

**Rational** software

# IBM Rational需求定义 与管理解决方案 软件交付平台介绍手册(二)





## IBM Rational软件交付平台概览

在《软件开发平台整体介绍》文档中,介绍了IBM Rational软件交付平台主要包括的几个基本服务组件(如图):过程和组合管理、需求定义和管理、架构管理、质量管理和配置、变更和发布管理。在本文档中,主要介绍需求定义和管理部分。



IBM Rational软件交付平台集中体现了以下软件交付的基本原则:

- 有效的需求定义和管理: 强调软件开发由业务驱动和用例驱动, 通过有效地定义和管理需求及其变更来高效率的构建应用系统, 提升产品交付速度和能力;
- 敏捷和迭代化开发: 有效控制项目风险、增加项目预见性、尽早地发现软件产品中的缺陷;
- 以架构为中心: 采用可视化建模技术来构建以构件为基础的系统框架, 有效地管理系统的复杂度, 增强系统的灵活性和可扩展性;
- 持续质量验证: 在整个产品生命周期中持续地验证软件质量, 确保产品满足客户的需求, 并且构造一个高性能、高可靠的软件系统;
- 管理软件资产和变更: 在整个产品生命周期中管理好企业的软件资产, 并对所有的变更请求进行管理, 支持团队的并行开发。

本文将介绍IBM Rational软件交付平台的需求定义和管理服务组件, 通过有效的需求定义和管理能够保持需求和业务目标的一致性, 提高软件开发质量, 加速软件交付过程。

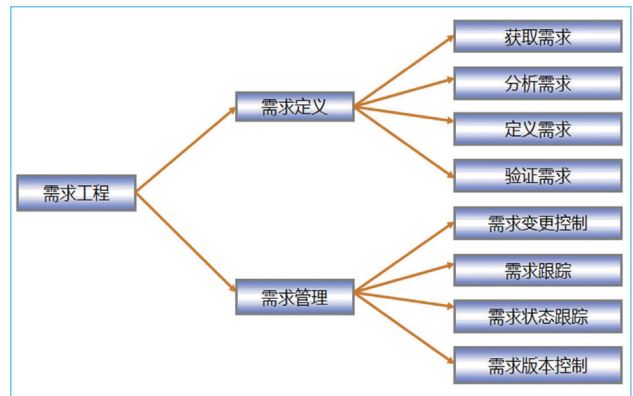
下一章节我们先探讨一些需求工程的基本概念和最佳实践, 以便充分理解IBM的需求定义和管理服务组件是如何有效支撑和实现这些最佳实践的。

## 需求工程基本概念与最佳实践

需求工程是应用已证实有效的技术、方法进行需求分析, 确定客户需求, 帮助分析人员理解问题并定义目标系统的所有外部特征的一门学科。它通过合适的工具和记号系统地描述待开发系统及其行为特征和相关约束, 形成需求文档, 并对不断变化的需求演进给予支持。

需求工程是随着计算机的发展而发展的, 在计算机发展的初期, 软件规模不大, 软件开发所关注的是代码编写, 需求分析很少受到重视。后来软件开发引入了生命周期的概念, 需求分析成为其第一阶段。随着软件系统规模的扩大, 需求分析与定义在整个软件开发与维护过程中越来越重要, 直接关系到软件的成功与否。人们逐渐认识到需求分析活动不再仅限于软件开发的最初阶段, 它贯穿于系统开发的整个生命周期。

需求工程包括需求开发和需求管理(包括变更管理)两个方面, 如下图所示:

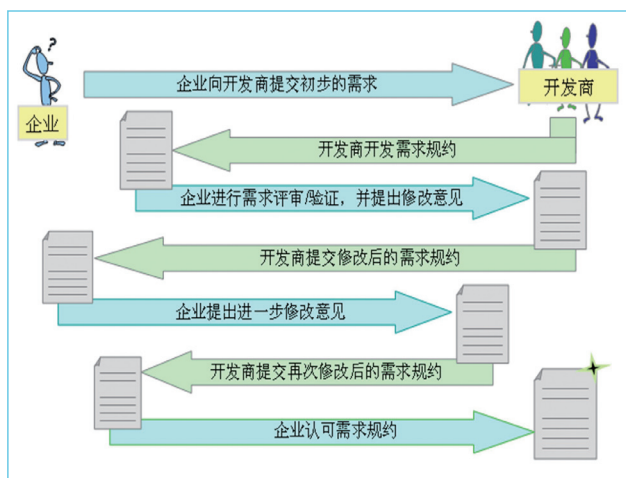


需求工程是软件工程中最复杂的过程之一, 其复杂性来自于客观和主观两个方面。从客观意义上说, 需求工程面对的问题几乎是没有范围的。由于应用领域的广泛性, 它的实施无疑与各个应用行业的特征密切相关。其客观上的难度还体现在非功能性需求及其与功能性需求的错综复杂的联系上, 当前对非功能性需求分析建模技术的缺乏大大增加了需求工程的复杂性。从主观意义上说, 需求工程需要方方面面人员的参与(如领域专家、领域用户、系统投资人、系统分析员、需求分析员等等), 各方面人员有不同的着眼点和不同的知识背景, 沟通上的困难给需求工程的实施增加了人为的难度。

## 需求定义

需求定义是一个不断反复的过程。传统的软件开发方法假设可以一次性得到完全正确的需求，长期的开发实践证明该假设过于理想化，在实际工作中是不成立的。现代软件开发实践表明，很难一次性得到完全正确的需求，项目的开发总是伴随着不完全正确的需求而进行的，关键在于如何有效降低需求不确定性的风险。

一个行之有效的方法就是不断反复地进行需求的开发与验证。典型的过程如下图所示：



## 需求管理

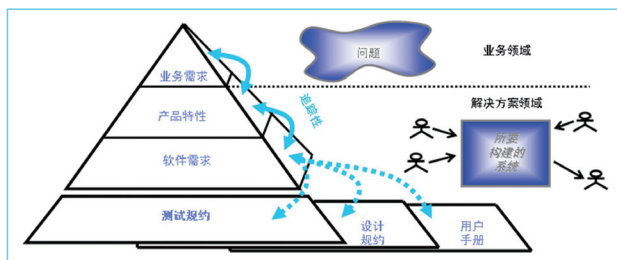
由于需求是正在构建的系统所必须遵照和符合的，而且是否符合需求决定了项目的成败，因此找出需求是什么，将它们记下来，进行组织，并在发生变化时对它们进行追踪，这些活动是非常有意义的，这个过程就是需求管理。

