

Rational software

质量管理 软件交付平台介绍手册(三)



要想让软件更好地为业务服务，至关重要的一步就是**实施一种能够对软件项目生命周期全过程进行管理的软件交付流程及支撑平台**。

IBM Rational质量管理方案

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(如图一)：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。在本文中，主要介绍质量管理部分(见图二)。



图一



图二

测试管理自动化

测试管理是指对系统测试活动的管理，其主要目的是测准(有效选择运行测试用例，发现系统的缺陷)和测全(保证所有需求对被测试过)。

系统测试的依据是系统需求，系统测试人员只有准确把握需求信息，才能进行有效的测试。同时，测试人员的职责不仅仅是发现缺陷，还有帮助开发人员重现并解决缺陷的义务，应该为开发人员提供缺陷的相关信息，以帮助开发人员快速定位并解决问题。因此，测试管理不仅仅以测试用例为核心，还应考虑对需求和缺陷信息的管理，并建立需求、测试用例、测试脚本以及缺陷的关联。

测试管理自动化的主要目的是通过工具自动获得软件质量以及测试过程的相关信息，从而及时有效地指导软件测试。通过测试管理自动化，测试人员能把更多精力关注在如何设计有效的测试用例，如何有效选择执行测试用例，从而保证系统质量。

功能测试自动化

功能测试是指通过对软件系统界面进行操作，验证系统功能是否正常。传统的手工功能测试是通过测试人员手工操作被测系统界面，并通过眼睛来检查系统界面输出，从而验证系统功能。当测试用例需要输入大量测试数据或需要被多次执行时，传统的手工测试方法将花费大量的时间，而且容易出错。通过引入自动化测试脚本，并把测试输入数据以及期望结果和测试脚本建立关联，就可通过运行自动化测试脚本来完成测试用例执行，大大节省测试时间。

性能测试

性能测试是指通过模拟大量虚拟客户端来访问被测服务器，从而发现系统在性能和可靠性方面的问题。软件系统的性能问题主要分为应用程序方面和系统配置方面。传统的性能测试通常在系统上线前才进行，这时即使发现系统在应用程序方面的性能问题，但迫于进度压力，也不可能去大量修改应用代码，而只能对系统配置进行升级或优化，有时会导致不得不增加硬件投资。因此我们建议在软件项目的早期就引入性能测试手段，对软件架构进行性能测试，从而尽早发现并及时解决系统在应用程序方面的性能问题。

强化代码级测试

开发人员测试通常也称为单元测试或代码级测试，是指由开发人员

对自己开发的代码进行测试。强化代码级测试,可尽早发现软件代码缺陷,保证每个模块的质量,大大缩短后期模块集成测试和系统测试的时间,提高项目计划的准确性。此外,通过对测试程序进行有效的管理,可在应用的单元级进行回归测试自动化,从而确保软件项目能快速、高质量地实现新需求。

Web应用安全测试

如今,许多公司依赖基于Web的软件和系统运营其业务流程、与供应商进行交易、向客户提供更加成熟的服务。在一个治理结构良好的组织中,检查部署的每个应用程序上构建安全性应该是软件和

系统交付业务流程的一部分。不幸的是,为了保持竞争能力的领先地位,许多公司对此漠不关心,而只是不断地加快新产品的面市速度。结果,这些弱点让黑客得以轻松访问或偷窃公司个人数据,可能导致整个公司处于风险之中。

Web应用安全测试是指在Web应用上线之前,对应用安全漏洞进行检查,防止黑客利用安全漏洞对系统进行攻击。

为了让软件项目能切实解决上述五个软件质量管理方面的问题,IBM Rational提供了相应的测试工具,帮助提高测试的效率。(见表一)

