 软件价格在整个项目投入中的比例不断升高;
  - 软件开发成本严重超标;
  - 软件开发周期大大超过规定日期;
  - 软件质量难于保证;
  - 软件修改、维护困难;
  - 失败的根本原因在于：开发人员写出的东西达不到用户要求 (人的问题、技术问题)

软件工程的定义

软件工程的定义

1968年

德国人 Bauer 在北大西洋公约组织会议上的定义：“建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法”

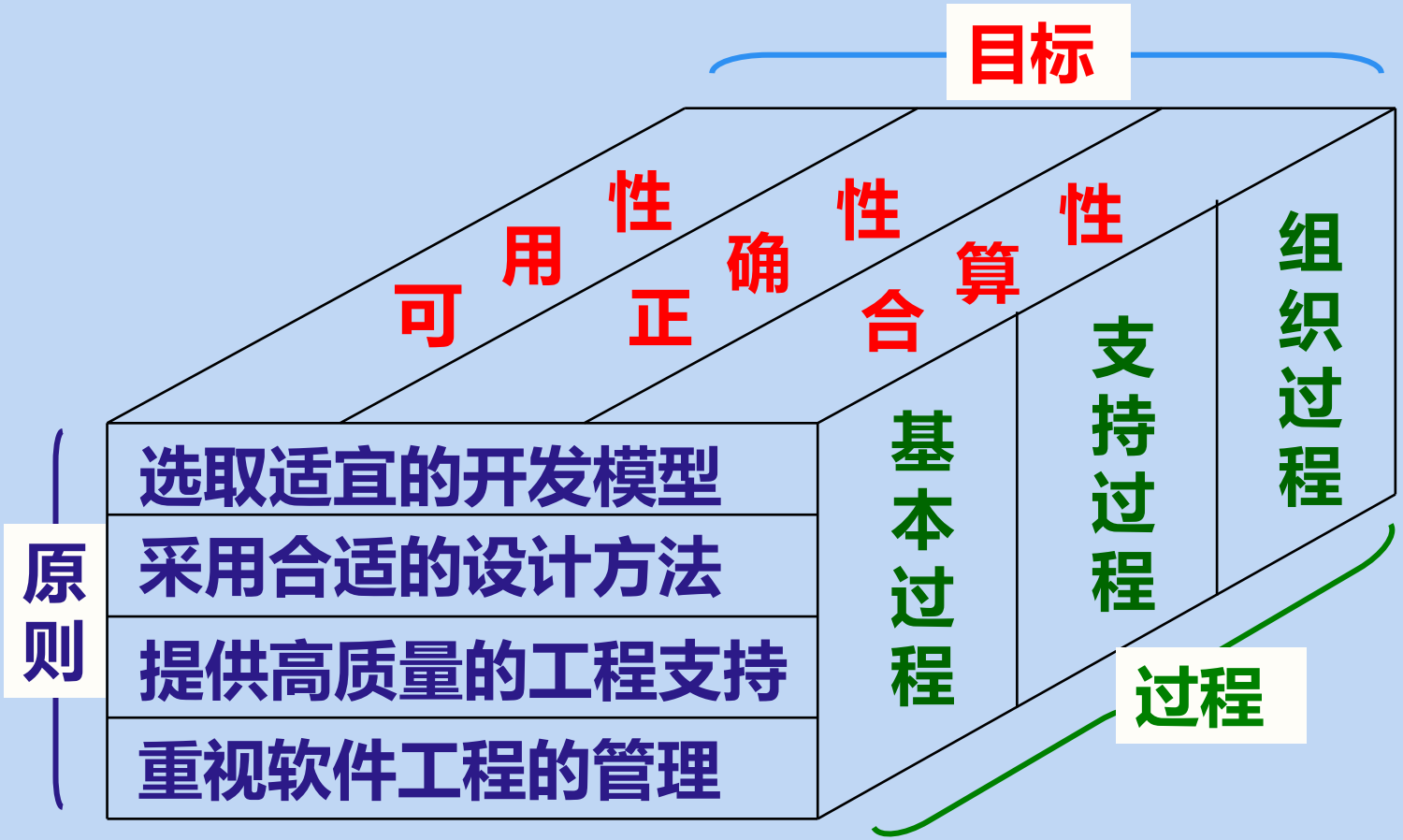
1983年

IEEE(美国电气和电子工程师协会)的软件工程定义：“软件工程是开发，运行 ，维护和修复软件的系统方法”。

1993年

IEEE 的一个更加综合的定义：“将系统化的，规范的，可度量的方法应用于软件的开发，运行和维护的过程，即将工程化应用于软件中”。

软件工程的框架


目标
可用性
正确性
合算性
组织过程
支持过程
基本过程
过程
原则
选取适宜的开发模型
采用合适的设计方法
提供高质量的工程支持
