



IBM Rational 需求工程技术方案

概述

需求开发与管理是软件项目中一项十分重要的工作，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。

对于需求进行有效的工程活动，将奠定项目成功的基石。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处，难免造成需求的遗漏，更无从进行有效的管理。为了解决需求的定义和确认，通过RRC提供的术语及词汇表维护、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实现了：

- ▶ 定义业务需求
 - ▶ 识别涉众
 - ▶ 捕获业务目标和对象
 - ▶ 流程优化
- ▶ 定义用户需求
 - ▶ 识别角色和用例
 - ▶ 用例优先级排序
 - ▶ 用例细化
- ▶ 定义软件需求
 - ▶ 从用户需求转化为功能需求
 - ▶ 定义非功能需求

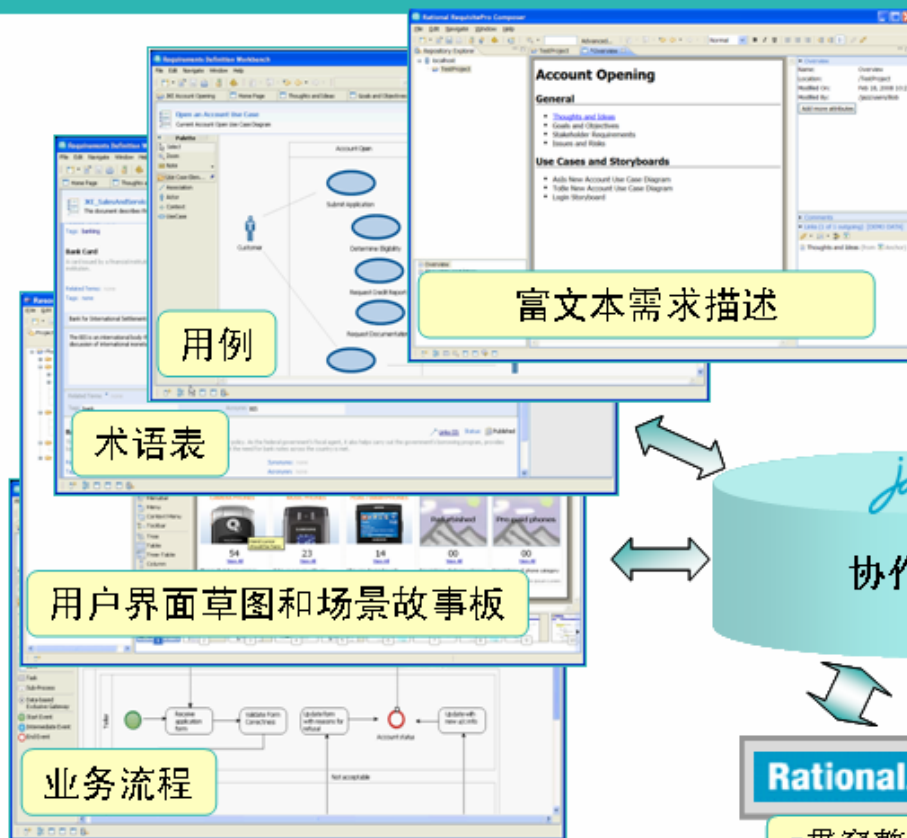
同时通过遵循必要的需求评估流程，对定义的需求进行确认和审批，形成需求规格说明并纳入需求管理。

系统越大越复杂，出现的需求类型就越多。一个需求类型不过是需求的一个类。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。

