IBM Rational

软件测试自动化技术

IBM Rational 技术白皮书

版本 1.0

Rational. software

IBM

目录

Rational. software

IBM

软件测试自动化技术白皮书

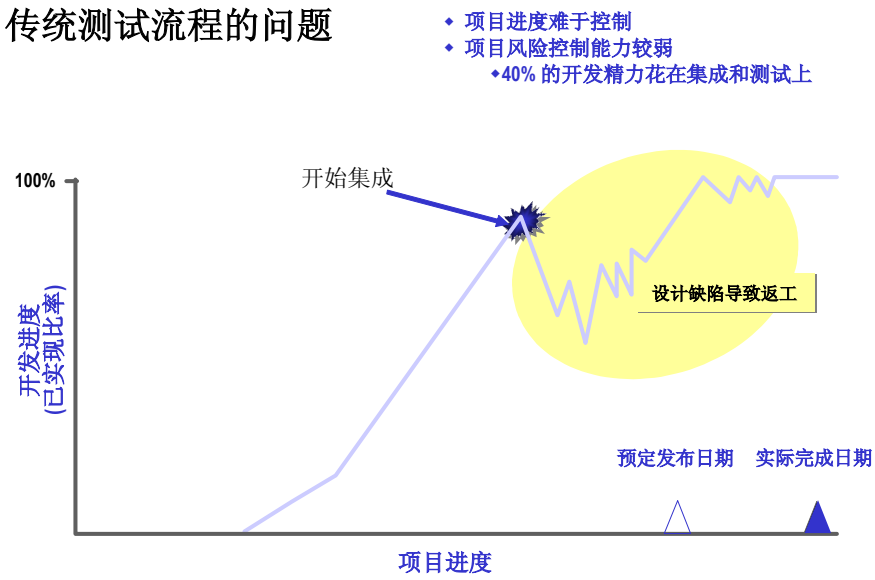
1. 传统软件测试过程中的问题

测试在所有的软件开发过程中都是最重要的部分。在软件开发过程中，一方面要求我们通过测试活动验证所开发的软件在功能上满足软件需求中描述的每一条特性，性能上满足客户要求的负载压力和相应的响应时间、吞吐量要求；另一方面，面向市场和客户，开发团队还要满足在预算范围内尽快发布软件的要求。

传统的软件测试流程一般是先在软件开发过程中进行少量的单元测试，然后在整个软件开发结束阶段，集中进行大量的测试，包括功能和性能的集成测试和系统测试。随着开发的软件项目越来越复杂，传统的软件测试流程不可避免地给我们的工作带来以下问题:

- 问题一：项目进度难于控制，项目管理难度加大

  如图一所示，大量的软件错误往往只有到了项目后期系统测试时才能够被发现，解决问题所花的时间很难预料，经常导致项目进度无法控制，同时在整个软件开发过程中，项目管理人员缺乏对软件质量状况的了解和控制，加大了项目管理难度。



