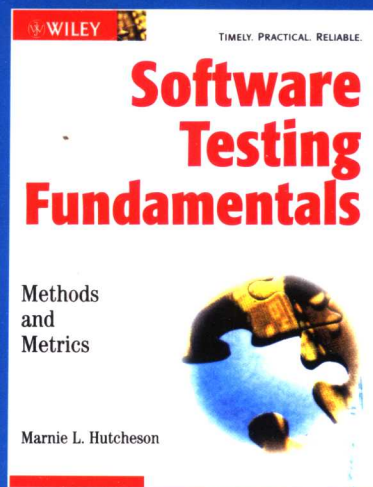


From Programmer, For Programmer

- 来自资深专家的软件测试者生存指南
- 全面论述基于风险测试的实用方法
- 实际开发项目的经典范例使您受益匪浅



软件测试 基础：方法与度量

[美] Marnie L. Hutcheson 著
包晓露 王小娟 贾有良 申来安 译

软件测试 基础：方法与度量

[美] Maric L. Richey 著
包晓露 王小娟 贾有良 申来安 译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

软件测试基础: 方法与度量 / (美) 哈奇森 (Hutcheson, M.L.) 著; 包晓露等译.

—北京: 人民邮电出版社, 2007.9

ISBN 978-7-115-16352-3

I. 软... II. ①哈...②包... III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2007) 第 082126 号

版 权 声 明

Marnie L. Hutcheson: Software Testing Fundamentals Methods and Metrics

ASP.NET 2.0 instant result

Copyright©2003 by John Wiley & Sons, Ltd.

All right reserved. This translation published under license.

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书中文简体字版由 John Wiley & Sons 公司授权人民邮电出版社出版, 专有出版权属于人民邮电出版社。

软件测试基础: 方法与度量

- ◆ 著 [美] Marnie L. Hutcheson
译 包晓露 王小娟 贾有良 申来安
责任编辑 俞 彬
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16
印张: 19 25
字数: 360 千字 2007 年 9 月第 1 版
印数: 1 - 4 000 册 2007 年 9 月北京第 1 次印刷

著作权合同登记号 图字: 01-2003-2750 号

ISBN 978-7-115-16352-3/TP

定价: 45.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

内容提要

本书通过具体的开发案例，详细地介绍了如何应用一种有效的软件测试方法——最重要测试（MIT）方法展开软件测试，并提供了相应的测试度量及评估模板，用于指导测试工作。此外，书中还详细地介绍了建立测试说明的步骤、将测试说明自动化的工具以及风险分析技术等。

本书第 1 章～第 5 章主要介绍了背景知识和基本概念。第 6 章～第 8 章重点讨论了测试说明以及如何建立测试说明。第 9 章～第 14 章介绍了测试评估、规划和计划的工具和分析技术。

本书注重实践，是一本关于软件测试的优秀书籍，对软件测试工作具有很好的指导意义，实用性较强，适合于软件测试人员、开发人员、项目管理人员阅读参考。

译者序

在软件测试越来越受到重视的今天，软件测试作为重要的质量保证手段被普遍实施，人们在不断地探索科学、有效的软件测试技术和方法。然而，正如本书作者 Marnie L. Hutcheson 所提到的那样，软件测试的现状不容乐观，很多测试人员对测试术语的定义并不清楚，并且在软件测试中很少使用度量技术。作者精辟地总结了目前软件测试存在的主要问题，并提出了最重要测试（Most Important Test, MIT）方法。MIT 方法是一种基于系统故障风险辅助规划测试工作的方法，主要用于自顶向下的系统测试、集成测试和功能测试中。其核心是建立一个测试统计表，测试人员通过几种技术识别需要测试的部分，评估项目的各个组件、特性和功能的风险。这些风险被转化为优先级系统，其中最重要的部分是重点测试部分。这样可在有限的时间内获得一个合理的测试覆盖率。MIT 方法对测试工作的效果进行评估，并基于这种评估的结果调整和改进以后的测试工作。

本书通过具体的开发案例，详细地介绍了如何应用 MIT 技术展开软件测试，并提供了相应的测试度量、评估模板可用于指导测试工作。此外，书中还详细地介绍了建立测试说明的步骤、将测试说明自动化的工具以及风险分析技术等内容。本书对软件测试工作具有很好的指导意义，实用性非常强。