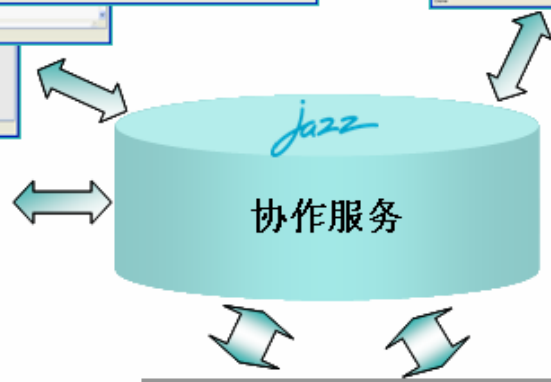
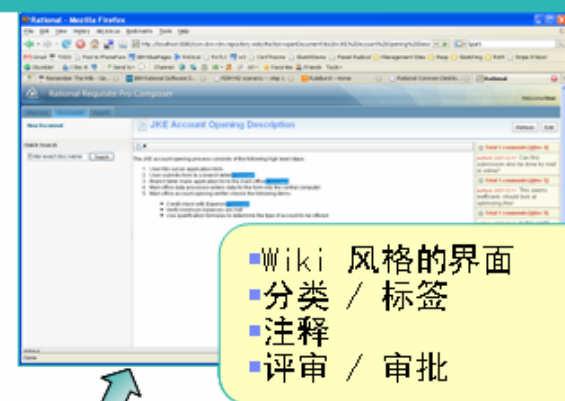
需求管理的重要内容之一就是需求跟踪。可以根据不同需求类型的导出情况，在不同的需求之间建立追踪关系。譬如，用户需要决定了要构建产品的特性，产品的特性又决定了产品的软件需求，等。在这些不同类型的需求之间建立关联，一旦其中的某些需求发生变化，就可以确定它可能带来的影响，从而制定相应的策略。

定义需求时无论怎样谨慎小心，也总会有可变因素。变更的需求之所以变得难以管理，不仅是因为一个变更了的需求意味着要花费或多或少的时间来实现某一个新特性，而且也因为对某个需求的变更很可能影响到其他需求。应确保赋予需求一个有弹性的结构，使它能适应变更，并且确保使用可追踪性链接可以表达需求与开发生命周期的其他工件之间的依赖关系。管理变更包括建立基线、确定需要追踪的重要依赖关系、建立相关项之间的可追踪性，以及变更控制等活动。

