

第07讲 RUP统一过程

SUST

2022.04

目录

1

4.1 RUP产生

2

4.2 基于统一过程的UML系统建模

3

4.3 二维开发模型

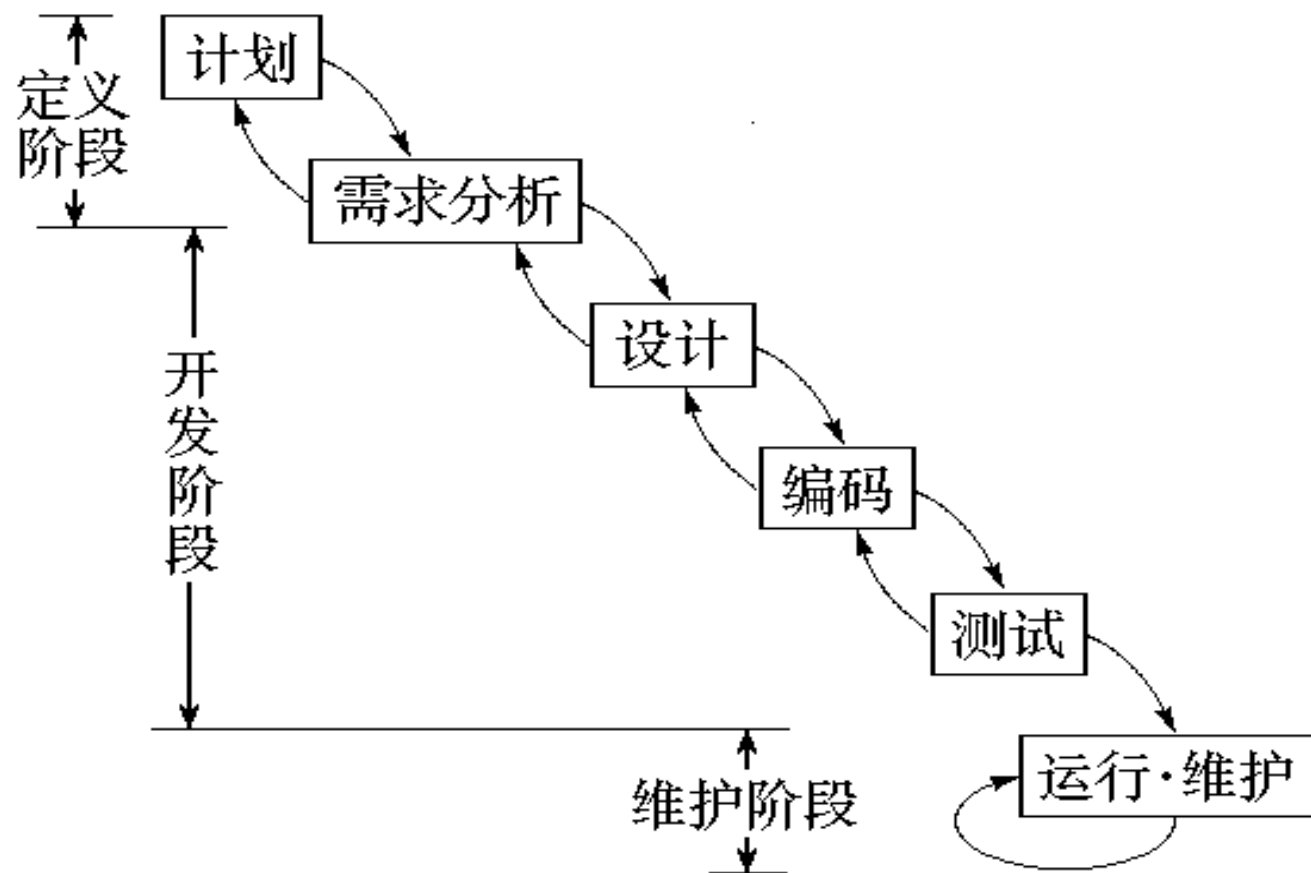
4

4.4 RUP开发过程
4.5 RUP核心 workflow
4.6 RUP十大要素和经验

- 牛顿迭代法 开根号
- 求n的平方根，先假设一猜测值 $x_0 = 1$ ，然后根据以下公式求出 x_1 ，再将 x_1 代入公式右边，继续求出 x_2 ...通过有效次迭代后即可求出n的平方根， x_{k+1}
- (迭代公式)

$$x_{k+1} = \frac{1}{2} \left(x_k + \frac{n}{x_k} \right),$$

- 关于软件开发模型有多种。大概有两类：
- 线性的:瀑布模型、原型模型
-
- 迭代的:螺旋模型、喷泉模型、进化树模型、迭代增量模型
- RUP就是迭代的模型。统一软件过程RUP是由Rational公司的Grady Booch、James Rumbaugh以及IvarJacobson联合制定的一种软件开发过程。