本书由包晓露组织翻译工作，参加翻译的有包晓露、王小娟、贾有良、申来安、朱国平、葛昱。译者力图体现原著的风格和思想，但由于时间仓促，译者水平有限，书中难免存在不妥和错误之处，恳请读者批评指正。

译者

2007 年 4 月

致 谢

当一门学科被很好地认知以后,可以用简单的几句话表达清楚,但在此之前却要经历漫长的道路。

多年以来,我从事着与本书内容相关的研究并将研究成果传播给测试人员及经理们。本书介绍的大多数方法和技术都是通俗易懂的,而好的答案不一定是难以回答的。这些方法中,有许多是老的方法并获有专利,有一些是新的方法。

本书旧版本的前 4 章在网上在线阅读已达 4 年之久。在这期间,访问的人数不断增长,目前每周大约有 350 个读者访问这些章节。来自各行业的超过 2 500 名软件测试人员和经理浏览了这一作品并提供了宝贵的反馈意见。对于所有参加讨论、咨询问题、指出错误以及报告他们的实践步骤及方法的人们,我都要再次表示感谢:“谢谢你们!希望你们继续多提宝贵意见,以便改进本书的内容。”

诚恳感谢以下人士: Minna Beissinger、Joseph Mueller、Eric Mink、Beate Kim、Joan Rothman、L. Gary Nakashian、Joy Nemittz、Adrian Craig Wheeler、Lawrence Holland、John Chernievsky、Boris Beizer、Dorothy Graham、Roger Sherman、Greg Daich 及 Tom Gillb。

特别感谢技术评论者、研究者及产品研发者 David Mayberry。如果没有他坚持不懈的努力,可能就不会有本书的问世。我还要感谢 Wiley 出版社的编辑们: Ben Ryan、Scott Amerman 和 Vincent Kunkemueller, 以及所有给予本书极大耐心和帮助的人们。最后,衷心感谢我的研究助理及艺术指导 Velvalee Boyd 和 Dane Boyd。

作者简介

Marnie L. Hutcheson 为微软公司 (Microsoft Corporation) 制作技术课件并在世界各地巡回为教授这些技术的讲师进行培训。她是软件开发、测试、质量保证及系统管理领域中的国际知名作者和讲演者。

她于 1987 年在 Prodigy Services 公司作为系统集成购置负责人开始了她的工程师生涯, 后来从事银行和金融行业。多年来, 她成为 Web 开发的先驱者并帮助了诸如 GTE 和 Microsoft 这样的公司开发和创建了几个主要的 Internet 技术。

在此之前, 她是一位从事 25 年表演艺术的学生。她在 Denver、Montreal、Boston 及 New York 的舞台上表演了 10 年的古典芭蕾、音乐剧、民乐以及歌剧。

在此期间她还在协会及学院教授舞蹈。在 20 世纪 70 年代后期, 她是美国奥林匹克溜冰教练 Don Laws 和奥林匹克舞美设计 Riki Harris 的舞蹈教练和助理舞美设计。她曾与美国奥林匹克的溜冰精英以及美国和加拿大优秀的体操运动员在一起工作过。

在现实的软件开发世界中，软件产品的开发并非总能与既定目标保持一致。“项目计划”通常就是某人在依于市场需求列出要实现的功能列表。管理人员在“项目计划”中加上开始和结束日期，时间的合理性存在很大的问题，甚至是根本不现实的。产品的设计与实现被开发人员隐藏起来。产品提交测试时经常事先无通知且滞后几周。测试人员难以制订测试计划，因为不能确定产品究竟能完成什么功能。唯一能确定的是产品必须按时（比如下周）交付。

这就是软件开发的混乱和无序，这种系统开发方式是初级的而且浪费巨大。本书提供了几种如何更好地理解、控制及如期在市场发布好产品的方法。这些方法适用于个人或开发团队，可为软件开发人员大大节省开销并提供更高质量的产品。

我经常奔波于用户对产品首次需要帮助的场所，并针对实际问题设计实际解决方案。此外，还要处理来自客户（由于产品某些方面的缺陷而蒙受经济损失）和一线技术支持人员所遇到的问题。一线支持人员常常因夹在具有某些特权的开发组和急于修复系统的客户之间而备受责备。