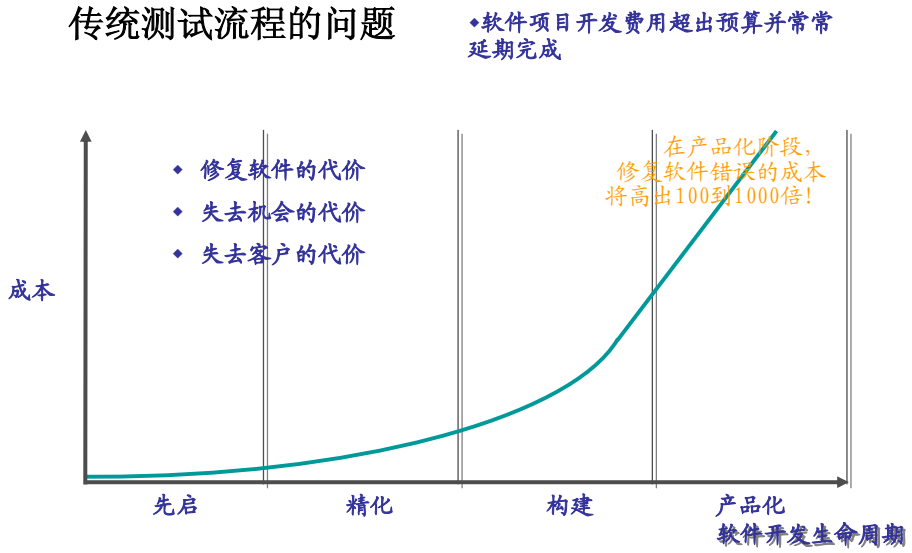
图一、传统测试流程中存在的问题

- 问题二：对于项目风险的控制能力较弱

  项目风险在项目开发较晚的时候才能够真正降低。往往是经过系统测试之后，才真正确定该设计是否能够满足系统功能、性能和可靠性方面的需求。

- 问题三：软件项目开发费用超出预算

  在整个软件开发周期中，错误发现的越晚，单位错误修复成本越高，如图二所示，错误的延迟解决必然导致整个项目成本的急剧增加。



图二、传统测试流程中存在的问题

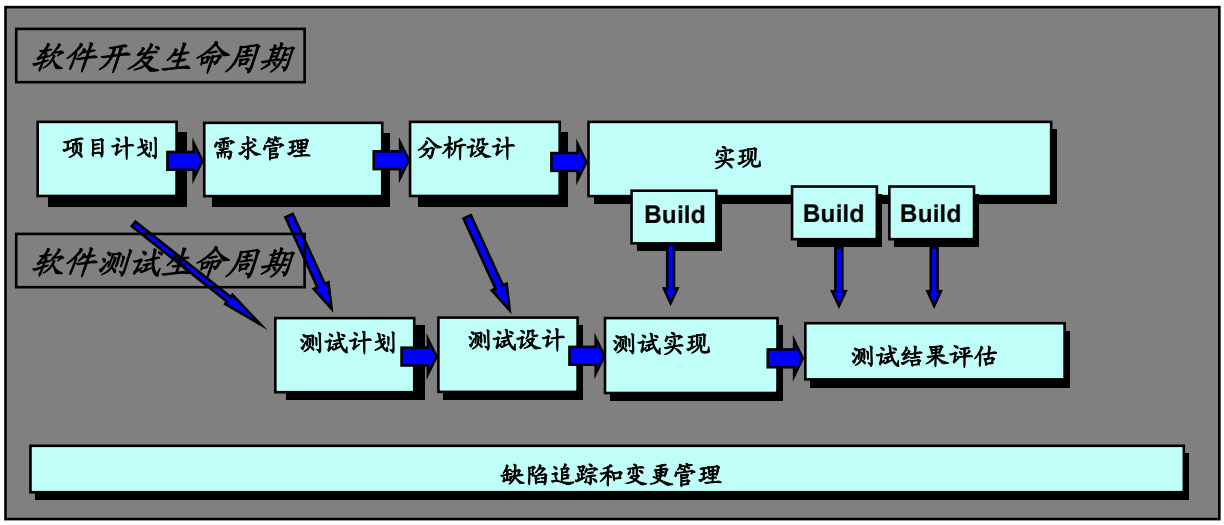
2. 采用 IBM Rational 软件自动化测试最佳成功经验解决传统测试问题

IBM Rational 软件自动化测试技术核心的三个最佳成功经验是：尽早测试、连续测试、自动化测试，并在此基础上提供了完整的软件测试流程和一整套的软件自动化测试工具，使我们最终能够做到：一个测试团队，基于一套完整的软件测试流程，使用一套完整的自动化软件测试工具，完成全方位的软件质量验证。

2.1 成功经验一：尽早测试

所谓尽早测试是指在整个软件开发生命周期中通过各种软件工程技术尽量早的完成各种软件测试任务的一种思想。IBM Rational 主要在以下三个方面为我们提供的尽早测试的软件工程技术：