丰富的需求定义方法



基于Web方式的需求评审和审批

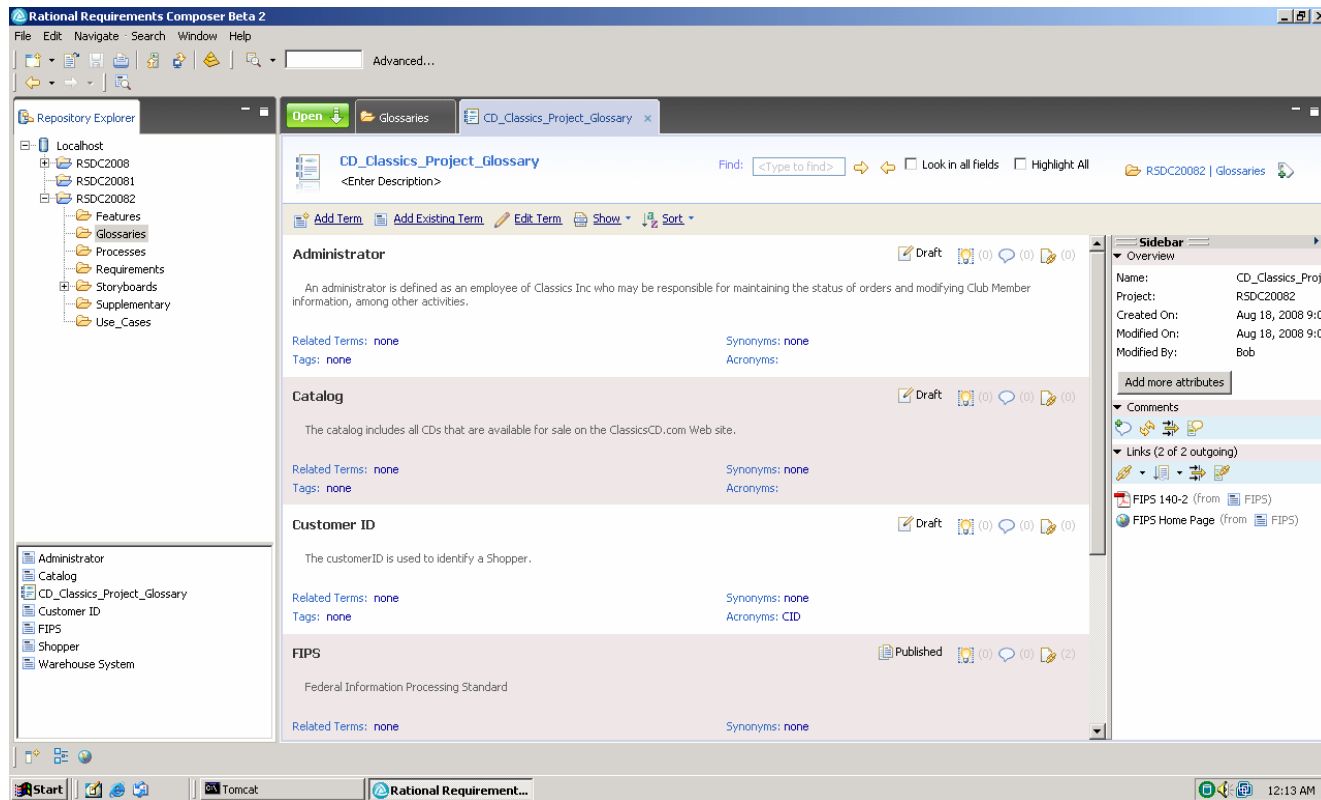


贯穿整个生命周期的需求管理

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。



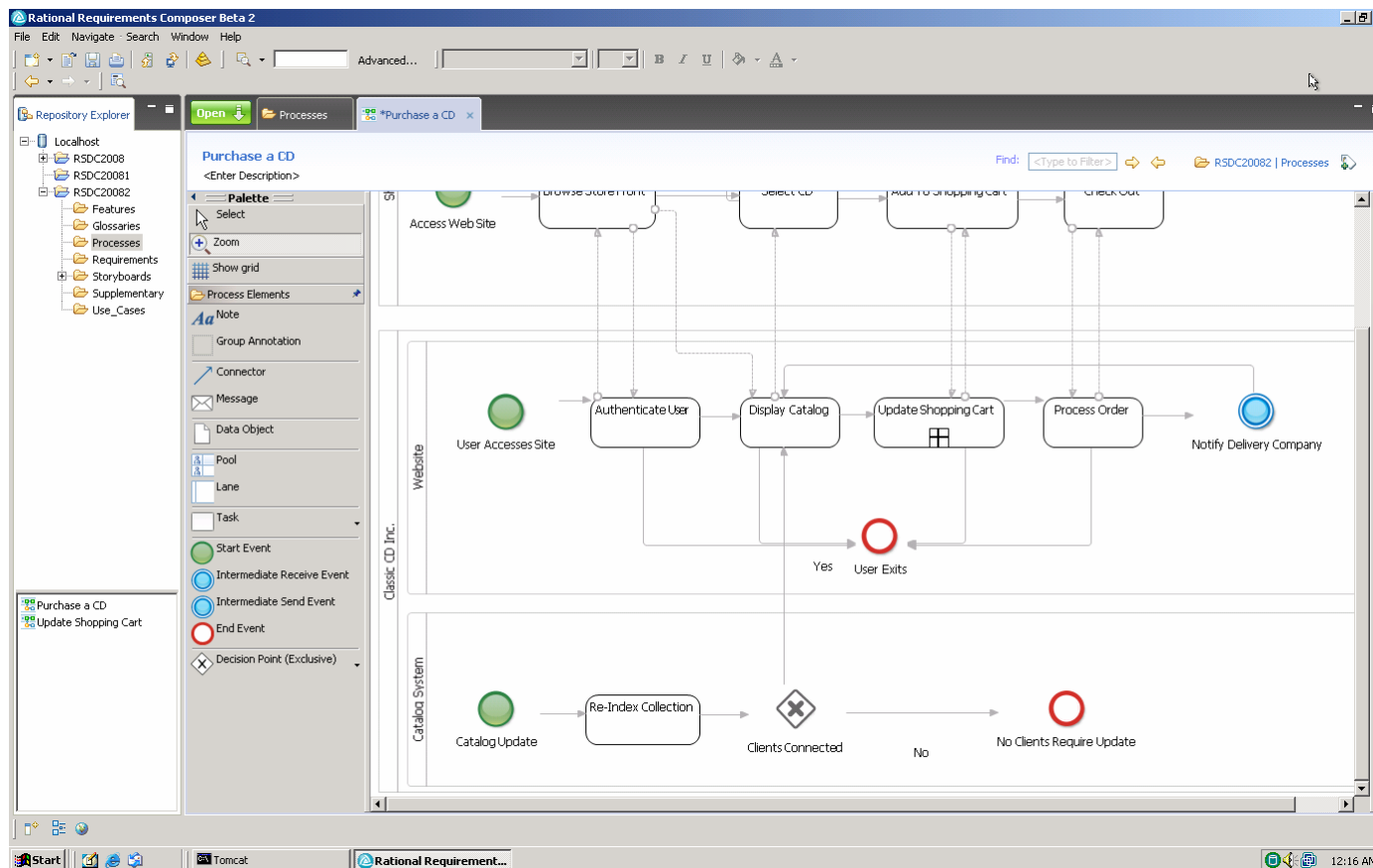
通过业务建模，深入理解用户的需求

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。

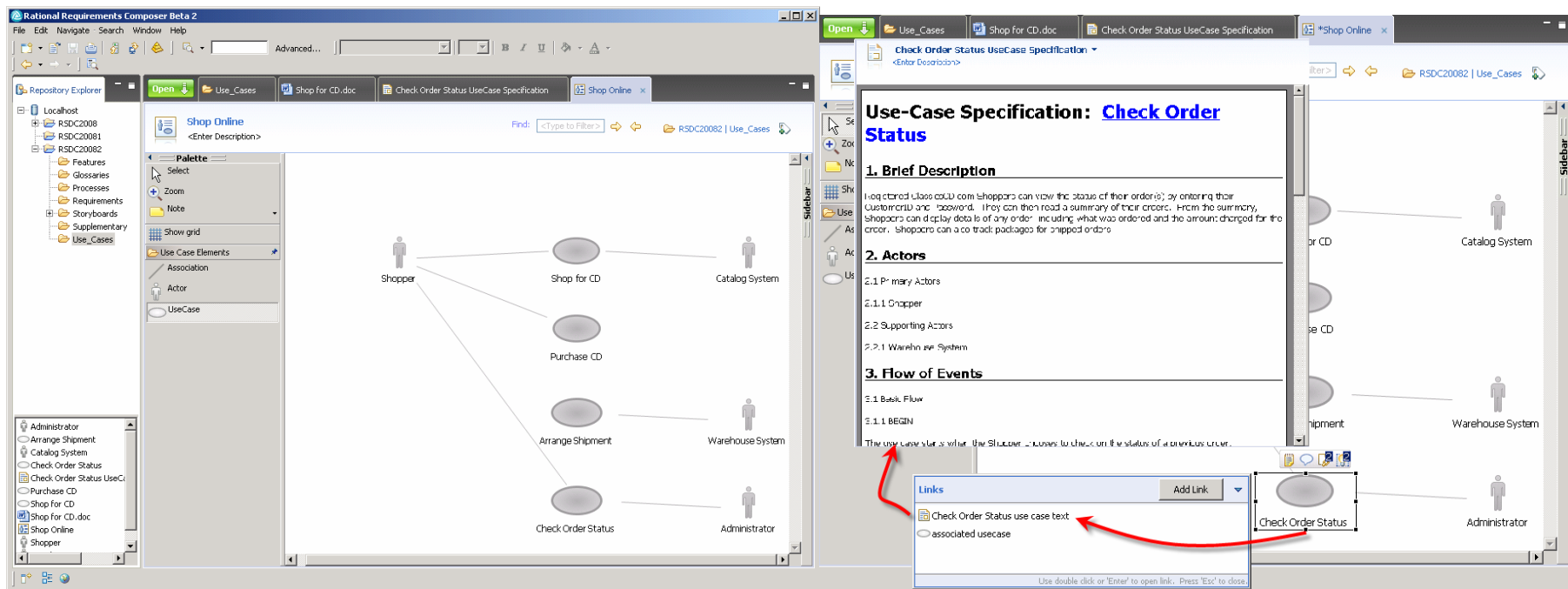


从用户角度出发，准确描述软件的功能需求

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(RichText)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。

Rational Requirements Composer Beta 2

File Edit Navigate Search Window Help

Advanced... Tahoma 10 B I U

Repository Explorer Open Features Storyboards Images Shop Story Shop Catalog Cart List Item

Localhost
 RSDC2008
 RSDC20081
 RSDC20082
 Features
 Glossaries
 Processes
 Requirements
 Storyboards
 Images
 Supplementary
 Use_Cases

Shop Story
 <Enter Description>

