



2008 IBM Rational  
软件开发高峰论坛

Rational software

# 开发的智慧 协作的力量



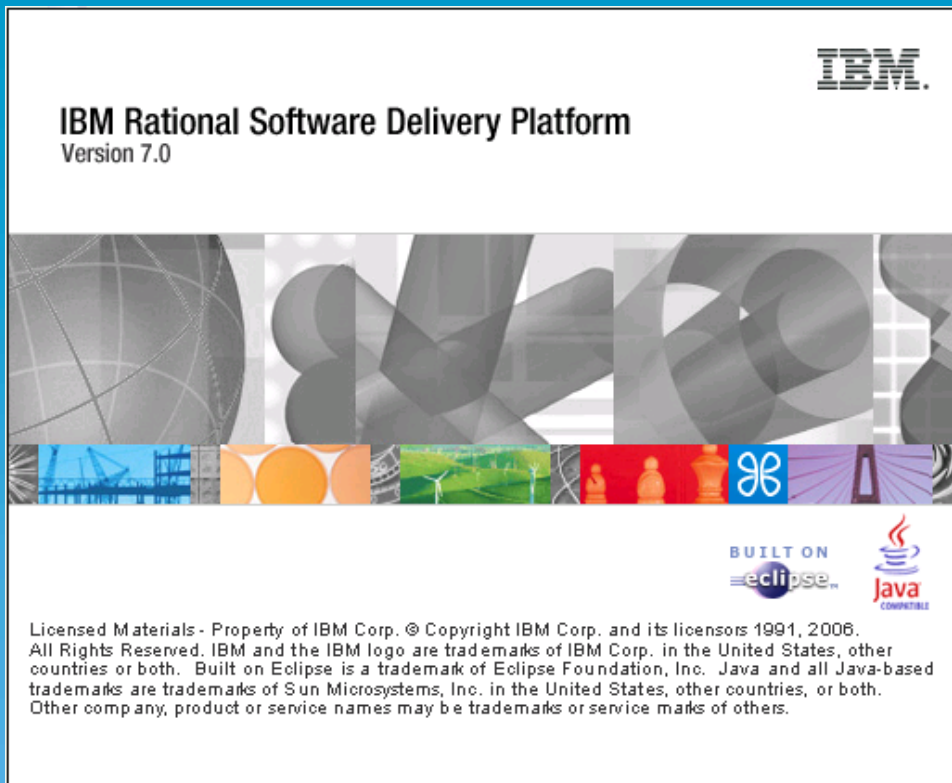


# 基于Rational Functional Tester 构建可复用的软件自动化测试系统

IBM中国软件开发中心  
高级软件工程师 孙志强 (sunzhiq@cn.ibm.com)

# 什么是Rational Functional Tester (RFT)?

IBM Rational Functional Tester 是一个面向对象的自动化测试工具,支持在 Windows及Linux平台下的图形化界面自动化开发与测试。



- 单独的IDE / 与.net开发平台集成
- Java 与 VB Script 两种脚本语言
- 提供对以下应用的测试
  - Java 应用程序
  - .net应用
  - Web应用
  - 模拟终端应用(3270/5250)
  - 特定应用： Siebel & SAP

