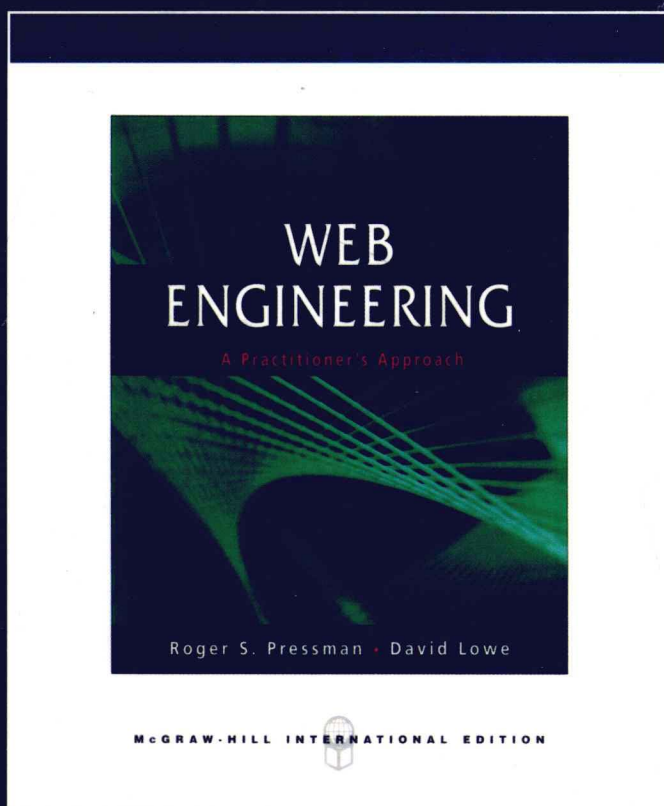


计 算 机 科 学 丛 书

Web 工程 实践者的研究方法

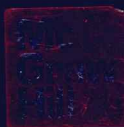
Roger S. Pressman David Lowe 著 霍秋艳 闫兵 陈静玉 等译



Web Engineering
A Practitioner's Approach



机械工业出版社
China Machine Press



TP393.09
P906-2

机 科 学

Web工程

实践者的研究方法

Roger S. Pressman David Lowe 著 霍秋艳 闫兵 陈静玉 等译

Web Engineering
A Practitioner's Approach



机械工业出版社
China Machine Press

TP393.09
P906-2

本书全面、系统地阐述了当前 Web 工程技术的各个方面,概括性地对基于 Web 的系统和应用的工程化实践过程进行了全面的介绍。主要内容包括:Web 工程过程、沟通、计划、建模活动、WebApp 分析建模、交互设计、信息设计、功能设计、构建和部署、设计模式、技术和工具、测试、变更管理和内容管理,以及 WebApp 的发展趋势。本书贯穿了一个应用实例,并以轻松诙谐的方式展现相关内容。

本书可作为计算机相关专业本科生、研究生的教材或参考书,也可供从事 Web 工程的相关人员参考。

Roger S. Pressman and David Lowe: *Web Engineering: A Practitioner's Approach* (ISBN 978-007-126377-1) .

Copyright © 2009 by The McGraw-Hill Companies, Inc.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2009-1620

图书在版编目(CIP)数据

Web 工程:实践者的研究方法/(美)普瑞斯曼(Pressman, R. S.)著;霍秋艳等译. —北京:机械工业出版社,2010.1

(计算机科学丛书)

书名原文:Web Engineering: A Practitioner's Approach

ISBN 978-7-111-28250-1

I. W… II. ①普… ②霍… III. 主页制作-程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2009)第 161259 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:周茂辉

北京市荣盛彩色印刷有限公司印刷

2010 年 1 月第 1 版第 1 次印刷

184mm×260mm·19.25 印张

标准书号:ISBN 978-7-111-28250-1

定价:49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱:hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心

