

TOP100

全球软件案例精选集

NOKIA Connecting People **SONY**



HUAWEI



NARI



Microsoft
TECHNOLOGY



Alibaba.com

Human Factors



ORACLE
SOFTWARE POWERS THE INTERNET



Autodesk

ThoughtWorks



Nokia Siemens
Networks



Letv 乐视网
要有你的看法



TREND
MICRO



大众点评
dianping.com



H3C



JKTCOM

Research
微软亚洲研究院

Panasonic



CNC
中国网通

SOSO 搜搜

中国平安

PING AN

intertrust



sina 新浪

用友
yongyou

msup



CISCO



金山
KINGSOFT

lenovo 联想

eico



RayooTech

科大讯飞



IBM

Kingdee



Baidu 百度
Google



Kinect

Alcatel-Lucent

Tencent 腾讯

NEUSOFT
东软

当当网

MARVELLOU

淘宝网

搜狐
sohu.com

方正科技

the 第九城市

ca



invent



揭示研发管理白金定律，分享那些激动人心的创新与变革，使团队获得更多源动力与更大的推动力！



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

内 容 简 介

《TOP100 全球软件案例精选集》源于全球软件案例研究峰会，精选了第一届峰会的精品案例。涉及软件研发中心的五个维度（质量管理\产品创新\架构设计\开发流程\团队管理），测重对案例的深入分析研究，挖掘案例成功的关键因素，总结成功案例的经验，反思失败案例的教训，以此给软件从业者带来思考，促进整个行业的进步，为国内软件研发中心的快速成长找到捷径。

本书有别于市面上其他的科技类书籍，书中的内容全部来源于知名软件研发中心的实际案例，案例的作者都是项目的一线带头人，他们从案例出发，详细讲述如何解决产品创新、团队建设、架构设计、流程再造、测试管理中遇到的问题，提供了一整套解决问题的方法，对问题中的关系进行量化，并进行解决问题的实践操作。从事不同工作的作者们试图用多种案例揭开知名研发中心的神秘面纱，展示业界先进管理、探索创新的一面。

本书主要面向 IT 从业人员，初级程序员、核心技术骨干或是中高层管理人员都能从这些最佳实践案例中学习 IT 行业最前沿最成熟的科技信息。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

TOP100 全球软件案例精选集 / 麦思博（北京）软件技术有限公司编著. —北京：电子工业出版社，2014.1
ISBN 978-7-121-21668-8

I. ①T… II. ①麦… III. ①软件开发—案例—世界 IV. ①TP311.52

中国版本图书馆 CIP 数据核字（2013）第 243126 号

策划统筹：王龙元

责任编辑：鄂卫华

印刷：中国电影出版社印刷厂

装订：中国电影出版社印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本：787×1092 1/16 印张：19 字数：424 千字

印次：2014 年 1 月第 1 次印刷

定价：56.00 元



凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

P R E F A C E

“他山之石，可以攻玉”。在分享成为主流的科技界，在知识无国界的今天，在新技术不断推进的现今，中国的软件研发团队日益壮大。在与国外优秀公司和团队带头人的接触中，我们发觉中国各个研发中心在理论学习和趋势研究上并没有落后于国外。举个例子，云计算和大数据这两项技术，国内的一些公司做的非常不错，至于像淘宝在“双十一”期间完成的过百亿的销售额背后的技术处理方法，更是让国外的公司感到惊叹。究竟国内在什么环节有所缺失造成了研发中心成长的瓶颈呢？

答案是案例研究。案例研究起源于哈佛大学法学院，之后随着案例研究的可操作性和蕴含的巨大价值延伸到其他领域。随着 IT 的快速发展，案例研究迅速切入这一行业，并产生了更为深远的影响。譬如创新圣地——硅谷，每周都有很多技术沙龙，主讲人都是在分享他们在各自岗位上所做的工作经验、方法、启迪等。正是这些无私的共享，不断地传播，才让成功的经验在美国如病毒一般传递，才在北美大地上盛开了无数创新之花，也带动了整个行业的迅速发展。