简单的说，需求管理是：

- 一种获取、组织并记录系统需求的系统化方法；
- 一个使客户与项目团队对不断变更的系统需求达成并保持一致的过程。

## 需求管理模型

需求是分不同层次和类型的，如下面的“需求金字塔”所示，不同层次的需求和同一层次的需求之间都可以建立追踪关系：



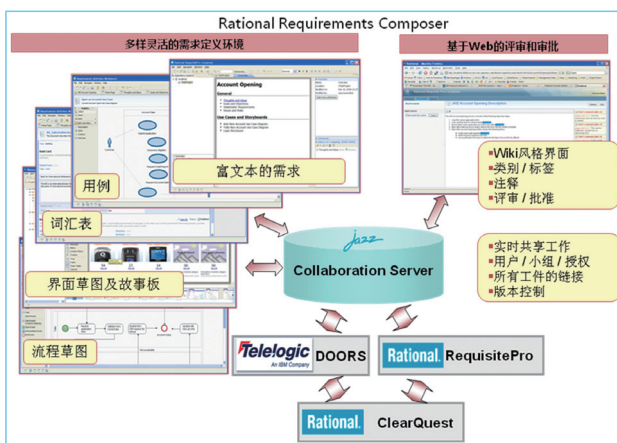
为了建立一个真正满足客户需求的系统，项目团队首先必须确定系统要解决的问题。然后，团队必须确定涉众，从中获得业务和用户需要，对其进行描述，并区分它们的优先级。从这一组高层期望或需求出发，对产品或系统特性集达成一致意见。而后，由产品特性来抽取软件需求，在IBM Rational的需求管理模型中，软件需求是以用例模型的方式来描述。从测试的角度来看，测试项一定来自于软件需求，即软件需求中确定了哪些需求项，测试就要根据这些需求项来制定和实现。

通常一类需求可以分解成其他类型的需求，对需求进行分类可以使项目更容易管理。

## IBM Rational需求工程解决方案

需求定义与管理是软件项目中一项十分重要的工作，对于需求进行有效的工程活动，将奠定项目成功的基石。据调查显示在众多失败的软件项目中，由于需求原因导致的约占到45%，因此，需求工作将对软件项目能否最终实现产生至关重要的影响。

基于需求工程最佳实践，IBM Rational提供了如下的需求工程解决方案，它由Rational Requirements Composer (简称RRC)、RequisitePro、DOORS和变更管理工具组成，具有如下的架构：





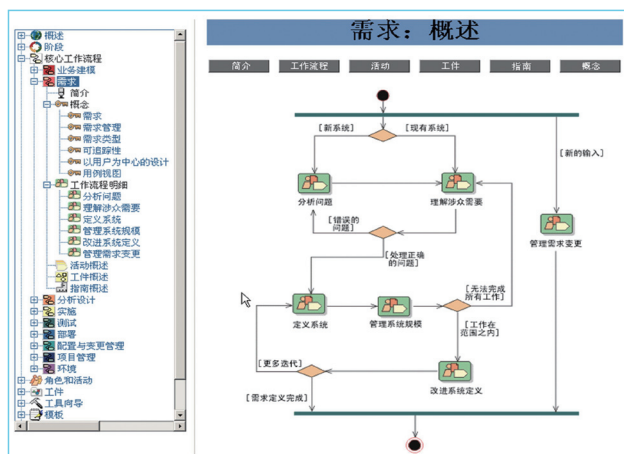
其中, RRC用于定义需求, DOORS和RequisitePro用于管理需求, ClearQuest用于管理需求变更。

## 需求工程最佳实践

需求工程最佳实践对于保证需求的质量具有重大意义:

### ■ 明确定义需求工程过程

多数软件项目都是基于团队协作的, 为了保证团队所有需求相关人员能够步调一致地进行合作, 必须为他们提供统一的工作流程和方法, 以便相互之间能够顺利地进行沟通与协作, 确保在统一的流程和方法指导下进行协同的需求开发和管理。



### ■ 业务建模——保证完全理解客户或者行业业务过程

业务建模架起现实世界与计算机系统之间的桥梁, 有效的业务建模可以为需求的正确性、无二义性、易于理解等质量维度提供保证。

通过业务建模手段来描述组织的组织架构, 描述当前的业务过程, 找出当前存在的问题并确定改进的可能性。而改进通常都是通过自动化现行业务过程中的若干环节, 来优化用户的业务过程, 提高业务效率和/或业务质量, 以期带来更多的商业价值。

### ■ 用例建模——站在用户的立场上来描述需求

用例建模是被业界广泛接受的需求开发与描述方法, 有效的用例建模可以为需求的全部质量维度(正确性、完备性、一致性、无二义性、易于理解)提供保证。

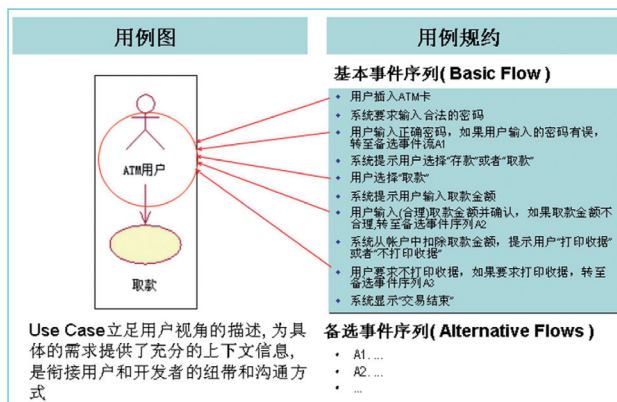
用例建模站在用户的角度, 从目标系统的外部以一种可观测和可

验证的方式描述目标系统的预期行为, 描述系统如何与最终用户以及/或其他系统进行交互。与简单的需求列表方法不同, 用例按照叙事的方式讲述用户可能怎样使用系统。因此, 用例易于被业务部门理解, 可以清晰定义目标系统的边界。用例模型在系统功能与功能的最终用户之间建立了明确的关联, 便于用户业务部门安排合适的需求评审人员对其相关需求进行评审, 既可以提高需求评审的效率, 又可以保证需求的正确性。

用例主要描述系统的功能性需求, 系统的非功能性需求则通过需求补充规约进行描述。

用例模型的主要元素包括用例图和用例规约。用例图以图形的方式描述主角(Actor, 即系统外部与系统交互的用户或其他系统)和用例的关系。可以从不同的角度建立多个用例图来帮助直观理解系统的功能需求。用例规约则以叙事的方式讲述用户可能怎样使用系统。

下图是ATM用户取款的事用例模型示例:



用例是衔接用户业务与开发团队的纽带, 用例同时也是项目开发计划、测试案例以及系统文档撰写的基础, 因此在项目中扮演举足轻重的作用。

### ■ 采用统一的需求描述手段

采用统一的需求描述手段可以增强需求的可理解性和一致性, 便于对需求内容进行审核, 保证需求的正确性。

统一的需求描述手段包括项目前景文档、用例图、用例规约、补充规约等, 并建立规范的需求文档模板, 采用集中手段管理需求(如借助需求管理工具将全部需求信息存于关系数据库中)。