译者序

作为一本 Web 工程著作，本书全面、系统地阐述了当前 Web 工程技术的各个方面，并清晰地对基于 Web 的系统和应用的工程化实践过程进行了全面介绍。本书的问世，将对这个全新的工程学科产生重要的影响。

Web 工程是一门新兴学科，综合讲述的相关书籍寥若晨星。本书的焦点是 Web 工程，告诉我们关于基于 Web 的系统和应用的工程化实践过程，强调一个敏捷过程和一些已在行业应用中被证明过的简单而实用的方法。对于基于 Web 的特定工具和技术的讲解相对较少，因为已有数以千计的强调各种特定工具和技术的书籍、文献和基于 Web 的资源，但它们几乎都没有用联系的观点来看待 Web 工程。

本书是 Web 工程学科的及时雨，是一本非常好的 Web 工程读物，认真阅读它会使你受益匪浅。自本书原版面世以来，我就是它的忠实读者，它是我从事“Web 工程”课程教学的重要参考书。在此，郑重推荐给读者。

本书的特点如下：

- 强调敏捷的过程和一些已经在行业应用中证实的简单而实用的方法。
- 各章后面提供了相应的参考文献信息。
- 全书贯穿了一个应用实例（基于 Web 的住宅安全系统——SafeHomeAssured.com），并以对话的形式逐步展开这一开发过程，使本书更具有可读性，引发读者的阅读兴趣。
- 以非正式的会话的风格，以回答问题的形式展现内容。

本书在前言中指出了其目的是提供一本可供实践者、大学生或者研究生使用的书。除此之外，教师可以把本书作为“Web 工程”课程的教材。在作为教材时，可能还需要自行补充一些材料：习题和课程实习。对于教师或者有一定 Web 应用开发经验的研究生和专业人员，本书提供了全面的、丰富的、代表当前发展水平的 Web 工程知识。

参加本书翻译工作的主要有：闫兵（第 1~10 章、第 13~14 章）、王蕾（第 11~12 章）、陈威（第 16~17 章）完成了翻译初稿。本人完成了第 15 章的翻译，并对全部译稿做了仔细的审核与大量的修改，闫兵和陈静玉对全部译稿也进行了仔细的审核。

尽管本书的翻译工作历时超过半年，但仍然感觉时间太紧，而且 Web 工程作为一门新的学科，很多概念和术语的翻译还很不成熟。限于译者水平，我们对内容的理解和中文表达难免有不当之处，在此敬请读者批评指正。

霍秋艳

2009 年 10 月

前言

在开始计划写这本书的时候，我们曾担心它会淹没在数以百计（不！是数以千计）的相关书籍中，已经有大量关于“Web 设计”、HTML、Java、XML 的技术书，以及构建成功的基于 Web 的系统和应用（WebApp）必须要理解的无数的技术书籍。奇怪的是，我们发现一个至关重要的主题（其他的技术都被应用的过程）被覆盖得相对较少。我们称这个过程为 Web 工程（Web engineering），同时我们相信应用这个技术更可能构建出满足用户需要并给他们客户的业务或者组织带来真正收益的 WebApp。

强调 WebApp 会是现实中所有的业务和组织成功的关键已经是陈词滥调。然而，很多的 WebApp 仍然是这样构建的：基本不考虑问题分析、有效的设计、可靠的测试以及变更管理的基本原则。所带来的后果就是，很多 WebApp 无法满足最终用户的需要和委托给开发者的业务目标。

现在，为了应对下一代的基于 Web 的系统和应用所提出的挑战，我们要从守旧的方法转变到 Web 工程。这个行业正向更加实效的 Web 工程过程（展示出敏捷性和适应性）的方向发展。同时，这个过程必须要传递一个学科方法的完整性。

