

信息科学与技术丛书

黄文高 编著

QTP 自动化测试 与框架模型设计

- 
- 深入剖析 QTP 每个功能的“内幕”，入门、提高不再困难
 - 重点介绍当前行业最流行的自动化框架模型——混合测试框架模型
 - 将书中的框架模型与实际工作相结合，与笔者一起“悟道”
 - 测试吧工作室 www.testingba.com 倾情打造



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书

QTP 自动化测试与框架模型设计

黄文高 编著



机械工业出版社

本书分基础篇、提高篇和框架篇三大部分：基础篇介绍了软件自动化测试概述和自动化测试工具 QuickTest 的使用；提高篇详细介绍了自动化测试中其他技术的使用；框架篇介绍了关键驱动框架，重点介绍了混合测试框架模型，并通过具体实例介绍了如何使用自动化测试工具 QuickTest 进行自动化测试。

本书适用于想了解自动化测试的读者和 QuickTest 的初学者，也希望书中框架和实践部分的内容对中高级自动化测试工程师有所帮助。

图书在版编目（CIP）数据

QTP 自动化测试与框架模型设计/黄文高编著. —北京：机械工业出版社，2011.6

（信息科学与技术丛书）

ISBN 978-7-111-35003-3

I. ①Q… II. ①黄… III. ①软件工具-测试 IV. ①TP311.56

中国版本图书馆 CIP 数据核字（2011）第 108480 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：郝建伟

责任印制：乔 宇

三河市国英印务有限公司印刷

2011 年 7 月第 1 版·第 1 次印刷

184mm×260mm·20.25 印张·498 千字

0001-3500 册

标准书号：ISBN 978-7-111-35003-3

ISBN 978-7-89433-000-0（光盘）

定价：49.00 元（含 1 CD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

社服务中心：（010）88361066

门户网：<http://www.cmpbook.com>

销 售 一 部：（010）68326294

教材网：<http://www.cmpedu.com>

销 售 二 部：（010）88379649

读者购书热线：（010）88379203

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展，人类每时每刻都会面对层出不穷的新技术和新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术不断地受到挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着Internet 延伸到人类世界的方方面面，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络 and 工程应用等内容，注重理论与实践的结合，内容实用、层次分明、语言流畅，是信息科学与技术领域专业人员不可或缺的参考书。

目前，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前言

写作背景

有资料显示，目前软件外包业务的承揽企业主要来自印度、中国和东欧等国家和地区，尤其印度企业拥有全球软件外包市场 40% 以上的份额，而中国企业的市场占有率却还不足 5%。但是，这种状况正在改变，一些印度公司受到金融危机冲击，财务状况出现问题，同时，其他国家公司担心印度存在资金不安全问题，因此逐渐开始把外包向中国市场转移。

这给了中国的 IT 外包承揽企业一个极大的机会。在以服务业转移为标志的新一轮全球产业转移浪潮中，中国正逐步成为以信息技术、业务流程和知识处理为新重点的全球现代信息服务外包转移的重要目的地之一。2009 年，国务院确定了大连、北京、天津、上海、重庆等 20 个城市为中国服务外包的示范城市，深入开展承接国际服务外包的业务，并对这 20 个示范城市实行鼓励措施。

目前，在国内软件自动化测试尚处于起步阶段，绝大多数的 IT 企业并没有自动化测试工程师，但随着中国软件测试外包行业拥有全球外包市场份额的不断增多，自动化测试在中国将越来越受重视，软件自动化测试行业正在悄然兴起。可以预见，在未来中国的软件测试行业中，软件自动化测试将占据半壁江山。

写作目的

随着软件自动化测试在中国软件测试行业的不断发展，手工测试工程师受到越来越大的挑战，未来软件测试工程师必须了解自动化测试才能更好地适应行业发展，而谈到自动化测试就不能不谈 QuickTest Professional。

不论是自动化测试，还是自动化测试工具的杰出代表——QuickTest Professional，在自动化测试尚未完全遍及的今天，许多朋友并没有真正掌握。很多朋友虽然在测试过程中使用 QuickTest 进行自动化测试，但测试结果并不令人满意。大多数朋友只停留在使用 QuickTest 进行简单的测试，更多只是进行录制回放这种简单的测试过程，而无法很好地使用自动化测试框架，更无法开发自动化测试框架。