# 进行图形化界面(GUI)进行自动化测试的一般方法



## 录制

- 录制脚本
  - 捕捉用户与被测系统的交互操作



## 改进

- 改进脚本
  - 对脚本进行手工修正以满足特定要求



## 执行

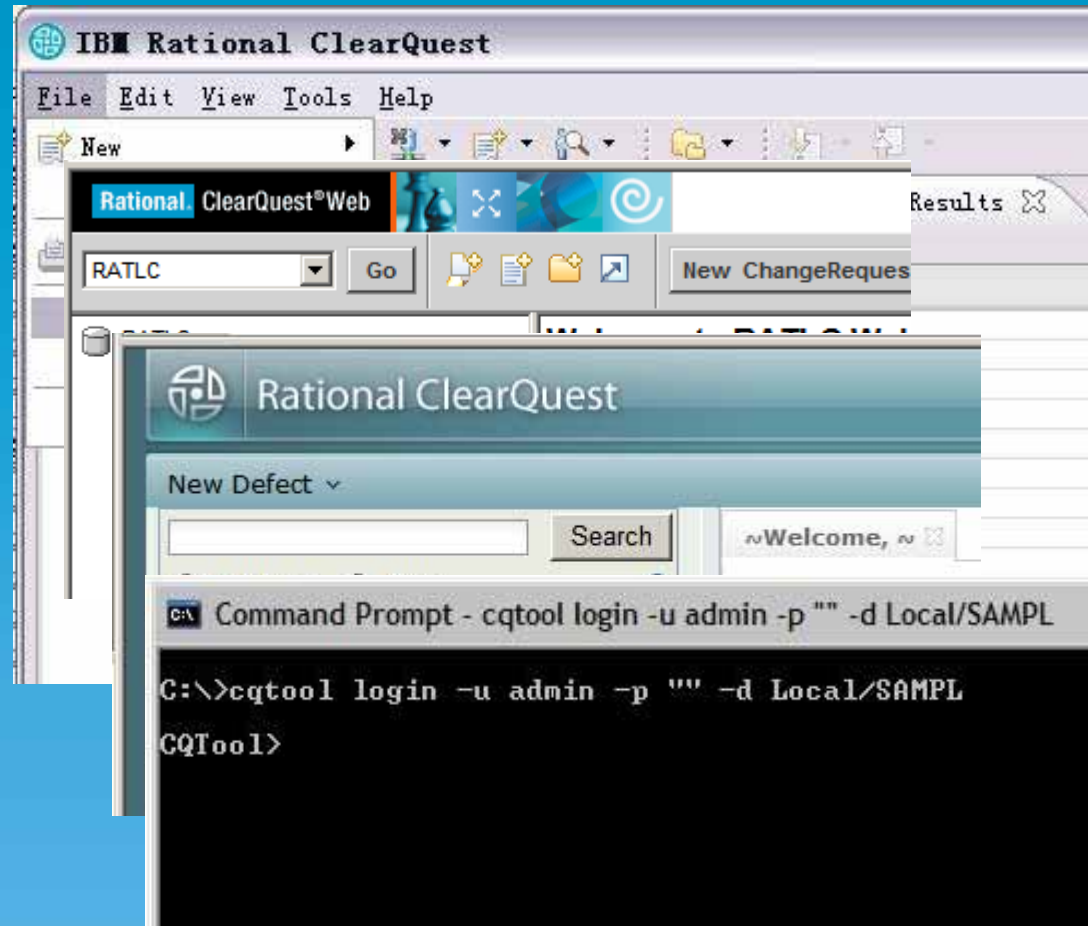
- 执行脚本
  - 针对被测系统执行测试脚本,验证被测系统的功能

# 应对更高的需求

- 测试复杂的大型应用
- 持续性测试
- 重用测试资产

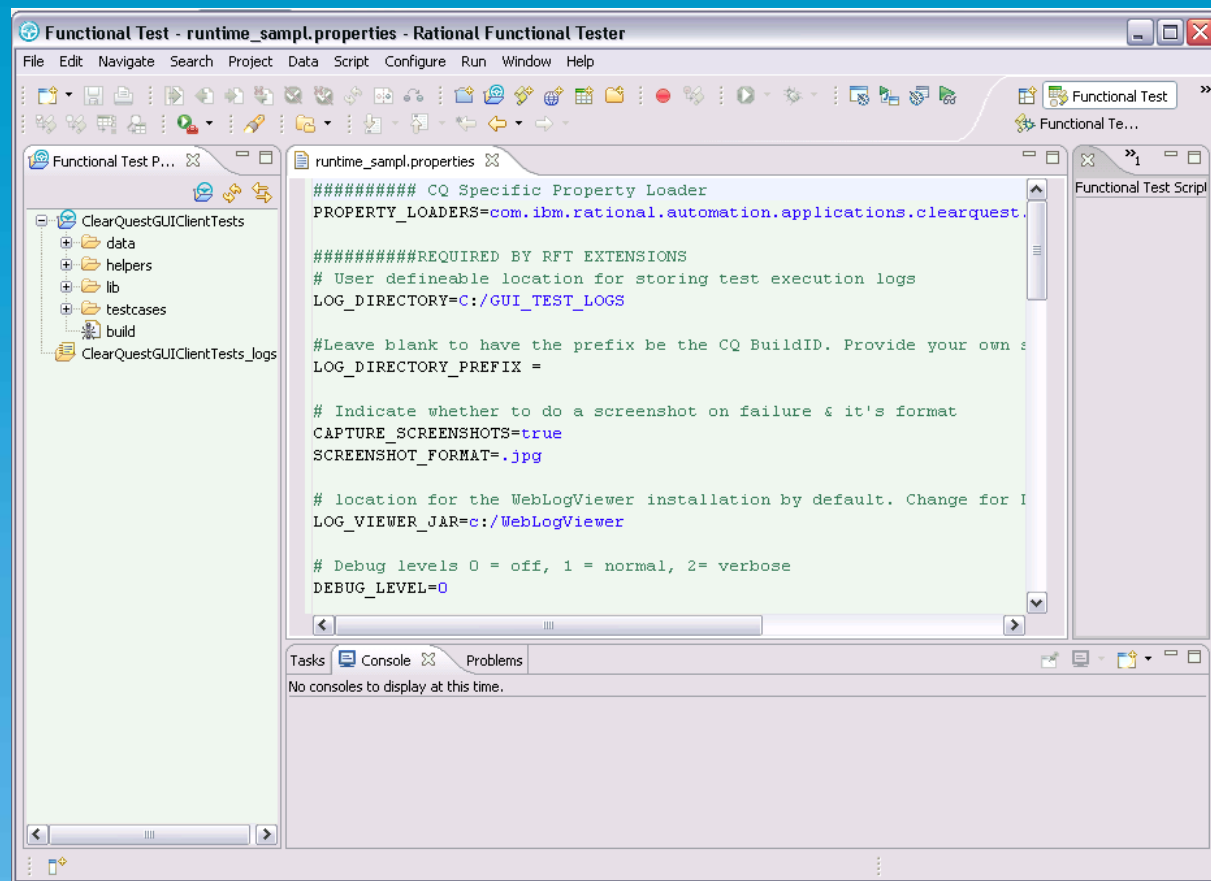
# 实例: Rational ClearQuest的自动化测试需求

- 多客户端
- 业务逻辑复杂
  - 界面动态变化
  - 用户场景多变
- 实现了全球化
  - 界面
  - 测试数据
- 客户端/服务器结构
  - 需要有效管理测试环境



# 针对ClearQuest产品的自动化测试系统

- 一套脚本，涵盖多客户端
- 功能强大
- 配置灵活
- 易于扩展
- 实时监控



# 如何构建可重用的自动化测试系统？

- 开发方法
  - 基于业务逻辑建模
  - 结构分层
  - 动态实现系统的操作与验证
- 基础架构构建
  - 在RFT基础上进行功能扩充
  - 构建通用的工具集





