

利用 Rational Unified Process 达到 CMM 级别 2 和 3

Rational Software 白皮书

TP 174

目录

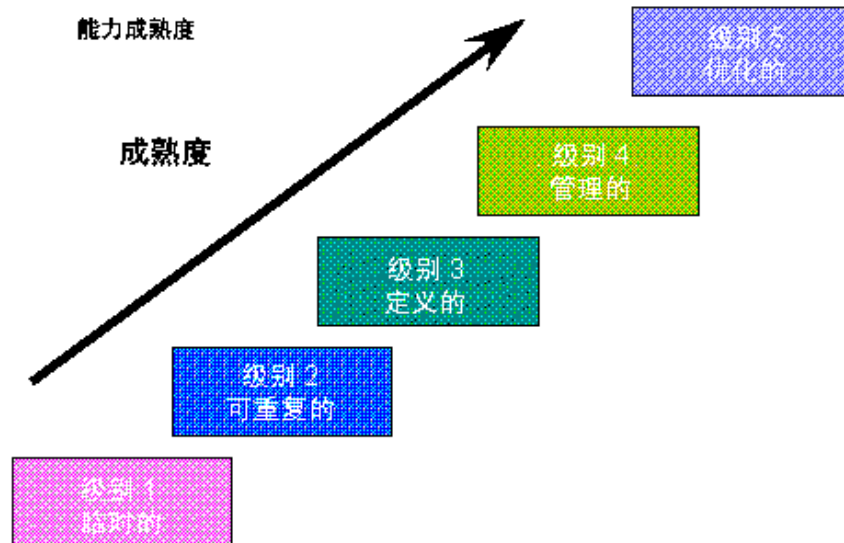
摘要1
简介1
级别 2 , 可重复的2
需求管理2
软件项目规划4
软件项目跟踪与勘察5
软件分包管理6
软件质量保证6
软件配置管理7
级别 3 , 已定义的8
组织流程重点8
组织流程定义8
培训计划9
集成软件管理9
软件产品工程10
组间协作10
平级复审10
参考资料11

摘要

软件工程协会 (SEI) 的能力成熟度模型 (CMM) 提供了一种著名的软件流程成熟度基准。CMM 已经成为了许多领域内的流行工具,用于评估一个组织的软件流程的成熟程度。本白皮书说明了 Rational Unified Process 如何支持正在努力达到 CMM 级别 2 (可重复的) 和级别 3 (已定义的) 的组织。

简介

软件工程协会 (SEI) 的能力成熟度模型 (CMM) 是一个描述有效软件流程元素的框架; 请参阅本文档末尾标题为“参考资料, [REF 1]”的部分。CMM 描述了一条从临时的、未成熟的流程向成熟的、规范化的流程演进的途径。CMM 覆盖软件开发和维护的规划、工程以及管理经验。这些关键的经验提高了组织实现成本、进度、功能性和产品质量等目标的能力。CMM 有五个成熟级别: 从级别 1 到级别 5。如下图所示。每个成熟级别由关键流程领域 (Key Process Areas, KPA) 组成, 每个 KPA 确定一组相关活动。当这些相关活动一起开展时, 它们完成一系列被认为对在该成熟级别建立流程能力有重要影响的目标。



在可重复级别, 应建立管理软件项目的策略以及实施这些策略的过程步骤。新项目的规划和管理是以类似项目的经验为基础的。达到级别 2 的目标就是为了使软件项目的有效管理流程制度化, 从而让组织重复在过去的项目中获得的成功经验, 即使项目实施的具体流程可能存在差异。有效流程的特征可以归纳为熟练的、文档化的、加强的、培训的、评测的和可以改进。级别 2 的组织的项目已经安装了基本的软件管理控制。符合现实的项目承诺是根据对以前项目的观察结果和当前项目的需求做出的。项目的软件经理跟踪软件成本、进度和功能, 并确定在履行承诺期间出现的问题。创建软件需求和为满足这些需求开发的工作产品的基线, 并控制它们的完整性。定义软件项目标准后, 组织确保这些标准得到不折不扣的执行。如果有分包商, 则软件项目可以和分包商合作, 建立牢靠的顾客供应商关系。