不会开发自动化测试框架，就相当于“武功”被废了 80%，自然无法很好地完成自动化测试工作。笔者希望通过本书来引导读者进行自动化测试框架设计，而且欢迎读者使用本书的自动化测试框架，更希望读者在学习 QuickTest 的过程中“悟道”，通过对本书的学习，能够开发出适合当前工作需要的自动化测试框架模型。这也是笔者的初衷。

章节构成

书中使用了大量的源代码，并对测试过程中需要注意的地方进行了详细的注释，目的是通过源代码来介绍函数和方法，希望读者在学习过程中少走弯路。书中的一些实例或许正好与读者所面临的问题存在相似之处，但笔者不希望读者依葫芦画瓢地去解决问题，而是学会 IV

动脑思考这些问题，并对解决方案进行改进，也只有这样才能更快地提高自己的水平。

本书分为基础篇、提高篇和框架篇三大部分。

基础篇包括 11 章内容，介绍软件自动化测试概述和自动化测试工具 QuickTest 的使用。

第 1 章介绍自动化测试的相关基础知识。

第 2 章介绍脚本的录制及回放。

第 3 章介绍脚本录制完成后如何对脚本进行编辑和增强。

第 4 章介绍脚本的调试方法。

第 5 章介绍对象库的常用操作。

第 6 章介绍常用 Windows 控件对象的测试。

第 7 章介绍常用 VB 控件对象的测试。

第 8 章介绍常用 Web 控件对象的测试。

第 9 章介绍 .NET 插件的管理和扩展功能，以及 .NET 插件对象。

第 10 章介绍同步等待的原理和方法。

第 11 章介绍 QuickTest 中常用的内部对象。

提高篇包括 7 章内容，详细介绍自动化测试中其他技术的使用，以及一些常用的自动化测试开发技巧。

第 12 章介绍 Windows 脚本宿主技术在自动化测试中的使用。

第 13 章介绍 HTML DOM 技术在自动化测试中的使用。

第 14 章介绍正则表达式在自动化测试中的使用。

第 15 章介绍 XML 在自动化测试中的使用。

第 16 章介绍在 QuickTest 中如何使用 Excel 对象。

第 17 章介绍在 QuickTest 中如何使用 Word 对象。

第 18 章介绍在 QuickTest 中如何读取和查找 PDF 文件中的内容。

框架篇包括 3 章内容，介绍关键驱动框架，重点介绍混合测试框架模型，并通过具体实例介绍如何使用自动化测试工具 QuickTest 进行自动化测试。

第 19 章介绍自动化测试框架的发展，并提供一个简单的关键字驱动框架供读者参考。

第 20 章对混合测试框架模型进行详细的介绍。

第 21 章通过实例介绍混合测试框架模型在自动化测试中的应用。

书中很多内容是笔者的经验总结。由于笔者水平有限，书中错误和不妥之处在所难免，欢迎读者批评指正。读者在阅读本书的过程中如有任何问题或批评建议，可以到测试吧工作室论坛（www.testingba.com）中寻求解答，作者将尽力答疑解惑。