## ■ 综合运用多种需求定义技术

辅助的需求开发技术有：用户界面原型和需求研讨会。

通过用户界面原型可以及早收集业务部门的反馈，验证对需求理解的正确性，及早发现并纠正问题，有效降低纠正需求理解错误的代价。

需求研讨会在项目的早期是非常有效的需求开发手段。用户方面的与会者包括管理者、项目经理、业务部门代表等；项目开发团队方面的与会者包括项目经理、业务分析员、架构师、开发人员代表等。通过需求研讨会可以集思广益，协助开发团队理解用户的业务过程，收集用户的业务需求，及早纠正对需求的错误理解，把握需求的优先级等。

## ■ 分类并且量化管理需求

系统越大越复杂，出现的需求类型就越多。需求类型对需求进行分类，通过确定需求类型，团队可以把大量需求组织成意义明确且更容易管理的组。在一个项目中建立不同类型的需求有助于团队成员对变更请求进行分类，并使相互之间的沟通更为清楚明确。不同类型的需求之间要建立起追踪关系，以便方便的进行追踪和影响分析。

为了对需求进行进一步的管理，我们还可以通过对需求的深入分析，给每条需求添加必要的属性，如优先级、来源、稳定性、成本、难易程度、风险等等，实现需求的量化管理。通过对需求进行量化管理，有助于加深对需求的理解，方便对需求进行查询和排序。

## ■ 管理不同类型需求之间的追踪关系

用户的业务需求逐步会转化为目标系统的软件需求，但业务需求与软件需求之间并非简单的一对一的关系，相反往往是一个多对多的复杂关系，即某个业务需求可能与多个软件需求相关，反过来，某个软件需求可能也与多个业务需求关联。有效管理业务需求和软件需求之间的追踪关系可以确保所有的业务需求都被软件需求覆盖了，从而保证了需求的完备性。

管理业务需求和软件需求之间的追踪关系还可以帮助开发人员了解需求的来源，帮助项目经理管理项目的规模，评估需求变更对项目的影 响，管理需求的变更，评估测试失败对需求的影响等。

可以借助需求管理工具来自动化管理业务需求和软件需求之间的追踪关系。

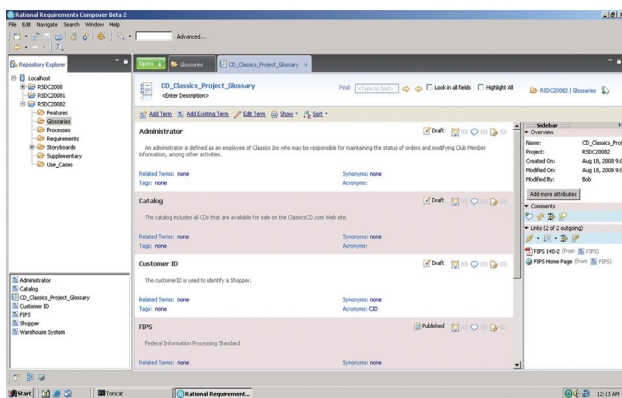
## 需求定义

实现有效的需求工程，第一步就要解决需求的定义问题。项目的干系人众多：客户、业务部门、开发人员、维护人员等等，造成了需求的来源众多，而且渠道多样：开会收集的、市场调研的、电话沟通的、电子邮件交流的等等。但由于缺乏必要的系统平台，于是各种各样的需求就四散在各地，难免造成需求的遗漏，更无从进行有效的管理。RRC提供了术语及词汇表维护、业务流程建模、用户界面草图、场景描述、故事板和富文本格式的需求规格，有效解决了需求的定义和确认过程中的各种问题。

## 获取定义业务词汇

在需求定义中，必须使用业务中最常用的术语和表达方法来定义常用词汇。此后您还应该 在业务的所有文本说明中始终如一地使用常用词汇。这样可以 从需求工程的一开始就保持需求领域说明的一致性，从而避免项目成员对术语的使用及其含义产生误解。

RRC提供了术语和词汇表的定义、同义词关联、缩略语对等以及状态等级等功能，同时通过方便的超链接功能使词汇表的履历与项目需求的定义保持动态的一致。并且可以通过标签(Tag)功能对术语和词汇表进行基线化。



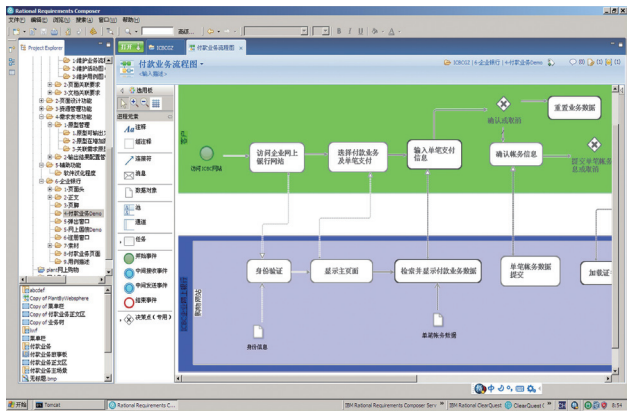
## 通过业务建模, 深入理解用户的需求

由于企业业务活动的多变性与复杂性，要准确理解目标系统的业务需求，首先必须了解目标系统的应用环境，也就是企业的业务过程与协作流程。因此，对于收集后经过评估和分派的原始客户需求，就需要由领域专家对客户 的业务流程进行业务建模，与客户深入交流，以深入理解客户现有的工作流程，并对其进行相应的优化，发现可

以通过IT系统实现的环节,进而将业务需求转化为软件需求,从而为今后的设计、实现指明方向。

RRC提供基于Business Process Modeling Notation (BPMN) 2.0的子集符号协助客户业务人员和开发人员创建和优化业务流程,同时将业务流程与业务目标和业务对象相关联,并通过泳道图对象流时序流信息流等建模方式构建业务模型,从而实现:

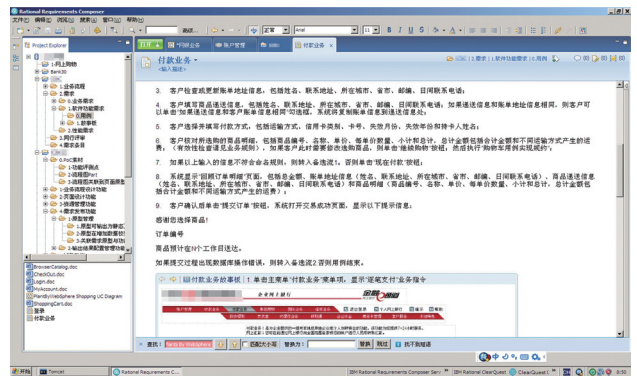
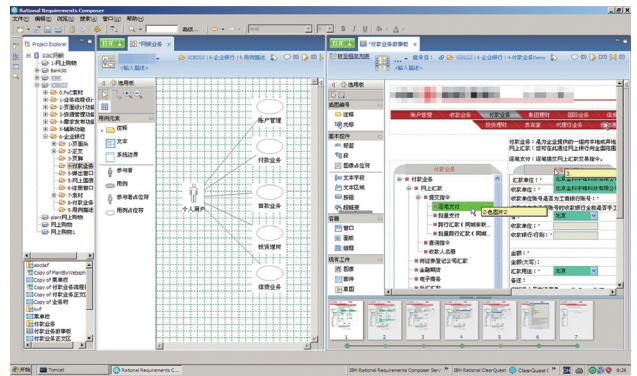
- 调研客户各部门的业务活动情况,了解客户的组织中当前存在的问题并确定改进的可能性;
- 确保客户、用户和开发人员就项目的业务需求达成共识;
- 导出客户所需的业务需求;
- 有助于明确项目或产品的用户需求和功能性需求。