级别 2 组织的软件流程能力可以用规范化来概括, 因为软件项目的规划和跟踪是稳定的, 以前的成功经验可重复使用。项目的流程受项目管理系统的有效控制, 遵守根据以前项目的性能制定的符合现实的计划。

级别 2 KPA 是：

- ☐ 需求管理
- ☐ 软件项目规划
- ☐ 软件项目跟踪与勘察
- ☐ 软件分包管理
- ☐ 软件质量保证
- ☐ 软件配置管理

在已定义的级别上，组织开发维护软件的标准流程已做记录（包括软件工程和管理流程），而且这些流程都集成到一个连贯的整体中。标准流程在整个 CMM 中始终是指组织的标准软件流程。在级别 3 建立的流程用于（必要的时候可修改）帮助软件经理和技术人员更有效地执行任务。组织在建立标准化的软件流程的时候，利用了有效的软件工程的经验和方法。有一个小组负责组织的软件流程活动，如软件工程或SEPG。为了确保员工和经理具有完成分配给他们的任务必须掌握的知识和技能，需要在整个组织范围内实施培训计划。

项目对组织的标准软件流程进行定制，开发它们自己定义的软件流程，使项目具有独一无二的特点。这个定制流程在 CMM 中是指项目的已定义软件流程。已定义的软件流程包含了定义明确的软件工程和管理流程的一个一致、完整的集合。明确定义的流程其特征可以归纳为包含准备就绪的准则、输入、执行工作的标准和过程、验证机制（如平级复审）、输出和完成标准等。由于明确定义了软件流程，管理层对所有项目的技术进展都有深刻的了解。

级别 3 组织的软件流程能力可以用标准一致来概括，因为软件工程和管理活动不仅稳定而且可重复。在已建立的产品线内，成本、进度和功能都受到控制，并且软件质量获得跟踪。这一流程能力建立在整个组织对已定义的软件流程的活动、角色和责任形成共同理解的基础上。

级别 3 KPA 是：

- ☐ 组织流程重点
- ☐ 组织流程定义
- ☐ 培训计划
- ☐ 综合软件管理
- ☐ 软件产品工程
- ☐ 组间协作
- ☐ 平级复审

本文各节描述 Rational Unified Process 特性、方法、程序和工件如何实现KPA 目标。

本文是为关心达到 CMM 框架内的组织成熟级别 2 和级别 3 的组织人员而编写的。

级别 2，可重复的

需求管理

需求管理的目的是为了在客户和处理客户需求的软件项目之间建立共识。与客户达成的统一认识是软件项目规划（如软件项目规划 KPA 所述）和管理（如软件项目跟踪

与勘察 KPA 所述)的基础。对客户关系的控制依赖于执行有效的变更控制流程(如软件配置管理 KPA 所述)。

Rational Unified Process 的关键特性之一在于它是用例驱动的。用例代表了获取、组织和传达用户需求的一种系统化方案。它们提供了记录功能性需求的方式,而功能性需求是项目开发、测试和迭代规划的基础。在 Rational Unified Process 中,用例在用例模型中进行维护,并在项目的整个生命周期里统一引用,从分析到测试一直到维护。

在工程环境中获取需求的 Rational Unified Process 工件是:

- ☐ 由用例和用例包构成的用例模型
- ☐ 非功能性的“补充规约”
- ☐ 用例模型调查
- ☐ 用例报告
- ☐ 词汇表

在管理环境中使用的、说明待开发用例及场景(需求)的 Rational Unified Process 工件包括:

- ☐ 迭代计划
- ☐ 集成构建计划
- ☐ 项目计划
- ☐ 软件开发计划

所有这些工件都建立了基线,并受某个变更管理规定的制约。

目标 1 :对分配给软件的系统需求进行控制,以便创建软件工程和管理基线。

Rational Unified Process 主张对所有演进的工件进行连续的配置控制,然而,“正式的”基线与以下里程碑对应。

- ☐ 生命周期目标里程碑(先启阶段)
- ☐ 生命周期构架里程碑(精化阶段)
- ☐ 初始操作能力里程碑(构建阶段),以及
- ☐ 产品发布里程碑(移交阶段)