Frame List Create your storyboard frame list below. Pre

1. Enter CDClasses Web Site
Main page
2. View CD Catalog
The user browses through the catalog of CDs
3. Search for Beethoven
The user searches for Beethoven CDs
4. Search Results
The user is returned only the CDs from Beethoven
5. Search Results
The user clicks a search result to get more details
6. View Details of Beethoven CD
The full details of the CD are displayed
7. Add Beethoven CD to cart
The user adds a CD to the cart
8. Checkout
The user proceeds to checkout

Catalog
 <Enter Description>

Palette

- Select
- Marquee
- Note
- Cursor
- UI Elements
- Label
- Paragraph
- Image
- Text Field
- Text Area
- Hyperlink
- Checkbox
- Radio
- List
- Dropdown
- Item
- MenuBar
- Menu
- Context Menu
- ToolBar
- Tree
- Table
- Tree-Table
- Column

ClassicsCD.co

Shopping Cart

Item Name	Short	Price
Bach: Violin Concertos	Short	\$16.99
Beethoven: Symphonie Nr. 7	Short	\$12.99
Beethoven: Symphonie Nr. 5	Short	\$12.99

Storyboard

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Rational Requirements Composer Beta 2

File Edit Navigate Search Window Help

Advanced...

Repository Explorer

- Open
- F...
- S...
- I...
- *...
- *... x
- C...
- C...
- C...
- G...
- U...
- S...

Shop <Enter Description> RSDC20082 | Storyboards

Localhost

- RSDC2008
- RSDC20081
- RSDC20082
 - Features
 - Glossaries
 - Processes
 - Requirements
 - Storyboards
 - Images
 - Supplementary
 - Use_Cases

Administrator

- Arrange Shipment
- Catalog System
- Check Order Status
- Check Order Status UseC...
- Purchase CD
- Shop for CD
- Shop for CD.rnr