### 从用户角度出发,准确描述软件的功能需求

用例建模是被业界广泛接受的需求开发与描述方法,用例建模站在用户的角度,从目标系统的外部以一种可观测和可验证的方式描述目标系统的预期行为,描述系统如何与最终用户以及/或其他系统进行交互。用例易于被业务部门理解,可以清晰定义目标系统的边界。用例模型在系统功能与最终用户之间建立了明确的关联,便于企业业务部门安排合适的需求评审人员对其相关需求进行评审,既可以提高需求评审的效率,又可以保证需求的正确性。用例主要描述系统的功能性需求,系统的非功能性需求则通过需求补充规约进行描述。

RRC以基于用例(Use Cases)的需求组织方式和及其灵活的超链功能,将最初的基于富文本(Rich Text)的需求描述按照客户的业务流程逐步细化,并结合以原型、词汇术语的有机组合,最终以立体的方式展现客户的软件需求规格。



### 利用原型准确把握用户界面技术,使系统达到最佳的可用性

原型作为一种软件需求定义方法,代表了开发者对于如何实现一个特定的用例的一种观念。另一方面用户对原型的评价可以指出软件需求中的许多问题,在你开发真正产品之前,可以最低的费用来解决这些问题。

同时原型作为一种软件需求定义工具,用它可以探索不同的用户界面技术,使系统达到最佳的可用性,并且可以评价可能的技术方案。

RRC提供了故事板(Storyboards)、草图(Sketch)、屏幕流(Screen Flow)和组件(Part)等多样性的手段,为开发客户原型提供了最佳平台。

当用户在考虑原型中所提出的功能可否使他们完成各自的业务任务时,原型使用户所探讨的问题更加具体化。您可以利用故事板并结合“行为原型”(behavioral prototype)在屏幕上显示出用户界面的正面像,并允许这些界面之间的一些导航,在包含必要的功能基础上对把对话元素细化为特定的屏幕显示、菜单和对话框。一旦确定了软件需求并勾画出屏幕的大体布局,你就可以从最佳使用的角度设计每一个用户界面元素的细节。比起直接从用例的描述跳



跃到完整的用户界面的实现，然后在软件需求中发现重大错误，利用逐步求精的方法所花费的努力将会更小。

利用草图(Sketch)可以摒弃书面原型(paper prototype)以及简单的画图工具的缺陷，它可以把一个系统某部分是如何实现的呈现在用户面前。成为探索特定的用户界面设计的合适工具。同时结合组件(Part)功能可以使你迅速地实现并更改用户界面，而不管在界面后面的代码效率的高低。

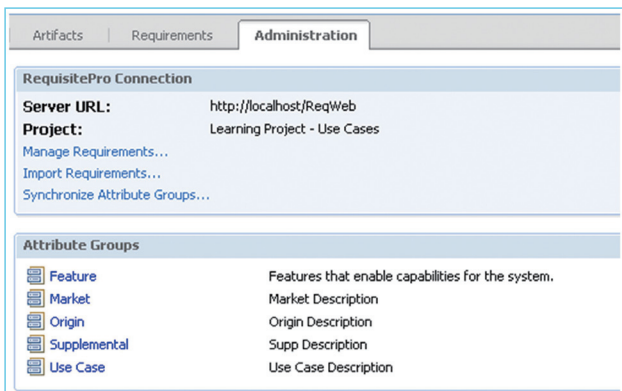
最终您可以将以上的各种原型元素以屏幕流(Screen Flow)的方式来展现用例中的活动顺序，提供一个基于文字需求的满足业务场景的功能场景。



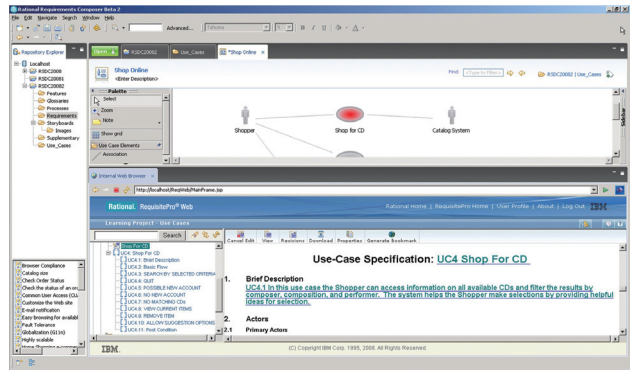
## 需求管理

### 基于RequisitePro实现需求管理

对于已经开发出来的需求，需要进行有效的管理。业界领先的需求管理工具RequisitePro能够与RRC完美集成，如下图所示，在RRC中可以管理RequisitePro中的需求，可以将RRC定义好的需求导入到RequisitePro中，并能够实现双方需求属性的同步：

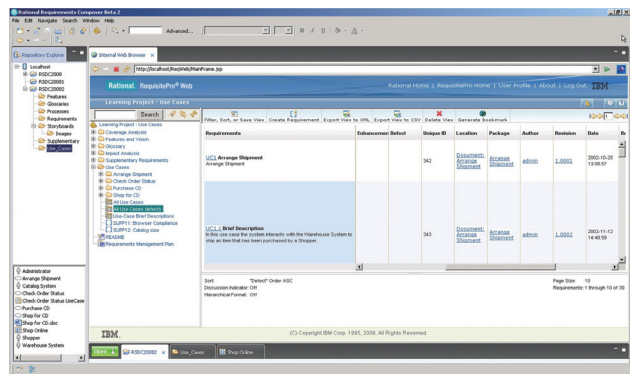


同时对于特定的需求我们还可以将其展现在Microsoft Word文档中，并且可以通过与RequisitePro的集成，直接在Word中对后台需求库中需求进行编辑修改。而对于一般用户、业务人员以及企业领导，RequisitePro还提供了方便易用的Web界面，可以查看项目需求的进展情况。