重视软件工程的管理

02

Part two

软件生存周期与常用模型

软件生存周期的定义

软件生存周期的定义

软件生存周期概念

- 软件生存周期模型是软件工程思想的具体化，是跨越软件生存周期的系统开发、运行、维护所实施的全部活动和任务的过程框架。
- 软件生存周期包含三个阶段：软件定义、软件开发及软件运行维护。
- 常用的软件生存周期模型有瀑布模型，演化模型，螺旋模型，增量模型，喷泉模型，演化开发模型，敏捷模型。

瀑布模型

- 各项活动按自上而下，相互衔接的固定次序，如同瀑布逐级下落，每项活动均处于一个质量环（输入-处理-输出-评审）中。
- 阶段间具有顺序性和依赖性。
- 每个阶段必须完成规定的文档；每个阶段结束前完成文档审查。
- 核心思想是采用“**工序**”的思想将复杂的软件开发问题简化。将软件设计与软件功能的实现分开由不同的阶段完成，便于分工协作。采用结构化的分析与设计方法，将逻辑实现与物理实现分开。

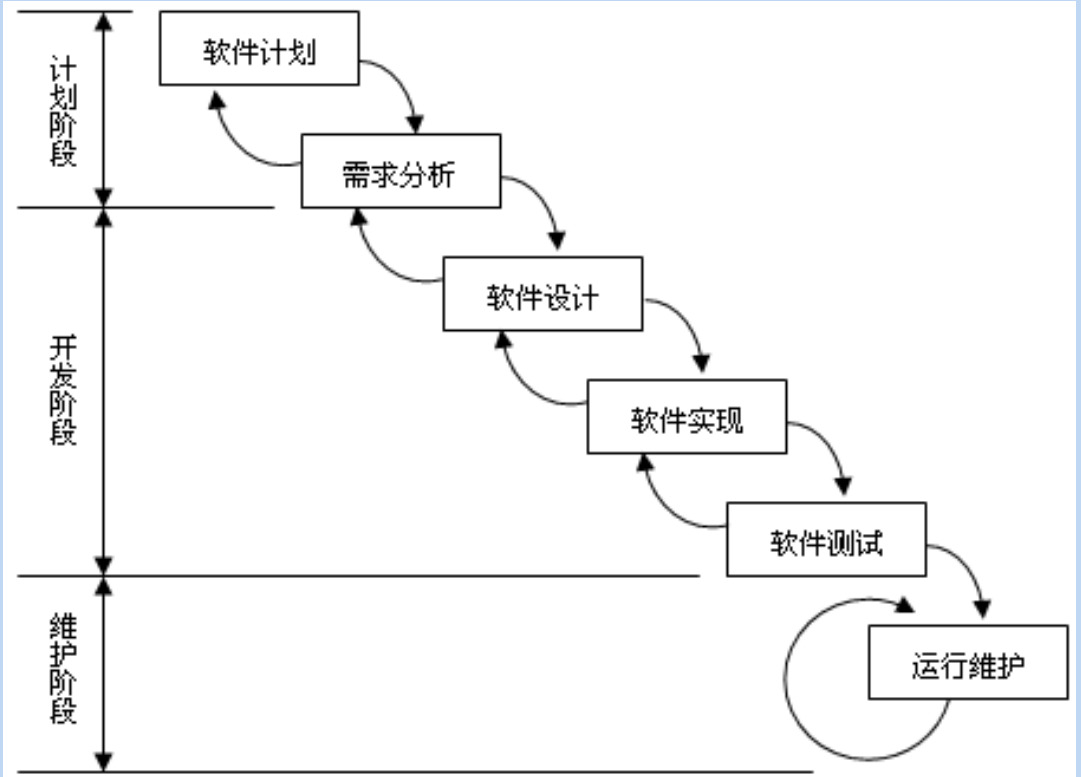


图 2-2 瀑布模型

瀑布模型

主要特点:

（1）上一阶段活动的成果作为本阶段活动的输入；

（2）得到本阶段活动的工作成果，再传递传给下一阶段活动；

（3）在每一阶段活动结束时，对阶段活动的工作成果进行评审，如果工作成果得到确认，则继续下一项活动，否则返回前一阶段，甚至更前阶段的活动进行返工。

主要缺点:

（1）由于开发模型呈线性，所以当开发成果尚未经过测试时，用户无法看到软件的效果。这样，软件与用户见面的时间间隔较长，增加了一定的风险。

（2）重新进行分析和测试会导致成本增加和时间延迟。

（3）在软件开发前期未发现的错误传到后面的开发活动中时，可能会扩散，进而可能会造成整个软件项目开发失败。

（4）在软件需求分析阶段，要完全确定用户的所有需求是比较困难的，甚至可以说是不太可能的。

适用项目类型:

瀑布模型主要适合于需求明确，且无大的需求变更的软件开发，例如，编译系统、操作系统等

增量模型

- 融合了瀑布模型的基本成分和原型实现的**迭代**特征。采用随着时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“**增量**”。
- 当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。

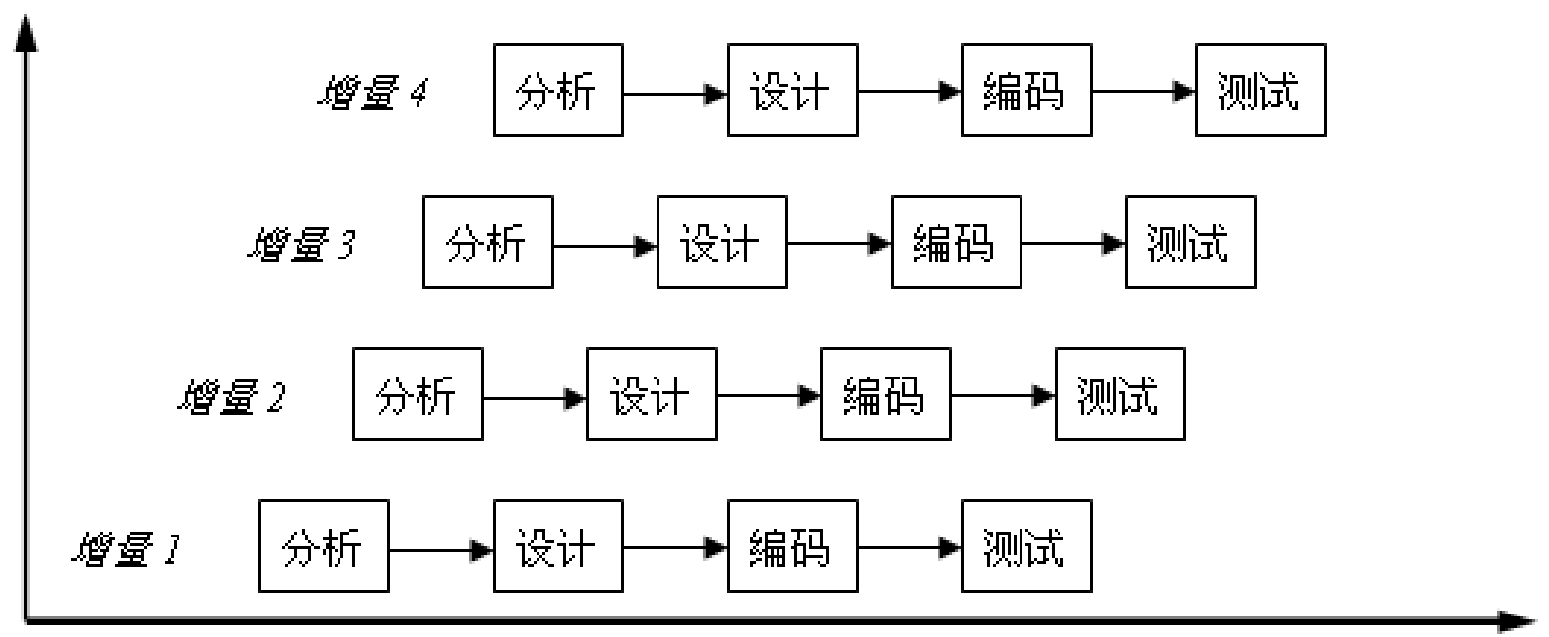


图 2-5 增量模型

增量模型

主要特点:

（1）人员分配灵活，刚开始不用投入大量人力资源，如果核心产品很受欢迎，则可增加人力实现下一个增量 。

（2）当配备的人员不能在设定的期限内完成产品时，它提供了一种先推出核心产品的途径，这样就可以先发布部分功能给客户，以满足客户的局部需求 。

（3）随着高优先级功能被首先开发，项目整体风险降低了，能够有计划地管理技术风险

主要缺点:

（1）每个增量后没法做完整的系统测试。

（2）这种以客户交付增量为基础的软件变化频繁，因此软件没有很好的架构。

（3）如果增量包之间存在相交的情况且不能很好地处理，就必须做全盘的系统分析。

适用项目类型:

增量模式是一种能够快速构造可运行产品的方法，也适用于市场竞争激烈，需要快速发布产品项目。

演化模型

演化模型

- 这是一种增量式、迭代式的模型，而且还强调了其他模型均忽略了的**风险分析和评价**。
- 演化模型包括螺旋模型、原型模型、并型开发模型

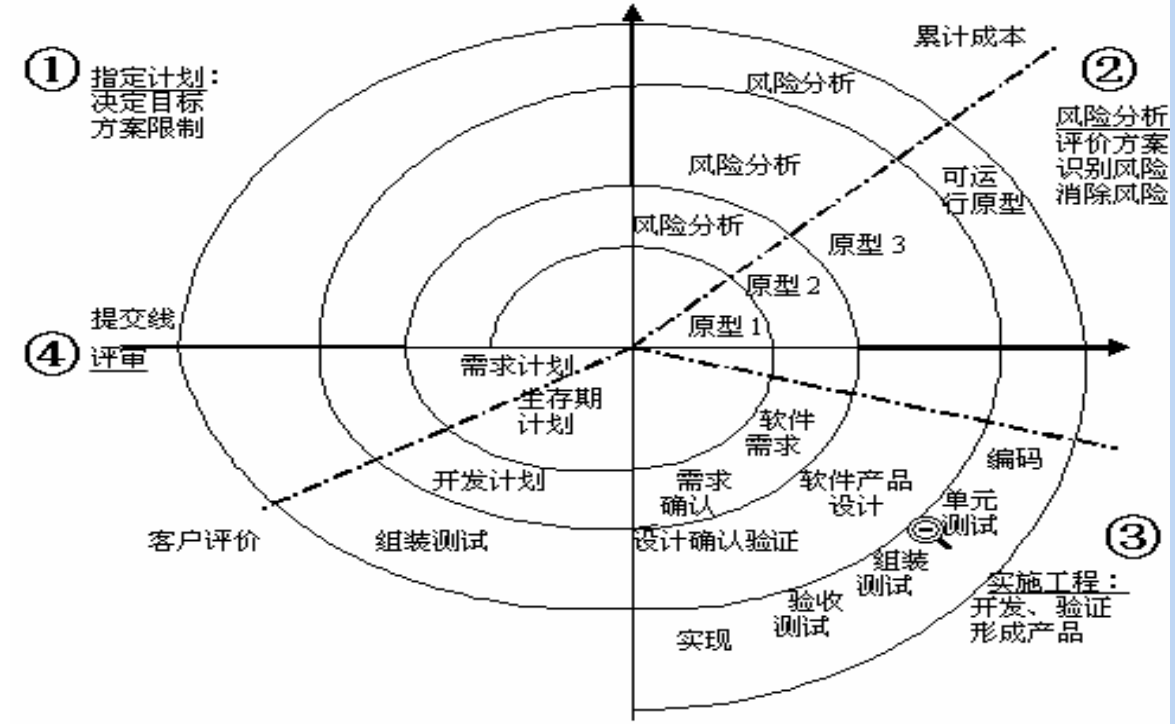


图 2-3 螺旋模型

螺旋模型

螺旋模型是增加了风险分析和规避措施的“原型”+“瀑布”的迭代开发模型，它沿着螺线旋转，自内向外每旋转一圈便开发出更完善的一个新版本。

① 制定计划
② 风险分析
③ 实施工程
④ 客户评估

演化模型

主要特点：

（1）在“瀑布模型”的每一个开发阶段前，引入一个非常严格的风险识别、风险分析和风险控制。它把软件项目分解成一个个小项目，每个小项目都标识一个或多个主要风险，直到所有的主要风险因素都被确定。

（2）螺旋模型支持用户需求的动态变化，为用户参与软件开发的所有关键决策提供了方便，有助于提高目标软件的适应能力，为项目管理人员及时调整管理决策提供了便利，从而降低了软件开发风险。

主要缺点：

（1）过多的迭代次数会增加开发成本，延迟提交时间。

（2）采用螺旋模型，需要具有相当丰富的风险评估经验和专门知识。在风险较大的项目开发中，如果未能够及时标识风险，势必造成重大损失。

适用项目类型：

演化模式适用需要进行适当风险分析和间歇式软件交付的项目。如风险很高的硬实时应用程序或者高风险概率的软实时系统项目。

敏捷模型

- 一种**轻量级**的开发方法，”敏捷过程模型”是指基于迭代开发的软件开发方法。敏捷方法将任务分解为较小的迭代，或者部分不直接涉及长期计划。在开发过程的开始就确定了项目范围和要求。事先明确定义了有关迭代次数，每次迭代的持续时间和范围的计划。敏捷方法强调**适应性**而非预测性，强调以**人为中心**而非以流程为中心，强调对变化的适应和对人性的关注。

