



2008 IBM Rational
软件开发高峰论坛

Rational software

开发的智慧 协作的力量



在跨地域全球化的环境下 如何有效的敏捷开发



高伟斌

IBM Rational ClearQuest 产品开发工程师

gaowb@cn.ibm.com

什么是敏捷开发



- 一种迭代化不断进化的高协作开发方式
- 创造高质量软件并能节省成本和及时满足变化的需求
- 主要原则
 - 减少浪费
 - 恰到好处
 - 持续的测试和验证
 - 稳固的团队协作
 - 快速应对变化
 - 客户参与
 - 及时交付



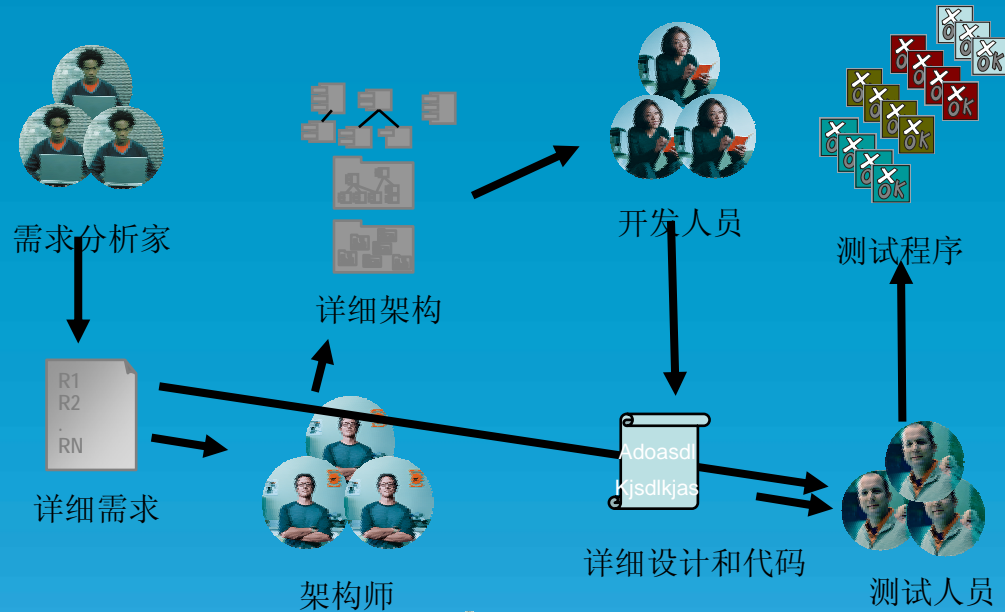
跨地域全球化敏捷开发经验共享

- IBM Rational ClearQuest 产品开发
 - 有十几年历史的 Rational 旗舰大产品
 - 开发主要分为核心、Eclipse 客户端和 Web 客户端等开发
 - 架构和开发语言平台复杂
- 开发团队
 - 中国、美国(东西部)、日本、印度
- 2007年初开始全面敏捷开发