软件开发模式

- 瀑布模型：（从本质来讲，瀑布模型是一个软件开发架构,重复应用）

（核心思想：按工序将问题化简，将功能的实现与设计分开，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，依照软件生命周期自上而下，相互衔接的次序<如同瀑布流水逐级下落>）

软件开发模式

- 快速原型模型：（需要迅速造一个可以运行的软件原型，以便理解和澄清问题）

快速原型模型允许在需求分析阶段对软件的需求进行初步的非完全的分析 and 定义，快速设计开发出软件系统的原型（展示待开发软件的全部或部分功能和性能

（过程：用户对该原型进行测试评定，给出具体改善的意见以及丰富的细化软件需求，开发人员进行修改完善）

软件开发模式

- 优点：

克服瀑布模型的缺点，减少由于软件需求不明确带来的开发风险

缺点：

A、所选用的开发技术和工具不一定符合主流的发展

B、快速建立起来的系统加上连续的修改可能会造成产品质量低下

软件开发模式

- 增量模型：（采用随着日程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”，第一个增量往往就是核心的产品）
 - 与其他模型共同之处：它与原型实现模型和其他演化方法一样，本质都是迭代
 - 与原型实现模型不同之处：它强调每一个增量均发布一个可操作产品，（它不需要等到所有需求都出来，只要某个需求的增量包出来即可进行开发）

软件开发模式

- 优点：

- 1、 人员分配灵活，一开始不需要投入大量人力资源
- 2、 当配备人员不能在限定的时间内完成产品时，它可以提供一种先推出核心产品的途径，可现发布部分功能给用户（对用户起镇静作用）
- 3、 增量能够有计划的管理技术风险

- 缺点：

- 1、 如果增量包之间存在相交的情况且未很好处理，则必须做全盘系统分析

软件开发模式

- 原型模型：（样品模型，采用逐步求精的方法完善原型）

主要思想：

先借用已有系统作为原型模型，通过“样品”不断改进，使得最后的产品就是用户所需要的。原型模型通过向用户提供原型获取用户的反馈，使开发出的软件能够真正反映用户的需求，

软件开发模式

- 采用方法：

原型模型采用逐步求精的方法完善原型，使得原型能够“快速”开发，避免了像瀑布模型一样在冗长的开发过程中难以对用户的反馈作出快速的响应

软件开发模式

- 优点：

(1) 开发人员和用户在“原型”上达成一致。这样一来，可以减少设计中的错误和开发中的风险，也减少了对用户培训的时间，而提高了系统的实用、正确性以及用户的满意程度。

(2) 缩短了开发周期，加快了工程进度。

(3) 降低成本。

软件开发模式

- 缺点：
 - 1、当重新生产该产品时，难以让用户接收，给工程继续开展带来不利因素。
 - 2、不宜利用原型系统作为最终产品。采用原型模型开发系统，用户和开发者必须达成一致。