这种情况正好契合了本文第一句话，也是古人先贤留给我们的至理名言——“他山之石，可以攻玉”。多年服务于软件研发中心的我们发现可以通过丰富的案例而汲取著名企业的精华和经验。“青出于蓝而胜于蓝”，后来者居上。案例把一个个具有特殊性的具体开发情景展现在从业者的面前，为从业者提供了一个真实的练兵场，使从业者有机会接触到大量的实践经验。而这些实际问题的解决方法正是案例研究的精髓所在。帮助更多人了解这些精髓，掌握成功解决问题的方法，这些正是成立组委会召开全球软件案例研究峰会的初衷。

回想全球软件案例研究峰会的组织、准备阶段，真是一段让人激动又艰难的岁月。激动的是当大会在各个媒体发布召开通知后，各个知名研发中心的负责人都本着无私、分享的精神，发来了众多的案例，业界同仁的支持让我们倍感欣慰，更加觉得把这个峰会办好是一件非常有意义的事情。艰难的是发来的案例如此之多，而每一个案例都凝聚了各个团队的全年努力，每一个案例都是那么的鲜活，而我们却因为大会时间的限制，不得不从这些不分伯仲的案例中“吹毛求疵”，只能选出 TOP100，直到现在，想到那些被“淘汰”的案例，心里都有不舍。在案例选取中，我们力争筛选出真实、新鲜的案例，避免夸大案例或选用不适合国情的外国案例和过时案例，并采用评审人机制，让业界权威人士运用最新理论成果分析案例，打通理论和实践之间的围墙。

大会的成功在我们意料之中。可是大会仅仅是一个开始，大会的谢幕不等于案例研究的结束。这是我在整理大会资料时的感悟，特别是当我的桌子上摆满了 TOP100 案例分享者的案例文稿的时候，这些文稿就如同全球软件案例研究峰会的纪录电影一样，记录着中国 IT 界在这个年度的技术总结。梳理这些文稿的过程，其实正是梳理中国 IT 界年度成功经验的过程。我忽然发现我不能“私吞”这些财富，这些财富必将帮助更多的研发中心走上快速发展的康庄大道。于是，在案例大会结束后，我们又开始紧锣密鼓地整理大会的资料，并联系案例分享者多次对案例文稿进行修正。并且对这 TOP100 案例再次进行筛选，才有了《TOP100 全球软件案例精选集》这本书的诞生。

如今这本书终于要和大家见面了，在此我要对全球软件案例研究峰会的分享者表示感谢，没有你们的奉献精神，我们也无法走出案例研究的第一步。在我写本书前言的时候，本年度的全球软件案例研究峰会召开在即，现在正处于案例的筛选阶段，我很欣喜地看到更多高质量的案例被征集，这也从侧面反应了中国软件研发中心这一年又做出了不少成绩，同时感觉在案例研究的道路上，我们还可以做得更多。这也是我们明年努力的方向。

全球软件案例研究峰会组委会秘书长
麦思博（北京）软件技术有限公司 CEO
刘付强

《TOP100 全球软件案例精选集》是当今中国信息技术行业的一线企业，如外企 500 强的微软、谷歌、英特尔、亚马逊等，民企 500 强的华为、百度、腾讯、阿里巴巴等，国企 500 强的东软、浪潮、中国移动、中国电信等亲践者的经验合集。

——新维创新私塾创始人、《半面创新》作者 周宏桥

在这个知识大爆炸的时代，与其伏案钻研理论，不如抬头借鉴经验；在这发展大提速的时刻，与其自己摸石过河，不如复制成功基因。

正如马奇在《经验的疆界》中所说，组织学习模式中试错、模仿、天择都是低效的，而故事、模型才是高效的。本书汇聚了 IT 行业精彩纷呈的“案例故事”，同时也是过去一年中国 IT 精英们的经验之旅，定能给你带去“读一卷书，似十年路”的价值。

——中国系统分析员顾问团(CSAI)软件工程首席顾问、
《软件需求最佳实践》作者 徐 峰

案例学习是一种很高效的学习方式。国际上流行的法学、商学、医学的理论都是通过案例来说明。本书收集了信息产业具有代表性的案例，为同行们搭起了实践与理论的桥梁。这本案例精选集不但适合团队的管理者，也适合团队的每一个成员。