同样,Rational Unified Process 在需求、需求管理、跟踪及创建基线上与 CMM 一致。

目标 2 :软件计划、产品和活动与分配给软件的系统需求保持一致。

该 CMM 目标重点在于确保交付的系统满足用户需求。Rational Unified Process 通过两种方式帮助组织实现这一目标:

- ☐ 用例方案确保用户需求得到理解并被获取。获取需求后,需求向下流动到各个“可视的”Rational Unified Process 模型(用例、设计、实施和测试),以保证一致性和连贯性。

- **控制的迭代式开发方案**是一种风险降低策略，藉此项目风险能够及早得到理解和研究，早得到理解和研究，每一次累进迭代，通过不断集成新增的功能，及早揭示风险。若使用传统的瀑布式方法，则这些风险直到开发生命周期的后期才能够被发现。及早识别风险对项目管理有直接好处，可重新定义需求规模，或者提出其他战术改变。

Rational Unified Process 管理文档包括：

- 商业理由
- 软件开发计划
- 评测计划
- 风险列表
- 项目计划
- 迭代计划
- 迭代评估和状态评估。

有效的变更控制和管理是 Rational Unified Process 的另一特性，它确保了软件根据分配的、受到跟踪的指定需求来开发。

Rational Unified Process 主张每个项目都应设立一个变更控制委员会(CCB)，对提议的变更或者开发过程中发现的缺陷在规模及影响方面（预算、技术或时间安排）作出公断。为了协助 CCB 的运作，Rational Unified Process 建议使用强大的配置管理和版本控制工具/环境。

软件项目规划

软件项目规划的目的在于建立合理的计划，执行软件工程和管理软件项目。这些计划是软件项目管理（如 *软件项目跟踪与勘察* KPA 所述）所必不可少的。没有符合现实的计划，就不可能实施有效的项目管理。

目标1：对软件估算进行记录，以便用于规划和跟踪软件项目。

Rational Unified Process 的目标之一是确保各方面的期望都同步进行并且保持一致。它通过在项目生命周期内进行定期评估来确保完成，并记录在 *状态评估报告* 中。报告需要对资源（人员配备和财政）、首要的十大风险、技术进步的追踪数据，通过指标和主要的里程碑结果来进行测量。

Rational Unified Process 利用了以下类别的指标：

- 进度（代码行、类的个数、每次迭代的功能点、返工）
- 稳定性（返工类型、需求或实施变更率）
- 适应性（返工成本）
- 模块度（返工影响范围）
- 质量（缺陷发现频率、密度、继承深度、返工指示符）
- 成熟程度（每次故障的测试时间）
- 资源耗费配置文件（计划的与实际的）

目标 2：计划并记录软件项目活动和承诺。

获取项目计划和承诺的 Rational Unified Process 文档包括：

- 商业理由

- 软件开发计划
- 评测计划
- 风险列表
- 项目计划
- 迭代计划
- 迭代评估，以及
- 状态评估。

目标 3：涉及的组和个人同意与软件项目相关的约定。

在 Rational Unified Process 中，软件开发计划定义了项目的总计划；迭代计划详细定义了迭代中要完成的目标。Rational Unified Process 所要求的迭代计划复审会向所有项目干系人公布迭代计划，以在迭代开始前达成一致意见。根据协商的迭代计划，项目经理会生成一系列工作单，向涉及的项目组和个人详细传达迭代计划的目的。项目经理就这些工作单与涉及的人员达成协议，这样迭代就可以进行了。

软件项目跟踪与勘察

软件项目跟踪与勘察的目的在于建立实际进度的适当可见度，以便管理人员在软件项目的执行极大偏离软件计划时采取有效的措施。

目标 1：对比软件计划追踪实际结果和性能。

如*软件项目规划*一节所述，Rational Unified Process 有几个级别的项目计划和一个状态评估报告。状态评估报告对比计划与实际的结果，从而进行评估。为特定里程碑生成状态评估报告是项目经理的职责。