### 对需求进行多角度分析, 实现量化管理

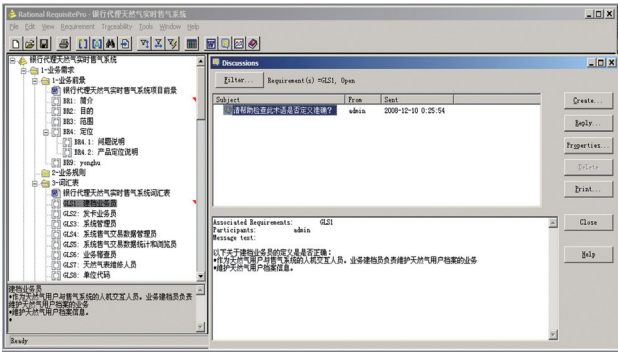
为了对需求进行进一步的管理，我们还可以通过对需求的深入分析，给每条需求添加必要的属性，例如：优先级、来源、稳定性、成本、难易程度、风险等等。然后，我们就可以利用RequisitePro提供的数据库的强大检索能力，通过选择不同的属性作为过滤的条件，建立不同的视图，让我们能够从不同的视角对现有的需求加以审视和权衡，以确定项目的范围和开发的重点。



### 记录需求变化的历史, 交流需求修改的意见

在RequisitePro中对需求采取了条目化的管理方式，能够保留每条需求的变化历史，为以后的过程分析积累了数据。另外RequisitePro中还提供了项目需求基线的管理功能，可以建立项目级的需求基线以作为开发的基础，需求的基线可归档到集成的配置管理系统中(例如ClearCase)，以保留项目级需求基线的版本。通过

RequisitePro提供的需求基线的比较功能,可以方便的查看项目需求基线的变化情况。另外,RequisitePro中针对每条需求还提供了类似BBS的讨论页面,可以记录对于需求的看法以及修改的建议等等,并且经过配置,相应的讨论内容还可以通过Email通知相关人员,这样大大促进了对于相应需求的理解和沟通。



### 基于DOORS实现需求管理

对于定义好的需求,需要进行进一步的分解、跟踪和管理,除了RequisitePro之外,还可以借助于在业界排名第一的需求管理系统DOORS。关于RequisitePro与DOORS的定位与区分,请参见后续章节的专门阐述。在RRC中定义的需求,可以放在DOORS中进行跟踪管理,以确保需求被正确地实现。

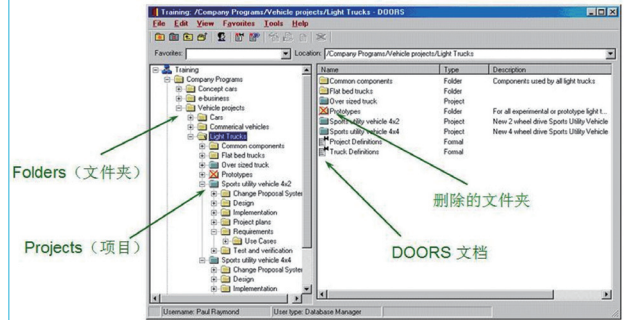
#### 集中的需求存储库

DOORS的数据库可以存放组织中的所有系统和项目的开发过程文档数据,包括用户需求、系统需求、系统架构、设计、测试计划、测试案例、测试结果等。

在DOORS的数据库视图中,用户可以自己定义组织和项目的结构。

#### DOORS数据库视图

没有层数限制,可灵活伸缩



组织你的项目

### 维护需求的追踪性,分析变更造成的影响

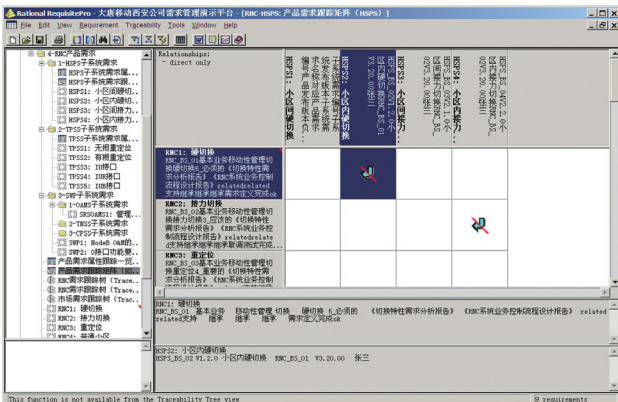
根据需求管理的基本理论,需求具有不同的层次:业务需求、产品特性、软件需求、测试需求,这些层次的需求间具有密切的联系,高层次的需求发生改变必然会影响到其下各层相关的需求,因此为了进行有效的需求管理,必须实现对需求追踪性的管理。

RequisitePro中对需求的追踪性管理提供了多种方式,如:需求追踪矩阵视图,需求追踪树视图,需求追踪属性页等等。通过需求追踪矩阵视图可以方便地设置需求间的追踪关系,对于已建立追踪关系的需求,一旦发生改变,RequisitePro会自动在相应的追踪箭头上打上红色的斜杠加以标识,这样通过追踪矩阵视图或需求追踪树视图就可以对需求的变更进行影响性分析了,以防止具体的需求未能及时根据变化进行更新。尤其是需求追踪树视图,可以非常直观的展现受某条需求影响的所有需求,是说服客户慎重变更需求的有效手段。

通过需求信息的集中管理,可以保障所有相关人员在同一个需求视图上。

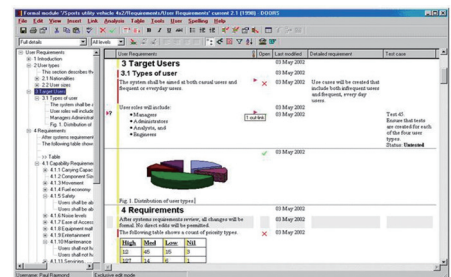
#### 结合文档和电子表格的单一可视化视图

在DOORS的文档(Formal Module)界面中,可以同时展示类似Word文档界面和类似电子表格界面的单一可视化视图。



#### DOORS文档界面

在一个窗口中你可以做所有的事情!