本书面世了，它全面展示了工程化的基于 Web 的系统和应用的过程。为了在这个全新的工程学科中引导读者，其内容是以一种非正式的、会话的风格，以问答的形式进行展示。

在全书中，我们强调一个敏捷的过程和一些已经在行业应用中证明过的简单而实用的方法。同时，我们有意较少涉及基于 Web 的特定的工具和技术。这不是因为我们认为它们不重要，而是因为已经有了数以千计的强调它们的书籍、文献和网上资源，而它们几乎都没有用联系的观点来看待 Web 工程。因此，我们自然把重点放在 Web 工程上。我们的目的就是提供一本可供实践者、大学生或者研究生使用的书。

Web 工程过程强调一种敏捷方法，并展示一些简单而有效的方法，以获取和分析问题需求，设计一个有效的解决方案，然后构建和测试一个高质量的 WebApp。但是，这个过程不仅仅是关于技术的。我们还将呈现一些已经被证实的用于项目管理、变更和内容管理以及质量保证的技术。在全书中，我们用一个精心设计的案例来展开叙述我们介绍的方法和技术。Web 站点 www.SafeHomeAssured.com 对这个案例补充了一些深入的细节，同时还提供了其他额外信息。

本书受益于许多讨论构建高质量的 WebApp 的准则和技术的出版物和 Web 资源。我们对本书中引用到的原始资料的作者以及在过去的几年中对我们的想法进行指导的数以百计的同行和作者表示衷心的感谢。同时，要特别感谢 Didar Zowghi、Norazlin Yusop、Xiaoying Kong 和 Rechatrin Tongrungrrojana。

在全书中，一些引用的文本、图片以及 SafeHome 这个案例来自于 Roger Pressman 的《Software Engineering: A Practitioner's Approach》（sixth edition）^①。我们经过允许直接使用或加以改造，以满足特定 Web 工程师的需求。

我们两个作者的家庭都有 4 个成员，在此我们想对他们在这件事情上做出的努力表示感谢。

① 该书由机械工业出版社引进出版，中文书名为《软件工程：实践者的研究方法》（原书第 6 版），书号为 7-111-19400-4。——编辑注

VI

我们杰出的妻子（Barbara 和 Catherine）忍受了为出版一本书所花费的大量用于写、修订和四处奔波的时间。Roger 的儿子（Mathew 和 Michael）已经长大了，有他们自己的事情要做，并且每天都要使用因特网和 Web。David 的儿子（Oliver 和 Dominic）很年轻，都有他们自己的未来，同时他们将来的职业生涯中会有相当长的一部分时间能见证未来网络的发展。我们希望本书所展现的思想可以对他们的职业成长有所帮助。