软件开发模式

- 喷泉模型：（以用户需求为动力，以对象为驱动的模式，主要用于采用对象技术的软件开发项目）

它认为软件开发过程自下而上周期的各阶段是相互迭代和无间隙的特性

相互迭代：软件的各个部分常常被重复工作多次，相关对象在每次迭代中随之加入渐进的软件成分

无间隙：它在各项活动之间没有明显边界（如分析和设计活动之间）

软件开发模式

- 优点：

- 1、 可以提高软件项目开发效率，节省开发时间，适应于面向对象的软件开发过程

不便之处：

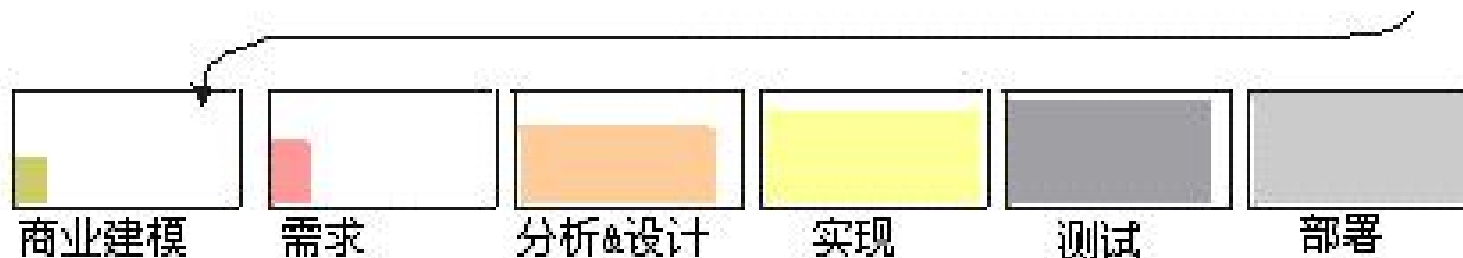
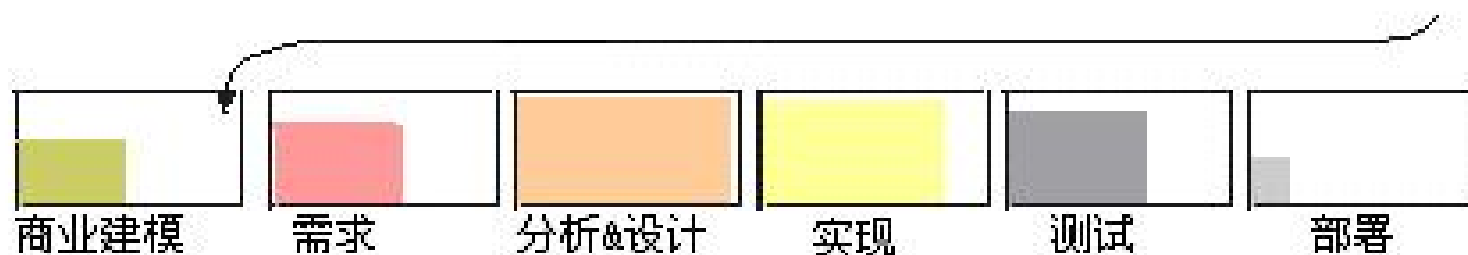
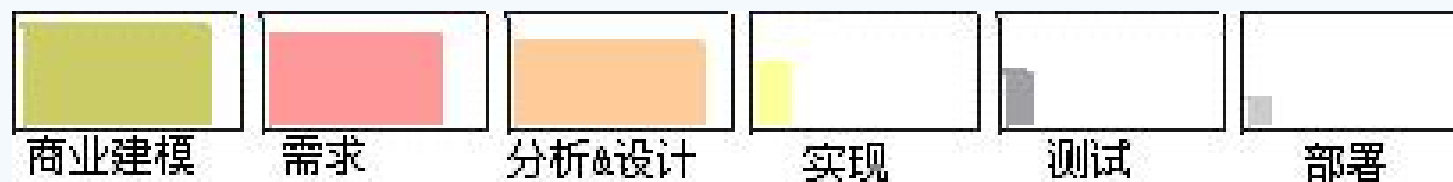
- 1、 由于喷泉模型在各个开发阶段是重叠的，因此在开发过程中需要大量的开发人员，因此不利于项目的管理。
- 2、 这种模型要求严格管理文档，使得审核的难度加大，尤其是面对可能随时加入各种信息、需求与资料的情况

软件开发模式

- 螺旋模型：（适用需求经常变化的项目）
它主要是风险分析与评估，沿着螺线进行若干次迭代，
过程：
 - 1、 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件
 - 2、 风险分析：分析评估所选方案，考虑如何识别和消除风险
 - 3、 实施工程：实施软件开发和验证；
 - 4、 客户评估：评价开发工作，提出修正建议，制定下一步计划。

软件开发模式

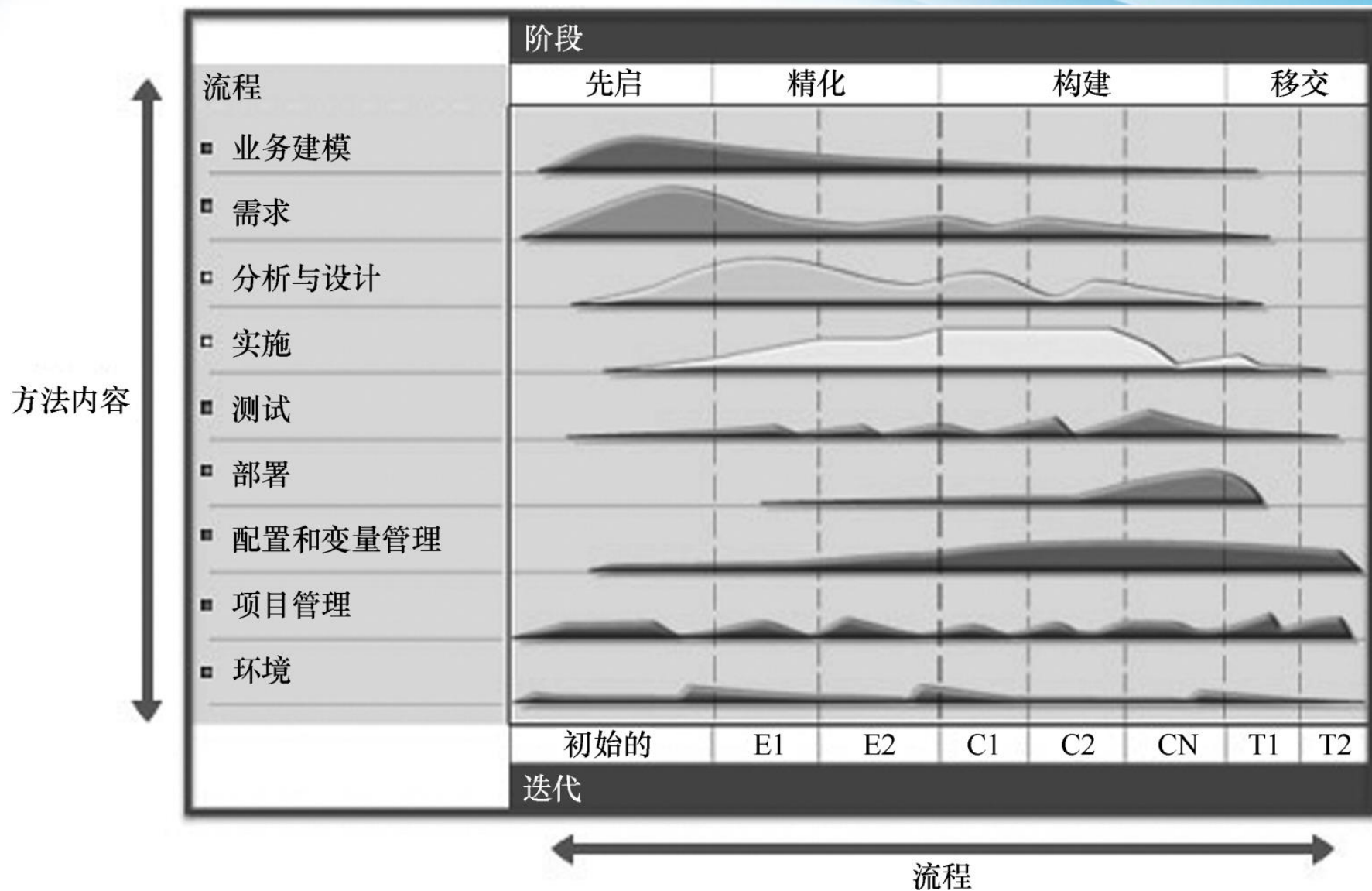
- 优点：
 - 1、 它由风险驱动，强调可选方案和约束条件从而支持软件的重用，有助于将软件质量作为特殊目标融入产品开发中
- 缺点：
 - 1、 难以让用户确信这种演化方法的结果是可以控制的
 - 2、 建设周期长
 - 3、 除非软件开发人员擅长寻找可能的风险，准确的分析风险，否则将会带来更大的风险