# 使用业务对象来描述测试逻辑



```
Query myQuery= factory.createQueryObj(...);  
Query.create(...)
```

Query

CQECQuery

CQWebQuery

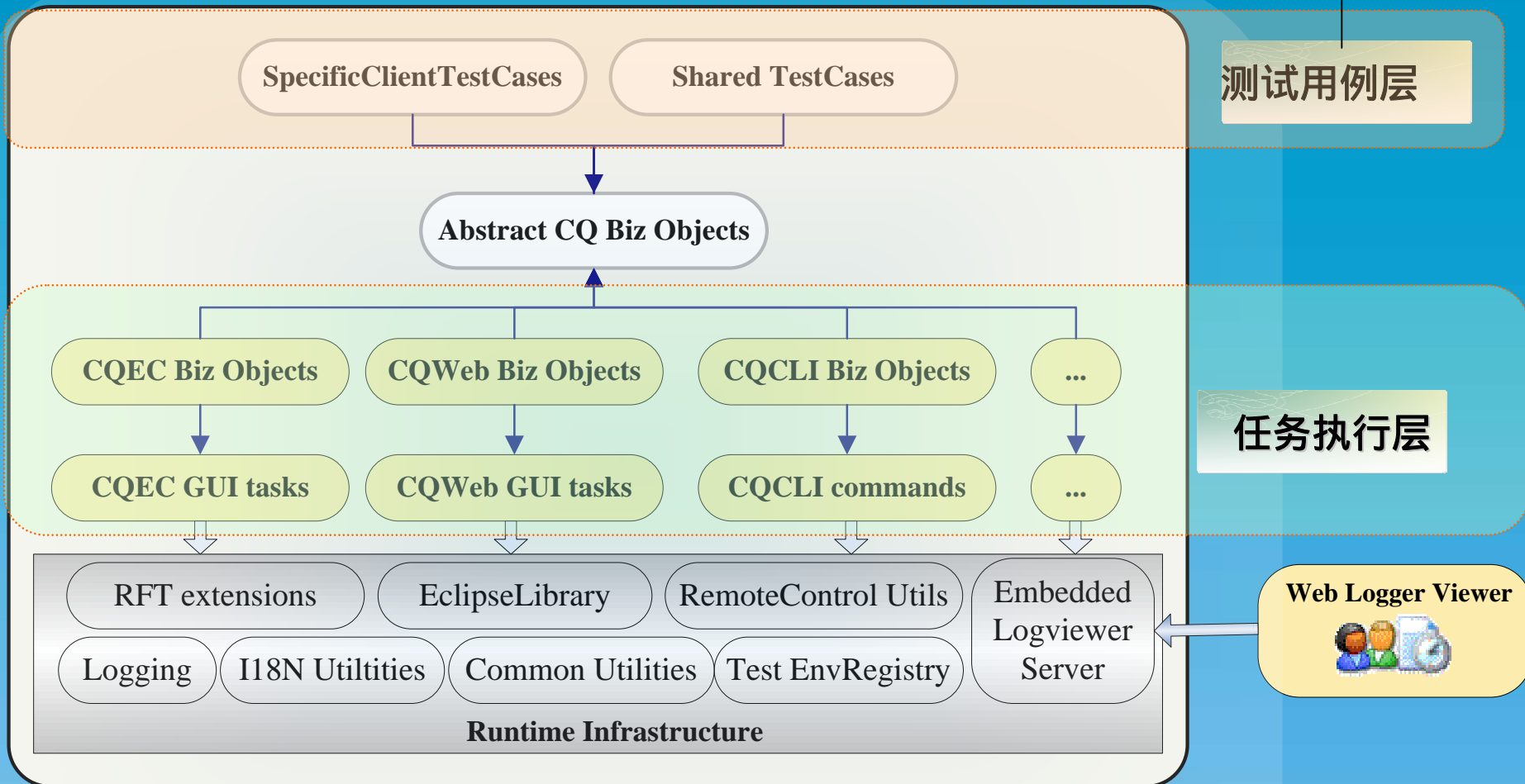
CreateCQECQuery

CreateCQWebQuery

Runtime Infrastructure

- 提高测试脚本的可读性
- 重用业务对象
- 重用测试逻辑

# 基于业务逻辑的分层结构



# 动态的系统操作与验证

- 动态操作
  - 图形控件的定位与操作
  - 操作动作序列的组织
- 动态验证
  - 表面的结果以及系统内部的状态
  - 对于验证结果的处理

# 构建基础架构

- 处理动态系统
  - 动态图形控件的查找与操作
  - 系统运行状态的动态验证
- 测试系统环境的管理
- 日志记录
- 运行监测
- 通用工具类支持
- 构建、初始化、执行脚本的过程