Roger Pressman

美国佛罗里达州贝卡拉顿市

David Lowe

澳大利亚悉尼

目 录

出版者的话	
译者序	
前言	
第 1 章 基于 Web 的系统	1
1.1 网络	1
1.2 Web 应用	2
1.2.1 一个学习案例	2
1.2.2 WebApp 就是计算机软件吗	3
1.2.3 WebApp 的属性有别于传统软件的属性吗	3
1.2.4 在 WebApp 演变的过程中会涉及哪些范畴	5
1.3 WebApp——一种哲学的观点	7
参考文献	7
第 2 章 Web 工程	8
2.1 什么是 Web 工程	8
2.1.1 敏捷意味着什么	8
2.1.2 什么是 WebE 框架	9
2.1.3 适应框架时应该遵循哪些原则	10
2.1.4 旧方法有什么价值吗	11
2.2 Web 工程的组件	11
2.2.1 软件工程如何发挥作用	11
2.2.2 为什么 WebE 过程的敏捷性如此重要	12
2.2.3 过程框架内有哪些 WebE 方法	13
2.2.4 工具和技术不就是 Web 工程的一切吗	13
2.3 Web 工程的最佳实践	14
2.4 本章小结	15
参考文献	15
第 3 章 Web 工程过程	17
3.1 定义框架	17
3.2 增量过程流	19
3.2.1 框架活动是如何进行的	19
3.2.2 框架如何进行精化	21
3.3 WebE 框架的一般动作和任务	22
3.3.1 沟通活动应当如何进行精化	22
3.3.2 制定一个增量计划需要哪些任务	23
3.3.3 什么是建模	24
3.3.4 可以应用哪些分析建模任务	24
3.3.5 设计模型的要素有哪些	25
3.3.6 可以应用哪些设计建模任务	26
3.3.7 应用哪些构建任务	27
3.3.8 WebApp 增量如何进行部署	28
3.4 保护活动	28
3.4.1 WebE 团队如何管理变更	28
3.4.2 如何保证增量的质量	29
3.4.3 如何管理风险	29
3.4.4 如何管理工作	29
3.5 本章小结	30
参考文献	30
第 4 章 沟通	31
4.1 沟通活动	31
4.2 规划	32
4.2.1 应当和谁沟通	32
4.2.2 可以使用哪些技术进行沟通	32
4.2.3 会有不同的视角吗	33
4.2.4 应该问什么问题	33
4.2.5 如何促进协作	34
4.3 提取需求	35
4.3.1 在提取需求的会话之前做什么	36
4.3.2 利益相关者如何进行准备	36
4.3.3 提取会话中执行哪些任务	37
4.3.4 WebApp 的用户种类	38
4.3.5 如何识别内容和功能需求	38
4.3.6 如何将约束和性能问题分离	39
4.3.7 什么是使用场景	40
4.3.8 什么是用例	41
4.3.9 如何创建用例	41
4.4 确认 WebApp 增量	45
4.5 协商	46
4.6 本章小结	47
参考文献	48

第 5 章 计划	49	5.8 跟踪项目	72
5.1 理解范围	49	5.8.1 进度问题有什么宏观指标吗	72
5.1.1 有哪些相关的沟通工作产品	49	5.8.2 可以使用哪些标准来跟踪进度	72
5.1.2 如需要更多的细节来理解 增量会怎么样呢	50	5.9 外包 WebE 工作	73
5.1.3 你的理解中仍然存在不足 怎么办	51	5.9.1 如何启动外包项目	73
5.2 精化框架活动	51	5.9.2 如何选择候选外包供应商	73
5.2.1 需要哪些动作和任务	52	5.9.3 如何评估报价的合理性和 估算的可靠性	74
5.2.2 生产哪些工作产品	53	5.9.4 需要什么程度的项目管理	74
5.2.3 评估质量的恰当方法	54	5.9.5 如何评估进度并管理范围	74
5.2.4 如何管理变更	55	5.10 本章小结	74
5.3 组建一个 WebE 团队	55	参考文献	75
5.3.1 如何识别一个优秀的 WebE 团队	56	第 6 章 建模活动	76
5.3.2 团队为什么没有凝聚力, 怎么改善	56	6.1 建模的概念	76
5.3.3 WebE 团队可以管理自身吗	57	6.1.1 如何评判一个模型的有用性	76
5.3.4 如何组建一个成功的团队	58	6.1.2 模型可以用于理解业务限制吗	77
5.3.5 一个好的团队负责人有哪些 特征	58	6.2 创建的模型	78
5.4 管理风险	59	6.2.1 过程告诉了我们关于建模 的哪些信息	78
5.4.1 如何识别风险	59	6.2.2 WebApp 告诉了我们关于建模 的哪些信息	78
5.4.2 如何评估风险	60	6.3 建模框架	79
5.4.3 如何开发应急计划	61	6.3.1 对 Web 来说,有建模框架吗	80
5.5 开发进度表	62	6.3.2 建模如何与 WebE 过程关联	80
5.5.1 什么是宏观进度安排	62	6.4 建模语言	82
5.5.2 什么是增量进度安排	63	6.4.1 应当具备哪些建模功能的能力	83
5.5.3 如何估算工作量和时间	64	6.4.2 应当具备哪些建模信息内容 的能力	83
5.5.4 如何描述任务之间的相互 依赖关系	65	6.4.3 在建模语言中,应当有哪些 通用能力	84
5.6 管理质量	66	6.5 已有的建模方法	85
5.6.1 团队可以使用哪些质量 保证机制	66	6.6 本章小结	87
5.6.2 什么是结对走查方法	67	参考文献	87
5.6.3 什么是团队走查方法	67	第 7 章 WebApp 分析建模	89
5.6.4 WebApp 存在质量标准吗	68	7.1 理解在 WebE 中的分析	89
5.7 管理变更	68	7.1.1 分析到什么程度才是足够的	89
5.7.1 如何评估变更的重要性和 影响	69	7.1.2 可以使用原型进行分析吗	90
5.7.2 何时推迟做出变更	70	7.1.3 分析和设计不同吗	91
5.7.3 所有的相关工作产品都应当 做出变更吗	71	7.2 对 WebApp 进行分析建模	91
		7.2.1 分析建模的输入是什么	91
		7.2.2 分析的输出是什么	93
		7.2.3 应当执行哪些分析任务	93