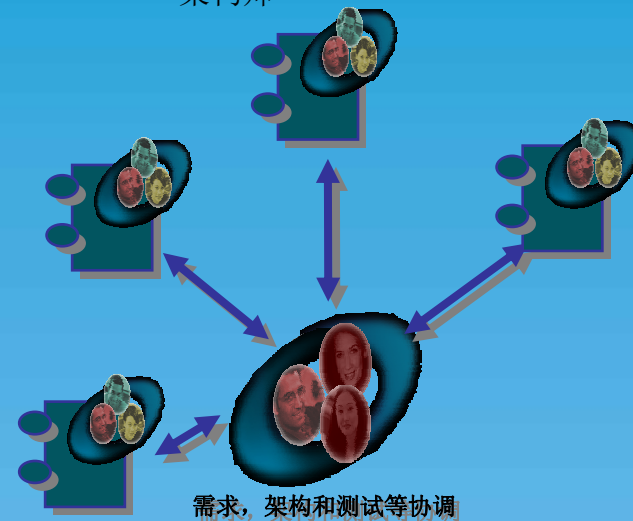


敏捷开发 — 减少浪费

- 浪费的产生
 - 不同部门之间太依赖繁重的移交

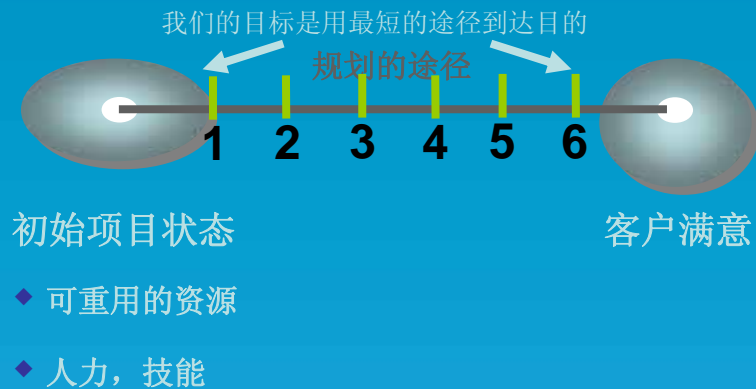


- 如何减少浪费
 - 跨部门用最少的移交紧密协作
 - 组织部门间松耦合



敏捷开发 — 恰到好处

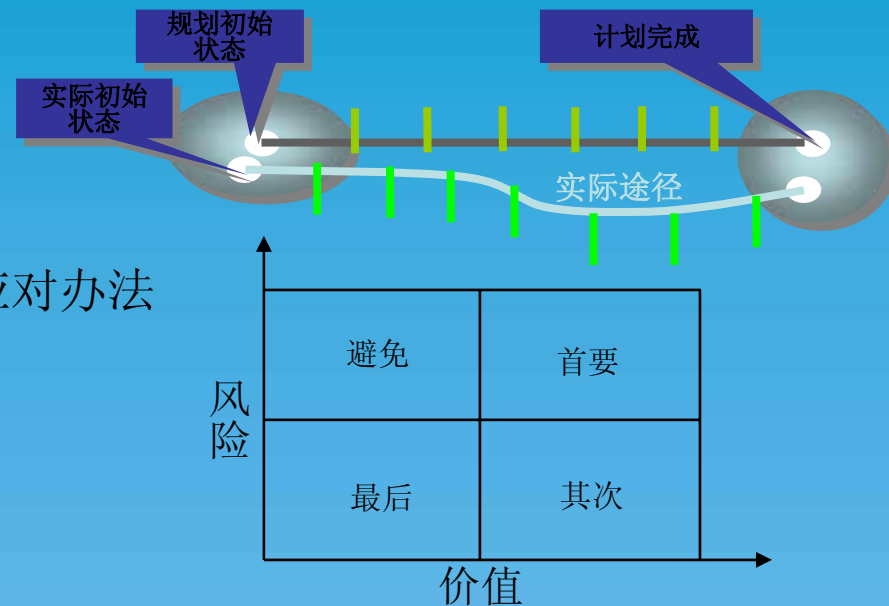
- 理想的高层开发计划



- 事实上都不是最短途径

- 如何尽可能做到最短途径

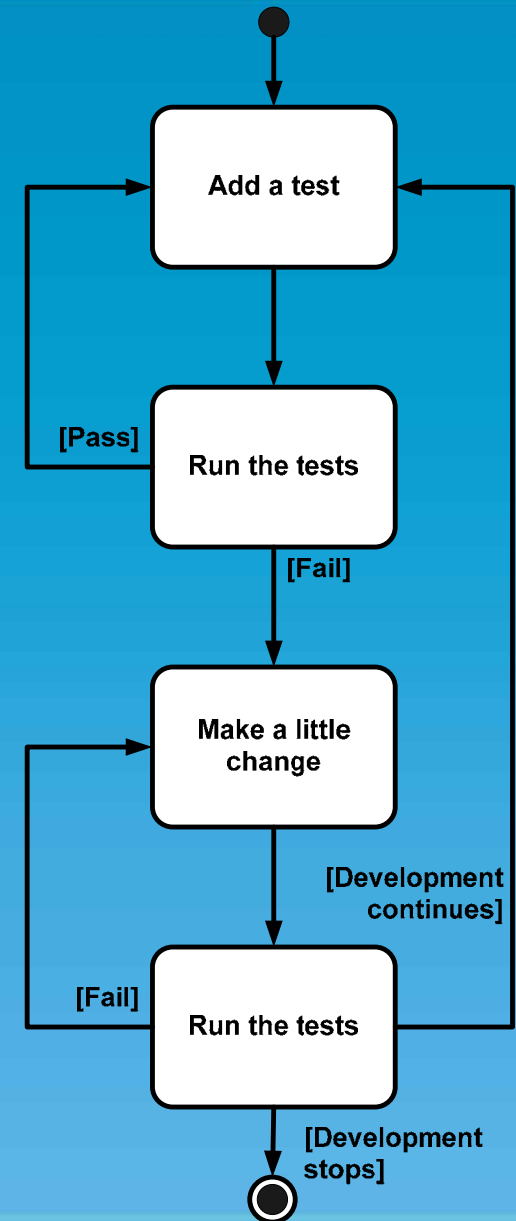
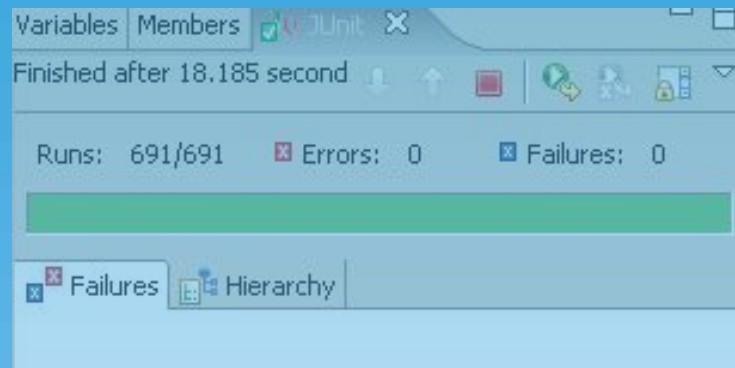
- 早期知晓高的风险, 并确保有应对办法
- 越重要的越早