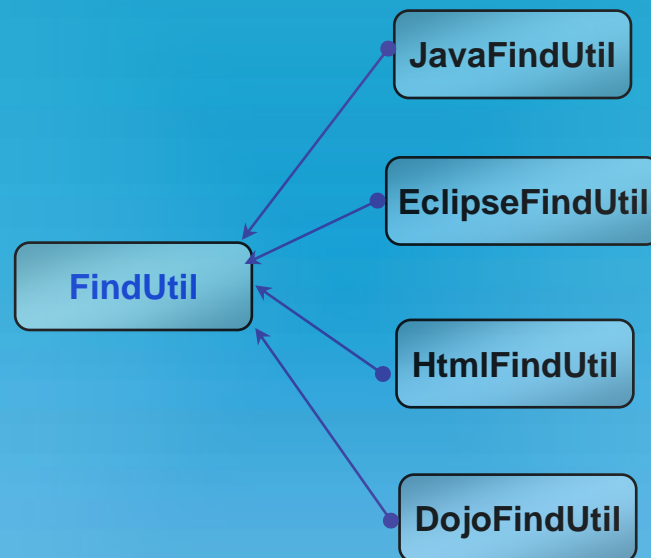
# 应对动态系统

- 动态查找
  - 动态界面的不确定性
  - I18N
- 动态验证
  - 界面上动态数据
  - 系统内部数据



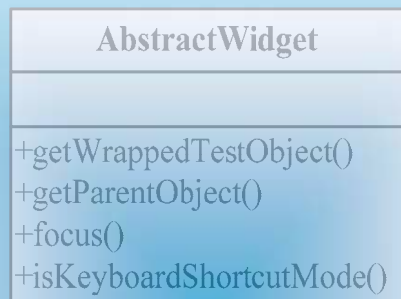
# 图形控件的定位

- 使用动态查找的方式定位对象
  - 基于RFT API
  - 停等/重试机制
  - 支持Java, Web, Eclipse, Dojo
- 使用简单



# 图形控件的操作

- 图形测试对象的封装
- 测试对象的使用
- 对鼠标模式及快捷键模式的支持



```
GuiTestObject loginTestObj =
    EclipseFindUtil.findCombo(...);
SWTCombo swtLogin =
    new SWTCombo(loginTestObj);
swtLogin.select(0);
swtLogin.select("server001");
```

# 图形控件的定位与操作 – 全球化界面识别问题的解决

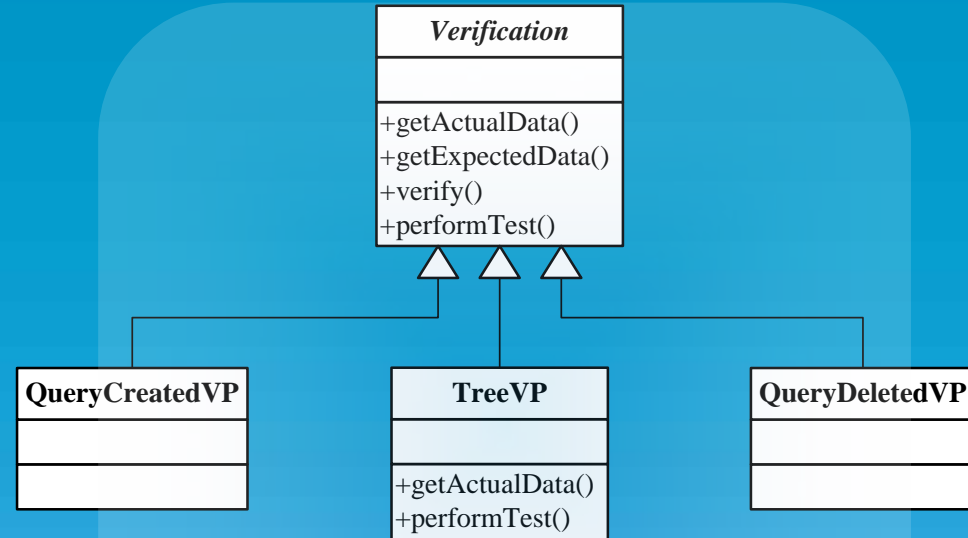
- 运行期间
  - 动态查找
  - 根据配置动态绑定资源文件，获取识别信息
    - Java与Eclipse应用，绑定程序自带的资源文件
    - Web应用，绑定开发期间生产的资源文件
- 开发期间
  - Java及Eclipse应用
    - 使用标签机制获取界面识别属性的资源位置
  - Web应用
    - 远程收集、处理资源文件，并使用程序生成统一的资源文件