——网易有道技术总监 蒋炜航

案例是最好的老师，偶像温伯格书籍中的案例至今还历历在目，草莓酱定律等故事还不时让我有所启发。《TOP100 全球软件案例精选集》搜罗了软件研发领域最新最全的案例，一个个生动鲜活的故事让你如临其境，将大大提升你的学习效率。

——支付宝成都研发中心负责人张林

This book collects more than 50 most inspirational, innovational and practical case studies, not only from China software industry, but from international technical leading companies. These authors use the real world cases to demonstrate, analyze and summarize how they solve technical challenges, and how they improve engineering efficiency in the research and development process, which are well worth reading and learning from.

——西雅图创业协会理事 Bill Liu

目 录

C O N T E N T S

测试管理/管理平台	1
基于 Java Script 产品的 TA 框架的六次改进	2
游戏中的自动化测试	9
游戏资源文件自动化测试平台	14
淘宝网探索式测试实践	20
测试卓越驱动电信领域持续集成	28
七拼八凑搭建测试框架	39
金蝶自动化测试发展之路	45
淘宝 PC 自动化测试框架 AutomanX	50
跨平台的自动化测试与工具	54
问题驱动的软件测试设计	61
产品创新/用户体验	71
智能家庭软硬整合平台架构设计	72
Hold 住你的用户体验	76
包豪斯——以用户体验的名义创新	80
网页游戏产品创作的挑战	88
互联网的社会学表达	95
围绕需求，解决问题，做有意义的设计	98
Agile UX 设计挑战	101
Kinect 体感技术在企业级的应用及展望	110
一云多端 移动互联	124
架构设计/技术战略	129
当当网打造个性化推荐&精准营销生态系统	130

构建企业级应用开发平台	134
公有云计算开发与设计	141
服务化与体系化解决大量定制小项目开发困境	150
前端的云时代	167
做最好的 HTML5 游戏	172
在产品开发期间接受突如其来的新需求而迭代出另一款新产品	177
构建多设备协同的互联网业务平台	181
开发管理/流程再造	187
如何尽快完成相似产品的开发	188
软件产品线工程的应用研究与安全检查系统软件产品线的构建	197
HTML5 安全攻防之道	203
软件研发流程管理的七宗罪	217
XPP	220
航天新一代产品数据管理系统研发实践	224
适合互联网的简约架构的演进	228
在无线终端产品研发中实施敏捷软件线	237
知识产权和软件产品保护	249
团队管理/组织发展	253
Team Smell 深度剖析及团队建设模型	254
互联网项目敏捷转型记	259
常见管理思路的甄别和批判性管理思想的培养	263
打造自组织团队	266
从 QA 到 EP	271
矩阵管理和东方文化下的研发团队能力建设案例	279
无线暴风的敏捷交付与持续优化	285
担当——如何组建靠谱的软件研发团队	288
国企敏捷一二三	294
后记	298

测试管理/管理平台

基于 Java Script 产品的 TA 框架的六次改进

一、产品和项目背景简介

商务智能是对客户提供的数据进行分析和处理,最终提供各种分析报告,为企业的各种经营活动提供决策信息。本项目的产品是对分析的结果进行可视化的转化,生成各种图表,例如,柱状图,饼图,以及一些可以供上层 Web 应用程序员调用的可视化控件。此产品用 Java Script 开发,并且支持多种浏览器,同时也支持移动设备上的应用,于 2011 年 3 月立项,并投入研发。

二、自动化测试框架的目标

软件测试的开发必须用回归测试来确保已完成的产品特性的质量。自动化测试(Test Automation, TA)框架将维护大量的回归测试用例,在保证产品质量的环节中具有举足轻重的地位。因此,我们在开始设计测试架构的时候便为测试自动化制定了如下的长期目标,以指导我们不断改进方向。

- ① 健壮的、稳定的、易用的。
- ② 能够高效地管理测试用例。
- ③ 测试用例执行时间和维护成本最小化。
- ④ 可以与开发团队共享测试用例以及其执行环境。