敏捷开发 — 持续的测试和验证

- 测试驱动的开发
 - 越早越好
 - 在循环迭代中不断完善
 - 高的覆盖率
- 每天都 **Build** 和集成
 - 编译代码
 - 集成测试

- 自动化



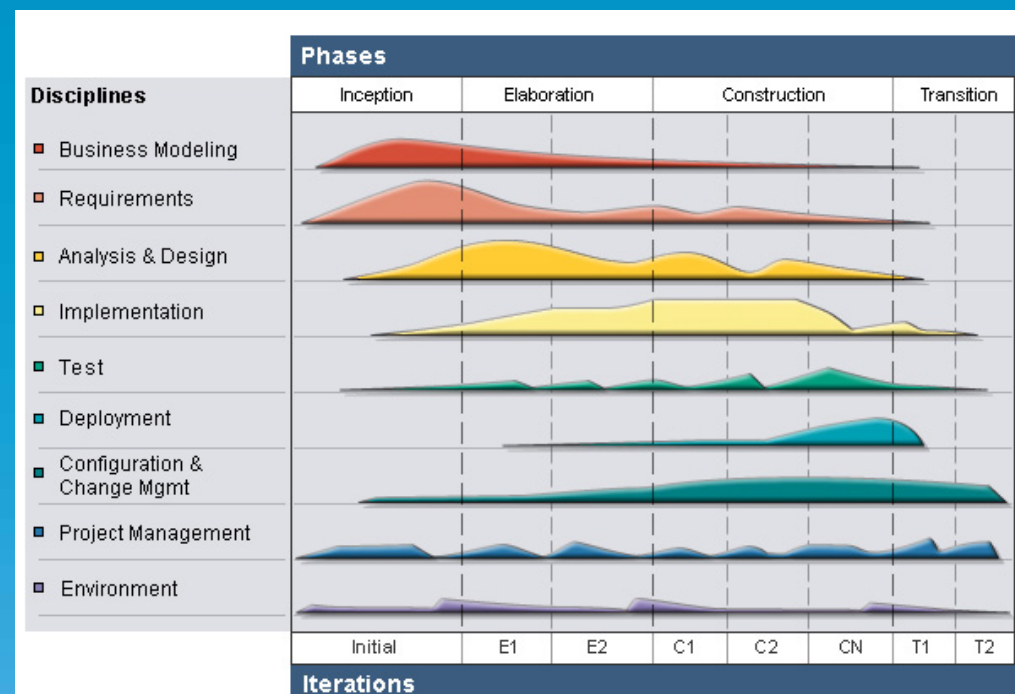
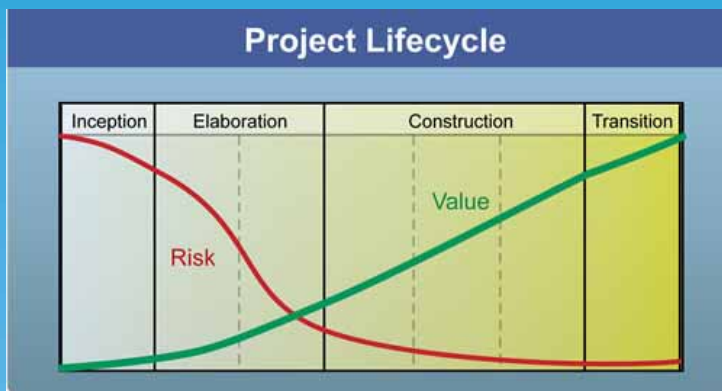
敏捷开发 — 稳固的团队协作

- 共同的目标
 - 清楚了解项目
- 所有的团队成员
 - 任务安排尽量自愿选择
 - 工作是乐趣
 - 自我导向
- 决策
 - 找最佳解决方案或行为
 - 行为统一



敏捷开发 — 快速应对变化

- 迭代化开发
- 精炼代码



敏捷开发 — 客户参与

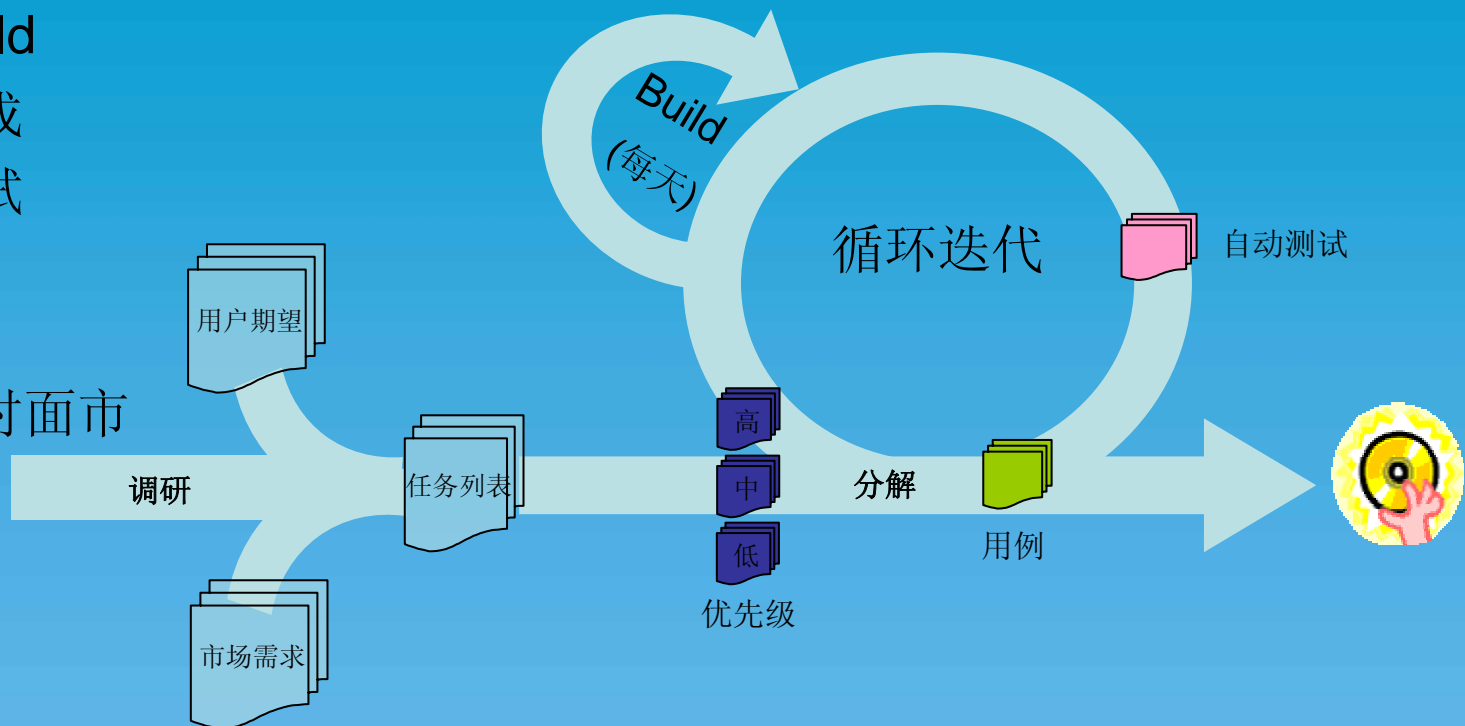
- 及早参与
 - 需求分析
 - 设计构建
 - 测试
- 客户满意
 - 功能
 - 质量
 - 共赢