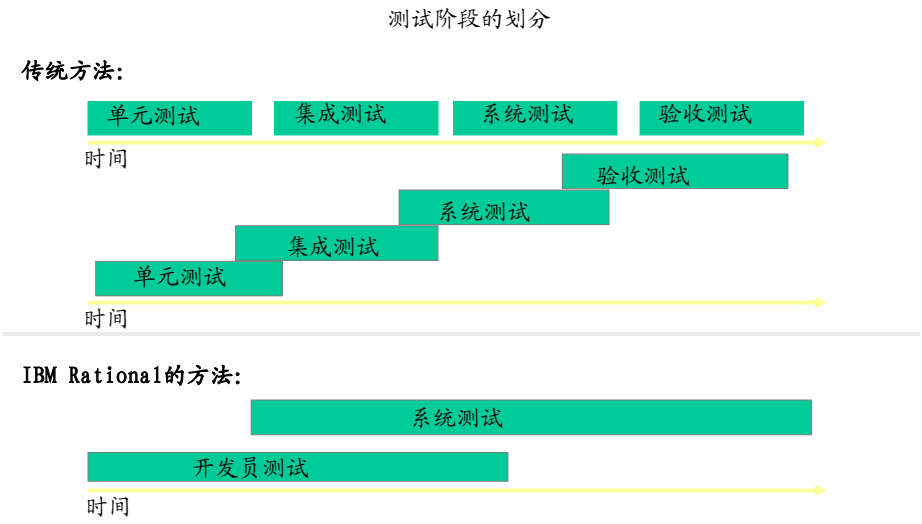
首先，软件的整个测试生命周期是与软件的开发生命周期基本平齐的过程，如图三所示，即当需求分析基本明确后我们就应该基于需求分析的结果和整个项目计划来进行软件的测试计划；伴随着分析设计过程同时应该完成测试用例的设计；当软件的第一个发布出来后，测试人员要马上基于它进行测试脚本的实现，并基于测试计划中的测试目的执行测试用例，对测试结果进行评估报告。这样，我们可以通过各种测试指标实时监控项目质量状况，提高对整个项目的控制和管理能力。



图三、软件测试生命周期

其次，通过迭代是软件开发把原来的整个软件开发生命周期分成多个迭代周期，在每个迭代周期都进行测试，这样在很大程度上提前了软件系统测试发生的时间，这可以在很大程度上降低项目风险和项目开发成本。

最后，IBM Rational 的尽早测试成功经验还体现在它扩展了传统软件测试阶段从单元测试、集成测试到系统测试、验收测试的划分，将整个软件的测试按阶段划分成开发员测试和系统测试两个阶段，如图四所示，它把软件的测试责无旁贷地扩展到整个开发人员的工作过程。通过提前测试发生的时间来尽早地提高软件质量、降低软件测试成本。

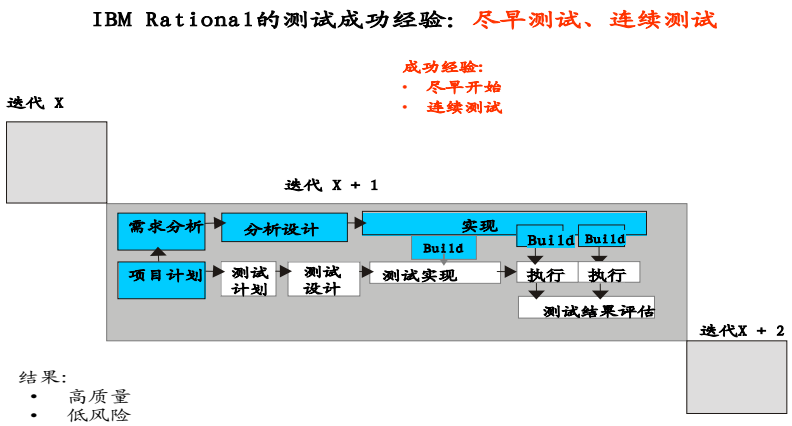


图四、IBM Rational 测试方法对测试阶段的划分

2.2 成功经验二：连续测试

测试成功经验连续测试是从迭代式软件开发模式得来。在迭代化的方法中，我们将整个项目的开发目标划分成为一些更易于完成和达到的阶段性小目标，这些小目标都有一个定义明确的阶段性评估标准。迭代就是为了完成一定的阶段性目标而从事的一系列开发活动，在每个迭代开始前都要根据项目当前的状态和所要达到的阶段性目标制定迭代计划，而且每个迭代中都包括需求、设计、编码、集成、测试等一系列的开发活动，都会增量式集成一些新的系统功能。通过每次迭代，我们都产生一个可运行的系统，通过对于这个可运行系统的测试来评估该次迭代有没有达到预定的迭代目标，并以此为依据来制定下一次迭代的目标。由此可见，在迭代式软件开发的每个迭代周期我们都会进行软件测试活动，整个软件测试的完成是通过每个迭代周期不断增量测试和回归测试实现的。

