

NEW PLAN

软考高级架构师

— 一段新征程 —





大纲介绍

项目管理

面向对象开发

软件工程

软件工程概述

软件成熟度模型

软件过程模型

逆向工程

需求工程

系统设计

测试基础知识

系统运行与维护

净室软件工程

基于构件的软件工程



软件过程概述(上个版本)



软件开发生命周期

- **软件定义时期**: 包括可行性研究和详细需求分析过程, 任务是确定软件开发工程必须完成的总目标, 具体可分成问题定义、可行性研究、需求分析等。
- **软件开发时期**: 就是软件的设计与实现, 可分成概要设计、详细设计、编码、测试等。
- **软件运行和维护**: 就是把软件产品移交给用户使用。



软件系统的文档可以分为**用户文档**和**系统文档**两类:

- 用户文档主要描述系统功能和使用方法, 并不关系这些功能是怎样实现的;
- 系统文档描述系统设计、实现和测试等各方面的内容。



软件过程概述(上个版本)



软件工程过程是指为获得软件产品包括以下4个方面活动:

- P(Plan): **软件规格说明**。规定软件的功能及其运行时的限制。
- D(Do): **软件开发**。开发出满足规格说明的软件。
- C(Check): **软件确认**。确认开发的软件能够满足用户的需求。
- A(Action): **软件演进**。软件在运行过程中不断改进以满足客户新的需求。



软件系统工具通常可以按软件过程活动将软件工具分为:

- 软件开发工具: 需求分析工具、设计工具、编码与排错工具、测试工具等。
- 软件维护工具: 版本控制工具、文档分析工具、开发信息库工具、逆向工程工具、再工程工具。
- 软件管理和软件支持工具: 项目管理工具、配置管理工具、软件评价工具、软件开发工具的评价和选择。



软件设计四个活动: **数据设计、架构(体系结构)设计、人机界面(接口)设计和过程(功能)设计。**



能力成熟度模型CMM

能力等级	特点	关键过程区域
初始级	软件过程的特点是杂乱无章，有时甚至很混乱，几乎没有明确定义的步骤，项目的成功完全依赖个人的努力和英雄式核心人物的作用。	
可重复级	建立了基本的项目管理过程和实践来跟踪项目费用、进度和功能特性，有必要的过程准则来重复以前在同类项目中的成功。	软件配置管理、软件质量保证、软件子合同管理、软件项目跟踪与监督、软件项目策划、软件需求管理
已定义级	管理和工程两方面的软件过程已经文档化、标准化，并综合成整个软件开发组织的标准软件过程。所有项目都采用根据实际情况修改后得到的标准软件过程来开发和维护软件。	同行评审、组间协调、软件产品工程、集成软件管理、培训大纲、组织过程定义、组织过程集点
已管理级	制定了软件过程和产品质量的详细度量标准。对软件过程 and 产品质量有定量的理解和控制。	软件质量管理和定量过程管理
优化级	加强了定量分析，通过来自过程质量反馈和来自新观念、新技术的反馈使过程能不断持续地改进。	过程更改管理、技术改革管理和缺陷预防



能力成熟度模型集成CMMI



CMMI：是若干过程模型的综合和改进，不仅仅软件，而是支持多个工程学科和领域的、系统的、一致的过程改进框架，能适应现代工程的特点和需要，能提高过程的质量和工作效率。



CMMI两种表示方法：

- 阶段式模型：类似于CMM,它关注组织的成熟度，五个成熟度模型如下：
- 连续式模型：关注每个过程域的能力，一个组织对不同的过程域可以达到不同的过程域能力等级。



阶段式模型

能力等级	特点	关键过程区域
初始级	过程不可预测且缺乏控制	
已管理级	过程为项目服务	需求管理、项目计划、配置管理、项目监督与控制、供应商合同管理、度量和分析、过程和产品质量保证
已定义级	过程为组织服务	需求开发、技术解决方案、产品集成、验证、确认组织级过程焦点、组织级过程定义、组织级培训、集成项目管理、风险管理、集成化的团队、决策分析和解决方案、组织级集成环境
定量管理	过程已度量和控制	组织过程性能、定量项目管理
优化级	集中于过程改进和优化	组织级改革与实施、因果分析和解决方案