7.2.4 可以使用哪些工具来建模	93	8.3.4 有评估内容质量的方法吗	121
7.2.5 如何确定建模是否必要以及 哪种方法最好呢	94	8.3.5 有质量检查表吗	121
7.3 理解用户	95	8.4 设计过程	123
7.3.1 为什么有必要重返用户等级	96	8.4.1 WebApp 设计的元素有哪些	123
7.3.2 不加修改地应用使用场景吗	97	8.4.2 设计过程的特征有哪些	124
7.4 内容模型	99	8.4.3 对设计活动而言, 一个增量 的 WebE 过程意味着什么	125
7.4.1 内容模型的结构元素有哪些	99	8.5 概念架构的初始设计	126
7.4.2 什么是信息交换, 如何进行 表示	100	8.6 技术架构的初始设计	129
7.4.3 如何确定内容对象	101	8.7 本章小结	129
7.4.4 有描述内容之间的联系和 层次的简单方法吗	102	参考文献	130
7.4.5 如何为 WebApp 选择和表示 分析类	103	第 9 章 交互设计	131
7.5 交互模型	104	9.1 界面设计原则和指导准则	131
7.5.1 用例从什么地方开始起作用	104	9.1.1 设计有效界面的原则	132
7.5.2 什么是序列图, 何时进行开发	104	9.1.2 有哪些实用的设计指导原则	135
7.5.3 状态图如何描述 WebApp 的行为	105	9.2 界面设计工作流	135
7.5.4 是否真的需要用例、序列图 和状态图	106	9.3 界面设计的前期工作	137
7.5.5 为什么创建一个界面原型 是很明智的	106	9.3.1 如何理解 WebApp 用户的特征	137
7.6 功能模型	107	9.3.2 如何详细描述已经识别的 内容对象	138
7.7 配置模型	108	9.3.3 用户完成哪些任务	138
7.8 关系 - 导航分析	109	9.3.4 如何详细描述已经确定的任务	140
7.8.1 如何建立内容对象和功能 之间的关系	110	9.3.5 如何为不同角色的不同用户 进行设计	141
7.8.2 如何分析导航需求	110	9.3.6 内容如何集成到界面描述之中	142
7.9 本章小结	111	9.4 界面设计步骤	143
参考文献	111	9.4.1 如何布局界面对象和动作	143
第 8 章 WebApp 的设计	113	9.4.2 如何设计界面导航机制	146
8.1 对 WebApp 进行设计	113	9.4.3 为什么界面的一致性如此重要	147
8.1.1 WebApp 设计者需要知道什么	114	9.5 美学设计	148
8.1.2 什么是逻辑设计	114	9.5.1 如何创建令人愉悦的布局	148
8.1.3 什么是物理设计	114	9.5.2 什么是好的图形设计	149
8.1.4 设计完成后生成哪些信息	114	9.6 可用性	150
8.2 设计目标	115	9.7 设计问题	151
8.3 设计和 WebApp 的质量	117	9.7.1 哪些因素影响响应时间, 怎么改进	151
8.3.1 用户如何理解质量	117	9.7.2 如何设计“帮助”设施	152
8.3.2 有以用户为中心的模型吗	117	9.7.3 界面应当如何处理错误	152
8.3.3 什么是质量框架	120	9.7.4 什么是“可访问性”? 如何 应用于界面设计中	153
		9.7.5 什么是“国际化”? 如何 应用到 WebApp 中	154
		9.8 本章小结	154
		9.9 参考文献	154