类别	工具	描述
测试管理工具	Rational Quality Manager	完成对需求、测试计划、测试用例和缺陷的全面管理,提供手工测试,并可和测试工具之间集成。测试环境管理帮助测试人员降低30%测试成本。
功能测试工具	Rational Functional Tester	基于Eclipse平台的功能测试自动化工具,支持Browser、Java、SAPGUI、Siebel图形界面和字符终端界面,以及利用Visual Studio .NET开发的图形界面应用,测试脚本基于Java或VB .NET语言。
性能测试工具	Rational Performance Tester	基于Eclipse平台的性能测试工具,支持HTTP、HTTPS、SAP、Siebel、Citrix等协议,测试脚本基于Java语言。
	Virtual Tester Pack	提供模拟大量虚拟用户支持。比如100 Virtual Tester Pack,能支持利用Performance Tester或Robot完成100并发用户的性能测试。
代码级测试工具	PurifyPlus	支持对传统C/C++、Microsoft Visual Studio (包括6.0, 2003/2005等版本)代码的运行时分析,提供内存分析、性能分析和代码覆盖分析功能。
	Logiscope	Logiscope进行软件质量分析,软件代码规范检查和软件测试用例覆盖率检查等重要特性,帮助您在系统崩溃前发现并修复软件缺陷。
	Rational Test Realtime	对嵌入式C/C++的代码审查、组件测试、系统测试和运行时分析。

表一

软件质量管理最佳经验

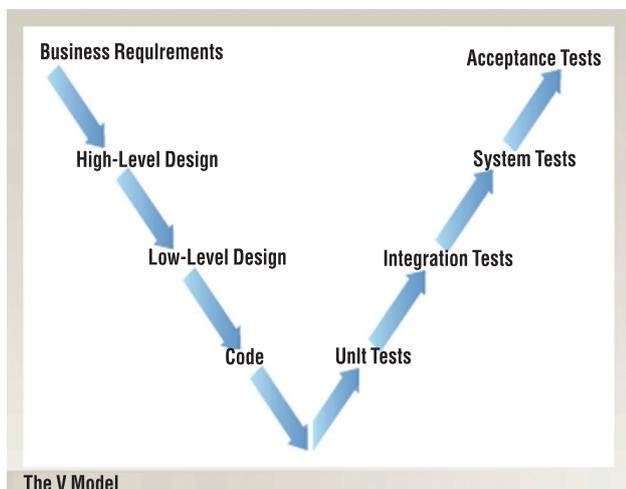
软件质量管理是指通过对软件的功能、性能、安全性和可靠性等质量要素进行验证,发现软件缺陷并帮助定位和修复缺陷,确保软件系统功能满足需求的活动。

IBM Rational根据多年软件工程研究和实践经验,认为通过在软件

项目中执行如下最佳经验,可以有效提高软件质量,确保项目进度。

■ 统一的需求变更管理流程

为了保证项目进度,根据V-模型,在需求阶段完成之后,开发机构在业务需求的基础上进行软件开发,而测试机构则应该基于同样的需求准备验收测试(Acceptance Test)。



然而,在实际的开发过程中,业务需求会不断发生变化,如果测试机构不能及时了解这些变化,将有可能导致大量的测试人力资源浪费。因此,为了避免这一问题,需要在业务部门、开发与测试机构之间建立起一套统一的需求变更管理流程,以保证在需求变化之后,测试机构可以马上对测试工作做出相应的调整。

■ 需求驱动测试

需求驱动测试组合了需求管理和测试管理在流程方面的最佳实践。测试有一些应该被重视的原则。它们从更高的层面上表示最佳实践(关于需求驱动测试方面)的元素。尽早计划和执行测试,撰写需求时就为每个需求计划测试。捕获需求的同时考虑测试,可以通过如何量化需求来改进表达需求这种方式。

需求驱动测试的最佳实践包括:

- 尽早计划测试
在需求编写时对每个需求的测试进行计划
- 尽早引入测试
在开发过程中尽早地执行测试
- 关联测试到需求
追溯测试到其所检查的需求
- 关联缺陷到需求
追溯缺陷到不被满足的需求
- 根据需求度量测试进度
设置目标,并根据那些被满足或不被满足的需求来度量测试的进度

需求驱动测试提供了一种需求管理和测试管理的集成方法,使得:

- 需求分析师能够交付包含全部验证标准的可测试需求
- QA或测试人员能够根据一致的需求集进行测试开发
- 发布管理能够基于需求质量度量而进行,而不是基于测试通过或失败的统计

IBM DOORS/RRC是市场和技术都领先的需求管理解决方案。通过改进需求的沟通和合作,从而提高系统工程和关键业务的IT项目的质量。

DOORS/RRC通过增加业务目标、客户需求、技术规格和规定的可见性来提高质量。

通过下列各种强大功能(捕获、链接、分析和需求变更及其跟踪),DOORS/RRC可以确保遵从要求和遵守规定/标准。

DOORS/RRC提供用于管理需求和测试用例的集成环境。通过DOORS/RRC和质量管理工具或测试管理工具的集成,企业可以创建从需求至测试的链接,所以您可以:

- 定义测试用例并将它们链接至原始需求;
- 开发测试用例以测试需求;
- 策划和记录测试运行的结果;
- 比较不同测试运行的结果,以查看发生变更的内容;
- 询问需求以了解需求覆盖率和测试状态。

■ 统一的、工具支持下的测试工作流程

作为有一定规模的测试团队,需要通过一个统一的测试流程来规范测试活动,明确每个角色的职责和所承担的任务,这样才能保证整个测试团队能有条不紊地完成测试项目。

由于实际测试中很可能会出现一个人同时测试多个子项目的情况,在这种情况下,如果没有适当工具的支持,是很容易出现错误的,也不利于测试进度的掌握和测试团队的管理。因此,最好将测试工作流程建立在成熟的工具之上建立相应测试平台,实现无纸化测试,即测试计划、测试用例、测试结果等文档的电子化流转。

■ 测试必须在稳定的版本上进行

被测试软件版本的稳定,是进行有效测试的重要保证,因此,除非特殊情况(如一个缺陷影响了后续测试工作的进行),否则,应在整个测试期间,保持被测版本的稳定,直到所有测试用例测试完成之后,再统一由开发中心提交包含了若干缺陷修复的被测版本,而不应频繁地更改被测版本,因为这可能导致新缺陷的引入。

■ 科学地利用回归测试工具以提高测试效率

在测试的初期,大多数测试脚本都是由测试人员来手工测试的,但当被测系统相对稳定之后,可能的话尽量利用回归测试工具对某些测试脚本进行自动化改造,以期在随后的测试工作中提高测试工作的效率,然而,时机的选择非常重要,过早地引入回归测试工具也将导致浪费。

在使用回归测试工具时,自动化测试框架将提高测试效率。所谓自动化测试框架,是由一些假设,概念和为自动化测试提供支持的实践组成的集合。例如,IBM框架。

IBM框架以前被称作为ITCL框架,由质量软件工程(Quality Software Engineering)和IBM中有经验的自动化团队合作开发而成的。这个框架由三层架构组成,架构的实现贯穿了应用对象、任务和测试用例包(IBM包)。

- **应用对象:** 储存有关应用程序中的GUI元素信息。同时在这里也可以编写Getter方法,这些Getter方法可以返回对象,使调用者能够对这些GUI元素进行查询和操作。一般情况下,这些方法在Task层中进行调用。
- **任务:** 在这里编写可重用的方法,这些方法在应用程序中执行一些可遍历的路径。任务中的方法可以被测试用例调用。
- **测试用例:** 导航一个应用程序,验证其状态,并记录其结果的方法。

采用自动化测试框架将有助于测试人员解决下列问题:

- 测试代码不能很好的被重用
解决界面对象、执行步骤和验证点掺杂在一个脚本中的问题
- 测试代码很难调试
解决脚本很长并且按顺序录制的问题
- 测试代码不容易维护
解决测试脚本和被测系统紧密关联的问题
- **尽早进行、持续进行压力测试**

压力测试往往被认为应该在项目的后期,由测试机构来进行。但实

际上,在开发机构,应尽早进行压力测试以确认系统架构是否存在设计缺陷,如果这些缺陷拖到项目后期,往往会导致项目的彻底失败。另外,在开发的过程中,开发机构应通过持续的压力测试,来发现系统设计的变更对整个系统性能是否有显著的影响。

在测试后期,测试机构也需要进行压力测试,但此时压力测试的目标将从应用程序扩展到整个系统,因此压力测试将更加全面和细致,尤其应充分考虑各种不同的测试场景(如业务高峰、月底年底等等)的不同负载情况,同时,应紧密监控各种系统资源的应用状况。另外,在系统维护运行的过程中,如需进行硬件、操作系统、数据库或中间件升级,都需要测试中心进行压力测试才能升级。