考试真题



()是系统分析阶段结束后得到的工作产品, ()是系统测试阶段完成后的工作产品:

A.系统设计规格说明

B.系统方案建议书

C.程序规格说明

D.单元测试数据

A.验收测试计划

B.测试标准

C.系统测试计划

D.操作手册



以下关于CMM的叙述中, 不正确的是()。

A.CMM是指软件过程能力成熟度模型

B.CMM根据软件过程的不同成熟度划分了5个等级, 其中, 1级被认为成熟度最高, 5级被认为成熟度最低

C.CMMI的任务是将已有的几个CMM模型结合在一起, 使之构造成为“集成模型”

D.采用更成熟的CMM模型, 一般来说可以提高最终产品的质量



软件过程模型



软件过程模型是一种**规划和组织软件开发过程**的方法。你可以把它想象成是一张路线图，它帮助我们在软件开发的整个旅程中找到正确的方向。就像你在旅行前会查看地图一样，我们在开发软件前也会选择一个合适的过程模型，以确保我们按照一定的步骤和顺序来构建软件。



过程模型通常由**一系列阶段和任务组成**，这些任务在软件开发的阶段中都有特定的目标。**每个阶段都有自己的任务和产出物**，这些产出物构成了软件开发的各个部分，最终汇总成完整的软件产品。



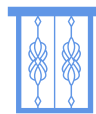
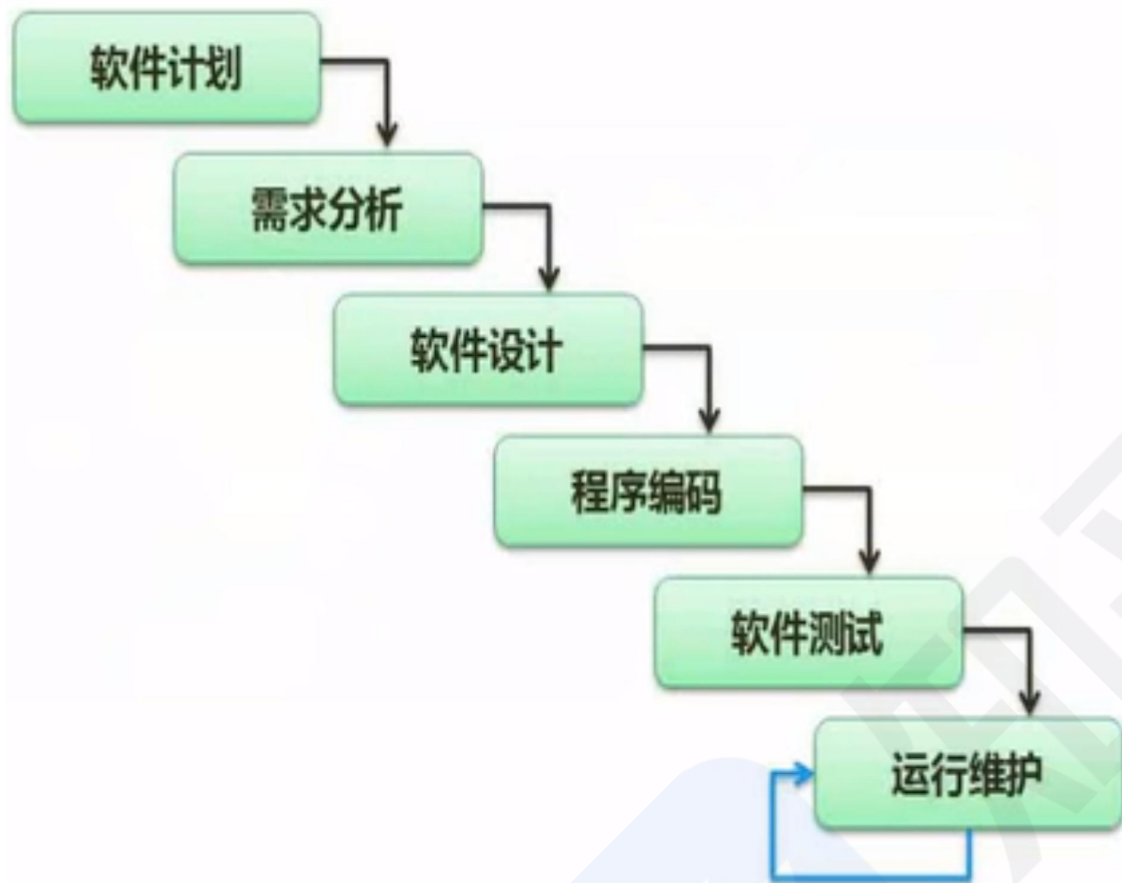
不同的软件项目可能需要不同的过程模型。有些项目可能需要非常详细的计划和严格的控制，因此会选择瀑布模型或V模型等传统的过程模型。而有些项目可能需要更加灵活和适应变化，因此会选择敏捷方法。