主要的 Rational Unified Process 里程碑对应着一个阶段（先启、精化、构建或移交）的结束，有明确指定的完成标准。一个阶段的每次迭代结束时，在次要的里程碑处都存在复审的机会，这也是决策点，是未来发展方向的经验教训。例如，精化阶段的目标是分析问题领域，建立一个坚固的构架基础，制定项目计划，消除项目中的风险最高的元素。必须对整个系统有了理解之后，才能做出构架决策。这就暗示着在描述大部分用例时会考虑一些约束：补充需求。为验证构架，实施一个系统，来证明所选构架是正确的，并执行意义重大的用例。

在精化阶段结束时，检查详细的系统目标和规模，以及选择的构架和确定的主要风险。当实际结果和性能极大地偏离软件计划时，采取纠正措施，并管理至项目结束。

*风险列表*是一个 Rational Unified Process 工件，它概括了项目中所有已知风险，并作为规划和项目评估的输入。每个风险都根据它的影响和应急计划来进行描述，应急计划是为降低风险而制定的。风险列表与业务案例一起制定，它们形成了“执行”或“不执行”项目的决策基础。风险列表在项目的整个生命周期都要进行维护。

目标 2：当实际结果和性能明显偏离软件计划时采取和管理更正操作来结束。

Rational Unified Process 需要两种级别的性能追踪：作为常规监测和项目控制的一部分，项目经理将使用收集的度量来生成状态评估。状态评估中指明的问题会直接由项目经理（通过工作单）根据项目的问题解决计划来处理，或者对于更大范围的问题，可以通过变更请求来处理。另外，每次迭代的结果遵循迭代评估和复审。这使之前迭代的经验能够通过变更请求指导后续迭代的计划和更正操作的管理。

目标 3：软件承诺的变更得到受影响的小组和个人的同意。

如 Rational Unified Process 所述，受控的迭代式开发流程确保项目干系人能经常看到项目进展情况以及为了保持项目不偏离轨道所作的任何必要的变更。提议的变更由变更控制委员会(CCB)进行复审，确保变更符合现实的，并且可以被项目的整体日程接受。

软件分包管理

分包管理的目的在于选择合格的软件分包商并对他们进行有效的管理。它综合考虑了需求管理、软件项目规划、软件项目跟踪与勘察等的基本管理控制，以及软件质量保证和软件配置管理之间的必要协调，并在适当时候对分包商施以控制。

目标 1：由主承包商选择合格的软件分包商。

目标 2：主承包商和软件分包商同意彼此承担的义务。

目标 3：主承包商和软件分包商保持连续不断的交流。

目标 4：主承包商针对其承诺追踪软件分包商的实际结果和执行情况。

这些目标超出了 Rational Unified Process 当前的范围，并且依赖于组织的具体情况。尽管 Rational Unified Process 并未对项目分包作明确说明，但它的工具、技术和机制都是以向下流动到分包商为前提的，因此流程仍属同类。

所有的分包决策都应该记录在商业理由中。与主承包商执行同一开发计划的分包商还参与技术交换、主要里程碑和状态评估等活动。

软件质量保证

软件质量保证的目的是为管理人员提供软件项目所用流程和正在构建的产品的可见度。软件质量保证是大多数软件工程和管理流程的一个构成部分。

Rational Unified Process 认为“质量”是所有项目员工共同的责任，它并非由组织本身体现出来。

目标 1：计划软件质量保证活动。

软件质量保证的规划是组织的一个责任。然而，Rational Unified Process 有许多属性用来塑造一个有效的项目质量保证计划。每个 Rational Unified Process 里程碑都有特定的完成标准，这些标准可作为审计的基础。Rational Unified Process 中的每个活动都有一个单独的复审活动。每次复审都有一组检查点与之相关，它们代表了在进入下一个活动之前必须“通过”的“关口”。Rational Unified Process 提供有关谁应该复审给定工件的指南。例如，设计员执行的“对象分析”的结果需要由一个独立的构架设计师、设计员、用例设计员和设计复审员来进行复审。如果有已定义的 Rational Unified Process 和工件复审标准，产品质量密切相关的目标实体应该能够轻松地评估是否遵守流程以及是否符合开发标准及指南。