# 动态验证

- 继承同样的抽象类，运行机制相同
- 界面数据的动态验证
  - 引入停/等机制
- 被测应用状态的动态验证



```
TreeVP treeHostVP = new TreeVP(ccTree, tagName, true);
treeHostVP.verify();

CQECPermissionInfoVP permissionVP =
    new CQECPermissionInfoVP(folderName, expectedPermission)
permissionVP.verify();
```

# 日志管理

- 与RFT自身统一的接口
  - 提供XML日志实现
  - 兼容其它日志实现
  - 提供灵活配置
- 单例实现，易于使用



# 测试环境管理

- 测试环境参数的配置
  - runtime.properties
- 测试环境参数读取与更改
  - 全局的注册变量
- 测试环境的管理 – 远程控制工具集
  - 初始化
  - 动态配置
  - 清理



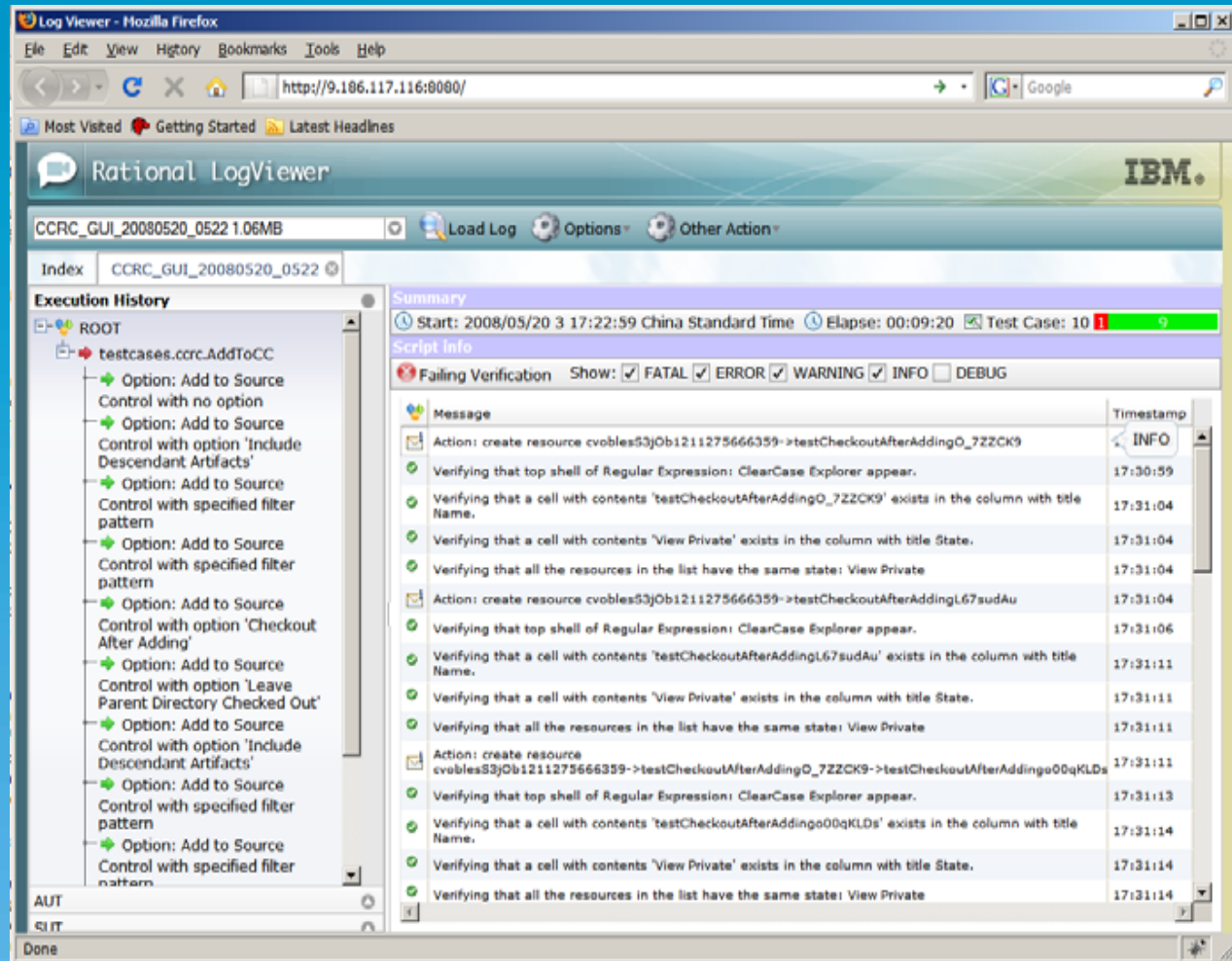
# 通用工具支持

- 文件及字符串操作
- 通用XML及专有XML数据文件解析
- 全球化测试字符串生成



# 基于Web的日志查看工具

- 清晰直观
- 内置的Web服务器
- 配置简单、自动启动
- 在线监控、实时更新



The screenshot displays the Rational LogViewer web interface. The browser window title is "Log Viewer - Mozilla Firefox". The address bar shows the URL "http://9.186.117.116:8080/". The page header includes the Rational logo and the text "Rational LogViewer". Below the header, there are navigation buttons for "Load Log", "Options", and "Other Action". The main content area is divided into two panes. The left pane, titled "Execution History", shows a tree view of test cases under the "ROOT" folder. The right pane, titled "Summary", displays test results for the selected test case "CCRC\_GUI\_20080520\_0522". The summary includes the start time (2008/05/20 3 17:22:59 China Standard Time), elapsed time (00:09:20), and test case ID (1019). The "Script info" section shows a "Failing Verification" status. The "Message" table lists various actions and verification steps with their timestamps.