通过选择适合项目性质和需求的过程模型，我们可以更好地管理项目，控制进度，提高质量，并确保我们按照既定计划前进。在我们的团队中，我们会根据每个项目的特点来决定使用哪种过程模型，以便让我们的开发过程更加有序和高效。



软件过程模型



瀑布模型(SDLC):瀑布模型是一个经典的软件生命周期模型，一般将软件开发分为：可行性分析(计划)、需求分析、软件设计(概要设计、详细设计)、编码、测试、运行维护等几个阶段。



瀑布模型特点

- 从上一项开发活动接受该项活动的工作对象作为输入，利用这一输入，实施该项活动应完成的工作内容。
- 给出该项活动的工作成果，作为输出传给下一项开发活动。
- 对该项活动的实施工作成果进行评审。若其工作成果得到确认，则继续进行下一项开发活动；否则返回前一项，甚至更前项的活动。尽量减少多个阶段间的反复。以相对来说较小的费用来开发软件



原型化模型**第一步就是创建一个快速原型**，能够满足项目干系人与未来的用户可以与原型进行交互，再通过与相关干系人进行充分的讨论和分析，**最终弄清楚当前系统的需求**，进行了充分的了解之后，在原型的基础上开发出用户满意的产品。
适合于需求不明确的情况



原型法认为在很难一下子全面准确地提出用户需求的情况下，原型应当具备的特点如下。

- 实际可行
- 具有最终系统的基本特征