如图五所示，采用连续测试的软件成功测试经验，不但能够持续的提高软件质量、监控质量状态，同时也使系统测试的尽早实现成为可能。从而有效的控制开发风险、减低测试成本和保证项目进度。



图五、IBM Rational 测试成功经验：连续测试

2.3 成功经验三：自动化测试

在整个软件的测试过程中要想实现尽早测试、连续测试，可以说完善的测试流程是前提，自动化测试工具是保证。IBM Rational 的自动化测试成功经验主要是指利用软件测试工具提供完整的软件测试流程的支持和各种测试的自动化实现。

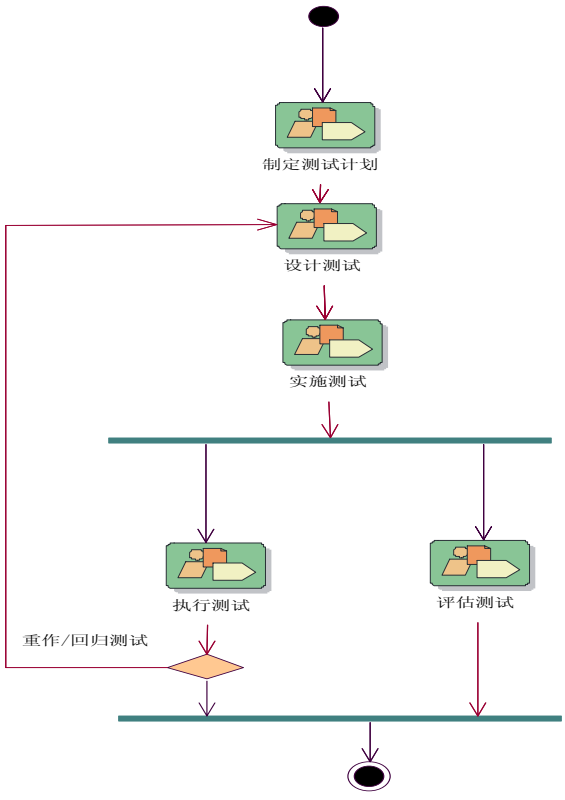
为了使各种软件测试团队更好地进行测试，IBM Rational 在提供了测试成功经验之外，还为我们提供了一整套的软件测试流程和自动化测试工具，使软件测试团队能够从容不迫地完成整个测试任务。

3. IBM Rational 软件测试流程

IBM Rational 的软件测试流程，不仅仅包含完整的软件测试流程框架，同时还提供了内嵌软件测试流程的测试管理工具的支持，包括完整的测试评测方法。

3.1 IBM Rational 软件测试流程框架

IBM Rational Unified Process(以下简称 RUP)提供了一套完整的测试流程框架，软件测试团队可以以它为基础，根据业务发展的实际要求，定制符合团队使用的软件测试流程。RUP 中的软件测试流程如图六所示:



图六、IBM Rational 软件测试流程

每个测试环节的具体阐述如下:

- 制定测试计划的目的是确定和描述要实施和执行的测试。这是通过生成包含测试需求和测试策略的测试计划来完成的。可以制定一个单独的测试计划，用于描述所有要实施和执行的不同测试类型，也可以为每种测试类型制定一个测试计划。
- 设计测试的目的是确定、描述和生成测试过程和测试用例。
- 实施测试的目的是实施（记录、生成或编写）设计测试中定义的测试过程。输出工件是测试过程的计算机可读版本，称为测试脚本。
- 执行测试的目的是确保整个系统按既定意图运行。系统集成员在各迭代中编译并链接系统。每一迭代都需要测试增加的功能，并重复执行以前版本测试过的所有测试用例（回归测试）。
- 评估测试的目的是生成并交付测试评估摘要。这是通过复审并评估测试结果、确定并记录变更请求，以及计算主要测试评测方法来完成的。测试评估摘要以组织有序的格式提供测试结果和主要测试评测方法，用于评估测试对象和测试流程的质量。

3.2 IBM Rational 软件测试的评测方法

软件测试的主要评测方法包括测试覆盖和质量评测。测试覆盖是对测试完全程度的评测，它是由测试需求和测试用例的覆盖或已执行代码的覆盖表示的。质量评测是对测试对象（系统或测试的应用程序）的可靠性、稳定性以及性能的评测，它建立在对测试结果的评估和对测试过程中确定的变更请求（缺陷）分析的基础上。