Message	Timestamp
Action: create resource cvobles53jOb1211275666359->testCheckoutAfterAddingO_7ZZCK9	17:30:59
Verifying that top shell of Regular Expression: ClearCase Explorer appear.	17:31:04
Verifying that a cell with contents 'testCheckoutAfterAddingO_7ZZCK9' exists in the column with title Name.	17:31:04
Verifying that a cell with contents 'View Private' exists in the column with title State.	17:31:04
Verifying that all the resources in the list have the same state: View Private	17:31:04
Action: create resource cvobles53jOb1211275666359->testCheckoutAfterAddingL67sudAu	17:31:04
Verifying that top shell of Regular Expression: ClearCase Explorer appear.	17:31:06
Verifying that a cell with contents 'testCheckoutAfterAddingL67sudAu' exists in the column with title Name.	17:31:11
Verifying that a cell with contents 'View Private' exists in the column with title State.	17:31:11
Verifying that all the resources in the list have the same state: View Private	17:31:11
Action: create resource cvobles53jOb1211275666359->testCheckoutAfterAddingo00qKLDs	17:31:11
Verifying that top shell of Regular Expression: ClearCase Explorer appear.	17:31:13
Verifying that a cell with contents 'testCheckoutAfterAddingo00qKLDs' exists in the column with title Name.	17:31:14
Verifying that a cell with contents 'View Private' exists in the column with title State.	17:31:14
Verifying that all the resources in the list have the same state: View Private	17:31:14