三、自动化测试框架的演进

(一) 最初版本

此产品于 2011 年 3 月在 SAP 上海研究院立项,到 2011 年 6 月有了第一个版本的自动化测试。在自动化测试框架中,测试人员采用



作者姓名: 耿晓倩

作者简介: Test Team Lead

六年的 SAP 质量工程师经验。领导团队进行测试自动化架构改进以及测试工具的开发工作。熟悉各种测试用例设计的各种方法和模型以及测试用例管理。致力于测试流程的不断改进。

研发团队规模: 40 人

研发团队职能定位: 产品研发

SilkTest 来打开一个指定 URL 的 HTML 页面，每个测试用例对应一个 HTML 页面，每个页面包含一个图标以及设定的图标属性信息，并通过截图比对的方式来验证测试用例是否正确。测试用例的管理则通过 excel 来完成，为避免测试的重复以及全面的覆盖率，测试人员通过一个交叉表来组织测试用例之间的关系，从而使得图表公共组件的用例可以得到复用。如图 1-1 所示，对于 Backgroud 这个特性，我们只需要在任意一个图表中进行测试就可以了，这样避免了测试用例的冗余。例如，layout 特性，在饼图中是没有的。图 1-1 可以让我们清楚地知道目前产品特性以及测试设计优化后的覆盖率是否合格。

	Pie	Bar	Line	Backgroud	Title	Legend	Layout	Tooltip	Theme	ValueAxis	Animator
Pie	y			y	y	y		y	y		
Bar		y		y	y	y	y	y	y	y	y
Line			y	y	y	y	y	y	y	y	y

图 1-1

第一个版本的测试框架使得我们告别了繁重的手工测试，但其明显存在着许多的问题，如测试数据的冗余、编写一个测试用例比较烦琐而且很容易出错、测试用例和 Excel 中的记录没有同步等。最突出和亟待解决的问题是，随着测试产品特性的增多，测试用例变得庞大，而每个用例用一个页面来维护，会导致成百上千的页面的后期维护工作变得巨大而难以控制。同时，截图比较的方法也因为维护成本过高，而让测试人员的维护工作苦不堪言。产品特性或是设计的稍稍改变都会带来大量的图片更改工作，这迫使我们通过工具来解决这个仅仅靠人手而无法解决的问题。

（二）第 1 次改进：Selenium 的引入

随着 Selenium 的成熟和改进以及 Selenium2（test driver）的推出，页面的控制可以交由程序来实现。那么对单个页面上的元素的测试也可以通过对 DIV 内容进行控制而将相似的内容整合在一张页面下。因此，在第二个版本中我们尝试着采用 Selenium 的测试驱动。这样，我们可以将相似实现方式的图表放在一起，例如有坐标系的图表，柱状图表等。Selenium2 的应用使我们的测试页面数量大大减少。同时，我们可以通过对 DOM 的控制来截取目标元素的图片而不是截取整个浏览器的屏幕作为结果检测的依据。

为适应 Selenium test driver 的使用，我们要找到一个测试管理的框架来对测试资源及用例进行有效的管理、执行和调度。毕竟，Selenium test driver 只是一个测试驱动而非测试工具。对于一套测试流程来说还有很多事情要做。因为人手有限，我们决定发掘公司内部资源。TestManager 是 SAP 内部项目组自己开发的一个测试架构。它以 XML 形式管理测试用例，通过调用测试驱动以及参数来完成测试步骤。虽然，测试用例用 XML 的形式存储，但为了直观，我们仍然使用 Excel 来记录和管理测试用例。与此同时，我们写了一个可以将 xls 内容转化成可执行的 XML 文件的

工具。

那么又随之而来了哪些问题呢？

(1) 不同的测试用例在测试页面中可能会用到相同的数据，而测试数据 **hard-code** 在页面中不能重用。这样不但有测试数据冗余问题，还会存在不同人写的测试用例的覆盖率因个人经验而不统一的问题。