我与开发人员共事过，开发人员的主要工作就是编写高质量的代码。他们没有时间完成质量保证所要求的所有规范或撰写测试及技术支持组所需要的操作文档。我也与测试人员共事过，测试人员的确不清楚系统背后发生的事情。测试人员不停地中断系统，但却不能为开发人员再现开发的问题。我还与文档编写人员共事过，他们不理解为什么在测试周期结束前的两周还会改变整个用户界面。

我的任务是通过自动化和优化过程防止故障发生及提高生产率，其主要工作对象是运行在大型网络中的应用程序。这些系统非常庞大，包含许多需要测试的组件。通常有许多面向对象模块及图

形用户界面（GUI）及基于浏览器的界面。这些应用程序通常与数据库、通信网络、专用服务器和嵌入式代码以及所有需要测试的专用硬件驱动进行交互。本书中的这些方法源于所有涉及这些领域项目的经验，其中有成功的，也有失败的。

本书还是一本针对“如何解决问题”的论著，涉及许多与人有关的因素。系统的设计、编码、集成、测试、发布和支持都是由人来完成的。我们不能忽视这个事实，即人的因素实际上在所有系统故障中起着主要作用。

本书主题

本书是软件测试人员管理软件测试过程的指南。尽管书中提到了一些有效的测试技术，但并非是关于测试技术的专著。本书涉及的测试方法具有防御性，提供了如何改进测试过程的方法和度量，而不管是否采用正规的测试技术。本书侧重于如何在测试过程中使用度量，如果不知道如何使用这些度量结果改进测试用例，或者甚至这些度量结果对你的测试起负作用的话，则采用度量技术是没有意义的。本书展示了如何使用度量技术进行探索，以及如何与其他人沟通这些探索的结果，并由此实施改进的方法。

不久前，我在一次会议上介绍了这些方法的主要内容，其中包括用例研究。应用这些方法设计测试说明，并进行系统风险分析，在上述用例中，在给定的时间和资源条件下，所确定的理想测试覆盖率为整个测试说明的 67%。

在我做完介绍的讨论会上，一个颇有名气的高个子同行问道，“原谅我的冒昧，你如此瘦小，但的确令我吃惊的是，你哪来那么大的勇气告诉你的主管，你仅打算测试系统的 67%？”

我的回答是：“我虽然身高只有 5.6 英尺，但实际上我足够高大。如果管理层给我足够的时间和资源测试所有的测试项，我愿意这样做。但如果让我在时间和资源不够的情况下进行测试，我只能委婉地告诉他们不可能实现 100% 的测试覆盖。如果时间和资源不能保证测试所有的内容，我认为需要确定的是应该执行最重要的测试（most important test）。”

我还打算告诉管理层所选测试的有效程度，测试过程发现的错误数，错误的严重程度以及修正错误的开销有多大。如果可能的话，还要告知由于发现和消除错误而节省的费用是多少。我将度量测试工作的效果，并可以说明在任意时刻我们是否符合进度要求，错误率是否太高或修正错误率是否太低。如果不能满足进度要求，我们可以为管理者提供用于决策的有用信息，特别是提供在这种情况下应该如何管

理的信息。

业界和技术趋势：为什么我认为现在是出版本书的最好时机

1980 年，我在 Prodigy 工作的时候开始研究这些方法。这些方法已经发展为适合大型、复杂实时系统快速开发环境的需求。这些方法被称为最重要测试方法（Most Important Test）或 MTI。MTI 很快在 Prodigy 成为测试方法的标准。1990 年，我开始编写和出版 MTI 项目的用例研究。1993 年，我离开 Prodigy 时将 MTI 也带了出来，并随着后续其他领域越来越多测试项目的展开而不断完善。在过去 10 年的大部分时间里，我竭尽全力帮助企业从集成大型系统及 Internet 网络中受益。

1993 年，我开发的 MTI 课件（基于 PowerPoint）主要源于我原来编写的《软件测试方法与度量（Software Testing Methods and Metrics）》一书的前 7 章。当时强调的是客户/服务器测试，而不是 Internet 测试，这在原书的章节中有所介绍。