● 统一的、高效的缺陷追踪流程

在v-model的四种测试当中,开发机构的开发组应负责单元测试(Unit Test),测试机构应负责集成测试(Integration Test)和系统测试(System Test),但是,如果任何一个团队发现了缺陷,都应通过统一的缺陷管理流程来将缺陷反映给开发团队,这可以提高测试机构与开发机构之间的沟通效率,提供对产品质量提供科学、准确的质量指标。

IBM Rational质量管理方案相关产品介绍

测试管理工具

IBM Rational Quality Manager (简称RQM):是协作性的、基于Web的质量管理平台,提供全面的测试计划、手工测试和与自动化测试工具的集成的能力。Rational Quality Manager适合于不同规模的测试团队使用并支持不同类型的用户角色,像测试经理、测试设计员、测试组长、测试员、测试环境管理员等,以及测试组织外的角色。IBM通过RQM实现了对整个软件测试生命周期的管理,使软件测试团队快速拥有以下能力:

■ 全面的测试计划管理

RQM中的测试计划支持分布式团队的协同工作,影响到整个项目生命周期的阶段。测试计划定义了测试的目标和测试的范围,并制定相应的标准帮助团队确定“系统是否可以被发布”?

■ 测试脚本的创建、执行和重用

RQM提供了手工测试编辑环境,供我们进行人工测试。通过使用关

键字的技术,增加了手工测试的重用和自动化的能力。并且RQM也支持与测试工具进行集成,管理和执行测试自动化脚本,这些工具包括Rational Functional Tester、Rational Performance Tester、Rational Robot和Rational Service Tester for SOA Quality等。

■ 测试分析和报告

Rational Quality Manager提供标准的测试报告帮助测试人员评估测试结果,提供测试各阶段的测试报告。

■ 团队协作

Rational Quality Manager使团队间的成员容易进行信息的共享。居于Jazz的工作项功能,RQM可以在团队成员间分配质量任

务和缺陷,并实时了解成员间完成的状态。测试计划的编写者和测试用例的设计者能分发这些工作由不同的人去完成,通过审核的能力追踪任务完成的状态,经过审批的结果在团队中被发布。团队成员可以看到新需求和变更的需求,并保证测试用例能覆盖那些需求。任何变化和测试的里程碑,团队成员都会被自动通知到。

■ 测试环境管理

Rational Quality Manager提供测试环境管理能力,测试人员在测试计划中创建并申请测试环境的请求,使用Lab manager对所申请的环境自动进行部署。通过测试环境管理,保证测试人员要进行测试时,测试环境是可用的。Lab manager能从一个集中的资源存储库追踪所有的测试资源,并对来自测试团队的请求进行支持。

软件质量管理产品	内容介绍
测试管理工具 IBM Rational Quality Manager	<p>IBM Rational Quality Manager是Rational提供的新一代软件测试管理工具,实现了测试需求、测试用例以及缺陷的集中管理,帮助中到大型团队协同工作、使团队共享项目信息,使用自动化的手段加速项目进度,并报告项目指标以决定项目或产品版本是否发布。该工具具有如下特点:</p> <ul style="list-style-type: none"> • 提供整个生命周期的测试计划,可定义角色、过程和交付物的属主、使过程和资产的流程自动化。 • 通过可定制的仪表盘可对显示内容进行裁剪,显示相关的质量信息。 • 使用Web 2.0技术,让分布式的团队能很好的进行协作。 • 跨越整个生命周期的共享访问和资产的追踪。 • 提供一个灵活的过程框架,可与多种工具进行集成。 • 提供手工测试的能力,使用富文本、内嵌图片进行手工测试的撰写,辅助数据输入和验证完成测试的定义和执行。 • 帮助用户完成被优化的测试组合环境的覆盖。 • 提供测试环境管理能力帮助用户在物理或虚拟的测试机器上追踪、确定进度和执行测试。 • 中心存储库存储可重用的测试资产和模板的版本历史。 • 不同的产品版本支持小、中、大不同规模的测试团队。

功能测试工具

Rational提供了Rational Functional Tester帮助测试人员完成功能测试。Rational Functional Tester(简称RFT): RFT支持对Browser界面、Java Application界面、以及SAP、Siebel界面的功能测试自动化,其测试脚本语言采用Java或Microsoft VB .NET,并可和Eclipse或Microsoft Visual Studio .NET集成。