3.2.1 覆盖评测

覆盖指标提供了“测试的完全程度如何？”这一问题的答案。最常用的覆盖评测是基于需求的测试覆盖和基于代码的测试覆盖。简而言之，测试覆盖是就需求（基于需求的）或代码的设计/实施标准（基于代码的）而言的完全程度的任意评测，如用例的核实（基于需求的）或所有代码行的执行（基于代码的）。

- 基于需求的测试覆盖

基于需求的测试覆盖在测试生命周期中要评测多次，并在测试生命周期的里程碑处提供测试覆盖的标识（如已计划的、已实施的、已执行的和成功的测试覆盖）。

测试覆盖通过以下公式计算:

测试覆盖 = $T^{(p,i,x,s)}$ / RfT

其中:

T 是用测试过程或测试用例表示的测试 (Test) 数（已计划的、已实施的或成功的）。RfT 是测试需求 (Requirement for Test) 的总数。

- 基于代码的测试覆盖

基于代码的测试覆盖评测测试过程中已经执行的代码的多少，与之相对的是要执行的剩余代码的多少。代码覆盖可以建立在控制流（语句、分支或路径）或数据流的基础上。基于代码的测试覆盖通过以下公式计算:

测试覆盖 = I^e / TIic

其中:

I^e 是用代码语句、代码分支、代码路径、数据状态判定点或数据元素名表示的已执行项目数。TIic (Total number of Items in the code) 是代码中的项目总数。

3.2.2 质量评测

测试覆盖的评估提供对测试完全程度的评测，对在测试过程中已发现缺陷的评估提供了最佳的软件质量指标。因为质量是软件与需求相符程度的指标，所以在这种环境中，缺陷被标识为一种更改请求，该更改请求中的测试对象与需求不符。

- 缺陷报告

一般，可以将缺陷计数作为时间的函数来报告，即创建**缺陷趋势**图或报告；也可以将缺陷计数作为一个或多个缺陷参数的函数来报告，如作为**缺陷密度**报告中采用的严重性或状态参数的函数。这些分析类型分别为揭示软件可靠性的缺陷趋势或缺陷分布提供了判断依据。

Rational Unified Process 以三类形式的报告提供缺陷评估:

- 缺陷分布（密度）报告允许将缺陷计数作为一个或多个缺陷参数的函数来显示。用于描述当前软件的质量状态。
- 缺陷龄期报告是一种特殊类型的缺陷分布报告。 缺陷龄期报告显示缺陷处于特定状态下的时间长短。我们一般用它来表示研发团队对质量的反应能力。
- 缺陷趋势报告按状态将缺陷计数作为时间的函数显示。趋势报告可以用来标识软件质量变化的趋势

➢ 性能评测

评估测试对象的性能行为时，可以使用多种评测，这些评测侧重于获取与行为相关的数据，如响应时间、计时配置文件、执行流、操作可靠性和限制。这些评测主要在“评估测试”活动中进行评估，但是也可以在“执行测试”活动中使用性能评测评估测试进度和状态。

主要的性能评测包括:

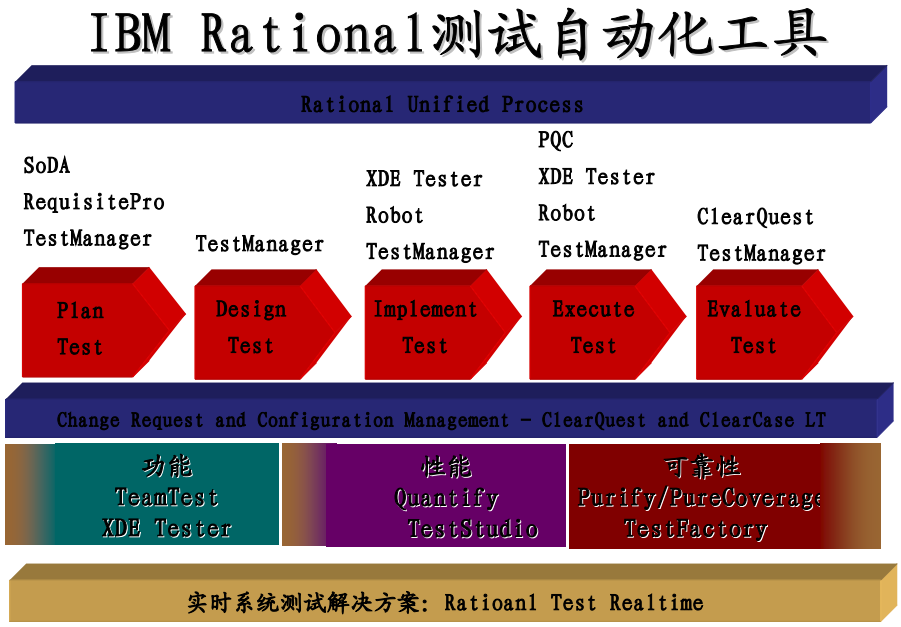
- 动态监测 – 在测试执行过程中，实时获取并显示正在执行的各测试脚本的状态。
- 响应时间/吞吐量 – 测试对象针对特定主角和/或用例的响应时间或吞吐量的评测。
- 百分位报告 – 数据已收集值的百分位评测/计算。