自 1993 年推出该教程后，每年都多次使用该教程教学。1997 年我将原教程的前 4 章放到了我的 Web 网站上。阅读这 4 章的人数连年增加。今年已有 17 000 个访问者下载了这些章节。最受欢迎的是第 2 章“基础方法（Fundamental Methods）”。鉴于其受欢迎程度以及众多读者的 E-mail 的鼓舞，已在此将该部分内容扩充到了第 3 章“管理软件测试的方法”和第 4 章“最重要测试（MIT）方法”中。

时代的变迁

2001 年，我花费了大量时间编写了 Microsoft 的 .NET 开发人员培训资料。在此过程中，我对 .NET 的大体内容非常熟悉。最基本的内容包括新的 .NET 结构及其统一的开发库，使得在开发语言上可以将“所有语言等同对待”，拷贝及运行应用程序以及 Web 服务使我想起了 1990 年在 Prodigy 时候所使用的方式。它们之间显著的差别是 Prodigy 小型而私有；.NET 是全球和公共的（包括私有的）。

我们真正开始注意软件开发生命周期中的这一最新技术对全球的影响可能是两年之后的事情了。的确，任何人都可以在全球范围内使用 Web 服务发布和销售软件，这种方式不用刻录 CD、编写手册或在杂志上做广告。从某种意义上说软件也将经受测试的考验。

当你考虑即将开始的 Internet 下一个浪潮——移动 Internet 时，这种情形将变得

更加有趣。PC 革命如同当今社会商业的发展一样，灵巧的电话及掌上电脑使得比以往更多的人能够通过微小的数据链路在小屏幕上访问动态应用程序。本书提及的方法也被发展到这种环境中，并且是可行的。

软件测试必须证明它的增值性，并且这是产品取得成功的必要条件。否则，市场压力将迫使竞争企业忽略测试，当他们可以将软件写入和拷贝到服务器时，就想尽快地将产品交给用户。

本书是关于基础方面的内容，幸运的是“基础”概念变化较慢，有时甚至是过时的。以前在 Prodigy 工作的例子在客户/服务器时代是过时的，在.NET 时代就更加过时了。在许多方面，为适应时代而修订版本实际上是对其原来状态的一种追溯。

本书范围及适用对象

本书是有关软件系统和应用开发的测试人员和管理人员的专业指导书。本书包括实用的解决方案，而不是一本纯理论书籍。本书介绍的理论和背景知识只是为了支持实际的解决方案。

本书是测试人员的生存指南，因为它帮助测试人员回答管理人员关注的问题。测试人员需要回答如下问题：“你为什么不能在下周完成测试？”。本书还是经理们的生存指南，因为他们必须解释事情为什么是这样的，完成任务的开销是多少以及为什么是这样的开销。

本书介绍的方法针对大型网络中的软件。通常在这些网络中运行基于 Web 应用程序和基于客户/服务器应用的组合，一些应用运行于公共的 Internet 网络中，一些应用运行于防火墙内的私有网络中。这些应用至少有一部分采用面向对象语言编写，并且所有应用均通过图形用户界面访问，作为被指定的客户端或浏览器中的 Web 页。这些测试方法也被广泛用于其他软件系统的测试，包括无线电通信、商业应用、嵌入式固件以及游戏软件。

本书描述的过程为自顶向下测试方法 (top-down approach)。这些方法可以用来进行单元测试，但用于集成测试、系统测试及端到端的测试更加有效。通常用在项目生命周期后期的负载测试、性能测试以及产生系统的监控中，也适用于自动测试或测试复用的场合。

总之，这是一本具有实效性的测试指导书籍。可以参考这些实践方法对过程进行改进，例如在测试跟踪中增加度量，或划分测试优先级以及跟踪每个任务的运行时间。

本书的组织

本书各章的组织与多数软件开发和测试工作的流程是并行的。后续章节建立在前面章节的基础上，因此跳跃式阅读也许不是上策，最好是按顺序阅读。

用例研究、有关自动化论述、测试技术、可用性问题以及人的因素等内容贯穿于本书的始终。本书分为 3 个主要部分：

- 第 1 章～第 5 章：介绍背景知识和概念。
- 第 6 章～第 8 章：重点讨论测试说明以及如何建立测试说明。
- 第 9 章～第 14 章：介绍测试评估、规划和计划的工具和分析技术。