4.2基于统一过程的UML系统建模

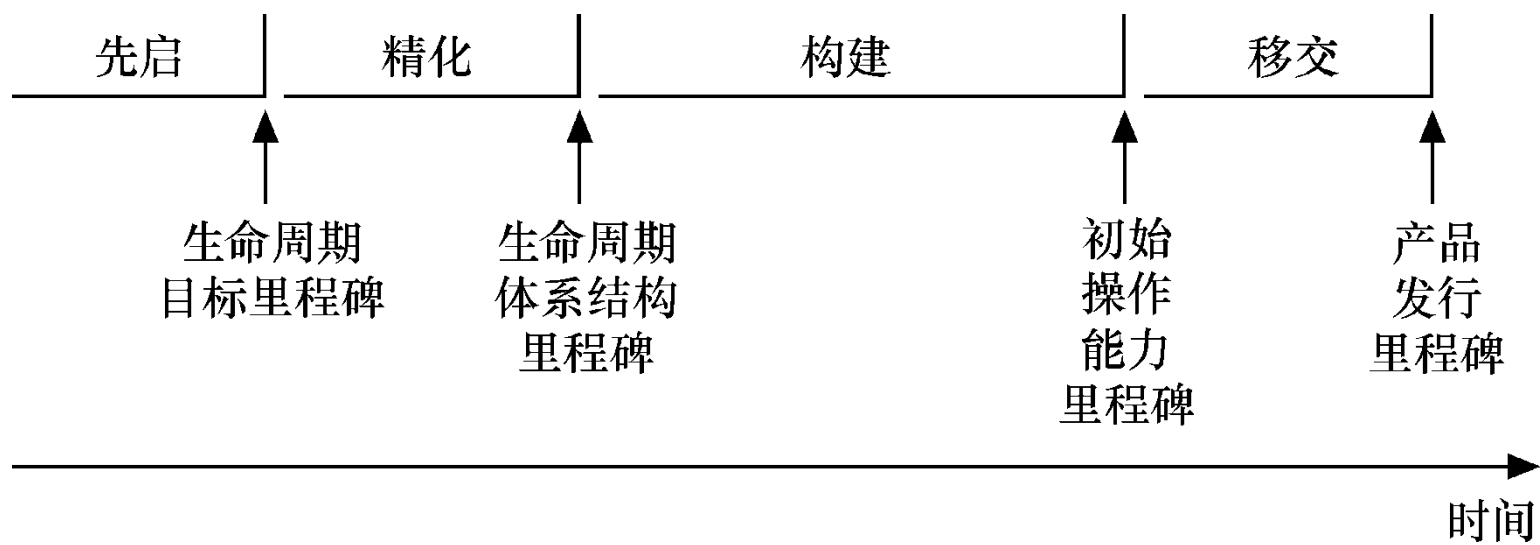
- RUP可使用UML来建立软件系统所需的各种模型。UML 是软件系统开发方法的一个组成部分, 融合了当前一些流行的面向对象开发方法的主要概念和技术, 成为一种面向对象的标准化的统一建模语言。UML与Rational 统一过程的结合将形成一种强大、高效的软件系统开发方法和技术。
- UML 在Rational 统一过程的使用中具有以下三个重要特征：统一过程以体系结构为中心；统一过程以用例驱动；UML 对迭代开发过程的支持。