- 比较报告 – 代表不同测试执行情况的两个（或多个）数据集之间的差异或趋势。

4. IBM Rational 软件自动化测试工具

在 IBM Rational 的软件自动化测试解决方案中，我们一直致力追求的一点就是测试工具和测试成功经验、测试流程的统一，上面阐述的每个测试成功经验和测试流程环节，我们都可以通过 Rational 的测试工具以及工具间的完美集成辅助完成。

IBM Rational 的软件自动化测试工具如图七所示，其最大特点是通过一套完整的软件测试工具在实现测试管理流程的基础上，同时涵盖了功能测试、性能测试和可靠性测试的自动化测试需求，通过工具之间的集成完成测试资源的整合，帮助测试团队实现 IBM Rational 的测试成功经验。



图七、IBM Rational 自动化测试工具

4.1 利用IBM Rational软件测试管理平台实现软件自动化测试流程

IBM Rational 在 RUP 测试方法论的基础上构建了软件自动化测试管理平台工具 TestManager，通过和测试需求管理工具 RequisitePro、缺陷追踪工具 ClearQuest 的完美集成，实现了对整个软件测试生命周期的管理，可以帮助软件测试团队快速建立软件测试平台和测试管理流程，使软件测试团队快速拥有以下能力:

- TestManager提供测试管理的核心平台，整合了从测试需求、测试计划、测试设计、测试实施、测试执行到测试结果分析、测试报告的自动生成等整个测试生命周期的管理活动。同时，统一组织各种Test Suite,Test Case,Test Script，方便地进行回归测试
- TestManager遵循RUP标准测试流程，使测试人员能够在统一的测试管理平台上、遵循统一的测试管理流程，完成对包括产品的功能性、可靠性和性能等全方位的质量测试。
- 作为一种集成解决方案，Rational TestManager与Rational 其他工具一起，提供从测试需求、到整个软件测试流程管理、缺陷追踪、测试结果评测的可追踪性，方便测试管理人员进行软件测试过程监控和有关软件质量的各种量化指标的采集、分析。

4.2 利用IBM Rational软件测试工具实现软件自动化的功能和性能测试

IBM Rational 的自动化软件测试工具的另一个特点就是: 通过 TestManager + Robot，在实现测试管理流程的同时，能够完成功能测试和性能测试，这会大大缩短测试团队对工具的学习过程，提高工具的易用性。

4.2.1 软件的自动化功能测试

功能测试主要围绕 Windows 图形界面、字符终端和 Browser 界面进行测试。客户端可以是 VC、VB、PB、Delphi 等编制的软件、各种字符终端软件或者运行浏览器 Microsoft Explorer 和 Netscape，通过自动录制形成测试脚本实现自动化的功能/回归测试。

IBM Rational 的功能测试解决方案的目标，是使功能性测试变得更简单、有效并可重复执行，从而快速提升软件测试团队的功能测试能力。它主要具有以下特点:

- 能够方便的对各种环境(IDE)中开发的应用程序、字符终端软件，完成包括测试计划、测试设计、测试实施、测试执行和测试结果分析等全部测试流程。
- 能够方便的录制或编写各种功能测试脚本，实现自动化的功能/回归测试。
- 利用数据池方便地解决大批量数据驱动的功能测试;
- 能够方便地完成分布式功能测试，可以一次测试多种测试平台;
- 能够自动完成功能测试需求覆盖，确保应用程序满足产品规格说明和测试计划的每一条业务需求;