(2) 由于不同的用例整合在一个页面中，让测试用例的粒度变大，那么就会带来用例不好调试，出错后互相影响的情况。

(3) 当然，即使测试的页面变少了，随着产品新特性的不断增加，测试页面还是会越来越多，多到无法控制！

(4) 没有了测试工具的管理，写测试计划的人需要了解各个测试驱动的用法。

(三) 第 2 次改进：数据驱动

在上一版本中我们提到了四个挑战，这个版本重点是在思考并解决这些问题。

首先，是解决数据冗余的问题。要解决数据冗余就要把测试数据从测试逻辑中抽离出来。这样做需要对调用数据的方法进行封装，一方面要提供一个统一的可以对产品进行设置的接口，另一方面要实现测试用例调用数据的统一接口供测试人员设计测试用例使用。实现这两个方法之后，我们可以通过一个入口 (**TestUtil**) 来调用，同时测试数据可以统一存储在数据库或者数据文件中。要实现这种方式，还要对开发人员提供的大量接口做封装。例如，我们要在一个图表中设置两个属性，一个是要改变图表布局的边缘填充宽度，即 **layout** 的 **padding** 值；另一个是要控制图例不显示在图表上，即 **legend** 的 **visible** 的值。如以下代码所示，这两个操作可以通过 **API** 的调用来完成：

```
updateChartsByProperties ([[ 'layout.padding', [50,50,50,50]], [ 'legend.visible', false ]]);
```

为了将创建图表的函数和设置图表属性的函数合在一起便于使用，可以通过下面的方法来实现：**testUtil.generateChart (viz/line; TestData.alalmlml; ['legend.visible', false])**。这样我们就可以把 **dataset** 放入数据库中，把属性数据放入 **JS** 文件中，实现测试数据和测试逻辑的分类。

使用这种统一的函数调用接口设置和操作图表属性还可以帮助我们解决测试页面太多的问题。我们可以用一个 **html** 模板页面，来加载要用到的 **JS** 方法，然后通过调用来实现图表的产生、改变和消亡。如此看来，大量的测试页面最后只变成一个页面了。这是多么令人振奋的一次变革！

测试框架的基本原则之一是编写测试用例尽量简单且不需要关心技术细节。因此我们要对杂乱繁多的测试驱动进行封装，从而达到通过关键字驱动测试步骤的目的。对于写测试用例的人员来说，只知道一个 **WebTestDriver** 就可以了，我们用 **Java** 实现了它。

(四) 第3次改进：全自动化

多数测试组为了节省资源，都希望电脑能完成的事情就不要手工去完成，从而解放更多的人力资源。对于网络应用程序的测试来说，因为存在很多自动化的测试难点，所以全自动化是个不小的挑战。我们还在这条追求的路上。这里举几个我们为之努力并取得成果的例子。

我们都知道鼠标事件以及移动设备上的特性需要事件触发器，因此需要模拟触发的过程，这为自动化测试前提条件增加了难度。尤其对于移动设备，要完成在 PC 端浏览器中的模拟测试，需要程序暴露出事件的相关信息。我们和架构师紧密合作，通过监听、模拟、触发、重现等步骤完成了这一过程。测试用例只需要包含坐标位置、事件类型以及时间戳，就可以将事件过程在浏览器中重现。这既解决了鼠标事件的自动化测试过程，又解决了因移动设备多样性而带来的自动化测试覆盖率的困扰。图 1-2 是录制测试用例的流程图。