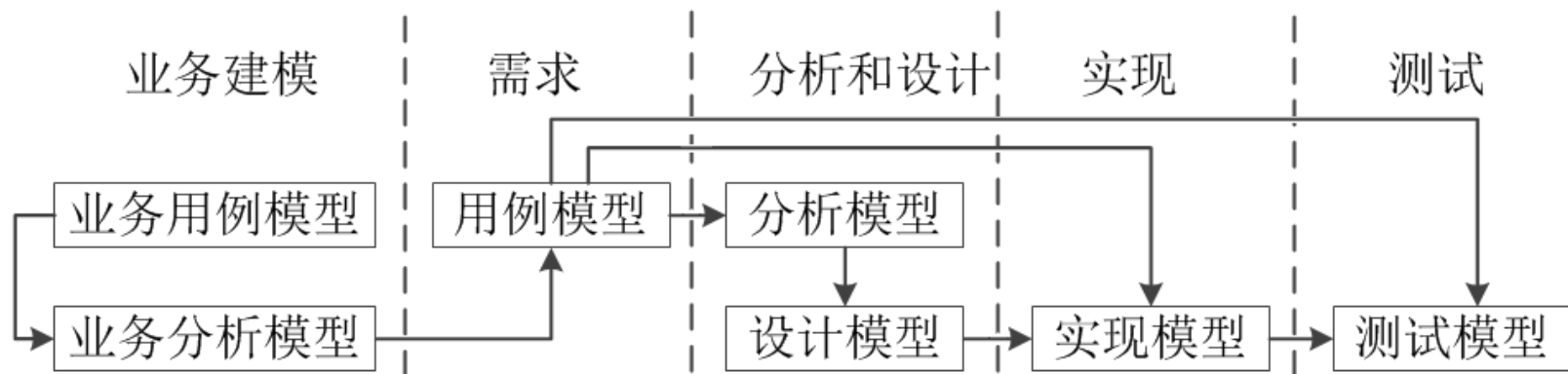
4.3 二维开发模型



4.4 RUP开发过程

- RUP中的软件生命周期在时间上被分解为四个顺序的阶段，分别是：
- 初始阶段（Inception）、细化阶段（Elaboration）、构造阶段（Construction）和交付阶段（Transition）。
- 每个阶段结束于一个主要的里程碑（Major Milestones）；每个阶段本质上是两个里程碑之间的时间跨度。在每个阶段的结尾执行一次评估以确定这个阶段的目标是否已经满足。如果评估结果令人满意的话，可以允许项目进入下一个阶段。