RFT提供了标准的脚本录制功能,并在无需编程的情况下,快速实现

测试数据的参数化,提高测试脚本的开发效率。由于采用标准的测试脚本语言,测试人员无须学习特定语言的语法和API,同时通过和开发环境集成,大大降低了工具学习成本,甚至开发人员也可以迅速掌握该工具,积极参与到自动化测试脚本的开发过程中。

此外,在测试脚本的开发方式上,测试人员使用RFT可从最初的基于录制和数据驱动技术,逐步过渡到基于测试框架的脚本开发技术。

软件质量管理产品	内容介绍
功能测试工具 IBM Rational Functional Tester	<p>Rational Functional Tester是一款先进的、自动化的功能测试和回归测试工具,它适合于测试人员和GUI开发人员</p> <ul style="list-style-type: none"> • 支持多种IDE: 为Java、Web、SAP和Siebel和Microsoft Visual Studio.Net程序 • 定制生成Java或Visual Basic.Net语言的测试脚本 • 为高级测试人员提供原汁原味的Java和VB.NET编辑器和调试器 • 使用ScriptAssure技术支持频繁的用户界面变更 • 多点验证,支持正则表达式的模式匹配 • 自动化的数据关联和数据驱动测试,消除手工编码的需要 • 先进的对象映射维护能力 • 支持Linux测试的编辑和执行 • 可用于测试3270/5250/VT100终端应用程序的附件支持

性能测试工具

Rational提供了Rational Performance Tester帮助测试人员完成性能测试。

Rational Performance Tester (简称RPT): RPT能支持HTTP/HTTPS、SAP、SIP、Siebel、Citrix和TCP Socket协议,同时客户可利用RPT提供的协议开发工具包(Protocol SDK),自主开发特殊协议的适配器,从而实现对其它协议的支持。RPT基于Eclipse平台,并基于Java脚本语言,能快速开发性能测试脚本、建立性能测试负

载模型、灵活运行性能测试并生成各种性能测试报告,具有易用性和开放性等特点。此外,RPT能实现对服务器资源的实时监控,并建立资源使用状况和服务器响应状况的关联,从而帮助快速定位在操作系统层面的性能问题。对J2EE应用,RPT还能进行性能深层次分析,更准确定位应用代码中的性能问题。在执行性能测试时,还需要Virtual Tester Pack许可证。具体的许可证数量取决于需要模拟的虚拟用户数量。

软件质量管理产品	内容介绍
性能测试工具 IBM Rational Performance Tester	<p>Rational Performance Tester是一款性能测试的创建、执行和分析工具,用于在最终部署前验证应用程序的可伸缩性和可靠性,特点如下:</p> <ul style="list-style-type: none"> • Web、SAP、Siebel和Citrix应用程序的多用户性能测试 • 可视化编辑器同时提供测试的高层视图和详细视图 • 不同用户群的灵活建模与仿真 • 低内存和处理器占用,实现了有限硬件资源下的大规模多用户测试 • 运行时的报告能够立即识别性能问题 • 自动识别和支持动态服务器响应 • 测试中用户负载动态变化 • 服务器资源数据的收集与可视化展现 • 采用浏览器样式显示测试中的每一张网页 • 能对服务器资源进行监控 • 能对J2EE应用性能问题进行深层分析

代码级测试工具

针对开发人员开发的软件代码的不同，Rational提供了如下不同的代码级测试工具。

Rational PurifyPlus: 是一款运行时分析工具，在运行被PurifyPlus插针后的软件后，能自动发现并定位软件在内存和性能方面的问题，并发现未被执行的代码。PurifyPlus包含三个功能组件，其中Purify能发现代码中的内存泄漏、内存越界访问错误等；而Quantify能发

现代码中的每行代码的执行时间和次数，从而发现代码的性能瓶颈；Coverage能统计出已经执行代码行的百分比，并指出未被执行过的代码行，这样开发人员可丰富测试用例再进行测试，从而确保测试的完备性。PurifyPlus能支持IBM AIX、Sun Solaris、HP-UX、LINUX等操作系统环境下自带的C/C++开发环境和gcc开发环境，以及Windows下的Microsoft Visual Studio开发环境。