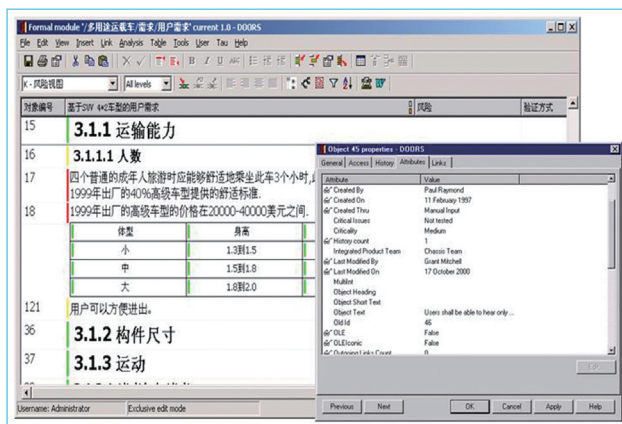
改善效率,减少错误,确保质量



通过结合文档和电子表格的显示视图，可以将一个角色所需要的相关信息都集中动态地展示出来，极大地提高了工作效率和质量，大大减少由于沟通不好带来的问题。

**没有限制的自定义属性**

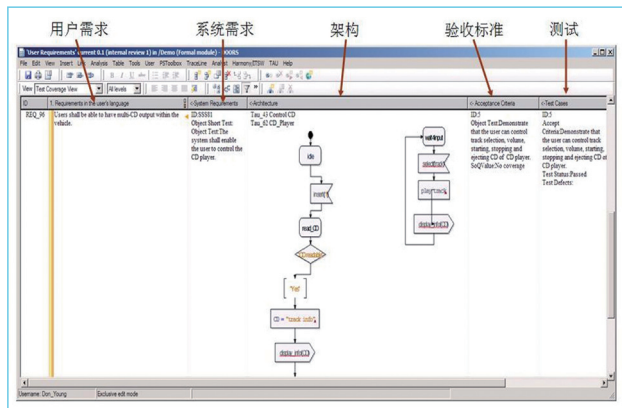
用户可以在DOORS中基于自己的开发过程定义不同数据类型的属性，支持字符串、文本、日期、枚举、数值等，没有数量限制。DOORS中的属性可以支持使用扩展脚本进行计算和统计度量。



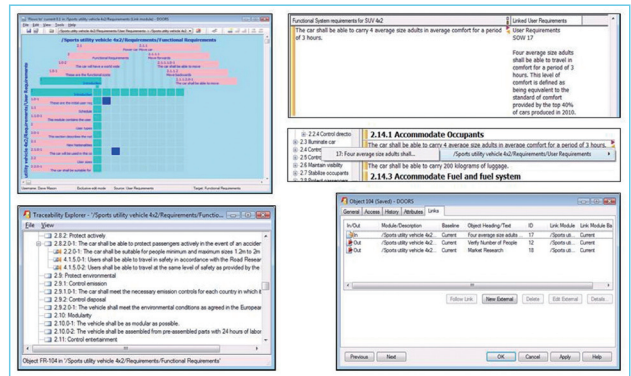
通过用户自定义的属性，可以进一步深入需求管理，如优先级、风险管理、成本管理、需求确认、需求评审等。

**丰富的追踪关系展示**

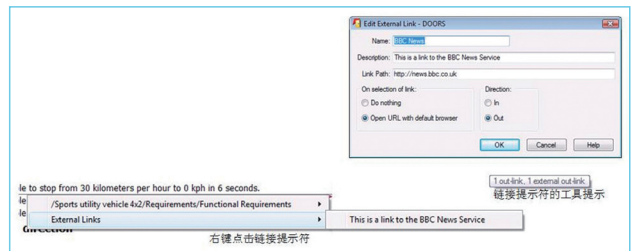
在DOORS中，可以在单一视图中显示丰富的追踪信息，例如：从用户需求追踪到系统需求，架构，验收标准，测试。



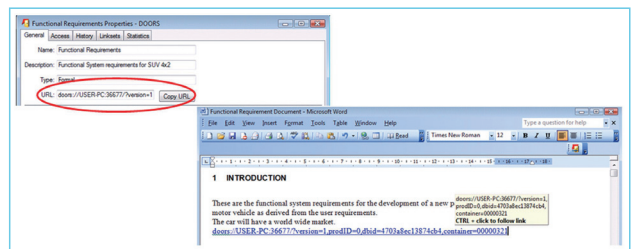
在DOORS中，可以使用多种方式查看追踪关系，包括如下方式：追踪矩阵，追踪视图，追踪树，追踪链接，追踪属性窗口。



在DOORS中，支持建立外部链接，如http、共享路径等。通过将追踪能力扩展至DOORS之外，鼓励更多的人按照需求进行工作。



DOORS中的数据库、项目、文件夹、文档或对象都具有DOORS URL，通过使用URL可以用户带到DOORS中。



DOORS除了项目内部的追踪关系，还可以管理不同项目之间的需求追踪关系。

通过丰富的追踪能力，DOORS可以将不同角色的人员带到一起，极大地加强了人员沟通和协作。

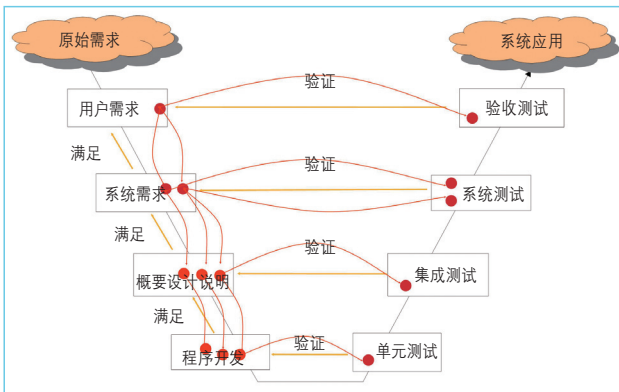
**追踪分析**

在DOORS中可以进行多种追踪分析。

**■ 影响分析**

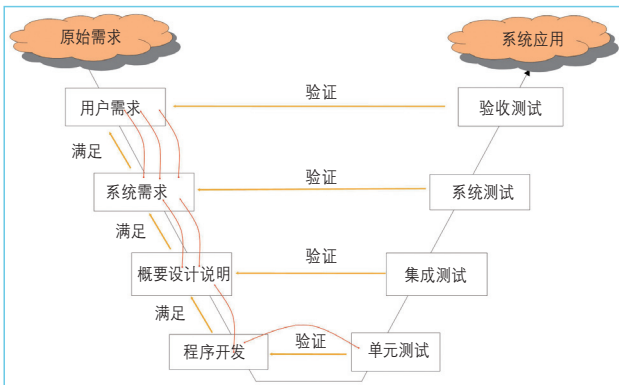
影响分析用来发现在项目生命周期中某个层次的需求发生变化时对

其它层次需求的波及影响。该影响分析应能够进行量化，并与原需求保持一致关联。



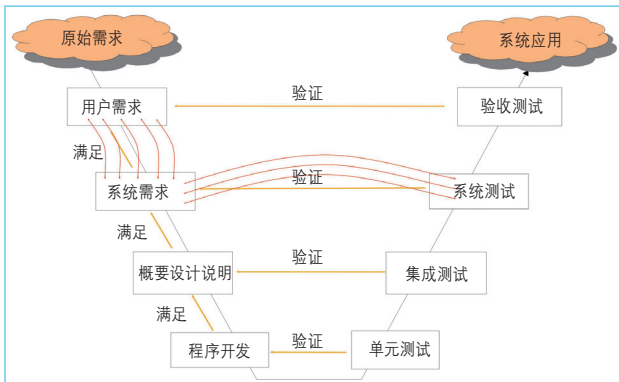
■ 来源分析

来源分析用来确定下层需求的来源是什么，以便在需求分析中很好地在不同层次需求之间进行沟通。



■ 覆盖率分析

覆盖率分析用来查看某个层次的需求是否都被下一个层次所覆盖，是否有缺漏，并且是否都已有测试来验证以及验证是否都已通过。



例如，为了实现一个项目需求，需要在多个系统中完成多个功能，可以通过自动化的方式检查该项目需求是否都已在应该涉及到的系统需求中得到了满足。

DOORS的可疑链接提示功能可以在需求项旁边直接产生标记来通知链接的另一方需求已经发生了变更。并且当被通知的一方相应调整后使得该链接不再是可疑链接时，用户能够将此可疑链接恢复为正常链接，从而不再作为可疑链接显示。