# 自动化测试系统带来的价值

- 对于QA部门：降低成本 提高效率
  - CQ全功能的回归测试：手工测试(3天) vs. 自动化测试(0.5天运行+0.5天分析)
  - 整合到端到端的测试平台，24\*7 全自动的配置与执行
  - 节省扩展成本：新增客户端的开发，30%开发成本的节省
  - 较小的维护成本
- 对于整个项目组织：实现开发成果的共享
  - 重用测试逻辑进行API测试
  - 重用公共的工具集

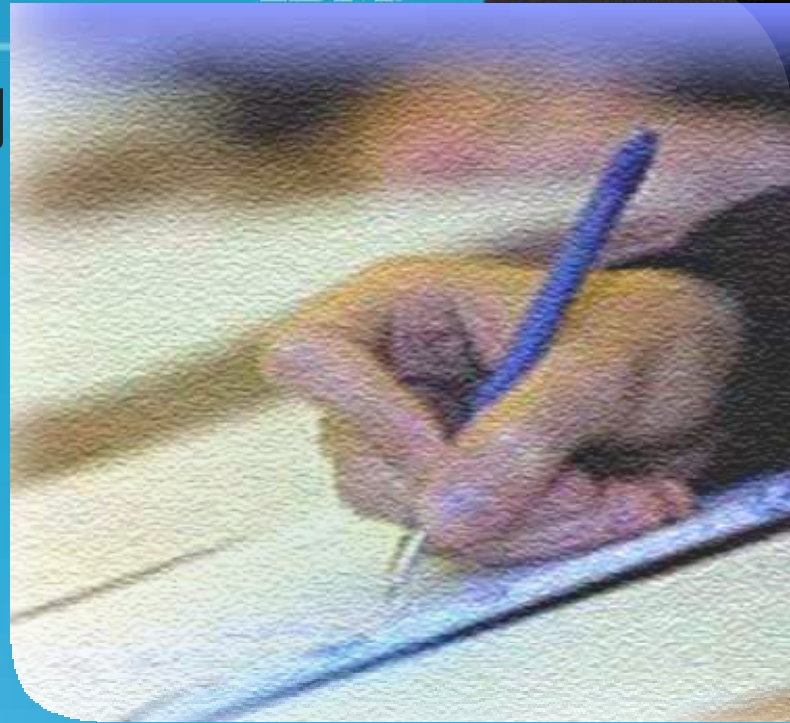
# 进行自动化开发的最佳实践

- 划定自动化测试用例范围
  - 什么是合适进行自动化测试？
  - 选取自动测试用例的原则
- 选择自动化开发良机
  - 被测产品可用前
  - 被测产品可用后
- 开发方法
  - 用例为准
  - 伪代码为先
  - 进行充分的单元测试



## 总结：基于RFT构建可重用的自动化测

- 开发方法
  - 业务建模
  - 结构分层
  - 分层开发
  - 提高重用
- 基础架构构建
  - 在RFT基础上进行功能扩充
  - 构建通用的工具集





# RFT V8 新功能

- 提供对更多应用域的支持
  - .NET 3.0
  - SAP gui 7.1, Siebel 8.0, Flex 3.0
  - DOJO Toolkit, Flex support on Firefox
  - PowerBuilder 11
  - Eclipse GEF
- 与其它产品的集成
  - 在RQM中管理与执行RFT脚本
  - 在RQM中查看分析RFT执行过程中产生的日志
  - 使用RQM集中的数据池
  - 同RTLM (Rational Test Lab Manager)集成



# 问题讨论



# 感谢您的参与!