编著者

目 录

出版说明

前言

第一部分 基 础 篇

第 1 章 自动化测试概述	2	2.3.6 Web 选项卡	22
1.1 自动化测试简介	2	2.3.7 Recovery 选项卡	23
1.1.1 自动化测试的目的与范围	2	2.4 场景恢复设置	23
1.1.2 自动化测试需要达到的程度	2	2.4.1 场景恢复的作用	23
1.1.3 自动化测试框架	3	2.4.2 Recovery Scenario Manager 设置	24
1.2 自动化测试的优点	3	2.5 脚本回放	28
1.3 自动化测试的缺点	4	2.6 本章小结	29
1.4 自动化测试时间选择	4	第 3 章 脚本编辑	30
1.4.1 自动化测试时间是第一要素	4	3.1 视图模式	30
1.4.2 一个极端的例子	5	3.1.1 关键字视图模式	30
1.4.3 一个定量的例子	5	3.1.2 专家视图模式	31
1.5 自动化测试普遍存在的问题	5	3.2 参数化	32
1.6 QuickTest 自动化测试工具介绍	6	3.2.1 DataTable 实现参数化	32
1.6.1 QuickTest 简介	6	3.2.2 环境变量实现参数化	35
1.6.2 QuickTest 的测试流程	7	3.2.3 外部数据源实现参数化	35
1.7 本章小结	8	3.3 检查点	41
第 2 章 脚本录制及回放	9	3.3.1 标准检查点	43
2.1 录制模式	9	3.3.2 文本检查点	45
2.1.1 普通录制模式	9	3.3.3 位图检查点	47
2.1.2 Analog 录制模式	10	3.3.4 数据库检查点	48
2.1.3 Low Level 录制模式	12	3.3.5 表检查点	52
2.2 Record and Run Setting 设置	13	3.4 Action 操作	54
2.2.1 Web 选项卡	13	3.4.1 新建一个 Action	55
2.2.2 Windows Applications 选项卡	14	3.4.2 以复用的方式调用外部 Action	57
2.3 Test Setting 设置	15	3.4.3 直接调用外部 Action	58
2.3.1 Properties 选项卡	15	3.4.4 分割 Action	59
2.3.2 Run 选项卡	15	3.4.5 Action 之间的参数传递	59
2.3.3 Resources 选项卡	17	3.5 添加语句	64
2.3.4 Parameters 选项卡	19	3.5.1 添加条件语句	64
2.3.5 Environment 选项卡	20		

3.5.2 添加循环语句	66	6.10 WinScrollBar 对象	106
3.6 本章小结	69	6.11 WinToolBar 对象	107
第4章 脚本调试	70	6.12 WinStatusBar 对象	108
4.1 语法检查	70	6.13 WinMenu 对象	108
4.2 断点调试	70	6.14 WinTreeView 对象	109
4.3 单步调试	71	6.15 WinCalendar 对象	111
4.4 Watch 使用	71	6.16 本章小结	112
4.5 Variables 使用	73	第7章 常用 VB 控件对象的测试	113
4.6 Command 使用	73	7.1 VbEdit 对象	113
4.7 VBSEdit 脚本编辑调试工具	76	7.2 VbEditor 对象	113
4.8 本章小结	76	7.3 VbButton 对象	114
第5章 对象库	77	7.4 VbComboBox 对象	114
5.1 对象库模型	77	7.5 VbListView 对象	115
5.2 添加新的对象	79	7.6 VbRadioButton 对象	116
5.3 导出对象库	80	7.7 VbCheckBox 对象	116
5.4 合并对象库	80	7.8 VbFrame 对象	117
5.5 共享对象库	83	7.9 VbTreeView 对象	117
5.6 对象侦察	84	7.10 本章小结	119
5.7 如何解决对象识别失败问题	85	第8章 常用 Web 控件对象的测试	120
5.8 描述性编程	94	8.1 WebButton 对象	120
5.8.1 使用描述性编程的必要性	94	8.2 WebEdit 对象	120
5.8.2 直接描述性编程	94	8.3 WebCheckBox 对象	121
5.8.3 描述性编程的后续强迫性	95	8.4 WebRadioGroup 对象	121
5.8.4 With 关键字的使用	95	8.5 WebList 对象	121
5.8.5 Description 对象的使用	96	8.6 WebElement 对象	122
5.8.6 ChildObjects 函数的使用	96	8.7 WebTable 对象	123
5.8.7 多个浏览器一起工作	97	8.8 本章小结	126
5.9 本章小结	100	第9章 .NET 插件测试	127
第6章 常用 Windows 控件对象的测试	101	9.1 .NET 插件介绍	127
6.1 Static 对象	101	9.1.1 .NET 插件管理	127
6.2 WinEdit 对象	102	9.1.2 .NET 插件扩展功能	127
6.3 WinButton 对象	102	9.1.3 .NET Windows Forms Spy	129
6.4 WinCheckBox 对象	102	9.2 SwfEdit 对象	130
6.5 WinRadioButton 对象	102	9.3 SwfButton 对象	131
6.6 WinComboBox 对象	103	9.4 SwfCheckBox 对象	131
6.7 WinList 对象	104	9.5 SwfRadioButton 对象	131
6.8 WinListView 对象	104	9.6 SwfList 对象	131
6.9 WinTab 对象	106	9.7 SwfComboBox 对象	133
		9.8 SwfTab 对象	134