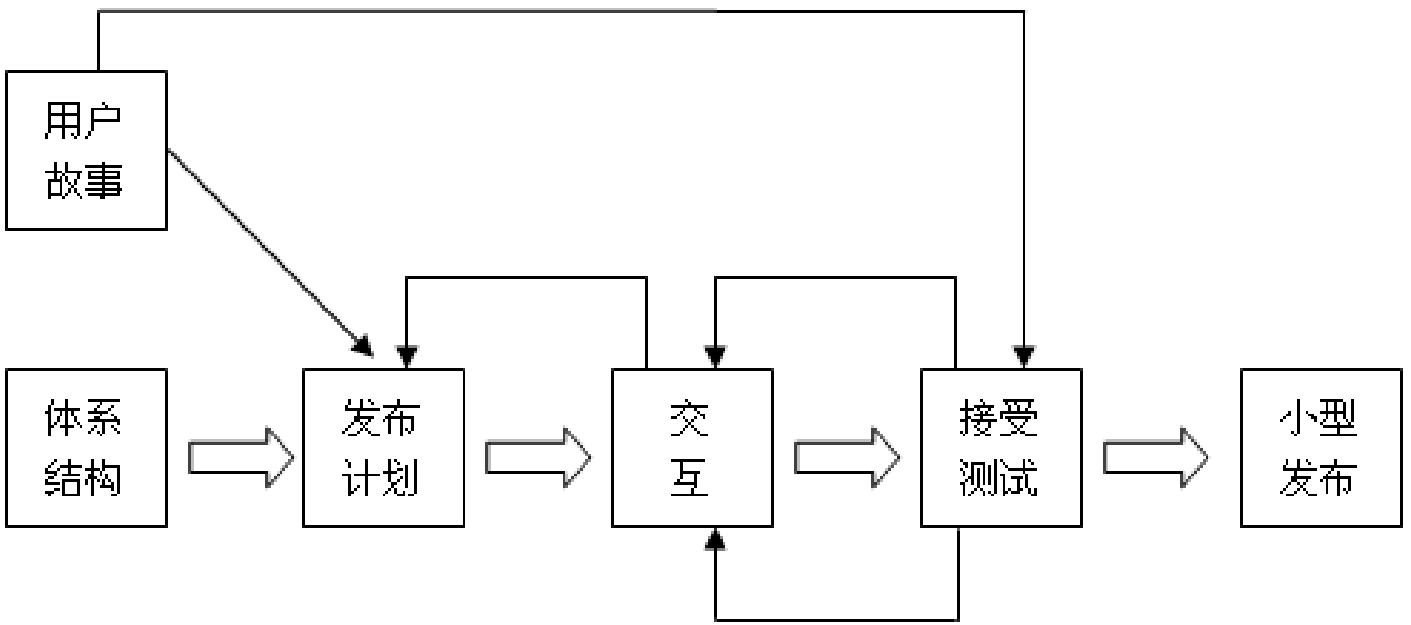


图 2-8 XP/敏捷方法

敏捷模型

主要特点:

（1）它的核心思想是**交流、简单、反馈和进取，轻文档**。敏捷模型强调小组内成员之间要经常进行交流；在尽量保证质量可以运行的前提下，力求过程和代码的简单化；在开发小组中不仅包括开发人员，还包括管理人员和客户，来自客户、开发人员和最终用户的具体反馈意见可以提供更多的机会来调整设计，保证把握正确的开发方向

（2）快递迭代

（3）让测试人员和开发人员参与需求讨论

主要缺点:

（1）不强调文档，不断变化的客户需求会导致频繁的更改。因此不适应大型项目

（2）让客户、开发人员和测试人员一起参与用户故事代价昂贵

适用项目类型:

软件需求经常变化或者需求变化比较大； 项目团队与用户之间进行沟通比较容易； 项目的开发风险比较高； 规模比较小，一般项目组成员在50 人之内；项目团队的成员能力比较强，而且具有责任感； 项目的可测试性比较好。如社交网站等。

思考题

练习题

1.假设要开发一个软件，该软件的功能对特定项目进行一项验证计算（假设计算方法十分确定、成熟），一旦实现后将用于该项目的测试验证中，由于项目特殊性，所以该软件产品在完成使命后将被抛弃。

练习题

2.假设项目组正在开发已经被广泛使用的字处理软件的新版本，由于市场竞争激励，公司规定了严格的完成期限。

练习题

3.假设项目组正在参与银行的ERP大型项目，项目过程中风险因素比较多

03

Part three

统一软件过程RUP

统一软件过程RUP

- 统一软件过程（RUP，Rational Unified Process）是基于面向对象统一建模语言UML的一种面向对象的软件过程模型。
- RUP遵循了逐步求精的、迭代的开发策略。RUP是以用例为驱动，以系统架构为中心的一个迭代式的增量过程。
- RUP分为初始、细化、构造和移交四个阶段，每个阶段又分为若干次迭代，每次迭代都经过一个核心工作流程。