敏捷开发 — 及时交付

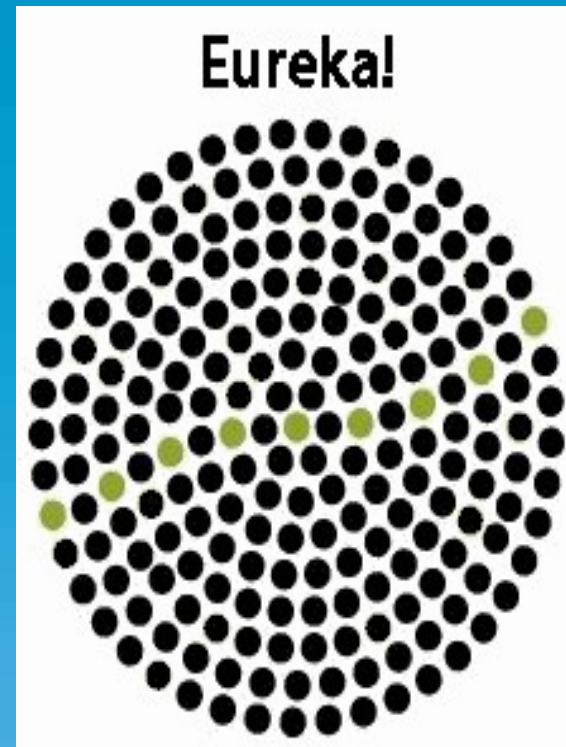
- 每天
 - 更新代码
 - 同步代码
 - Build
 - 集成
 - 测试

- 产品
 - 及时面市



敏捷开发后明显变化

- 迭代化更强
- 沟通更有效，响应及时
- 新的角色、职责和流程
- 开发人员不仅要去做单元测试，而且要做功能测试
- 系统测试和性能测试重点在像客户环境下的集成测试
- 产品质量提高了



跨地域全球化敏捷开发 — 开发流管理

- 开发团队使用 Rational ClearCase 统一变更管理 (UCM)
- 不同的流用来隔离变化和保证产品的稳定性
- 所有的流都要测试，向上交付时指定的测试必须要通过
- 测试、交付和同步都有相关的流程参考