9.9 SwfTable 对象	135	10.4 AJAX 同步点问题	147
9.10 SwfTreeView 对象	136	10.5 本章小结	157
9.11 SwfScrollBar 对象	137	第 11 章 Utility 对象	158
9.12 SwfCalendar 对象	138	11.1 SystemUtil 对象	158
9.13 本章小结	139	11.2 RandomNumber 对象	163
第 10 章 同步点技术	140	11.3 Reporter 对象	163
10.1 同步点原理	140	11.4 PathFinder 对象	167
10.1.1 默认同步时间	140	11.5 RegisterUserFunc 对象	169
10.1.2 手工插入同步点	141	11.6 Setting 对象	171
10.2 同步点方法	143	11.7 Crypt 对象	177
10.2.1 Sync 方法	143	11.8 OptionalStep 对象	178
10.2.2 WaitProperty 方法	143	11.9 Environment 对象	179
10.2.3 Wait 方法	145	11.10 XMLUtil 对象	180
10.2.4 Exist 方法	146	11.11 本章小结	181
10.3 脚本回放模式	146		

第二部分 提高篇

第 12 章 Windows 脚本宿主 (WSH) 技术	184	13.2 HTML DOM 对象	215
12.1 WSH 介绍	184	13.2.1 HTML 根节点	215
12.1.1 WSH	184	13.2.2 HTML 文档对象	217
12.1.2 WSH 的作用	184	13.2.3 HTML 元素集	218
12.1.3 WSH 的工作流程	185	13.2.4 获得页面元素	219
12.2 WSH 对象分析	185	13.3 HTML DOM 对象在 QuickTest 中的使用	220
12.2.1 WSH 对象模型	186	13.3.1 设置单选按钮	220
12.2.2 WSH 对象与相关任务	186	13.3.2 选中复选框	220
12.3 WSH 对象在 QuickTest 中的 使用	187	13.3.3 设置文本框的内容	221
12.3.1 WshArguments 对象在 QuickTest 中的使用	187	13.3.4 获取页面表格的内容	221
12.3.2 WshController 对象在 QuickTest 中的使用	188	13.3.5 判断页面中元素显示的情况	221
12.3.3 WshNetwork 对象在 QuickTest 中的使用	192	13.4 本章小结	222
12.3.4 WshShell 对象在 QuickTest 中的 使用	199	第 14 章 正则表达式的使用	223
12.4 本章小结	214	14.1 正则表达式介绍	223
第 13 章 HTML DOM 技术	215	14.1.1 正则表达式的发展	223
13.1 HTML DOM 介绍	215	14.1.2 正则表达式的定义	223
		14.2 RegExp 属性	225
		14.3 RegExp 对象	226
		14.3.1 Execute 方法	227
		14.3.2 Replace 方法	227
		14.3.3 Test 方法	227