RUP的生命周期

初始阶段

初始阶段的主要目标是为系统建立商业案例和确定项目的边界。

定目标

细化阶段

细化阶段的主要目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。

定功能

构建阶段

在构建阶段，所有剩余的构件和应用程序功能被开发并集成为产品，所有的功能被详尽的测试。

编码

交付阶段

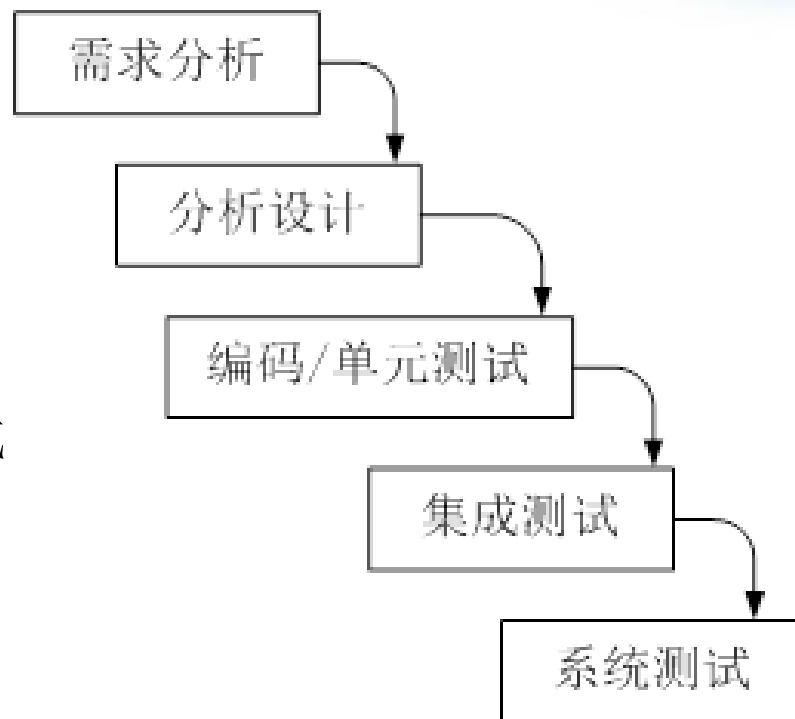
交付阶段的目的是将软件产品交付给用户群体。

交付

迭代软件开发

- 传统开发流程的问题

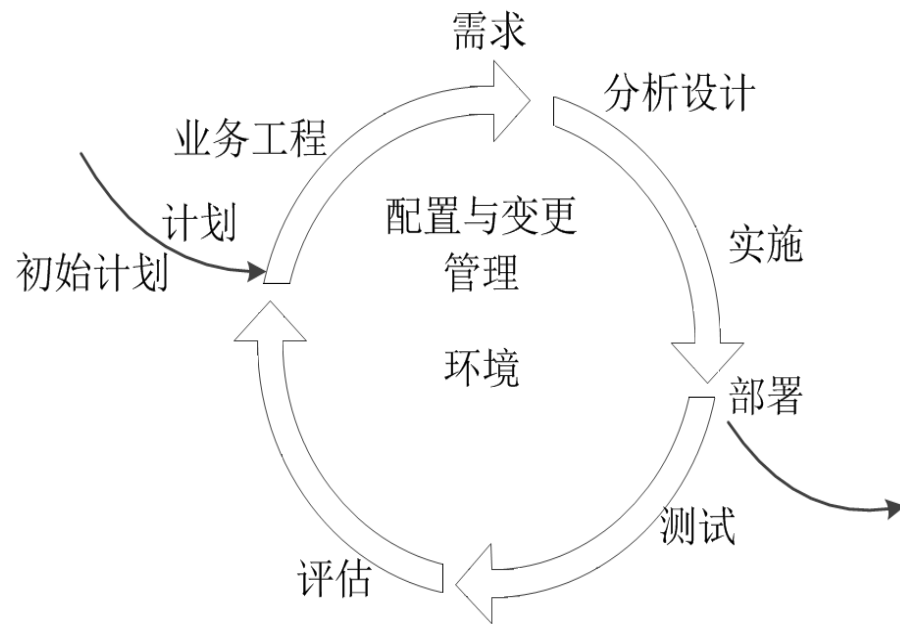
- 需求或设计中的错误往往只有到了项目后期才能够被发现
- 对于项目风险的控制能力较弱
- 软件项目常常延期完成或开发费用超出预算
- 项目管理人员专注于文档的完成和审核来估计项目的进展情况



迭代软件开发

- 迭代化开发的优势

- 允许变更需求
- 逐步集成元素
- 尽早降低风险
- 有助于提高团队的士气
- 生成更高质量的产品
- 保证项目开发进度
- 容许产品进行战术改变
- 迭代流程自身可在进行过程中得到改进和精炼