- 构造方便、快速，造价低。原型法的特点在于原型法对用户的需求是动态响应、逐步纳入的。



软件过程模型

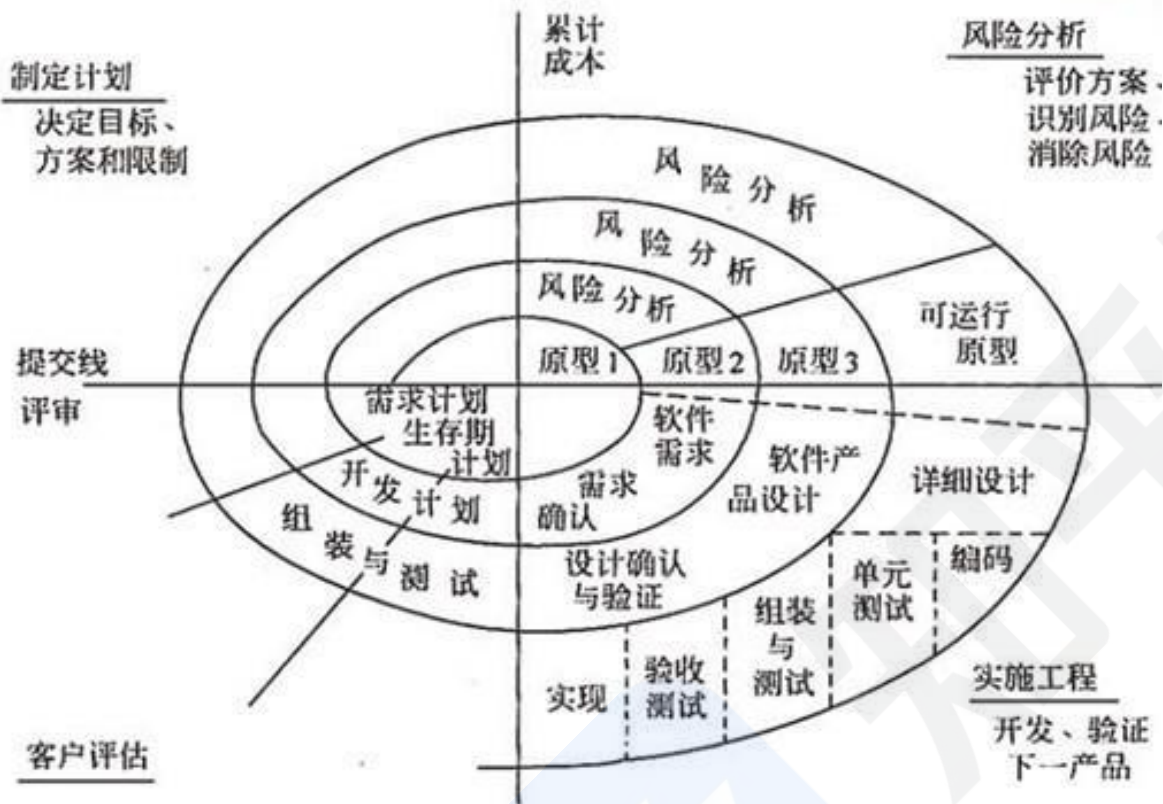


图 4-15 螺旋模型



螺旋模型是一个演化软件过程模型，将原型实现的迭代特征与线性顺序(瀑布)模型中控制的和系统化的方面结合起来。在螺旋模型中，软件开发是一系列的增量发布。

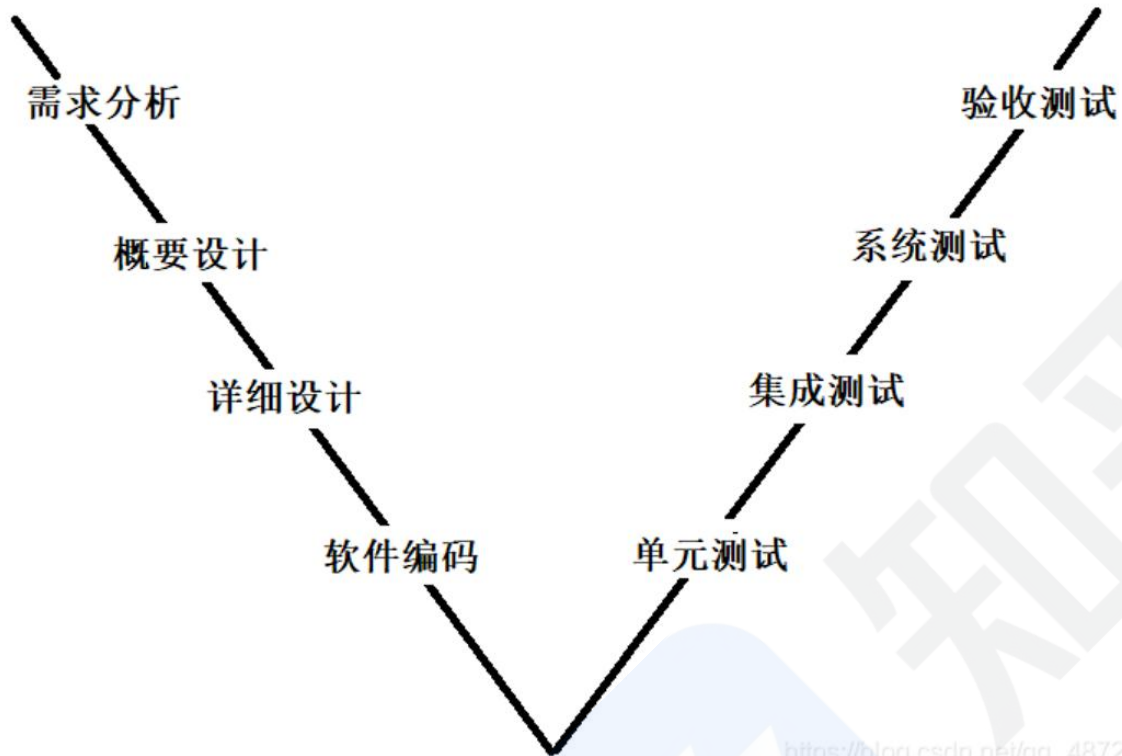


开发过程具有周期性重复的螺旋线状四个象限分别标志每个周期所划分的四阶段：制订计划、风险分析、实施工程和客户评估。螺旋模型强调了风险分析，特别适用于庞大而复杂的、高风险的系统。



软件过程模型

软件测试V模型



V模型从整体上看起来，就是一个V字型的结构，由左右两边组成。左边的下画线分别代表了需求分析、概要设计、详细设计、编码。右边的上画线代表了单元测试、集成测试、系统测试与验收测试。

V模型的特点如下：



- 单元测试的主要目的是针对编码过程中可能存在的各种错误；
- 集成测试的主要目的是针对详细设计中可能存在的问题
- 系统测试主要针对概要设计，检查系统作为一个整体是否有效地得到运行；
- 验收测试通常由业务专家或者用户进行，以确认产品能真正符合用户业务上的需要。
- **V模型适用于需求明确和需求变更不频繁的情形**