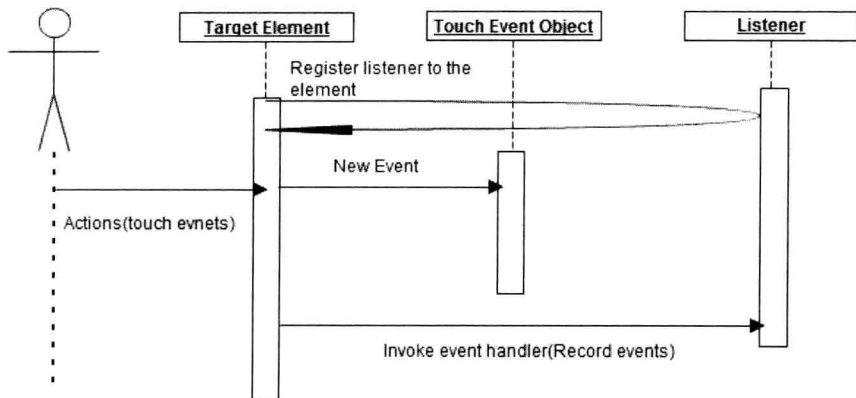


图 1-2

(五) 第4次改进：测试用例管理和持续集成

虽然在测试用例管理上我们已经做出了很多努力和进步，我们仍然需要引入专业的测试用例管理工具帮我们对大量的如一盘散沙般的用例进行高效的管理，以及根据不同的需求方便地遴选测试用例。我们引入开源和免费的工具 Testlink。

Testlink 有一个明显的优势在于，它暴露出了很多功能调用的接口可供我们直接与之通信。我们需要将测试用例导出成自动化测试框架可执行的测试计划文件。只要我们针对需要将导出的 XML 进行定制，就可以实现了。同时，导入的测试用例可以用 Excel 进行编辑，这对不懂技术的初学者或者新人是再好不过了。有了 Testlink，我们可以对每次测试的结果进行存储，以便于后期对产品质量进行缺陷分析，如图 1-3 所示。

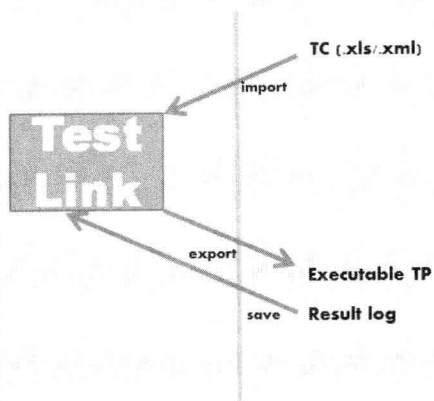


图 1-3

在 SCRUM 开发模式下，开发人员的持续集成为保证开发效率起到很重要的作用。为保证上传代码的质量，将测试框架集成到 CI Server 中也至关重要。因此，在这一版本中，我们将测试框架和虚拟机结合与代码服务器以及持续进程工具 Jenkins 进行了集成。到此，测试总体框架如图 1-4 所示。

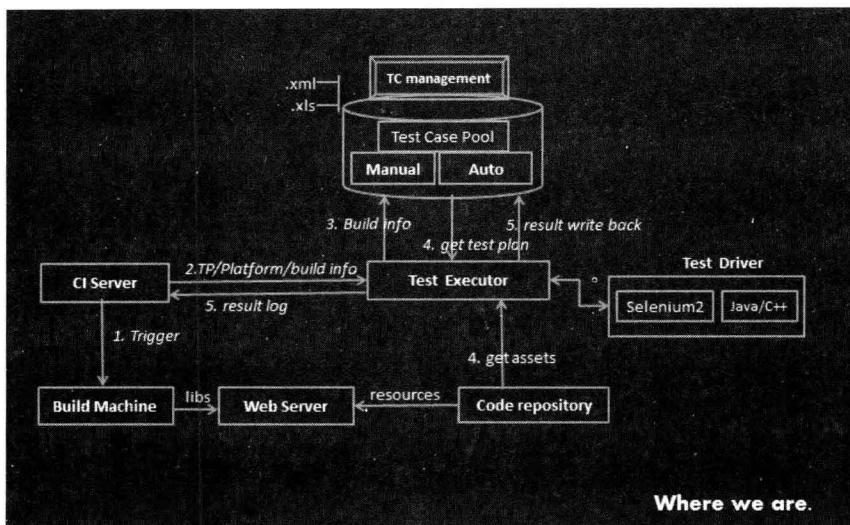


图 1-4

（六）第 5 次改进：像开发一样思考

在一个软件开发的团队中，测试人员的主要工作是对程序员写的代码进行测试，最终能够提

供产品的质量信息供程序员修复问题以及管理人员对发布产品的质量进行评测。为了做好测试工作，我们要了解产品的工作机制，以及程序员代码在软硬件环境中所起到的真正作用。

对于网络应用程序及产品，浏览器显示部分并不是产品代码所负责的功能部分。理论上，我们可以将其剥离开来，一来可以帮我们更加准确地测试自己的代码，二来可以剥离因浏览器的多样性而带来的渲染问题。如图 1-5 所示，我们从测试中去除了显示部分。