14.3.4 Match 对象	228	17.3.2 插入图片	252
14.4 RegExp 在 QuickTest 中的 使用	230	17.3.3 检查拼写错误	253
14.4.1 使用正则表达式选择 下拉列表框的值	230	17.3.4 检查对象属性拼写及语法 错误	253
14.4.2 使用正则表达式匹配 列表中的值	231	17.4 本章小结	254
14.4.3 使用正则表达式删除 字符串两侧的空格	231	第 18 章 PDF 的使用	255
14.4.4 正则表达式在 DOM 中的 使用	232	18.1 PDF 介绍	255
14.5 本章小结	233	18.2 AcroExch. App 对象	256
第 15 章 XML 的使用	234	18.2.1 AcroExch. App. CloseAllDocs 方法	256
15.1 XML 介绍	234	18.2.2 AcroExch. App. GetAVDoc 方法	256
15.2 XML 读写操作	234	18.2.3 AcroExch. App. GetNumAVDocs 方法	257
15.3 比较两个 XML 文件	237	18.2.4 AcroExch. App. MenuItemExecute 方法	257
15.4 将 XML 中的内容导出到 Excel 表中	239	18.2.5 AcroExch. App. ToolButtonIsEnabled 方法	258
15.5 本章小结	240	18.3 AcroExch. AVDoc 对象	259
第 16 章 Excel 的使用	241	18.3.1 AcroExch. AVDoc. Close 方法	259
16.1 创建 Excel 文件	241	18.3.2 AcroExch. AVDoc. FindText 方法	259
16.2 读写 Excel 文件	242	18.3.3 AcroExch. AVDoc. GetFrame 方法	260
16.3 其他相关的操作函数	243	18.4 AcroExch. AVPageView 对象	261
16.3.1 获取单元格的行数	243	18.5 AcroExch. PDDoc 对象	262
16.3.2 重写保存文件的方法	243	18.6 AcroExch. PDPage 对象	263
16.3.3 新增 sheet 方法	244	18.6.1 AcroExch. PDPage. CreatePageHilite 方法	263
16.3.4 对 sheet 进行重命名	245	18.6.2 AcroExch. PDPage. SetRotate 方法	264
16.3.5 移除 sheet	246	18.7 AcroExch. PDBookmark 对象	265
16.3.6 比较两个 sheet 的内容	246	18.8 AcroExch. PDTextSelect 对象	266
16.4 本章小结	248	18.9 本章小结	268
第 17 章 Word 的使用	249		
17.1 创建 Word 文档	249		
17.2 读写 Word 文档	250		
17.3 其他相关的操作函数	251		
17.3.1 添加表格	251		

第三部分 框 架 篇

第 19 章 自动化测试框架介绍	270	19.1.1 基于界面的软件自动化测试 框架的发展	270
19.1 自动化测试框架的发展	270		

19.1.2 自动化测试脚本类型的发展	271	21.4 测试方案及计划	292
19.2 关键字驱动测试	271	21.4.1 人力资源	293
19.3 本章小结	275	21.4.2 时间进度	293
第20章 混合测试框架模型	276	21.4.3 测试环境	293
20.1 框架模型	276	21.4.4 自动化测试模块选择及划分	294
20.2 文件结构	277	21.4.5 测试策略	294
20.3 数据交互层	280	21.4.6 测试数据准备	295
20.4 引擎驱动层	282	21.4.7 风险分析	295
20.5 输出层	283	21.5 测试用例	295
20.6 本章小结	286	21.6 脚本开发	300
第21章 混合测试框架模型在自动化测试中的使用	287	21.6.1 自动化测试框架开发	300
21.1 系统介绍	287	21.6.2 功能脚本的开发	301
21.2 自动化测试可行性分析	287	21.6.3 冒烟测试脚本开发	309
21.2.1 现在的测试流程	288	21.7 执行测试	312
21.2.2 手工测试工作量估算	288	21.7.1 执行冒烟测试	312
21.2.3 引入自动化测试后工作量估算	289	21.7.2 执行功能验证测试	312
21.2.4 收益	289	21.8 提交测试报告	313
21.2.5 风险	289	21.8.1 提交自动化测试用例执行表	313
21.3 自动化测试需求分析	289	21.8.2 提交自动化测试报告	313
		21.9 本章小结	313

第一部分

基 础 篇

基础篇主要介绍两部分内容：软件自动化测试概述和自动化测试工具 QuickTest 的使用。目的是通过基础篇的学习，使读者对软件自动化测试有一定的了解，并能熟练使用自动化测试工具 QuickTest。其中，第 1 章是自动化测试概述；第 2~5 章介绍 QuickTest 的使用以及如何录制一份测试脚本；第 6~9 章主要介绍不同种类的控件对象在 QuickTest 中的使用及控件的对象和属性；第 10 和 11 章主要介绍常用的增强测试脚本的技术——同步点技术和 QuickTest 中常用的内部对象。