本书中的标准定义

本书的术语定义的出处为韦伯美语新世界大辞典（Webster's New World Dictionary of the American Language）。任何一部好辞典应该是被公认的。当我引用其他著作的一个定义时，将会注明这本著作。由于篇幅所限，对于人人皆知的定义，在本书中将不再引用。

在我培训开发和测试人员时所遇到的一个主要困难就是他们对基本术语缺乏统一的认识。这一点在第 1 章讨论的测试调查和本书的附录 B 中有较详细的讨论。

缺乏统一认识的主要原因是由于业界出版了大量的标准，这些标准并非容易获得和理解。第二个原因是对这些标准定义有异议。我经常听到这样的争议：“不错，但它对此不适用”。辞典可作为解决这些问题的标准。它是大多数人获取知识的起点，因此再回到这个基本点也是必要的。在我研究的过程中，不断面对这样的事实，大多数人对下述常用术语的真正含义并不清楚，如测试、验证、确认、性能、有效性、效率、科学、艺术以及工程。

我所采用的阶梯方法与人类发展的需要相吻合，即人类在爬之前必须学会蠕动，在会走之前必须学会爬，在跑之前必须学会走。对开发中任意一个阶段的掌握程度直接依赖于对以前阶段的掌握程度。我将对读者知识掌握的程度尽可能保守地做一些假设。

在软件行业中，软件开发人员和测试人员来自于不同的方向、不同的学科以及不同的观点，因此统一认识是十分困难的。但是，我认为正是这种多样性给了我们力量。许多观点的相互影响不断地促进发明和创新。计算机工业的发展也许就是源于人类历史中的富有创造性的通力合作，这确实需要科学的管理方式。

第 1 章 软件测试现状	1
1.1 历史的回顾	3
1.1.1 20 世纪 80 年代: 蓝色巨人 (Big Blue) 和 大铁家伙 (Big Iron) 统治的时代	3
1.1.2 20 世纪 90 年代: PC 将计算带到 “每个桌面”	3
1.1.3 走进 Web 时代	5
1.1.4 目前的金融气候	6
1.1.5 此时你想到了什么? 最佳方法没有改变	8
1.2 当今测试人员所面临的挑战	8
1.2.1 没有规格说明就没有测试	9
1.2.2 市场第一: 迫于市场或企业压力而忽略测试	10
1.2.3 缺乏训练有素的测试人员	11
1.2.4 标准减少了对测试的需求	12
1.3 小结	15
第 2 章 软件测试环境中的质量保证	17
2.1 这一段场景的问题出在哪里	17
2.2 传统的质量保证存在的问题	18
2.2.1 质量的传统定义不适用了	19
2.2.2 有关质量的一些谬误	20
2.3 什么是不可行的: 控制质量的传统工具	23
2.3.1 传统 QA 和测试工具已不能满足要求	24
2.3.2 关于纸质文档的问题	24
2.4 解决方案: 改进质量过程	25
2.4.1 在你的环境中为质量选择正确的构成	25

2.4.2 正确选择适合你的开发环境的质量控制工具	26
2.5 小结	31
第 3 章 管理软件测试的方法	33
3.1 碰运气 (I-feel-lucky) 软件开发方法	33
3.2 关于艺术、科学和软件的一些说法	34
3.2.1 关于艺术的说法	35
3.2.2 关于科学的说法	35
3.2.3 关于软件的说法	36
3.3 工程方法	37
3.3.1 可审计性和履行职责	37
3.3.2 如何确定安全系数	40
3.4 集成和测试的折衷方法：艺术和工程的平衡	43
3.5 目前的测试策略	47
3.5.1 自顶向下宽度集成测试	47
3.5.2 定位测试组的组织策略	48
3.6 最佳方法	51
3.7 小结	52
第 4 章 最重要测试 (MIT) 方法	55
4.1 MIT 概述	55
4.1.1 MIT 能做什么	56
4.1.2 MIT 是如何工作的	56
4.1.3 如何用 MIT 获得成功	58
4.2 MIT 方法的步骤	59
4.3 现实世界的例子：重型、中型和轻型开发工作	64
4.3.1 项目概述	65
4.3.2 集成整个系统	71
4.4 小结	72
第 5 章 软件测试度量基础	73
5.1 度量和度量元	74
5.1.1 手工艺：艺术和工程之间的结合	74
5.1.2 可操作性定义：基础度量	74