Palette

- Select
- Transition
- Nodes
- UI Activity

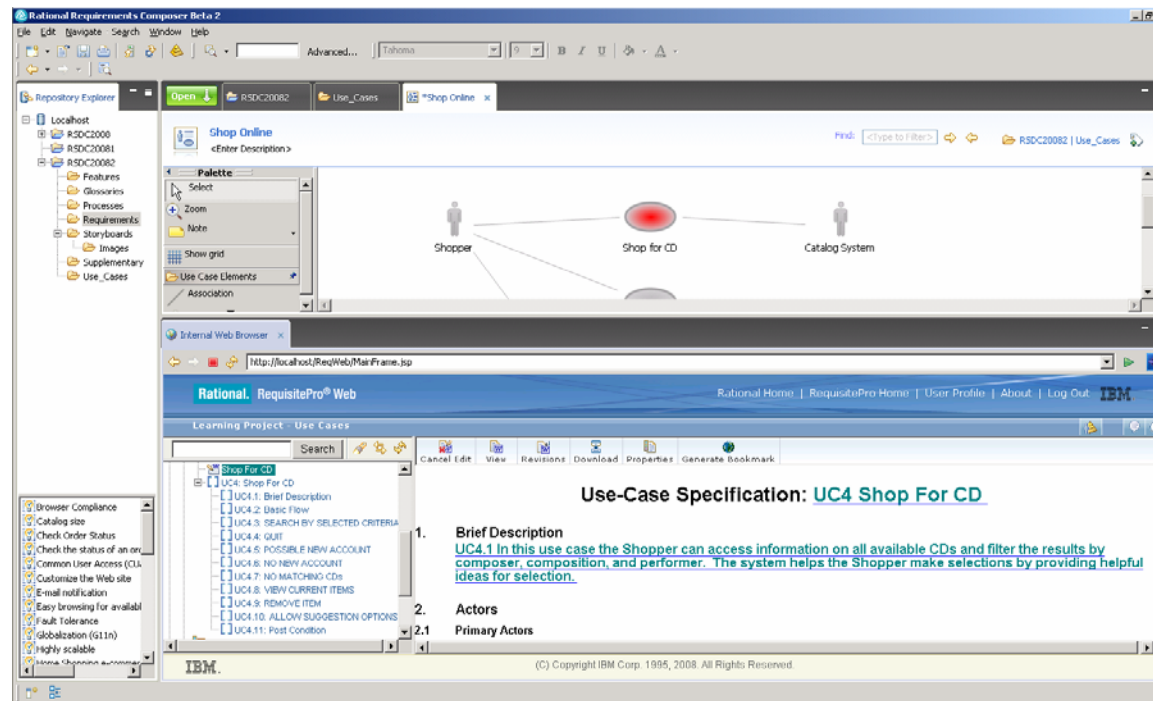
```
graph LR; Welcome[Welcome] -- shop --> Catalog[Catalog]; Catalog <--> ViewDetails[View Details]; Catalog -- checkout --> Checkout[Checkout];
```

Storyboard diagram showing transitions between screens:

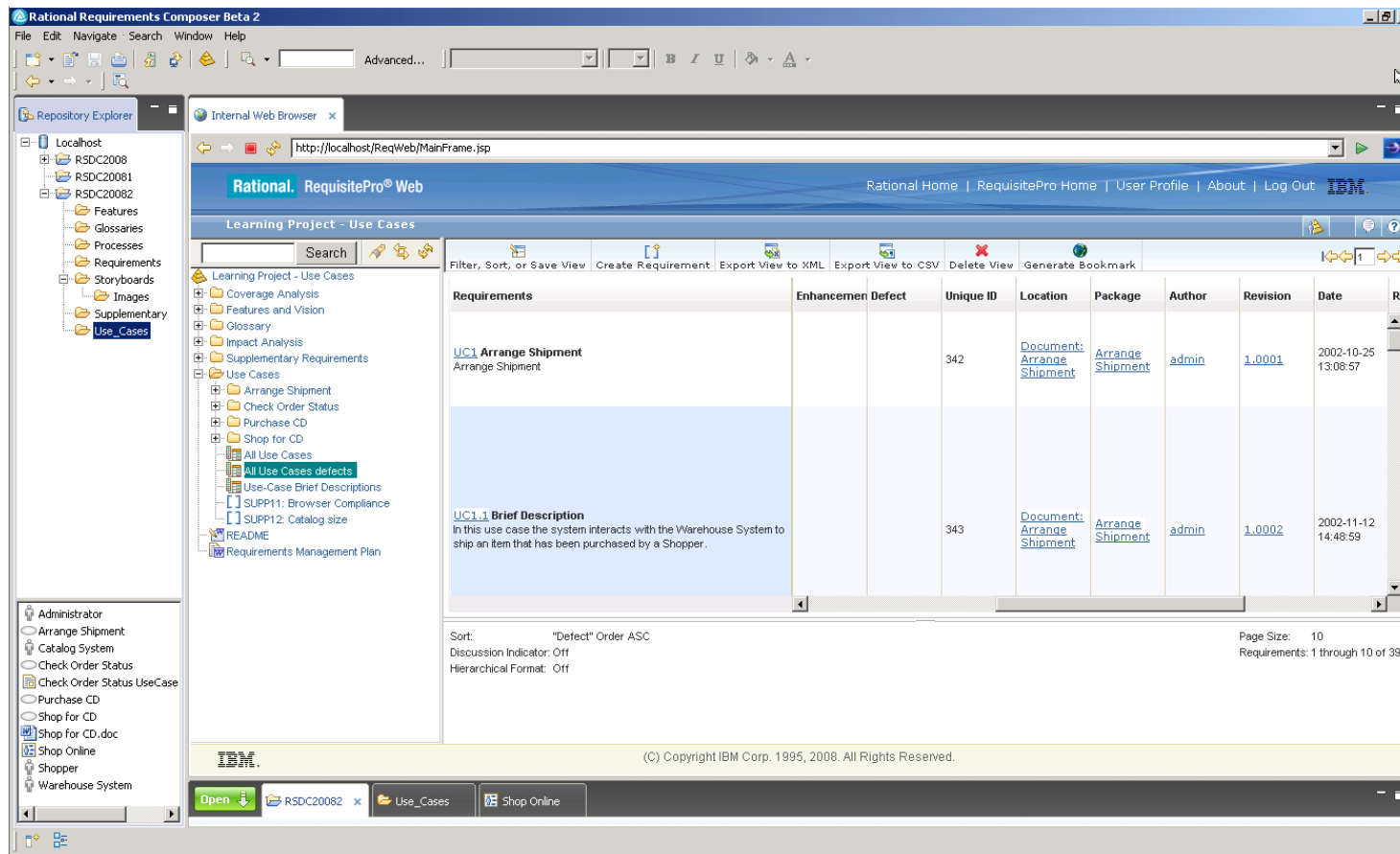
- Welcome screen transitions to Catalog screen via a transition labeled "shop".
- Catalog screen transitions to View Details screen via a transition labeled "View Details".
- Catalog screen transitions to Checkout screen via a transition labeled "checkout".

对需求进行多角度分析，实现量化管理

对于已经开发出来的需求，需要进行有效的管理，借助业界领先的需求管理工具 ReqPro，我们可以把 RRC 中的各类需求逐条记录到用数据库实现的需求库中，每条需求对应数据库中的一条记录，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 ReqPro 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，ReqPro 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。



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 ReqPro 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 ReqPro 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更历史，为以后的过程分析积累了数据。另外 ReqPro 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 ReqPro 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，ReqPro 中针对每条需求还提供了一个类似 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。