第1章 自动化测试概述

本章为自动化测试的概述，主要介绍自动化测试的定义、自动化测试时间选择以及自动化测试工具的杰出代表 QuickTest。

本章主要包括以下内容：

- 什么是自动化测试。
- 自动化测试的优点。
- 自动化测试的缺点。
- 自动化测试时间选择。
- QuickTest 自动化测试工具介绍。

1.1 自动化测试简介

自动化测试是使用软件来控制测试执行过程，比较实际结果和预期结果是否相一致，设置测试的前置条件和其他测试控制条件并输出测试报告。通常，自动化测试需要在适当的时间使已经形式化的手工测试过程自动化。

1.1.1 自动化测试的目的与范围

从自动化测试小组的角度来讲，自动化测试的目的是开发一套能够支持自动化测试的工具。自动化测试小组不仅要负责设计并实现数据驱动自动化测试框架，而且还要设计和构建用于回归测试的自动化测试套件。从企业的角度来讲，自动化测试的目的是提高测试效率，减少手工测试的工作量，进而达到节省测试成本的目的。

在自动化测试过程中，为了支持自动化测试脚本的开发以及与测试有关的维护活动，必须对自动化测试框架进行专门的部署。自动化测试框架必须支持单元测试、集成测试、系统测试以及回归测试。自动化测试工作的重心应放在某个特定领域的部署上。

所选择的部署方法应该能够覆盖自动化测试的所有工作。手工测试活动可以作为自动化测试的先导。手工测试的目的是使用手工测试的方法测试应用程序的所有特征。在手工测试过程中需要开发一些测试条件和测试数据，这些测试条件和测试数据可以通过回归测试的自动化测试框架来实现。

1.1.2 自动化测试需要达到的程度

自动化测试需要达到什么程度？这个问题在自动化测试工具发展的最初阶段就有人提出过。

首先，我们必须了解自动化测试工具的测试过程和被测试系统的测试过程（被测试系统的测

试过程是指手工测试时的测试过程)。测试工具与测试过程是不同的,工具是用于促进测试过程的,工具能被用于实现一个过程并执行测试过程的各种规范。在很多情况下,工具自带的内建程序可以被理解为过程,但它们往往是不完整的,不能正确地反映过程。最好的自动化测试工具是能够将工具与测试需求达成一致,并且提供高度可自定义的工作流程和跟踪报告能力。

其次,必须了解测试过程所涉及的几个环节:测试计划、测试设计、测试构建、测试执行、测试结果的捕获和分析、测试结果的验证和给出测试报告。在整个测试过程中这些活动都是密不可分的,只有将这些测试环节与自动化测试过程结合起来,才能更好地确定自动化测试需要达到的程度。

最后,所有领域的自动化测试都应该保证时间和成本适应于你的组织。实现的自动化程度越高,测试过程就越好、越有效,但同样其所带来的时间就越长、成本就越高。项目的进度与成本影响着自动化测试程度的高低。

1.1.3 自动化测试框架

自动化测试框架是假设、约束以及为自动化测试提供支持的工具的集合。自动化测试框架的最大优点是可以减少测试脚本实现和维护的成本。也就是说,如果测试用例更改了,则只需要修改测试用例文件,而不需要去更新脚本驱动程序和引擎驱动程序。自动化测试框架的优劣直接影响到自动化测试的成功与否。

假设是形成自动化测试策略的基础,下面几点是常用的假设条件。

- 集成工具套件必须是主要的测试管理、计划、开发和实现的工具。
- 工具套件必须用来指导和控制测试的执行并且用来捕获、分析、报告测试结果。
- 工具套件必须包括一个可选工具用于缺陷跟踪及解决。
- 工具套件必须包括一个可选的配置管理工具。
- 配置管理只能对手工测试和自动化测试产物进行配置管理。
- 所有上述工具必须与桌面工具(如 Microsoft Office)结合。
- 测试工程师需要的桌面一脚本一开发配置必须被定义并且被实现。
- 必须遵循测试标准,并且将测试标准以文档形式记录下来。

约束条件影响着自动化测试是否成功,如果不注意以下约束条件,自动化测试工作将很难成功。

- 自动化工具集资源必须独立于任何手工测试集。
- 自动化测试小组中是否有足够多的工作人员。
- 对于自动化工具的使用,软件开发小组的协调水平和管理水平不能太低。
- 在创建可测试应用中,需要与开发者协作和信息交流。
- 自动化测试的主要版本与自动化测试的发布进度安排有关。