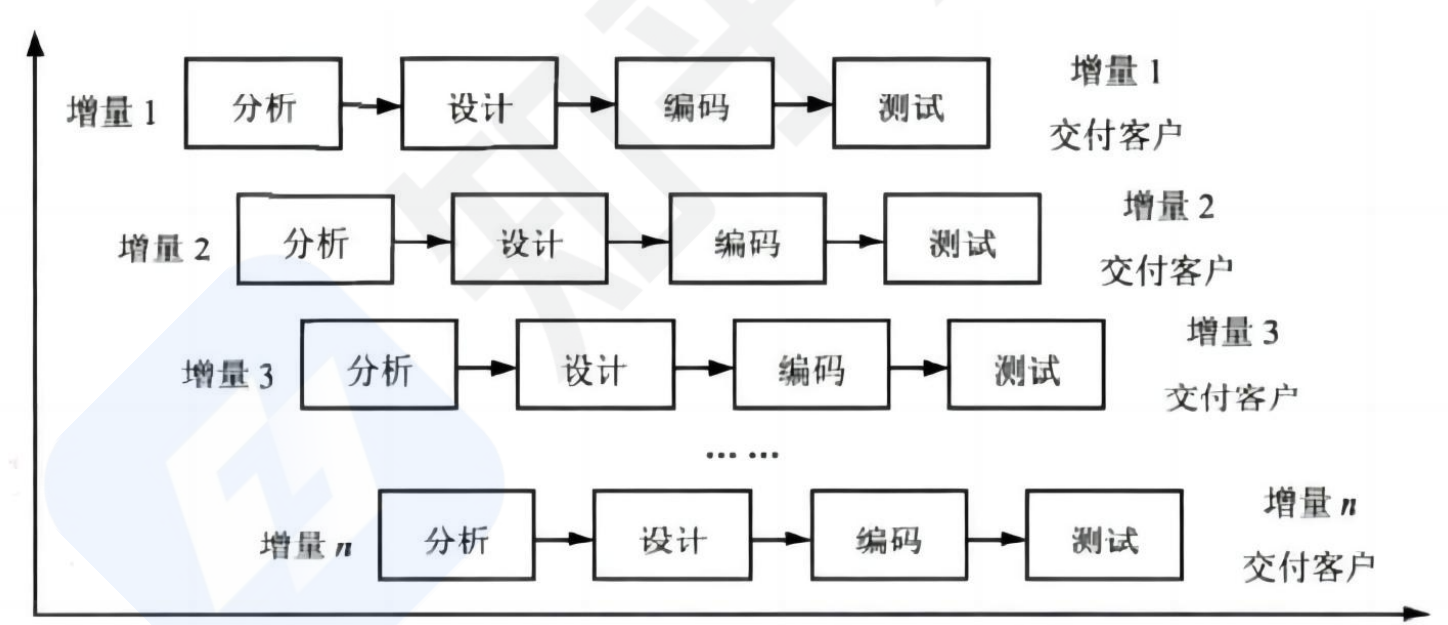
软件过程模型



增量模型：首先开发核心模块功能，而后与用户确认，之后再开发次核心模块的功能，即每次开发一部分功能，并与用户需求确认，最终完成项目开发，优先级最高的服务最先交付。



特点：但由于并不是从系统整体角度规划各个模块，因此不利于模块划分。难点在于如何将客户需求划分为多个增量。与原型不同的是增量模型的每一次增量版本都可作为独立可操作的作品，而原型的构造一般是为了演示





软件过程模型



喷泉模型：是一种以用户需求为动力，以对象作为驱动模型，适合于面向对象的开发方法。使开发过程具有迭代性和无间隙性。



基于构件的开发模型CBSD:也称之为快速开发模型，主要是利用预先包装的构件来构造应用系统。构件可以是组织内部开发的构件，也可以是商品化成品软件构件。特点是增强了复用性，在系统开发过程中，会构建一个构件库，供其他系统复用，因此可以提高可靠性，节省时间和成本。



形式化方法模型：建立在严格数学基础上的一种软件开发方法，主要活动是生成计算机软件形式化的数学规格说明。



考试真题



假设某软件公司与客户签订合同开发一个软件系统，系统的功能有较清晰的定义，且客户对交付时间有严格要求，则该系统的开发最适宜采用（ ）。

- A.瀑布模型 B.原型模型 C.V模型 D.螺旋模型



以下关于螺旋模型的叙述中，不正确的是()

- A.它是风险驱动的，要求开发人员必须具有丰富的风险评估知识和经验
B.它可以降低过多测试或测试不足带来的风险
C.它包含维护周期，因此维护 and 开发之间没有本质区别
D.它不适用于大型软件开发



软件过程模型-敏捷模型



开发宣言：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。



敏捷方法区别于其他方法的两个特点：

- 是“**适应性**”而非“预设性”。