目标 2：客观地验证软件产品及活动是否遵守适用的标准、过程和需求。

该目标可以通过挑选组织的质量保证人员来实现。然而，Rational Unified Process 提供了必要的复审清单和文档模板，它们可作为项目标准。

目标 3：将软件质量保证活动和结果通知受影响的小组和个人。

如软件项目规划所述，Rational Unified Process 的目标之一是确保各方的期望同步并且保持一致。除根据质量审计结果提供的输入之外，Rational Unified Process 还需要关于资源（员工配备和财政）、首要十大风险、用指标进行衡量的技术、进展以及主要里程碑结果等的报告。Rational Unified Process 指标计划提供了关于以下指标集合的指南：

- ☐ 进度（代码行、类、每次迭代的功能点）
- ☐ 稳定性（返工类型、易变性）
- ☐ 适应性（返工成本）
- ☐ 模块度（返工影响范围）
- ☐ 质量（缺陷发现频率、密度、继承深度）
- ☐ 成熟程度（每次故障的测试时间）
- ☐ 资源耗费配置文件（计划的与实际的）

目标4：在软件项目内无法解决的非兼容性问题由高级管理层负责处理。

这超出了 Rational Unified Process 的范围，属于组织的职责范畴。然而，Rational Unified Process 里描述的变更控制流程可以驱动某个机制，藉此记录非兼容性的问题并可以记录下来并向上提交以便解决。

软件配置管理

软件配置管理的目的是在项目的整个软件生命周期内建立并维护软件项目产品的完整性。软件配置管理是大多数软件工程和管理流程的一个构成部分。

目标 1：计划软件配置管理活动。

如 Rational Unified Process 所述，可靠的配置管理是受控的迭代式开发方法中一个必不可少的元素。既然软件是分阶段演进的，因此以前开发的软件版本可以用于后续开发是非常重要的。在每一阶段规划如何开发指导性软件是 Rational Unified Process 的核心。

Rational Unified Process 有两个主要手段，用于规定如何维护项目的软件开发资产以及如何集成这些资产：

- ☐ 配置管理计划，以及
- ☐ 集成构建计划。

在先启阶段启动的配置管理计划描述以下内容：

- ☐ 管理软件的版本化和处理；
- ☐ 保存指定的 Rational Unified Process 模型，将它们分成多个配置项；
- ☐ 使用变更控制方法管理变更和发布。

集成构建计划提供了关于待构建的配置项的详细信息以及它们在某个特定的迭代中的集成顺序。

目标 2：确定、控制所选的软件工作产品，并使之可用。

Rational Unified Process 配置管理计划需要一个对配置控制和管理流程的说明，确保确实确定和控制工作产品，并使之可用。

目标 3：对已确定软件工作产品的变更进行控制。

Rational Unified Process 主张，项目应有一个变更控制委员会(CCB)和一个变更管理系统，以便有效地管理、跟踪和实施变更请求，并计算其变更成本。

目标 4：将软件基线的状态和内容通知受影响的小组与个人。

Rational Unified Process 提倡使用电子方式维护需求、设计和实施基线以及它们之间的可追踪性。基线的所有变更分别由不同级别的项目控制团队来裁定。例如，变更控制委员会(CCB)负责考虑需求级别的变更所带来的 - 影响。规模较小的设计和实施变更，由相应级别的技术权威进行复审。批准、控制级别以及它们传达的方式分别在 *配置管理计划* 和 *软件开发计划* 中说明。

级别 3，已定义的

组织流程重点

组织流程重点的目的是建立软件流程活动的组织职责，这些活动提高了组织的整体软件流程能力。组织流程重点活动产生的主要结果是一组软件流程资产，这些资产在组织流程定义中有描述。如集成软件管理所述，软件项目使用这些资产。