开发人员 workflow

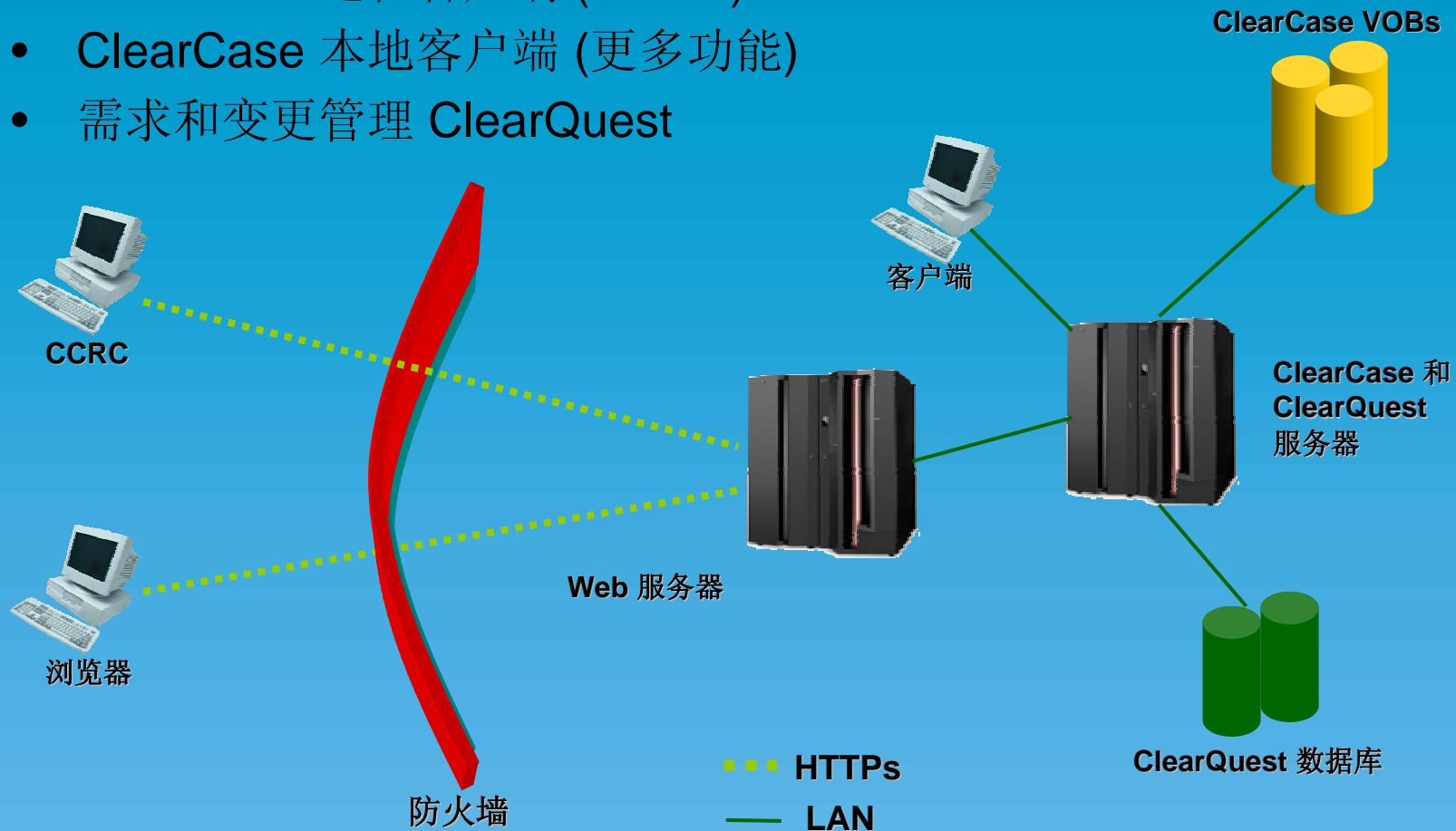
功能流

产品流

集成流

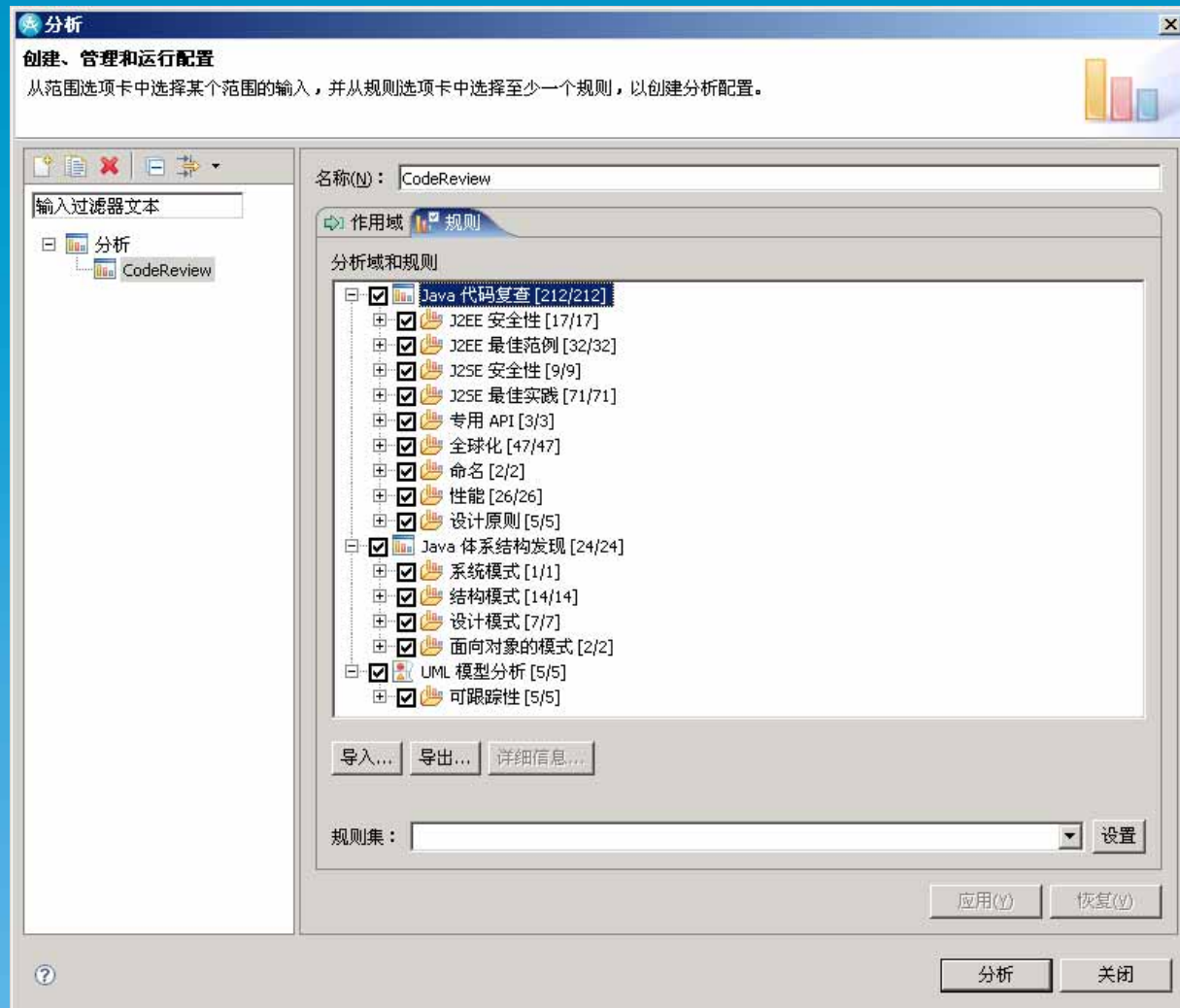
跨地域全球化敏捷开发 — 开发环境

- ClearCase 远程客户端 (CCRC)
- ClearCase 本地客户端 (更多功能)
- 需求和变更管理 ClearQuest



在 RSA/RAD 中配置代码复审保证质量

- 根据实际业务灵活定制
- 添加自定义的复审规则
- 通过导出导入统一复审规则集
- 提供改进建议和自动快速修正

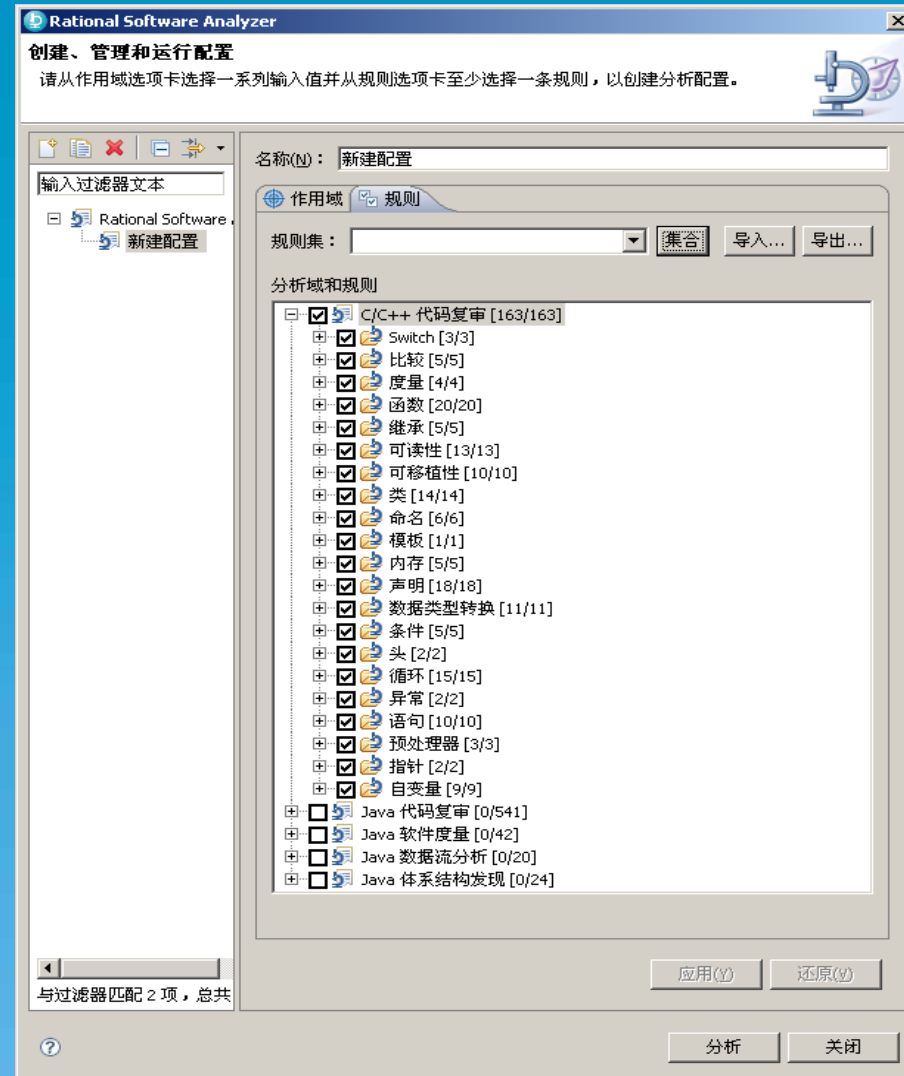


RSA/RAD 代码复审结果实例

The screenshot displays the Rational Software Architect interface for a Java project named 'CBEFormatter.java'. The main window shows the source code with a highlighted line: `new Integer(cmLogRecord.getThr`. The left-hand pane, titled 'CodeReview [Java 代码复查, Java 体系结构发现, UML 模型分析]', shows a hierarchical tree of analysis results. The 'Java 代码复查' section is expanded, listing various findings such as 'J2EE 安全性', 'J2SE 安全性', '命名', '全球化', '设计原则', '性能', '避免使用 new 来实例化布尔值', '避免使用 new 来实例化字符串', '避免使用静态集合', '避免在循环中使用 java.lang.String + 运算符', and '当从基本类型转换为 java.lang.String 时, 应避免实例化不必要'. The right-hand pane shows the '大纲' (Outline) view, listing the package structure: `com.ibm.rational.common.l`, `导入声明`, `CBEFormatter`, `cmLogRecord : CMLog`, `CBEFormatter()`, `Format(LogRecord)`, and `convertLevelToSeveri`. The bottom status bar indicates '只读' (Read-only), '智能插入' (Smart Insert), and '121 : 84'.