第 10 章 信息设计	156	11.4 功能架构	188
10.1 信息架构	156	11.4.1 功能架构是什么样的	189
10.1.1 什么是信息架构	157	11.4.2 如何开发功能架构	190
10.1.2 信息架构的元素有哪些	157	11.4.3 处理异常的功能如何开发	191
10.1.3 好的信息架构有什么特征	158	11.4.4 在功能设计时可以使用架构 模式吗	192
10.1.4 如何开发信息架构	159	11.5 详细功能设计	193
10.2 组织内容	160	11.5.1 WAE 建模如何用于详细 设计中	193
10.3 设计信息空间	161	11.5.2 WebML 为什么适合于工作流 建模	194
10.3.1 有哪些可能的信息结构	161	11.6 状态建模	196
10.3.2 什么可以产生好的结构	164	11.7 本章小结	198
10.4 蓝图：为结构添加细节	165	参考文献	198
10.5 访问信息	167	第 12 章 构建和部署	200
10.5.1 如何确保用户可以理解 上下文并且不会迷路	167	12.1 WebE 过程中的构建和部署	200
10.5.2 如何帮助用户在信息 结构内移动	168	12.1.1 构建和部署之间的相互作用 是什么	201
10.5.3 实现搜索机制的指导准则	169	12.1.2 部署环境扮演什么角色	201
10.5.4 搜索机制会产生问题吗	170	12.2 构建	204
10.6 线框模型	170	12.2.1 有通用的构建任务吗	204
10.7 导航设计：创建详细结构	171	12.2.2 什么是重构？如何应用它	205
10.7.1 信息设计和导航设计模型 是如何演化的	171	12.3 构建的原则和概念	206
10.7.2 RMM 模型如何用于导航设计	172	12.4 部署	207
10.7.3 如何使用 WebML 创建导航 设计	175	12.4.1 是否存在通用的部署任务	207
10.7.4 可能创建链接了内容和功能 的模型吗	176	12.4.2 指导 WebE 团队的部署原则	208
10.7.5 Web 自身的结构有影响吗	176	12.4.3 如何使用版本控制和 CMS	209
10.8 总结设计过程	177	12.5 构建和组件的使用	210
10.9 本章小结	178	12.5.1 什么是通用组件	211
参考文献	179	12.5.2 如何定义面向对象组件	211
第 11 章 功能设计	181	12.5.3 如何定义传统组件	212
11.1 WebApp 功能	181	12.5.4 好组件有哪些特征	212
11.2 WebApp 功能的性质	182	12.6 组件级设计指导准则	213
11.2.1 功能示例	182	12.7 组件的设计步骤	215
11.2.2 功能分类	182	12.8 本章小结	216
11.2.3 信息和功能总是可区分的吗	184	参考文献	217
11.3 设计过程中的功能设计	185	第 13 章 设计模式	219
11.3.1 功能设计过程包含哪些要素	185	13.1 模式：理解概念	219
11.3.2 需要多少功能设计	186	13.1.1 什么是模式	219
11.3.3 SafeHomeAssured.com 的初始 功能设计应该如何进行	186	13.1.2 模式怎么描述	220
		13.2 WebApp 模式：设计焦点和粒度	221
		13.2.1 如何使用设计焦点来确定 模式	221