目标 1：在组织范围内协调软件流程开发和改进活动。

Rational Unified Process 是一个迭代式流程，它依赖于在多次迭代中重新制定“同一”已定义的流程。流程制定的重复性质、状态指标的评估、以及每一阶段和迭代获得的经验教训都提供了在连续的各次迭代中,选择适用的流程-的机会。

目标 2：软件流程的优缺点根据相对于流程标准进行鉴别。

Rational Unified Process 代表了一个整体软件开发流程，可对其进行定制，以便在某一类型的项目中更有效地使用它。 *环境工作流程* 提供关于如何定制 Rational Unified Process 的指南。除了技术和管理复杂性外，在项目中可用于确定流程“形状”的一些流程判别标准有：

- ☐ 业务环境（投机或者内部的合同）
- ☐ 软件开发工作量的大小
- ☐ 创新程度
- ☐ 应用类型

目标 3：计划组织级别的流程开发和改进活动。

该级别 3 目标完全依赖于采用该级别的组织。

组织流程定义

组织流程定义的目的在于开发和维护一套适用的软件流程资产，提高项目的流程性能，为组织提供不断积累的长期利益的基础。这些资产提供了一个通过培训等机制实现制度化的稳定基础，这在培训计划中进行说明。

目标 1：为组织开发并维护一个标准的软件流程。

Rational Unified Process 在这方面居于领先地位，用作组织的基线软件开发流程，可对其进行发展、定制和维护。

目标 2：收集、复审与软件项目使用组织的标准软件流程有关的信息，并使之可用。

这一目标需要得到采用 Rational Unified Process 的组织的支持。

培训计划

培训计划的目的在于发展个人的技能和知识，以便他们能高效地履行其职责。培训是组织的职责，但软件项目应该先确定所需要的技能，并在项目需求独特时提供必要的培训。

目标 1：计划培训活动。

这个目标只有采用 Rational Unified Process 的组织才能实现。然而，Rational Unified Process 是一个“行业最佳方案”知识库，它提供了关于如何开展各种软件开发活动的指南、概念和详细的分步说明。因此，Rational Unified Process 本身就是一个优秀的培训材料来源。

然而，Rational Unified Process 还需要相关的支持课程，包括：

- ☐ Rational Unified Process 概述，包括需求、分析设计、实施、测试、构架、流程配置、管理、工具等几个模块，以及对面向对象的介绍。
- ☐ 通过用例来实现需求管理(RMUC)
- ☐ 面向对象的项目管理(OOPM)
- ☐ 面向对象的设计分析(OOAD)
- ☐ 软件质量自动化
- ☐ 配置管理
- ☐ 软件构架和迭代式流程

目标 2：为培养履行软件管理和技术职责所需的技能和知识而提供培训。

目标 3：软件工程组和其他软件相关小组的个人接受必要的培训，以便履行其职责。

采用 Rational Unified Process 的组织需实现这些培训计划目标。然而，Rational Unified Process 提供了一系列的课程，如上节所述。

集成软件管理

集成软件管理的目的在于将软件工程和管理活动集成到一个一致、确定的软件流程中，该流程是根据组织的标准软件流程和相关流程资产定制的，这在组织流程定义中有描述。如软件产品工程所述，该定制是根据业务环境和项目的技术需要进行的。集成软件管理是从级别 2 的软件项目规划和的软件项目跟踪与勘察演进得到的。

目标 1：项目的已定义软件流程是组织标准软件流程的一个定制版本。

与 Rational Unified Process 环境工作流一致，Rational Unified Process 的标准交付版本是可配置并且可以根据各种类型的项目调整使用规模。

目标 2：根据项目的已定义软件流程计划并管理项目。

这一目标需由采用 Rational Unified Process 的组织来说明。

软件产品工程

软件产品工程的目的是为了统一执行一个明确定义的软件工程流程，将所有的软件工程活动进行集成，以便高效地生产出正确、一致的软件产品。软件产品工程描述项目的技术活动，如需求分析、设计、代码和测试等。