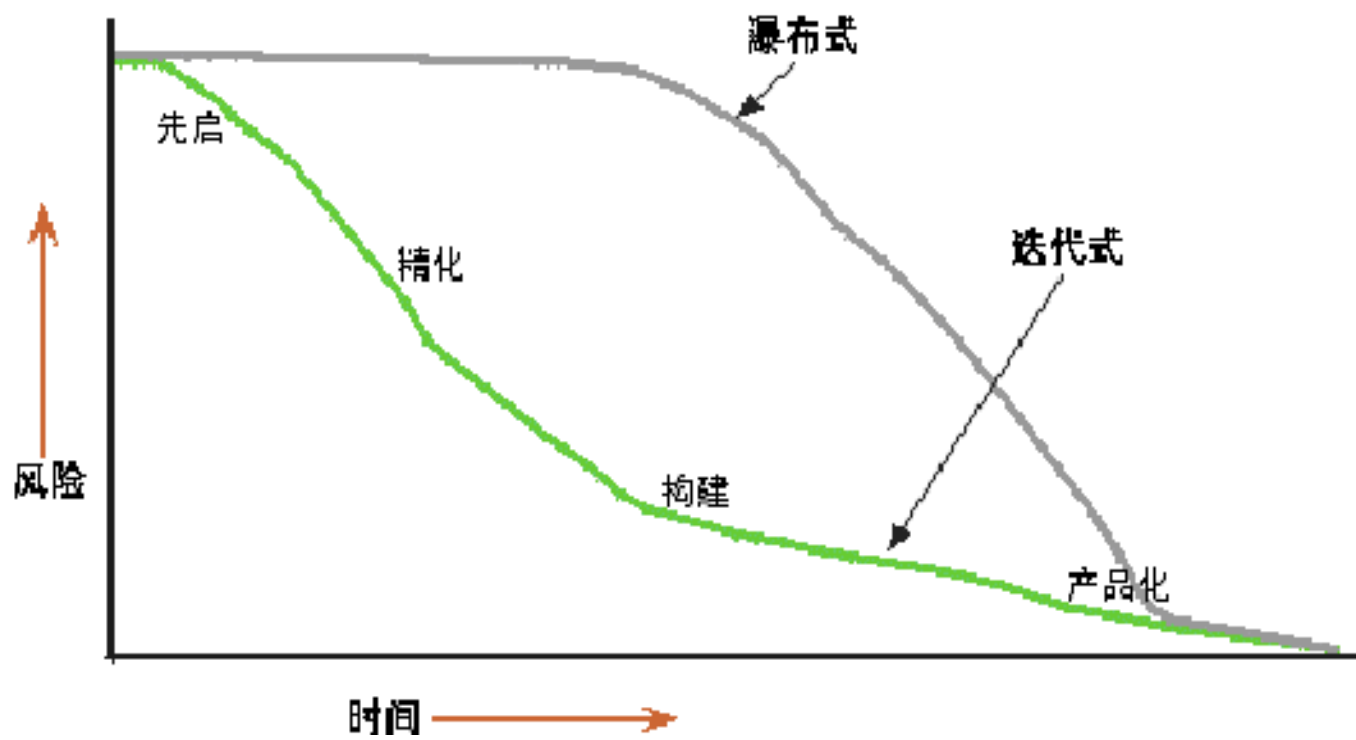


迭代软件开发

- 迭代方式开发软件

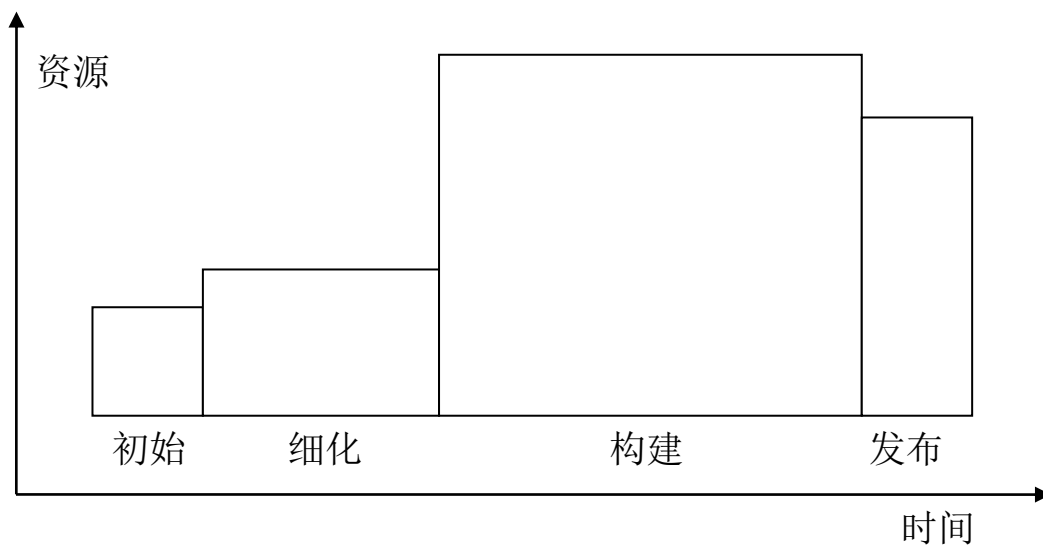
- 当我们在实际工作中实践迭代化思想时，**RUP**就可以给予我们实践的指导。**RUP**是一个通用的软件流程框架，它是一个以架构为中心、用例驱动的迭代化软件开发流程。**RUP**是从几千个软件项目的实践经验中总结出来的，对于实际的项目具有很强的指导意义，是软件开发行业事实上的行业标准。

- 迭代过程具有以下优点：
- 降低了在一个增量上的开支风险。如果开发人员重复某个迭代，那么损失只是这一个开发有误的迭代的花费。
- 降低了产品无法按照既定进度进入市场的风险。通过在开发早期就确定风险，可以尽早来解决而不至于在开发后期匆匆忙忙。
- 加快了整个开发工作的进度。因为开发人员清楚问题的焦点所在，他们的工作会更有效率。
- 由于用户的需求并不能在一开始就作出完全的界定，它们通常是在后续阶段中不断细化的。因此，迭代过程这种模式使适应需求的变化会更容易些。



迭代软件开发

- 关于开发资源的分配



迭代软件开发实践

- **【例】** 本节将应用火车票订购管理系统的功能来说明，火车票订购管理系统中的系统功能如表及优先级别如下表：

序号	功能名称	功能需求标识	优先级
1	申请订票	TOS-F01	高
2	订票确认	TOS-F02	高
3	统计	TOS-F03	中
4	到票登记	TOS-F04	高
5	领票	TOS-F05	高
6	查询	TOS-F06	中
7	我的火车票	TOS-F07	中
8	修改密码	TOS-F08	低
9	导出Excel	TOS-F09	中
10	导入学生信息	TOS-F10	低