■ 历史版本和基线管理

在DOORS中的每个需求对象在修改时都会自动记录历史版本，历史版本可以进行比较、恢复和查找。

DOORS中的Formal Module文档可以建立基线，并进行基线查看、比较和电子签名。

DOORS中支持对项目中的部分文档建立统一的基线集定义，在项目不同里程碑将不同文档基线加入项目基线集中，以保证一致的项目需求基线。

■ 权限管理

DOORS既能够创建用户，又能够根据角色建立用户组。DOORS能够对各级数据库结构(包括文件夹，项目夹，文档，需求项，属性等)设定只读，修改，删除，创建和管理五个不同的权限，这些权限即可以对单个用户设定，也可以对用户组设定(使得用户组中的所有用户具有相同的权限)，从而使不同的用户或用户组对数据库每一层结构具有不同的权限控制。

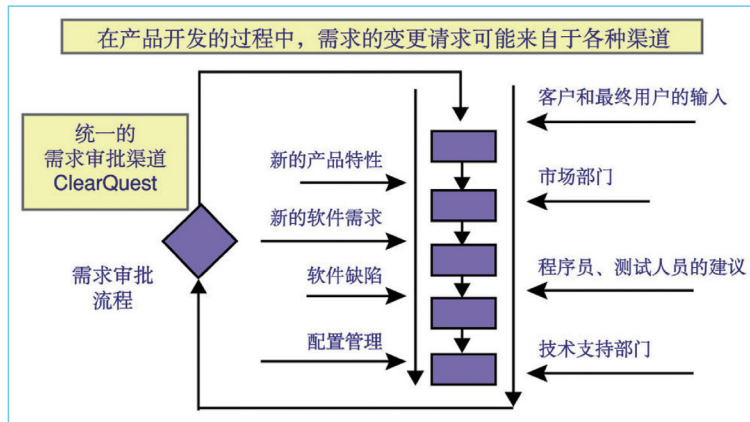
■ 二次扩展脚本支持

DOORS支持二次扩展脚本DXL。通过DXL，DOORS可以扩展界面显示、统计计算、自动化处理、扩展用户功能，以及与其他外部系统进行集成。

需求变更管理：控制需求的变更，防止需求蔓延

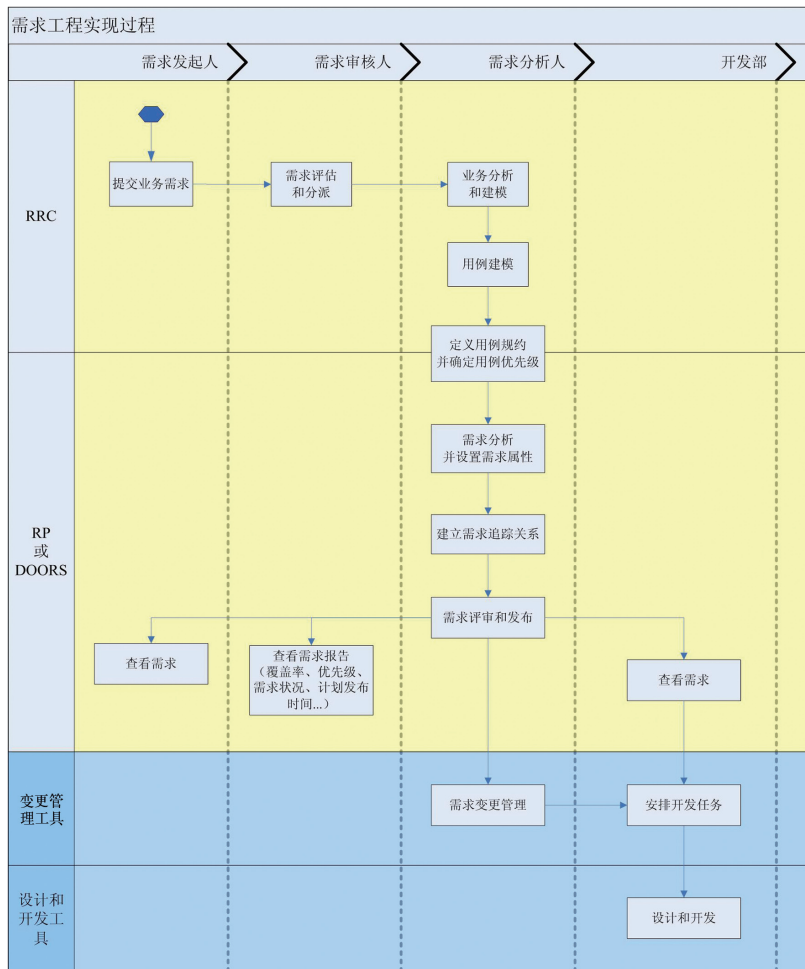
对需求变更的严格管理是最大限度降低变更风险，减少运营过程中出现差错的有效方法。在项目的开发过程中，需求的变化是不可避免的，但是该如何处理需求的变更呢？除了前面所述的进行需求变更的影响性分析以外，建立必要的审批流程，防止任意的和无休止的修改需求，也是对需求进行有效管理的必要环节之一。

对于需求变更的管理，我们选择使用ClearQuest来建立相应的需求变更管理流程，对所有的需求变更请求进行统一的评估和审批。ClearQuest通过流程定制、界面友好的客户端和Web等不同的操作模式，对来源广泛的需求变更请求进行管理。