目标 1：定义、集成并统一执行软件工程任务，以便生产软件。

Rational Unified Process 活动以及对每个角色需要什么定义，在项目必备计划工件的背景下，确保确定、分配和完成任务。Rational Unified Process 内在的迭代式开发流程可以迅速证明软件开发团队的效力，并提供对最终产品的评估。

目标 2：软件产品互相保持一致。

工程模型（用例模型、设计模型、源代码和可执行组件）之间的可追踪性通过环境进行维护。

组间协作

组间协作的目的是为软件工程小组积极参与其他工程小组的工作提供一种方法，以便项目能更好地高效满足客户的需要。组间协作是集成软件管理的跨学科的方面，它超越了软件工程的范围。不但软件流程应该集成，软件工程小组与其他组的交互也必须协调和控制。

目标 1：客户的需求得到所有项目涉及的团队的同意。

使用用例而不是其他“正式”需求规约方法作为需求获取和说明的依据的一个重大好处在于，用例容易被项目干系人理解。同样，Rational Unified Process 用例需求获取方法代表所有项目干系人可以对需要执行的任务达成一致意见。这在流程中得到进一步的贯彻执行，并反映在作为软件开发基础的模型和复审中。

目标 2：工程组之间的承诺得到项目涉及的团队的认可。

这个目标需由采用 Rational Unified Process 的组织来说明。然而，Rational Unified Process 可视模型有助于人们理解产品开发各个阶段（从需求获取到部署）有什么要求。Rational Unified Process 变更和配置管理流程确保提议的变更得到适当评估，并将评估结果向所有项目干系人传达。

工程组确定、追踪并解决组间问题。Rational Unified Process 迭代式开发流程有助于通过连续集成所有已开发软件来及早发现软件问题。为提出和解决团队间的问题，对多个团队开发的软件进行集成时产生的问题可作为一个“公共空间”。Rational Unified Process 检测和变更请求流程支持这种观点，该流程提供了一种正式机制，用于捕获、追踪并解决项目开发问题。

平级复审

平级复审的目的是为了及早有效地排除软件工作产品的缺陷。一个重要的结果就是加深了对软件工作产品以及可以防止的缺陷的理解。平级复审是软件产品工程提出的一个重要而有效的工程方法。

目标-1：计划平级复审活动。

如在级别 2 的质量保证目标中所述，Rational Unified Process 内的每个活动都有一个独立的复审活动。

由于早期问题检测能降低整体成本，因此 Rational Unified Process 提倡“及早并经常”所有工件，尤其是关键工件进行平级复审。Rational Unified Process 提供了每一阶段、每一模型内要复审的重要特性清单。

目标- 2：识别和排除软件工作产品中的缺陷。

Rational Unified Process 工件复审员需要确定用于下一个开发阶段的工件是否准备就绪。如果工件未达到复审“合格”标准，根据 Rational Unified Process 指标计划，需要获取以下方面的详细资料：

- ☐ 稳定性（返工类型、易变性）
- ☐ 适应性（返工成本）
- ☐ 模块度（返工影响范围）
- ☐ 质量（缺陷发现频率、密度、继承深度）
- ☐ 成熟程度（每次故障的测试时间）
- ☐ 资源耗费配置文件（计划的与实际的）

参考资料：

[REF1] Mark C. Paulk et al, "Key Practices of the Capability Maturity Model - Version 1.1", Software Engineering Institute - Carnegie Mellon University.



两家总部：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
电话：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
电话：(781) 676-2400

免费电话：(800) 728-1212

电子邮件：info@rational.com

Web: www.rational.com

全球网址：www.rational.com/worldwide

Rational、Rational 徽标和 Rational Unified Process 是 Rational Software Corporation 在美国和 / 或其他国家或地区的注册商标。Microsoft、Microsoft Windows、Microsoft Visual Studio、Microsoft Word、Microsoft Project、Visual C++ 和 Visual Basic 是 Microsoft Corporation 的商标或注册商标。其他所有名称均仅用于标识目的，它们是其相应公司的商标或注册商标。ALL RIGHTS RESERVED.

Copyright 2006 Rational Software Corporation.
如有更改，恕不另行通知。