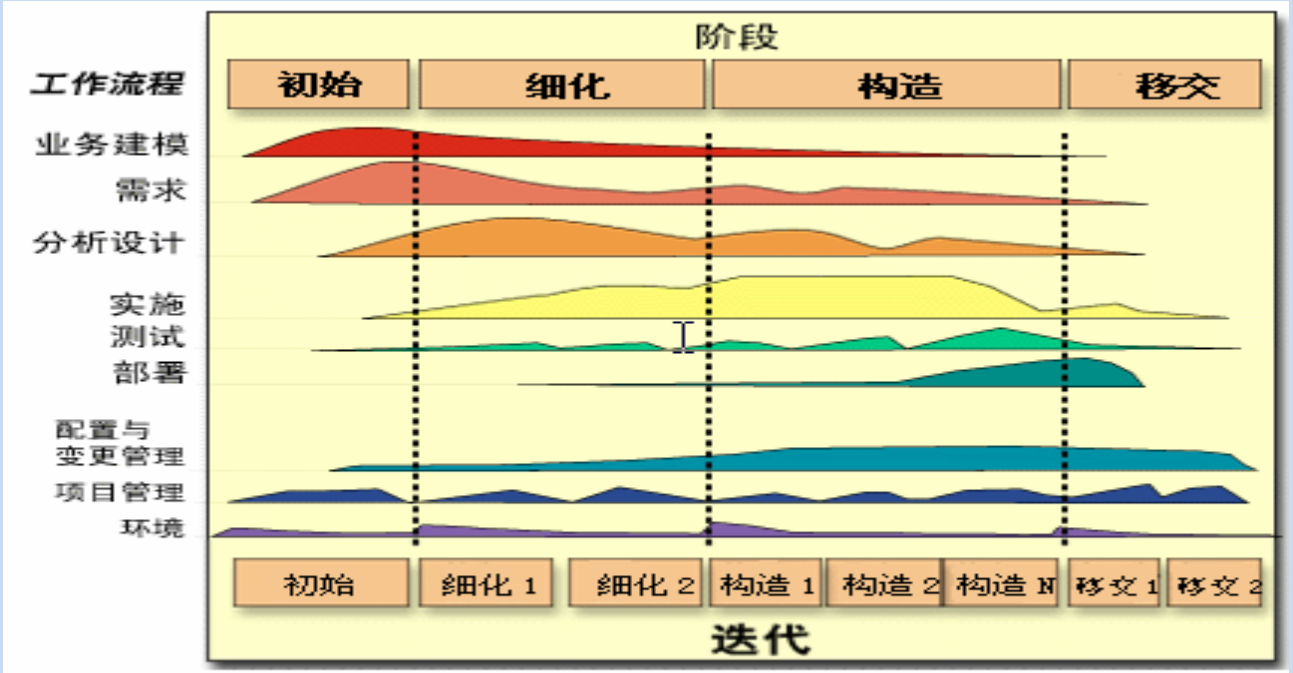


图 2-9 RUP 二维空间

统一软件过程RUP

统一过程中，有6个核心工作流

- 业务建模工作流：用商业用例为商业过程建立文档。
- 需求工作流：目标是描述系统应该做什么，确保开发人员构建正确的系统。为此，需明确系统的功能需求和非功能需求（约束）。
- 分析和设计工作流：其目标是说明如何做。结果是分析模型和设计模型。
- 实现工作流：用分层的方式组织代码的结构，用构建的形式来实现类，对构件进行单元测试，将构件集成到可执行的系统中。
- 测试工作流：验证对象直接的交互、是否所有的构件都集成了、是否正确实现了所有需求、查错并能改正。
- 部署工作流：制作软件的外部版本、软件打包、分发、为用户提供帮助和支持。

04

Part four

统一软件模型的生命周期阶段

RUP初始阶段

RUP初始阶段：确定项目是否可行

- 明确项目规模，了解环境已及最重要需求和约束
- 划分主要的子系统，给出系统的体系结构概貌
- 考虑时间、经费、人员、技术、项目计划和效益等因素
- 分析项目执行的风险

该阶段的焦点是需求和分析工作流

RUP细化阶段

RUP细化阶段：识别出大部分用例（80%），建立健全的体系结构集成，编制项目计划，细化风险评估。

- 用例模型需要完成80%
- 创建软件结构的描述性文档
- 创建可执行的系统原型
- 细化风险列表
- 创建建立整个项目的开发计划

该阶段的焦点是商业建模和需求工作流

RUP构造阶段

RUP构造阶段：每一次迭代开发都进行分析、设计、编码、测试和集成过程，最终得到客户需求的产品。

- 优化资源，使开发成本降到最低
- 尽快达到质量要求
- 尽快完成有用的版本
- 完成所有功能的分析、开发、测试
- 迭代式、递增地开发随时可以发布的产品
- 确定准备好软件系统的外部环境

该阶段的焦点是实现工作流

RUP交付阶段

RUP交付阶段：完成最后的软件产品和产品验收测试，并编制用户文档，进行用户培训工作。将完整的系统部署到用户所处环境，确保软件对最终用户是可用的。

- 按用户的要求验证新系统
- 替换旧的系统
- 对用户和维护人员进行培训
- 与用户达成共识，部署基线与评估标准一致

该阶段的焦点是测试和部署工作流

感谢您的观看