为了提高对 Java 和 Web 开发的应用软件功能测试的支持，IBM Rational 的功能测试的解决方案还提供了 IBM Rational XDE Tester，它主要用于在 Windows 和 Linux 平台上基于 Java 和 Web 开发的应用软件的功能测试，尤其适用于使用 IBM WebSphere Studio、Eclipse 和 Rational XDE Developer 等开发平台进行软件开发的团队。它的三个最重要的自动化测试的特性是:

- 专业的自动化测试脚本创建环境: 测试平台扩展嵌入到 IBM WebSphere Studio、Eclipse 和 Rational XDE Developer 开发平台，统一了测试和开发环境;
- 测试脚本在回归测试方面具有很强的灵活性和可维护性: ScriptAssure 是 IBM 提供的针对 Java 和 Web 应用程序测试时的一组高级能力，它能够帮助创建灵活、可重用的测试脚本，大大提高了脚本的可维护性。对象地图（Object

mapping）提供了核心对象库，测试人员可以基于它进行被测程序中被测对象的修改和验证，并根据修改自动更新所有相关的测试脚本。可以自己设置被测程序中用来表示被测对象的对象属性集，这使得少量对象属性的变化不会影响测试脚本的正常回放。同时，可以创建针对动态数据的验证点，通过正则表达式更容易对动态的数据进行验证;

- 强大的测试脚本语言：使用标准的测试脚本语言Java，可以充分利用工业标准语言的优点进行测试。

4.2.2 软件的自动化压力测试

IBM Rational 压力测试工具主要目标是快速提升软件测试团队的性能测试能力，包括负载测试，压力测试等等。Rational 性能测试解决方案可以方便灵活地模拟各种负载模型，完成以查找响应时间瓶颈、系统吞吐量、最大并发虚拟用户等为目地的各种要求的性能测试。包括:

- 利用 TestStudio 可以完成对压力测试的测试需求、测试计划、测试设计、测试实施、测试执行和测试结果分析等整个测试生命周期的管理;
- 利用 TestStudio 中的 Test Suite，能够方便的完成压力测试对负载模型的各种要求，包括:
  - 建立复杂的Scenario模型;
  - 准确模拟复杂负载的时序控制;
  - 基于Transaction的负载分析;
  - 建立面向目标的事务负载模型，例如：100事务/秒
  - 响应时间精确到1/100秒;
  - 支持不同虚拟用户的不同IP地址模拟;
  - 准确的波特率模拟;
  - 利用TestStudio，能够方便地完成压力测试过程中各种指标的观测;
  - 利用TestStudio，能够方便地完成压力测试结果分析和各种结果报告的生成;

4.3 利用 IBM Rational 软件测试工具实现软件自动化的可靠性测试和单元测试

IBM Rational 软件测试工具 PurifyPlus 主要用于帮助软件测试团队在短期内快速提升单元测试能力和可靠性测试能力的团队，其主要特点是：见效快、使用方便、门槛低、培训时间短，开发人员 2 小时内即可完全掌握该软件进行测试。

PurifyPlus 包含 Rational Purify、Quantify 和 PureCoverage 三个产品，主要功能如下：

- Rational Purify主要针对软件开发过程中难于发现的内存错误、运行时错误。在软件开发过程中：
  - 自动地发现错误;
  - 准确地定位错误;
  - 提供完备的错误信息;

从而减少了调试时间， 帮助开发团队找出缺陷并最终形成高质量的产品，使您能真正做到更快地发布更好的软件。

- Rational Quantify主要解决软件开发过程中的性能问题。在软件开发过程中：
  - 方便地查明并显示应用程序的性能瓶颈，从而确保整个应用程序的质量和性能。
  - Rational Quantify 给开发团队提供了一个性能数据的全局图形化视图，使您从开发流程的开头起就注重性能问题，真正做到更快地发布更好的软件。
- Rational PureCoverage提供应用程序的测试覆盖率信息。在软件开发过程中：
  - 能自动找出代码中未经测试的代码，保证代码测试覆盖率;
  - 帮助开发人员确保开发质量，并使质量保证人员能够评价测试工作的效果。
  - 可针对每次测试生成全面的覆盖率报告，可以归并程序多次运行所生成的覆盖数据，并自动比较测试结果，以评估测试进度。