5.2 在软件测试中如何度量	75
5.2.1 基本测试度量：它有多大	76
5.2.2 错误	78
5.2.3 度量测试工作的度量元	82
5.2.4 测试它的开销是多少	85
5.2.5 衡量效果的度量：它值得吗	85
5.3 小结	87
第 6 章 测试说明	89
6.1 目标：成功的测试工作	89
6.2 计划适当的测试：测试多少算充分	90
6.3 确定测试范围：它有多大	92
6.3.1 单元测试	93
6.3.2 一个测试脚本：多种测试类型	94
6.3.3 使用历史数据评估工作	95
6.4 使用测试说明的理由	97
6.5 测试说明中的测试来源	98
6.5.1 基于需求的测试	99
6.5.2 分析技术	99
6.5.3 非分析方法	102
6.6 小结	103
第 7 章 如何构建测试说明	105
7.1 从需求入手	105
7.1.1 结构 RAD 项目的测试说明示例	106
7.1.2 一个 Web 项目的初始测试说明	112
7.2 确定并列出测试环境	113
7.2.1 测试一个大型系统	114
7.2.2 测试多种环境	116
7.3 增加细节和步骤层次	117
7.3.1 会谈：需要什么和如何准备	118
7.3.2 执行会谈	121
7.3.3 分析结论：所得教训	123
7.3.4 准备按你所学地行动	124

7.4	小结	127
7.5	进一步的研究	128
7.5.1	产品 A	128
7.5.2	产品 B	129
第 8 章	测试说明自动化工具	131
8.1	测试说明的展开	133
8.1.1	简要过程	133
8.1.2	PowerPoint 中的测试说明	134
8.1.3	文档中的测试说明	136
8.1.4	电子表格中的测试说明	142
8.1.5	高功能网站中的测试说明	144
8.2	小结	147
第 9 章	风险分析	149
9.1	风险分析过程概述	150
9.2	风险分析的益处	151
9.2.1	确保正确的测试工作重点	151
9.2.2	公开测试人员的假设	151
9.2.3	通过其他组受益来提高重用性	152
9.2.4	为测试工作提供强有力的项目管理工具	152
9.3	项目管理策略及风险	155
9.4	MIT 风险分析	157
9.4.1	定量和定性分析	158
9.4.2	MIT 等级索引	158
9.4.3	MIT 等级准则	160
9.4.4	在辩论中使用风险等级	163
9.5	案例研究：401(k) Web 项目	163
9.5.1	应用程序	164
9.5.2	我的测试	164
9.5.3	报告我的测试结果	164
9.5.4	计算测试的价值	165
9.5.5	关于这个项目的一些想法	166
9.6	小结	166

第 10 章 风险分析应用	169
10.1 在测试说明中应用风险分析	170
10.1.1 测试评估过程	171
10.1.2 MIT 总计工作表	173
10.1.3 规划工作表	176
10.1.4 协调测试工作	181
10.1.5 不要忽视开发人员修正错误	182
10.1.6 测试合同	183
10.2 选择测试的风险准则	184
10.2.1 确认和验证系统需求	184
10.2.2 检查系统中最重要的功能	185
10.2.3 检查最重要的路径和数据	185
10.2.4 重点投资	185
10.3 小结	186
第 11 章 路径分析	187
11.1 弥诺陶洛斯迷宫的传说	187
11.1.1 线性无关	191
11.1.2 case 语句	192
11.1.3 一连串判定语句	195
11.2 计算通过一个系统的路径数	199
11.2.1 什么是逻辑流程图	200
11.2.2 用于计算路径的方程式	202
11.3 小结	205
11.4 为进一步探索而提出的路径分析的例子	206
第 12 章 路径分析应用	209
12.1 通过深度理解校正问题：这一切如何开始	209
12.1.1 显示类比的游戏	210
12.1.2 获取地图	211
12.2 分而治之	214
12.3 未经计算的路径	215
12.4 快速建立路径测试评估	215