Rational Software Analyzer (RSAR) 中代码复审

- 更专业的代码复审工具
- 支持 C/C++ 代码复审
- 更多 Java 代码复审规则
- 可分析软件度量和数据流



跨地域全球化敏捷开发 — 代码视察

- 指定专门的内部和外部视察资深工程师
 - 有流程为不同时区的有效视察
 - 视察通过并且测试没问题后才可交付
 - 视察有记录可追溯

Repeatability	<input type="text"/>	▼
Code Reviewed By:	gaowb	▼
Product Blocked:	<input type="text"/>	▼

- 必要时通过 Lotus Sametime 或电话会议及时沟通交流已尽快达到共识

```
:: Zhen Jie Shen joined the chat. :: 20:42:11
:: Ravi Gopalakrishnan joined the chat. :: 20:42:11
:: Chen JC Jiang joined the chat. :: 20:42:11
Zhen Jie Shen I think that we may have some risks on two defects. 20:42:22
Ravi Gopalakri... ok, which ones? 20:42:29
Zhen Jie Shen RATLC01037958 Repeated deletes cause client crash Fred 20:42:30
RATLC01222188 When deleting a large number of records from the query view, CQ hangs, needs to be restarted. Jim
    ◀ These two. 20:42:33
    ◀ They are both memory problems. 20:42:48
    ◀ Fred has reproduced one some days before. But still can't find the root cause. 20:43:03
Ravi Gopalakri... yes, I am aware of them. Please coordinate with Michael Tabb and Morgan Stair if you get stuck 20:43:08
gaowb@cn.ibm... It is good to nail down the causation in CQEC or CQ Core? 20:44:01
Zhen Jie Shen Ok. I got it. I'll ask them to contact Michael and Morgan. 20:45:29
```