软件质量管理产品	内容介绍
代码级测试工具 IBM Rational PurifyPlus	Rational PurifyPlus是一套完整的运行时分析工具，用于提高应用程序的可靠性和性能。特点如下： <ul style="list-style-type: none"> • 查明难以发现的错误 • 突出性能瓶颈 • 识别未测试代码 • 对整个应用程序或仅对选择的模块操作 • 直接集成Microsoft Visual Studio .NET • 支持Java、Visual C# .NET、Visual Basic .NET、Visual C/C++ 6、Visual Basic 6和ANSI C/C++ • 支持Windows、UNIX和Linux环境 • 无论有无源代码均可工作

Logiscope: 提供软件代码静态分析, 软件代码规范检查和软件测试用例覆盖率检查等重要特性, 帮助您在系统崩溃前发现并修复软件缺陷。Logiscope包含三个模块, Logiscope RuleChecker根据为项目定制的规则自动地检查代码编程规则, 可以避免错误陷阱和代码误解。预定义220多个的编程规则, 用户可以从这些规则中选择, 也可以定义新的规则, 用所选的规则对源代码进行验证, 指

出所有不符合编程规则的代码, 并提出改进源代码的解释和建议。Logiscope Audit帮助定位错误的代码模块。一旦发现错误代码模块, Logiscope Audit可提供基于软件度量和图形的质量信息, 能够帮助用户诊断问题和做出判断(是重写模块还是做更彻底地测试)。Logiscope TestChecker分析代码测试覆盖率和显示未覆盖的代码路径。发现未测试源代码中隐藏的Bugs, 来提高软件的可靠性。

软件质量管理产品	内容介绍
代码级测试工具Logiscope	<p>Logiscope是基于代码级别的软件质量分析、软件代码规范检查的静态测试工具。特点如下:</p> <ul style="list-style-type: none"> • 软件质量分析Logiscope Audit帮助定位错误的代码模块。Logiscope Audit提供基于软件度量和图形的质量信息, 能够帮助用户诊断问题和做出判断。 • 软件代码规范检查Logiscope RuleChecker, 根据为项目定制的规则自动地检查代码编程规则, 可以避免错误陷阱和代码误解。 • 软件测试用例覆盖率检查Logiscope TestChecker, 分析代码测试覆盖率和显示未覆盖的代码路径。发现未测试源代码中隐藏的Bugs, 来提高软件的可靠性。 • 管理与控制你的软件复杂度。Logiscope提供了很多方式来可视化文本代码, 例如: 调用图、控制图与继承关系图。你可以清楚地看到系统的代码结构与行为, 并使你能够快速辨识与解决复杂的问题。 • 优化你的测试流程。高效的测试可以确保缩短上市时间及提高产品质量。Logiscope可以帮助你辨识低效率的测试, 例如重复的测试用例与没有被测试覆盖的源代码。 • Logiscope可以为基于SEI/CMM二级(可重复级)并且想要成为三级(可定义级)或更高级的软件组织提供一套跟踪软件质量的技术。 • Logiscope遵循DO-178B的软件系统E级到A级标准, 帮助软件开发组织对源代码进行“审核和分析”和“结构的覆盖率分析”。 • Logiscope遵循ISO/IEC 9126标准为软件开发组织提供分析软件特征的功能。Logiscope也遵循ISO-9001标准支持软件开发组织做接受测试和质量记录。

Rational Test RealTime (简称RTRT): 是针对嵌入式C/C++软件的开发人员测试工具。能对C/C++代码进行自动化代码评审、对C函数或C++方法进行组件测试以及运行时分析功能。同时通过其特有的Target Deployment Port (TDP)技术, 能支持目前主流的嵌入式软件开发系统。

软件质量管理产品	内容介绍
代码级测试工具 IBM Rational Test RealTime	<p>Rational Test RealTime是一种跨平台的嵌入式软件组件测试和运行时分析工具。特点如下:</p> <ul style="list-style-type: none"> • 在单一测试环境中对宿主和目标自动进行组件测试和运行时分析 • 自动创建和部署组件测试装置、测试存根和测试驱动程序 • 描述内存使用和性能、分析代码覆盖、可视化地说明运行时跟踪情况 • 基于宿主的测试很容易适应不同的目标, 不必重新编写测试程序 • 直接在目标上进行测试和分析。支持所用常见平台, 无论是8位的微芯片还是64位的RTOS • 提供安全性和任务关键型验收所需要的详细的代码覆盖信息 • 提供代码自动评审功能, 并和Eclipse集成





Rational software