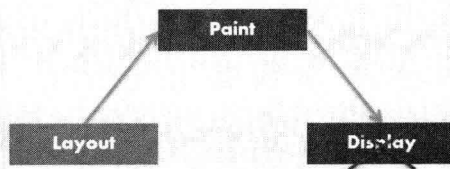


图 1-5

当然这一实现，也归功于 PhantomJS 的帮助。PhantomJS 通过程序对 webkit 的调用来画出页面的布局等元素，因为没有启用浏览器，对于结果验证可以通过后台截图比较的方式。这样做对每个截图的屏幕的分辨率便不再有严格的要求。关于 PhantomJS 的用法，请参见 <http://phantomjs.org>。

（七）第 6 次改进：和开发环境集成

在开篇的目标中最后一点谈到希望能够做到和开发团队共享测试用例以及执行环境。这一点是出于在敏捷开发模式下对时间方面的考虑。一般，敏捷开发周期很短，最常见的为一个月。在一个月中，要使开发人员所完成的程序达到产品发布水平，需要测试尽早介入。但由于代码初期的不稳定性，过早介入会给测试人员带来很多浪费。如果环境和用例可以让开发人员很容易地使用，那么在程序员开发的过程中就可以自己做简单的回归测试，进一步保证上传代码的质量。

Eclipse 的 plugin 可以将测试工具集成到 Eclipse 运行环境中来满足测试集成于开发环境的需求。基于 JUnit 框架进行扩展来对测试用例进行管理和执行。同时，借助于前面讲到的 PhantomJS，开发人员甚至可以通过对 webkit 的调用来实现 headless 的测试执行。这样开发人员和测试人员便可以共同维护测试用例。

四、ROI 分析

六次自动化测试架构的改进的投入产出分析可以从几个不同的方面看，例如，测试执行时间、测试用例生成时间、测试用例管理复杂度、测试环境准备时间，以及测试结果后续分析所需要的时间等。简单来看，1000+个测试用例从准备测试环境到执行到发布测试结果报告的时间花费从

之前的四个半小时缩短到了现在的 47 分钟。这让原本只可以晚上执行的测试用例集合可以在需要的时候随时执行。

五、成功因素小结

专注：一年半的时间，我们一直专注在自动化测试框架的改进上。这让我们不断积累经验，增加知识，做到厚积薄发。

合作：在本次改革的流程中，任何一项成功都是合作的结果。不光是测试人员内部的合作，还要和开发人员进行合作，还要和公司的不同部门（做类似产品的团队）进行合作。

持续改进：本着追求进步的思想才能持续不断地改进。遇到问题，要有积极的解决问题的态度。工具在改进的同时，我们每个人都在成长和进步。

360 度学习：我们说“三人行必有我师”。大多情况下，人们习惯于和水平类似的人沟通，以及向比自己强的人请教。但其实每个人身上都有值得我们学习的优点。谦虚的态度是进步的先决条件。

最后，总结我们的启示为：不被传统的测试方法束缚；从产品用到的技术中找到突破；持续改进的思想；不断尝试的理念。

游戏中的自动化测试

摘要：案例分享，策划配置检查，Dump 自动分析

关键词：测试、自动化测试、测试开发、测试平台

一、引言

游戏软件和传统软件都是电脑软件，它们有很多共同点，也有很多不一样的地方。相对来说，游戏要比传统软件复杂很多，游戏涉及场景、美术、动画、音效、故事背景、数值、历程等复杂的元素，而这些都是一般软件产品所不具备的，从而使得游戏测试要比传统软件的测试复杂很多，但测试的思想在不同的软件领域都是相通的，本案例分享的方法在传统软件领域同样适用。