13.2.2 粒度为什么是模式的一个 重要特征	221	15.1.2 WebApp 环境中发生什么 类型的错误	243
13.3 模式仓库	222	15.1.3 应该应用什么测试策略	243
13.3.1 什么是模式库	222	15.1.4 多少测试计划是必需的	244
13.3.2 Web 工程师可参考哪些 模式资源	222	15.2 测试过程概述	245
13.3.3 WebE 团队能够创建自己的 模式集吗	223	15.3 内容测试	247
13.3.4 如何发现和使用模式	224	15.3.1 内容测试的目标是什么	247
13.4 模式示例	225	15.3.2 数据库测试如何用于验证 内容	248
13.4.1 有可能定义解决业务层 问题的模式吗	225	15.4 用户界面测试	249
13.4.2 因为交互很普遍,就必然 有很多交互模式吗	226	15.4.1 界面测试策略	249
13.4.3 有哪些可用的导航模式	229	15.4.2 如何测试界面机制	250
13.4.4 内容和表示模式适合使用的 地方	232	15.4.3 如何测试界面语义	252
13.5 本章小结	233	15.5 可用性测试	252
参考文献	234	15.6 兼容性测试	255
第 14 章 技术和工具	235	15.7 组件级测试	255
14.1 常见问题	235	15.8 导航测试	256
14.1.1 分而治之如何影响工具和 技术	235	15.8.1 如何测试导航语法	257
14.1.2 选用哪种技术——开源还是 私有	236	15.8.2 如何测试导航语义	257
14.1.3 WebE 技术应用种类的影响	237	15.9 配置测试	259
14.2 实现工具和技术	237	15.9.1 如何测试服务器端	259
14.2.1 什么是应用框架	238	15.9.2 如何测试客户端	259
14.2.2 如何应用内容管理系统和 版本控制技术	238	15.10 安全性和性能测试	260
14.2.3 WebApp 怎样提供搜索能力	239	15.10.1 如何确定 WebApp 是安全的	260
14.3 开发工具和技术	239	15.10.2 如何测试 WebApp 的性能	262
14.3.1 怎样获得有助于建模活动 的工具	239	15.10.3 性能测试的目标是什么	262
14.3.2 专门针对 WebApp 的测试 工具	240	15.10.4 负载测试如何评估性能	263
14.3.3 支持 WebE 过程管理的工具	240	15.10.5 压力测试如何评估性能	263
14.4 本章小结	241	15.11 本章小结	266
参考文献	241	参考文献	266
第 15 章 WebApp 测试	242	第 16 章 变更管理和内容管理	268
15.1 测试概念	242	16.1 变更	268
15.1.1 什么是质量维度	242	16.1.1 变更的属性有哪些	268
		16.1.2 为什么需要变更	268
		16.1.3 WebApp 的哪些元素变更了	269
		16.2 Web 工程的变更管理	269
		16.2.1 为什么需要管理变更	269
		16.2.2 应该如何识别将要发生 变更的对象	271
		16.2.3 如何控制将要发生的变更	271
		16.2.4 如何管理一个 WebApp 或 其组件的不同版本	273

16.2.5 WebE 团队如何确保一个 变更恰当地实现了	274	17.1.1 基于 Web 的内容和功能的 交付会如何变化	282
16.2.6 如何让利益相关者知道 实施的变更	274	17.1.2 WebApp 将如何变化	283
16.3 内容管理	274	17.1.3 为了适应这些变化, Web 工程师必须做些什么	283
16.3.1 如何使用内容管理系统	275	17.1.4 Web 可以作为应用软件的 平台吗	284
16.3.2 CMS 的主要元素有哪些	275	17.1.5 未来的 Web 会是一个 操作系统吗	284
16.4 CMS 的实现标准	277	17.1.6 语义 Web 将如何改变事物 ...	285
16.4.1 数量如何影响内容管理	277	17.2 演化中的 Web 技术和 Web 2.0 ...	286
16.4.2 内容创作者的人数会影响 CMS 吗	278	17.2.1 什么是 Web 2.0	286
16.4.3 变更量如何影响变更管理的 正式性	278	17.2.2 哪些技术支持 Web 2.0	287
16.4.4 发布量如何影响内容管理的 正式性	279	17.2.3 在技术演化中应该考虑 哪些关键问题	290
16.5 本章小结	280	17.2.4 Web 2.0 将会如何发展	290
参考文献	281	17.3 对于未来的观点	291
第 17 章 未来趋势	282	17.4 Web 工程的变化特性	293
17.1 Web 和 WebApp 具有不断变化的 特征	282	参考文献	293