1.2 自动化测试的优点

自动化测试在最近几年之所以发展迅猛,是因为其有着下列手工测试所无法比拟的优点。

- 快速:脚本执行的速度远远快于手工测试执行速度。
- 可靠性:每次运行时都执行相同的操作,消除人为的错误。

- 可重复：可以对被测系统执行相同的操作。
- 可重用：可以使用测试脚本重复的测试应用程序的不同版本。
- 全面性：可以设计更多的测试用例进而提高每个功能的覆盖率。
- 高效率：测试人员可以更专注于验证新的功能或新修改的功能，而不需要花更多的时间验证以前测试版本中已经验证过的功能或模块。
- 无疲劳：随着测试时间的增长，所有的测试动作在每个版本间不停地重复，测试工程师心理越发疲劳，而自动化测试则没有这方面的问题。

1.3 自动化测试的缺点

自动化测试有着手工测试无法比拟的优点，但同时也具有诸多缺点。

- 自动化测试永远无法取代手工测试。
- 自动化测试发现的错误比手工测试发现的错误（Bug）少得多。
- 自动化测试对软件质量的依赖性太强。
- 自动化测试不能提高有效性。
- 测试工具并不像人一样具有想象力。

1.4 自动化测试时间选择

在制订自动化测试计划时常常会遇到这样一个问题，什么时候开始自动化测试？首先，在自动化测试开始之前必须先开发自动化测试框架，而开发自动化测试框架既费钱又费时，这个框架的创建和实现必须在所测试的应用程序（AUT）交付给品质控制员（QC）之前完成，因此在该项目的生命周期早期阶段就需要构建并测试。然而，一旦使用了该框架，你可能会发现，它并不适合所有的软件测试项目。这些问题都会影响到自动化测试的开始时间。

1.4.1 自动化测试时间是第一要素

在前期，建立自动化测试项目比建立手工测试项目花费的时间要多很多。自动化测试在开始阶段并没有捷径，如果要测试一个功能，在编写脚本的时候必须执行所有手工测试项目要执行的相同步骤。

自动化测试应在所有测试用例需求已经完全定义好，并在 AUT 的一个构建版本可以交付测试后进行，但这只是乐观的想法，因为在前期的测试版本中无法保证应用程序特征为 100% 有效且不发生变更。

此外，自动化测试要求被测试的应用程序特征必须有效，如果需求或应用程序特征不能 100% 有效，就不能根据它来创建有效的、完整的自动化测试。自动化测试的基本目的是一旦确定需求后，验证后继构建版本或 AUT 的修改版本是否正确实现了需求。因此需求何时确定会影响自动化测试的实施。

如果一个项目时间进度很紧迫，请不要考虑使用自动化测试，除非专门为该项目分配了充裕的时间。

上面这些因素决定了时间是自动化测试成功与否的第一要素。

1.4.2 一个极端的例子

首先,分析需要交付的AUT构建版本的数目和复杂度。如果只是一个普通规模和复杂度的构建版本,那么进行自动化测试可能并没有什么优势。在实际过程中,如果只是在维护期重用一次(当然这种情况很少见),这种情况进行自动化测试根本就不合理,即使项目本身复杂度较高,如果重建版本的数目和测试的交付物只限于修补,也不值得花时间去进行自动化测试。

如果该项目将重复地被交付测试,而新的特征集将在多个测试间隔交付且特征集很复杂,那么自动化测试可能会有很大的益处。大多数人的看法是,自动化测试要比手工执行测试多花费3~4倍的时间。如果项目早期计划超过3~4个重要的可测试构建版本,那么项目可以选择自动化测试。

但在实际测试过程中,并没有精确的公式能决定何时应该或不应该执行自动化测试。以下几点仅供参考使用。

- AUT中的特征集的复杂度。
 - AUT中的特征集在开发过程中是否变化很大。
 - AUT中的特征集是否按需求所述工作。
 - AUT中的特征集是否需要大量的数据组合来确认所有业务规则。
 - 自动化测试工具是否能正确地识别AUT特征集中的属性。
- 这些问题很复杂且很难判断,但下列实践准则可以用来决定是否进行自动化测试。