## 需求工程实现过程

下图概要说明了基于Rational需求工程解决方案实现需求工程的过程：

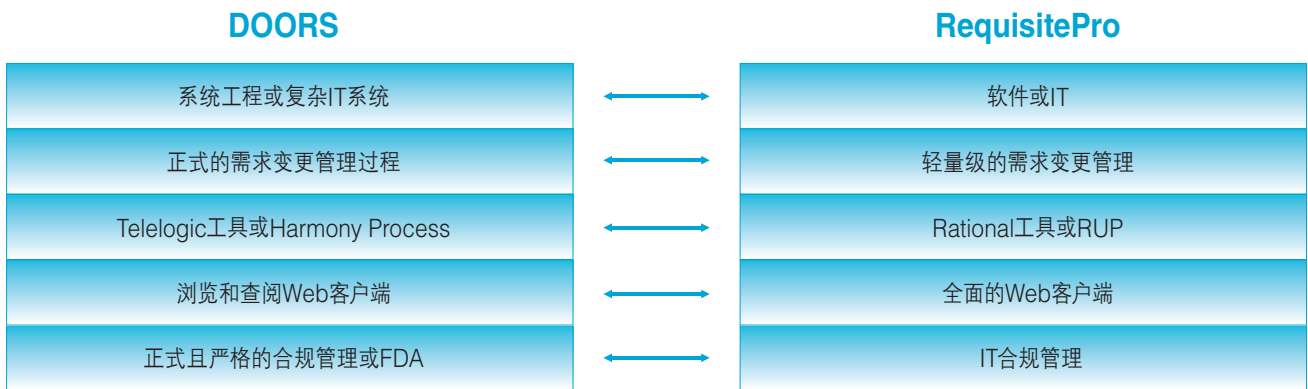


图中缩写说明：

- RRC-Rational Requirement Composer
- RP-RequisitePro

## 需求管理工具定位区分

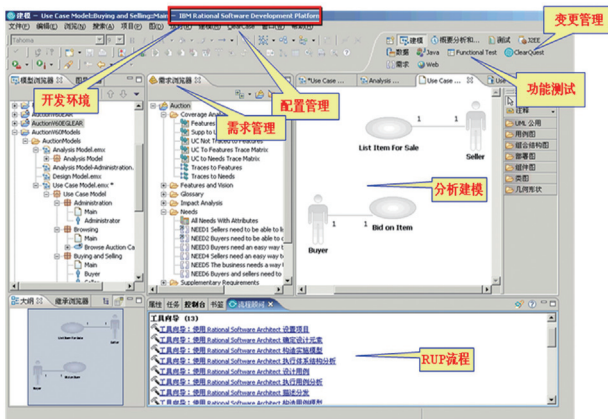
一种解决方案不可能适应所有客户或项目的需要, IBM Rational 提供不同的需求管理解决方案, 以适应不同客户和市场的差异化需求。以下是Rational需求管理系统RequisitePro和DOORS的定位区分说明。



一般来说, 客户管理需求的数量越大, 需求管理的过程越正式, 变更控制越严格, 那么选择DOORS越可能是一个正确的方案; 反之, 选择RequisitePro越可能是正确的方案。

## IBM Rational需求定义与管理解决方案的集成性

IBM Rational的需求工程解决方案可以在开发环境中得到完美的集成, 包括: 流程顾问、需求管理、分析建模、开发调试、测试管理、变更管理和配置管理, 给用户以全新的体验, 提供贯穿软件开发生命周期的端到端的完整解决方案。



## IBM Rational需求定义与管理解决方案的优势

大量的项目实践都表明, 很多质量问题往往是在需求阶段引起的, 客户的业务部门之间、业务部门和IT开发团队之间对于系统的需求理解并不完全一致, 开发过程中设计结果会逐渐偏离原始需求等等。毫不夸张地说, 需求管理的好坏直接关系到项目的成败。应该在需求分析阶段有足够的投入与参与。而业务团队参与需求管理的意识不强, 认为需求管理完全是开发团队的事, 只等着最后的项目验收确认, 其结果很可能就是项目的失败!

通过IBM Rational需求定义与管理解决方案, 能够给客户带来以下好处:

- 协助开发团队完全理解业务团队的业务需求。
- 解决由于业务要求的多变性与复杂性, 造成难以准确理解目标系统的业务需求的问题, 使业务人员和IT人员在了解目标系统的应用环境, 也就是业务过程与协作过程中。统一需求分析人员和业务人员的视角, 摒弃主观认识, 实现在完全理解业务需求的情况



下有序的进入设计、实现阶段,避免造成“差之毫厘,失之千里”的严重后果。

- 通过集中的需求存储库,使用协作、迭代的方法定义需求,改善跨业务和技术的沟通和协作,有助于加强对问题和解决方案的共同理解,从而促进所有涉众对需求达成一致。
- 通过跨生命周期的端到端的追踪性管理,增加整个开发过程中的需求状态和产品的可视性。
- 通过将需求与开发和测试关联到一起,防止客户需求被遗漏,并促使问题在生命周期较早期间被发现,从而减少返工成本,改进质量,提高客户满意度。

## 总结

### IBM Rational需求工程解决方案包含以下产品:

- IBM Rational Requirements Composer
- IBM Rational DOORS
- IBM Rational Change
- IBM Rational RequisitePro
- IBM Rational ClearQuest
- IBM Rational Unified Process

需求工程产品	内容介绍
需求定义工具 Rational Requirements Composer	<ul style="list-style-type: none"> <li>• 通过RRC能够使企业以及开发团队提高其需求定义过程质量。</li> <li>• 易于使用,高度可定制,提供各种可视视觉和协作工具,要求业务分析及开发人员能够捕获和完善的业务需求并提炼为明确的要求,从而推动需求质量和整个IT生命周期的提高。</li> <li>• 借助可视化的需求定义能力,有助于避免项目失败所要付出的高昂代价以及无休止的返工。</li> <li>• 提供实时协作以及可视的需求捕捉,借助流程回溯和优化、故事板、用户界面草图和富文本,从而更好地阐述和定义客户的需求。</li> <li>• 通过使用词汇表,在线评审和富文本编辑器,以更加明确需求的范围并进一步加强业务团队和开发团队的合作和沟通,使团队针对需求迅速达成共识并从而驱动后续的架构设计、编码、验证和测试并最终上线。</li> </ul>
通用IT系统需求管理 工具RequisitePro	<ul style="list-style-type: none"> <li>• 管理需求的追踪关系,提供需求追踪矩阵和需求追踪树的管理方式,并可自动标识需求变更产生的影响,进行需求变更影响性分析。</li> <li>• 为需求设置相应的属性,实现初步的量化管理,可查询需求的各种统计信息。</li> <li>• 通过RequisitePro的Web平台,给项目所有相关人员一个高效的沟通平台,让大家方便地了解到项目的需求。</li> <li>• 记录每条需求的修订历史,满足对于开发过程数据的审计需求。</li> </ul>
系统工程/复杂应用需 求管理工具DOORS	<ul style="list-style-type: none"> <li>• 管理复杂的追踪关系,提供跨生命周期的端到端的可视化追踪能力。</li> <li>• 通过结合文档和电子表格的单一视图提供丰富的信息展示能力。</li> </ul>
需求变更管理系 统工具ClearQuest/ Change	<ul style="list-style-type: none"> <li>• 建立需求变更管理流程,控制需求的变更。</li> <li>• 记录每条需求的修订历史,满足对于开发过程数据的审计需求。</li> <li>• 提供了电子签名和字段级的变更审计能力,满足遵从性(Compliance)要求。</li> </ul>
需求管理方法论 Rational Unified Process	<ul style="list-style-type: none"> <li>• 统一需求管理的过程和方法。</li> <li>• 统一需求描述模板和文档体系。</li> </ul>