二、案例分享

（一）策划配置检查

1. 案例解读

策划工程师在一款游戏的制作过程中起着举足轻重的作用，策划的好坏直接决定着这款游戏的成败，策划数值的正确与否也决定着游戏功能是否正确。这些数值通常以 tab 文件（文本文件）的形式保存在游戏中，每一个文件都有数十列、成百上千行，由于文件中的数据庞大，在编辑这些文件的时候经常会因为疏忽而引入一些 Bug，而这些 Bug 往往在产品完成构建之后的功能测试阶段才能被发现。发现 Bug 之后策划人员不得不重新去修改配置文件，重新集成构建，再执行回归测试以确认 Bug 是否被修复，这样一来，因为一个小小的 Bug 不得不重新执行整个流程，降低了工作效率，耗费了资源，增加了成本（如图 1-6 所示）。

我们需要将测试过程提前，希望能够在产品构建之前就发现配置文



作者姓名：白银祖

作者简介：高级质量经理。金山软件西山居游戏高级质量经理，曾先后在 WPS、金山毒霸、西山居等部门从事自动化测试工具的研究与开发、测试案例的设计与实现、新技术的研究与引入等工作，参与并开发了多套自动化系统，有着丰富的自动化测试经验。

团队规模：700 人

研发团队职能定位：

质量中心

件中所存在的 Bug，在第一时间就将其消灭，避免走重复流程，以提高工作效率。因此对于这些配置文件的检查就成了游戏测试中至关重要的一环。

以往这部分检查工作是由测试人员手工完成的，由于游戏的特殊性，这些配置文件通常都数量庞大、内容繁杂，表与表之间还经常存在相互关联的情况，测试人员往往需要耗费大量的时间和精力来完成手工检查的工作，检查过程也十分枯燥乏味。同时，这些检查对测试人员本身的资历和工作时的状态有比较高的要求，经验不足或者是当时工作状态不佳的测试人员可能会漏掉某些 Bug。

想要解决这些问题，最好的方式就是自动化。希望有个自动化的程序可以帮助我们自动检查这些配置文件，当发现配置文件有更改时立即开始执行相应的测试用例，并将测试结果发送给相关的开发人员、策划和测试人员，直接告诉他们 Bug 出现的地方和出错的原因（即图 1-7 中的 Runner）。这样一来，配置文件的检查工作不再依赖于产品的构建过程就能独立执行，更重要的是，自动化检查将测试人员从枯燥乏味的重复性工作中解放了出来，节省了人力，提高了工作效率，提高了检查的准确性。

图 1-7 中的 Report Mail 包含了详细的执行报告和错误日志，收到邮件的相关人员只需要查看错误日志就可以定位并找出导致 Bug 的原因，直接去修复问题即可，修复完成程序又会自动回归测试。

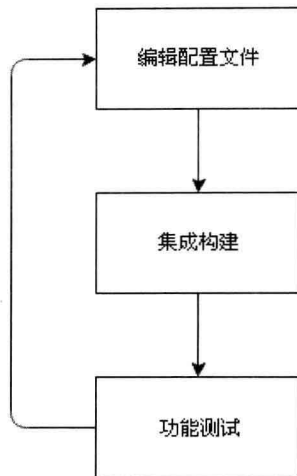


图 1-6

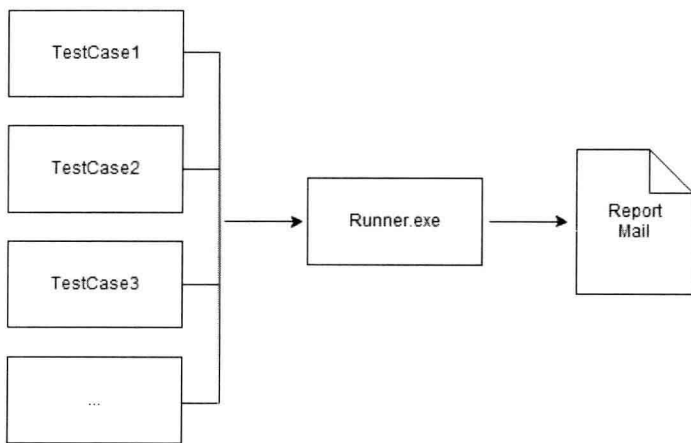


图 1-7

2. 案例成功要点

基于上文所提到的目标，我们最终实现了一套简单的检查框架，在这套框架的基础上可以很