- 是“**面向人的**”而非“面向过程的”。



敏捷方法的核心思想：

- 敏捷方法是适应型，而非可预测型。拥抱变化，适应变化。
- 敏捷方法是以人为本，而非以过程为本。发挥人的特性。
- **迭代增量式**的开发过程。以原型开发思想为基础，采用法代增量式开发，发行版本小型化。



软件过程模型-主要敏捷方法



- 极限编程(XP)。基础和价值观是交流、朴素、反馈和勇气，即任何一个软件项目都可以从4个方面入手进行改善：加强交流：从简单做起：寻求反馈；勇于实事求是。
 - XP是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期：通过积极的交流、反馈以及其他一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。
 - XP提倡测试先行，为了将以后出现bug的几率降到最低。



- 水晶系列方法。与XP方法一样，都有以人为中心的理念，但在实践上有所不同。其目的是发展一种提倡“机动性的”方法，包含具有共性的核心元素，每个都含有独特的角色、过程模式、工作产品和实践。



- 并列争球法(Scrum)。是一种迭代的增量化过程，把每段时间(如30天)一次的迭代称为个“冲刺”(Sprint),并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。



- 特性驱动开发方法(FDD)。是一个迭代的开发模型。认为有效的软件开发需要3个要素：人、过程和技术。有5个核心过程：开发整体对象模型、构造特征列表、计划特征开发、特征设计和特征构建。



软件过程模型-统一过程模型(RUP)



RUP描述了如何有效地利用商业的、可靠的方法开发和部署软件，是一种重量级过程。RUP类似一个在线的指导者，它可以为所有方面和层次的程序开发提供指导方针、模版以及事例支持。