- 如果AUT中的特征集的复杂度不大,不要采用自动化测试。
- 如果项目计划中只测试几个(3个或更少)构建版本,不要采用自动化测试。
- 如果特征不是100%有效,不管该AUT的规模或复杂度如何,不要对它进行自动化测试。
- 如果项目开发时间很紧,每次交付构建版本的时间间隔很短,这样就没有时间进行自动化测试。
- 如果一个特征不能通过自动化测试达到100%准确测试,不要进行自动化测试,除非它能节省大量的手工测试时间。

1.4.3 一个定量的例子

如果AUT的一个特征集要花费6h进行手工测试,那么使用自动化测试工具记录测试最少要花费6h(很可能更多,因为需要在手工测试和测试数据/测试脚本创建之间互换)。大多数情况是要另外花费12h来组织和建立自动化测试脚本。这样,这个特征集在自动化测试前期要花费18h,是手工测试时间的3倍。测试脚本执行时间大概为手工测试的1/10,大概为36min,这样在第4次自动化测试时,大概节省了5.5h,即从第4次自动化测试开始节省成本。

1.5 自动化测试普遍存在的问题

在引入自动化测试后,很多人发现自动化测试并没有达到人们想象中的作用,其主要原因

因有以下几点（这些是自动化测试过程中普遍存在的问题）。

（1）观念不正确，期望过高

对软件自动化测试过于乐观，对其期望过高，认为自动化测试能够代替手工测试，能够发现系统中大量的缺陷，不愿意花大量的时间做前期脚本的开发和自动化测试框架的开发，导致当自动化测试执行完成后，发现自动化测试并没有想象中那么完美，并不能帮助解决目前遇到的所有问题，甚至也并没有发现几个问题。

（2）缺乏具有良好素质和经验的优秀测试工程师

千里马需要伯乐，好的测试工具也需要优秀的测试工程师来使用。因为测试工具本身并没有想象力，而必须由测试工程师将测试计划和测试流程加载进去。只有将工具和人完美地结合起来，才能发挥其更大的作用。因此，要求测试工程师不仅要熟悉产品的特性和应用领域、熟悉测试流程，还要掌握测试技术和编程技术。

（3）脚本质量影响测试质量

在自动化测试脚本开发的过程中，并不会对脚本进行全面的测试，更多的是依赖测试工程师的经验，这样就无法保证脚本的质量。因此，当无法提供一种机制来保证脚本质量时，脚本将直接影响测试结果的正确性。

（4）没有对测试工程师进行充分的培训

在自动化测试开始前，需要就自动化测试工具对相关的测试工程师进行充分的培训；如果没有对工程师进行充分的培训，测试工程师将无法更深层次地去了解工具。这样，必然导致测试工程师对工具的使用效率低下，不能充分地发挥测试工具的作用。对工程师的培训不是上一两次培训课所能解决的，而应该对他们进行长期、系统的培训。

（5）盲目地引进测试工具

我们都清楚，不同的测试工具有着自身的特点和适用范围，并不是一个优秀的测试工具能适用于所有公司或所有项目的需求。在引入测试工具前，一定要认真分析该工具是否能解决公司的实际问题，否则工具引进反而成了摆设。例如，在整个开发过程中需求和用户界面变动较大，这种情况就不适合引入自动化测试工具，引入自动化测试工具无法提高测试效率。

（6）没有良好的使用测试工具的环境

建立良好的测试工具应用环境，需要对测试流程和管理机制进行适应性的调整，也只有这样，测试工具才能真正发挥其作用。

（7）其他方面

自动化测试中维护测试脚本的工作量比较大。在脚本开发过程中，一定要遵守一些相关的编码规范。这样，才能提高脚本的可重用性，也可以节省脚本的维护成本，提高工作效率。

1.6 QuickTest 自动化测试工具介绍

本节主要介绍自动化测试工具 QuickTest，并对其工作流程进行详细的介绍。

1.6.1 QuickTest 简介

Mercury QuickTest Professional 是企业级自动化测试工具，目前已经被惠普公司收购，正