跨地域全球化敏捷开发 — 实时测试

- 使用 **Rational PurifyPlus** 为可靠和高效的代码进行运行时分析
 - Purify: 自动定位内存相关错误, 如内存泄漏、不正确的存取管理
 - PureCoverage: 寻找未被覆盖的代码

Function: DisplayLocalTime

```

Line Coverage Line Number
13 void DisplayLocalTime(int nBuf
14 char buf[100])
15 {
16     SYSTEMTIME SystemTime;
17     GetSystemTime(&SystemTime);
18     int RetCode;
19     GetLocalTime(&SystemTime);
20     RetCode = sprintf(buf, "%d\\%d\\%d %d", "Currently ", SystemTime.wHour,
21     SystemTime.wMinute, SystemTime.wSecond, ...);
22     if (!DisplayPing && ((int)strlen(buf) == RetCode))
23     {
24         MessageBox(NULL, buf, "Local Time of Day",
25         MB_OK | MB_ICONINFORMATION);
26     }
27 }
28
29
30
31
32
33
34
35
36
37
  
```

Starting Purify'd hello.exe at 05/06/2002 15:16:04

Starting main

UMR: Uninitialized memory read in strlen (1 occurrence)

ABV: Array bounds write in WinMain (1 occurrence)

Writing 1 byte to 0x01c23992 (1 byte at 0x01c23992 illegal)

Address 0x01c23992 is 1 byte past the end of a 10 byte block at 0x01c23988

Address 0x01c23992 points to a malloc'd block in heap 0x01c20000

Thread ID: 0x5a0

Error location

WinMain [hello.c:30]

length = strlen(string2); // UMR because string2 is

for (i = 0; string1[i] != '\0'; i++) { // ABV's generated on this

string2[i] = string1[i]; // ABR generated on this

length = strlen(string2); // ABR generated on this

WinMainCRTStartup [winCRT0.obj]

Allocation location

malloc [malloc.obj]

WinMain [hello.c:25]

WinMainCRTStartup [winCRT0.obj]

ABV: Array bounds write in WinMain (1 occurrence)

ABV: Array bounds write in WinMain (1 occurrence)

ABV: Array bounds write in WinMain (1 occurrence)

ABV: Array bounds read in strlen (1 occurrence)

Displayed Errors: 5 of 5 Displayed Warnings: 2 of 2 Bytes leaked: 10+

- Quantify: 发现程序的性能瓶颈

Functions: [walk_annotations(win)]

Line time	L-D	Percent of Function time	Percent of I-D time	Line number	
0.01	C.C1	3.01	0.00	200	*****
37.86	3.34	43.43	15.44	98.10	201 {
					int i;
					for (i=0; i<SPECIMEN_LENGTH(specimen); i++)
					if (specimen.work_copy[i] == 'C'
					specimen.work_copy[i] == 'M'

Time spent in line alone

Time spent in line plus its descendants

Function: mark

Called: 1

Function time: 230.3

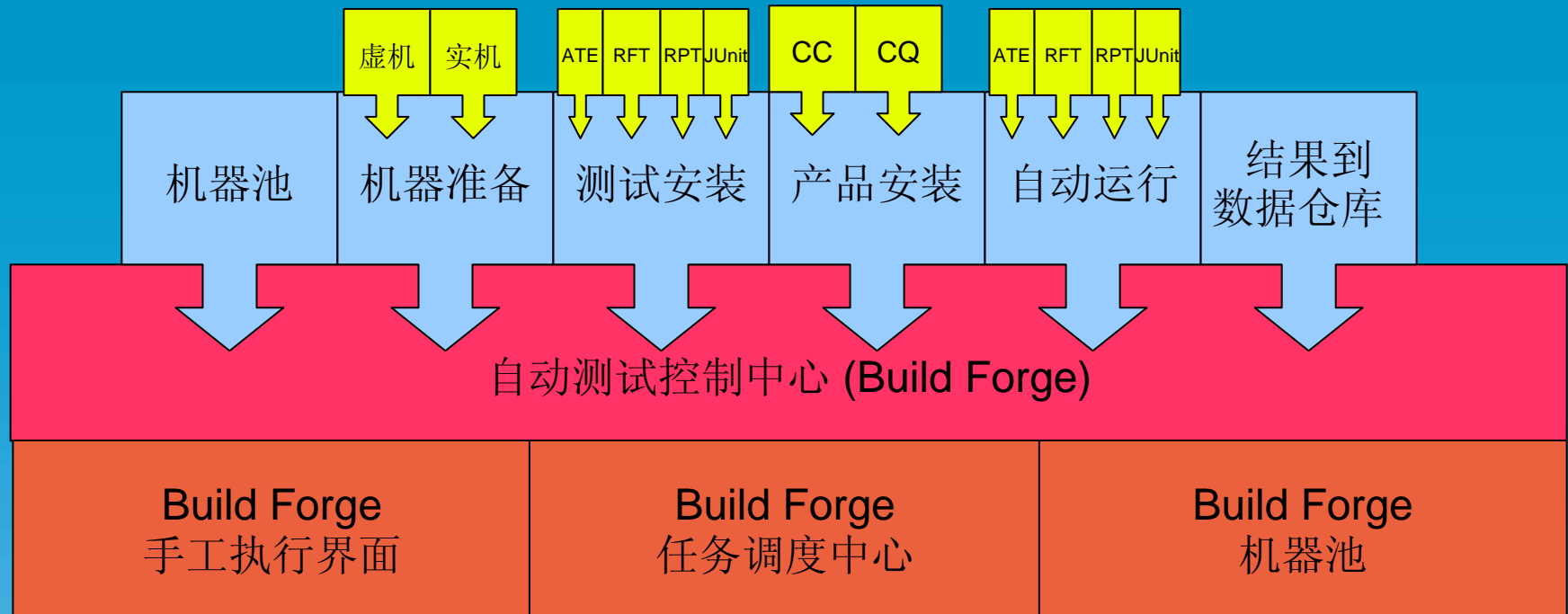
F+D time: 10.13

Distribution to Callers:

Called 1 times analyze_prebody



跨地域全球化敏捷开发 — 自动化测试框架



测试套件的实现在
IBM Rational Build Forge 任务中

Build Forge 中 Build 后触发测试示例

Baseline: 7.1.D080528 Product: ClearQuest

Platform	Test Name	Test Start Time	Host	Test Status	Test Fail Reason	Test Log
aix4_power	ATE: RE Accept	05/29/08 07:56:48	ryka	✓	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528
linux_x86	ATE: RE Accept	05/29/08 07:36:37	qvm1040	✓	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528
	ATE: Regression Test	05/29/08 08:15:25	qvm1013	✗	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528
nt_i386	RFT: CQ Designer	05/29/08 08:46:34	QVMW356	✗	-	http://qetools.lexma.ibm.com:8888/fullcontrol/index.php?mod=jobs&action=edit&bf_id=13885
	RFT: CQ Eclipse (RCP)	05/29/08 08:51:16	QVMW356	✓	-	http://qetools.lexma.ibm.com:8888/fullcontrol/index.php?mod=jobs&action=edit&bf_id=13885
	RFT: CQ Falcon	05/29/08 09:03:38	QVMW356	✗	-	http://qetools.lexma.ibm.com:8888/fullcontrol/index.php?mod=jobs&action=edit&bf_id=138845
	ATE: RE Accept	05/29/08 07:57:07	qvmw453	✓	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528
	ATE: Regression Test	05/29/08 08:47:04	qvmw189	✗	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528
sun5	ATE: RE Accept	05/29/08 07:36:21	qsun103	In Progress	-	https://ftasrv02.lexma.ibm.com/eea/trigger/7.1.D080528

Comments/Defects for All Platforms

Comments :

Platform Specific Details

None

运行状态

- 用的哪个Build?
- 结果自动上传到网页上
- 点击链接可查看详细日志

7.1.D080515			
[changes]			
Platform	Build Status	Spin Status	Accept Status
aix4_power	✓	✓	✓
hp11_ia64	✓	✓	-
hp11_pa	✓	✓	-
linux_390	✓	-	-
linux_ppc	✓	✓	-
linux_x86	✓	✓	✓
nt_i386	✓	✓	✓
sol_x86	✓	✓	-
sun5	✓	✓	✗

7.1.D080514			
[recommended baseline] [changes]			
Platform	Build Status	Spin Status	Accept Status
aix4_power	✓	✓	✓
hp11_ia64	✓	✓	-
hp11_pa	✓	✓	-
linux_390	✓	✓	-
linux_ppc	✓	✓	-
linux_x86	✓	✓	✓
nt_i386	✓	✓	✓
sol_x86	✓	✓	-
sun5	✓	✓	✓



do_eea Execute Cancel

Job Details

Job Steps

Project Parameters

Selector:

Class:

Tag Format:

Tag Example: DO_EEA_10

Project Tags

 Editable TagsB

Project Environment

picklists:

target

product

release

force_uninstall

stream

build

test_harness

scenario

Save Environment

测试机器

测试的产品

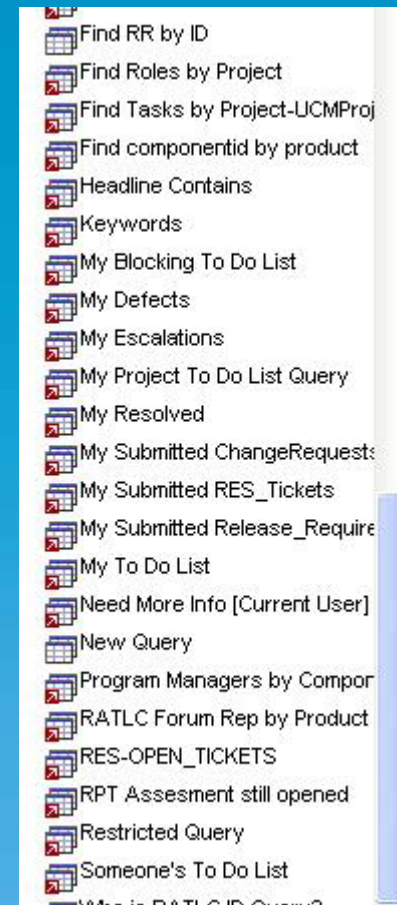
版本信息

测试环境

测试场景

跨地域全球化敏捷开发 — 任务选择和估算

- 开发人员自愿选择
 - 不过分依赖项目经理
 - 根据每个开发人员的侧重点和兴趣合理安排
 - 平衡每个开发人员的工作量和保证项目进度
- 通过 Rational ClearQuest 了解最新任务情况
 - “我的任务”
 - “她/他的任务”
 - “部门任务”
 - “产品/项目任务”
 - “产品缺陷”
- 每个任务和缺陷都有开发人员自己设置估计完成时间



Estimated Completion Date: 2008-08-15 21:22:27



问题讨论



多谢

了解更多:

- [IBM Rational 软件](#)
- [IBM Rational 软件交付平台](#)
- [过程与项目组合管理](#)
- [变更与发布管理](#)
- [质量管理](#)
- [架构管理](#)
- [Rational 使用版下载](#)
- [引领创新](#)
- [developerWorks Rational](#)
- [IBM Rational 视频](#)
- [IBM Rational 合作伙伴](#)