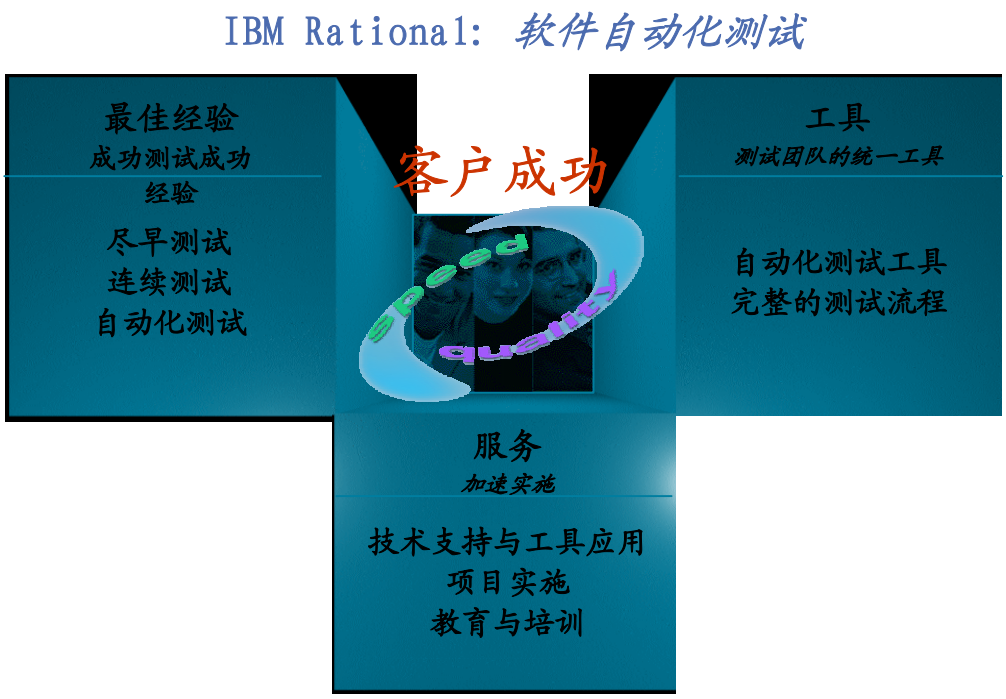
4.4 利用 IBM Rational 软件测试工具实现实时系统软件的自动化测试

IBM Rational Test Realtime 主要适合于开发实时系统和具有较高要求的非实时系统的软件开发，可以帮助测试团队快速建立起单元测试、集成测试、系统测试等测试能力。它提供的自动测试（包括单元测试、集成测试、系统测试）、代码覆盖、内存泄漏检查、性能分析以及 UML 跟踪等重要特性，帮助软件测试团队在系统崩溃前发现并修复软件缺陷。其主要功能特性如下:

- 自动生成测试脚本模板和测试程序（包括驱动模块和桩模块）：通过源代码分析，自动生成在目标上运行所需的测试脚本和测试程序。除了利用测试脚本指定测试数据外，不需要手工编码。而且在测试报告中，测试结果和源代码相联，简化代码修改;
- 通过代码自动插针进行代码覆盖率、内存泄漏以及性能瓶颈进行分析，并和测试用例建立关联;
- 通过把测试结果和观察结果和被测代码关联，把测试作为开发的一个重要部分，真正实现边开发边测试，边测试边观察，边观察边评估这一集成的开发测试过程;
- 通用的、低开销而且易于移植的目标适配技术（Target Deployment Port，TDP）：利用 TDP 技术，使得测试与编译器、连接器、调试器以及目标结构无关，实现了跨多开发环境、多目标结构;
- 模型观察和代码覆盖相集成：利用 UML Trace 功能观察应用运行状态，并通过状态机模型覆盖实现测试用例和模型的关联，充分利用了模型和代码级测试的长处;
- 与 ClearCase、ClearQuest 和 RUP 集成：在同一集成环境中完成对测试文件进行版本控制，提交和修改变更请求;

5. 小结

IBM Rational 主要为软件测试团队提供测试成功经验、自动化测试工具和全方位的咨询服务三方面的支持，如图八所示，最终实现：一个测试团队，基于一套完整的软件测试流程，使用一套完整的自动化软件测试工具，完成全方位的软件质量验证，这正是 IBM Rational 测试解决方案的精髓和终极目标。



图八、IBM Rational 的软件自动化测试解决方案