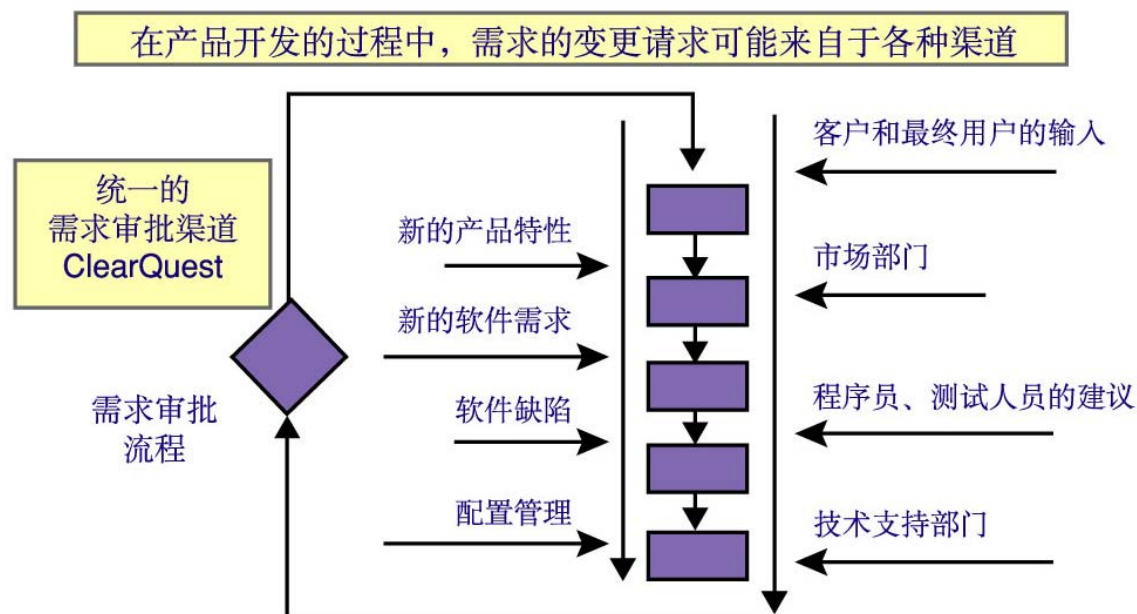
维护需求的追踪性，分析变更造成的影响

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。

ReqPro 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的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，ReqPro 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。

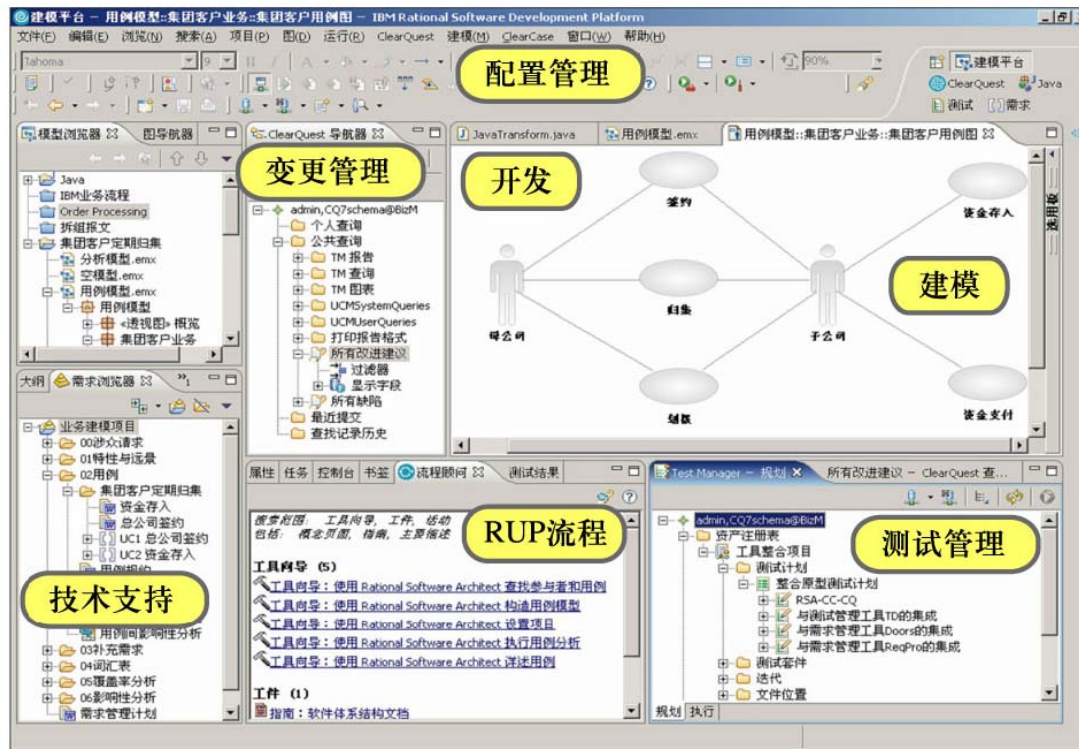
控制需求的变更，防止需求蔓延

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，我们仍然是使用 CQ 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。



紧密集成，贯穿软件开发生命周期

IBM Rational 的需求工程解决方案可以在 RSA 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.com/cn/software/rational