RUP软件开发生命周期是一个二维的软件开发模型，RUP中有9个核心工作流，如下：

- 业务建模：理解待开发系统所在的机构及其商业运作，确保所有参与人员对待开发系统所在的机构有共同的认识，评估待开发系统对所在机构的影响。
- 需求：定义系统功能及用户界面，使客户知道系统的功能，使开发人员理解系统的需求，为项目预算及计划提供基础。
- 分析与设计：把需求分析的结果转化为分析与设计模型。
- 实现：把设计模型转换为实现结果，对开发的代码做单元测试，将不同实现人员开发的模块集成为可执行系统。
- 测试：检查各子系统之间的交互、集成，验证所有需求是否均被正确实现，对发现的软件质量上的缺陷进行归档，对软件质量提出改进建议。
- 部署：打包、分发、安装软件，升级旧系统；培训用户及销售人員，并提供技术支持。
- 配置与变更管理：跟踪并维护系统开发过程中产生的所有制品的完整性和一致性。
- 项目管理：为软件开发项目提供计划、人员分配、执行、监控等方面的指导，为风险管理提供框架。
- 环境：为软件开发机构提供软件开发环境，即提供过程管理和工具的支持。



软件过程模型



RUP把软件开发生命周期**划分为多个循环**，每个循环生成产品的一个新的版本，**每个循环依次由4个连续的阶段组成，每个阶段完成确定的任务**。这4个阶段如下。

- 初始阶段：定义最终产品视图和业务模型，并确定系统范围。
- 细化阶段：设计及确定系统的体系结构，制订工作计划及资源要求。
- 构造阶段：构造产品并继续演进需求、体系结构、计划直至产品提交。
- 移交阶段：把产品提交给用户使用。



RUP中定义了如下一些核心概念，理解这些概念对于理解RUP很有帮助。

- 角色：Who的问题。角色描述某个人或一个小组的行为与职责。RUP预先定义了很多角色，如体系结构师、设计人员、实现人员、测试员和配置管理人员等，并对每一个角色的工作和职责都做了详尽的说明。
- 活动：How的问题。活动是一个有明确目的的独立工作单元。制品：What的问题。制品是活动生成、创建或修改的一段信息。
- 工作流：When的问题。工作流描述了一个有意义的连续的活动序列，每个工作流产生一些有价值的产品，并显示了角色之间的关系。



软件过程模型



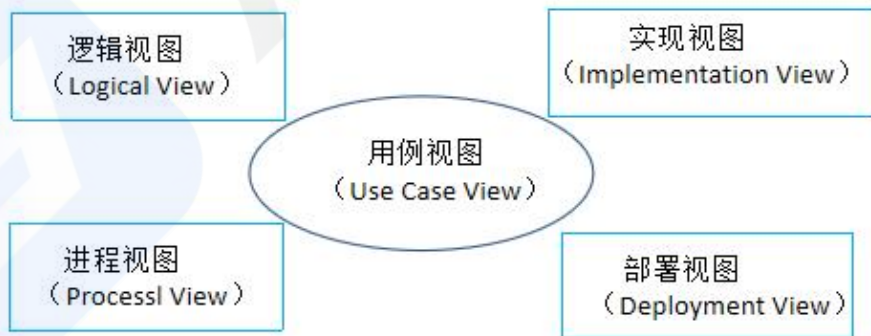
RUP的特点:

- 用例驱动：需求分析、设计、实现和测试等活动都是用例驱动的。
- 以体系结构为中心：包括系统的总体组织和全局控制、通信协议等。是一个多维的结构，会采用多个视图来描述。
- 迭代与增量。把整个项目开发分为多个迭代过程。在每次迭代中，只考虑系统的一部分需求，进行分析、设计、实现、测试和部署等过程；每次迭代是在已完成部分的基础上进行的，每次增加一些新的功能实现，以此进行下去，直至最后项目的完成。



典型的4+1视图模型中:

- 分析人员和测试人员关心的是系统的行为，会侧重于用例视图；
- 最终用户关心的是系统的功能，会侧重于逻辑视图；
- 程序员关心的是系统的配置、装配等问题，会侧重于实现视图；
- 系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题，会侧重于进程视图；
- 系统工程师关心的是系统的发布、安装、拓扑结构等问题，会侧重于部署视图。



T H E E N D

功不唐捐，玉汝于成！

• 开 启 新 征 程 •