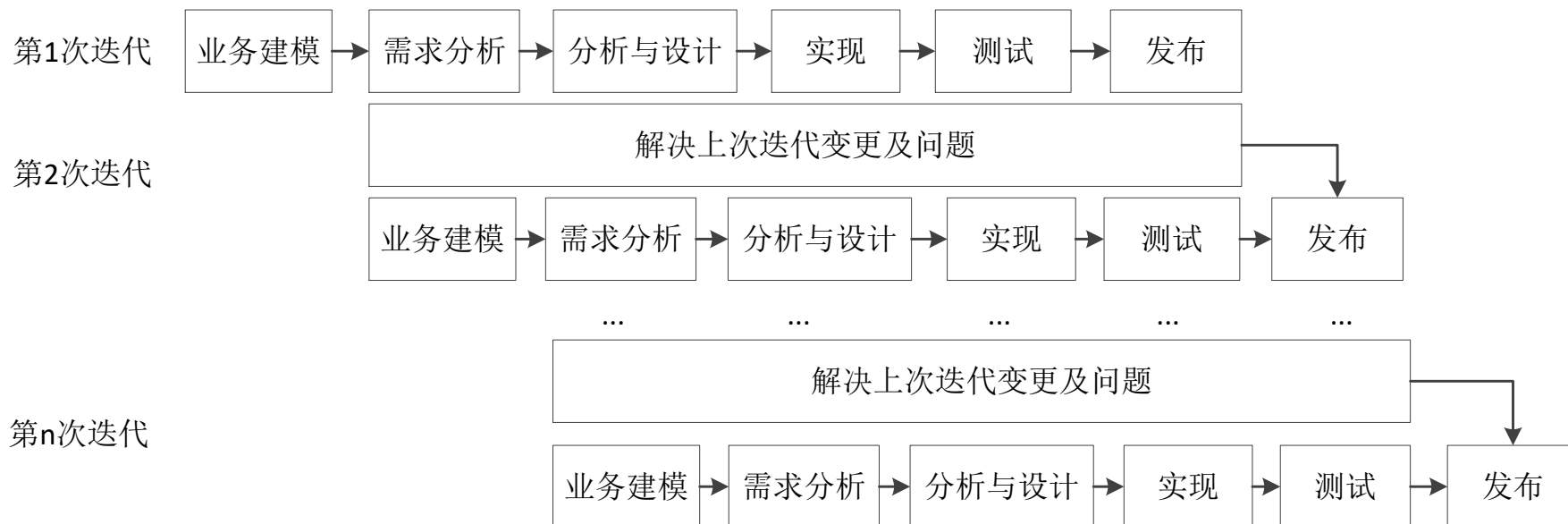
迭代软件开发实践

- 分析：假定，上表是目前仅已识别的系统功能，那么如何使用迭代来完成这个项目呢？如果完成一个迭代开发计划，分3次完成上述功能，迭代计划如表：

序号	迭代	功能名称	功能需求标识	优先级
1	第1次	申请订票	TOS-F01	高
2	第1次	订票确认	TOS-F02	高
3	第2次	统计	TOS-F03	中
4	第1次	到票登记	TOS-F04	高
5	第1次	领票	TOS-F05	高
6	第2次	查询	TOS-F06	中
7	第2次	我的火车票	TOS-F07	中
8	第3次	修改密码	TOS-F08	低
9	第2次	导出Excel	TOS-F09	中
10	第3次	导入学生信息	TOS-F10	低

迭代软件开发实践

- 得出的开发过程示意图：



4.5 RUP核心 workflow

- RUP中有9个核心 workflow，分为6个核心过程 workflow（Core Process Workflows）和3个核心支持 workflow（Core Supporting Workflows）。
- 尽管6个核心过程 workflow可能使人想起传统瀑布模型中的几个阶段，但应注意迭代过程中的阶段是完全不同的，这些 workflow在整个生命周期中一次又一次被访问。9个核心 workflow在项目中轮流被使用，在每一次迭代中以不同的重点和强度重复。

4.6 RUP十大要素和经验

- 1. 开发前景
- 2. 达成计划
- 3. 标识和减小风险
- 4. 分配和跟踪任务
- 5. 检查商业理由
- 6. 设计组件构架
- 7. 对产品进行增量式的构建和测试
- 8. 验证和评价结果
- 9. 管理和控制变化
- 10. 提供用户支持

- 最佳实践：
- 1. 迭代开发： RUP的开发过程建立在一系列迭代之上，每次迭代都有一个固定的时间限制（例如四个星期），称为"时间盒"，每次迭代结束的时候都发布一个稳定的小版本，该版本是最终系统的子集。"时间盒"是迭代开发中的关键概念：它意味着迭代周期的期限是固定的，如果目标没有完成，则放弃本次迭代的需求，而不是延长迭代的时间。
- 2. 管理需求
- 3. 使用基于组件的构架
- 4. 可视建模
- 5. 持续的质量验证
- 6. 控制变更

A decorative graphic at the top of the slide consisting of several overlapping, wavy blue lines that create a sense of depth and movement, resembling a stylized horizon or water waves. The colors transition from a light blue on the left to a slightly darker blue on the right.

Thank you !