第 1 章 基于 Web 的系统

我们先回顾一下计算机软件开发的前几十年的情况。在 20 世纪五六十年代的时候，几乎没有人意识到基于计算机的系统的重要性，事实上也没有人预见到计算机的软硬件会在 20 世纪末和 21 世纪初期对社会的各个方面产生全球化的影响。在早期，与计算机打交道的绝大多数人会陷入业务的困惑之中，这是因为创建计算机程序的过程中，混杂了许多非正式、紧迫、直觉和艺术等因素。当事情完成得好时，这种方式会带来在计算方面的重要提升。然而，事情不可能总是一帆风顺的。基于计算机的系统常常无法完成其应有的功能，导致交付延迟甚至无法交付，并很难（有时甚至不可能）在任何合理的时间框架内进行纠正、适应和改进。很遗憾，过去使用的是一种碰运气的方式。

但是旧式的思想已经建立起了一种文化，并很快变得难以改变。非正式、紧急、直觉和艺术是大多数基于计算机的系统开发者活动的驱动力。总之，非正式（informality）带来的是宽松的工作环境——你可以在其中做自己的事情。紧迫（urgency）带来行动和快速的决策。直觉（intuition）是一种让你在复杂的情形下“感觉”自己的解决办法的无形的素质。同时，艺术（art）会带来美学的形式和功能——使接触它的人们感到快乐的东西。也许你会问，那有什么不对呢？

当我们把焦点从过去转移到现在的时候，你就会发现这个问题的答案和 Web 工程有莫大的关系，同时它也是本书所讨论的主题。

1.1 网络

今天，我们生活在因特网快速发展的年代。关于这个时代令人振奋的描述已经很多了，所以如果不涉及这些已经算是陈词滥调的对话，就无法去讨论因特网和万维网的影响。你已经知道 Web “大”，很大。尽管 Web 的大小及其增长率令人吃惊，但是我们并不是指传统意义上的“大”（例如，网页和网站的数目、用户量和传输于网络的数据量），我们指的是社会和文化角度的“大”。

Web 已经成为贸易、商业、通信、教育、工程、娱乐、金融、政府、工业、媒体、医药、政治、科学和交通等方面不可或缺的技术，而这些只是影响你生活的诸多方面中的一小部分而已。但是，“不可或缺的技术”只是揭示了 Web 对我们每个人的表面影响。它已经改变了人们购物（电子商务）、与人见面（在线约会）、理解世界（门户）、获得新闻（在线媒体）、发表自己的观点（博客）、自我娱乐（从音乐下载到在线娱乐）以及上学（在线学习）等方式。

所有的这些影响有一个共性——它们需要传输工具，它可以携带与兴趣域相关的原始信息；以一种有意义的方式加以结构化；构建组织好的、美学的、处理过的和交互的（如果需要）包装的表示；同时以启动会话（conversation）的方式将包装的表示传送到 Web 浏览器。

你和一个 Web 应用之间的会话可以是主动的或者被动的。在被动会话中，你选择将要呈现的信息，但是不直接控制它的大小、类型或者结构。在主动会话中，你提供输入，因此所呈现的信息是通过定制来满足你的特定需要。

这个获取信息、组织信息、构建包装的表示并对其进行传输的工具称为 Web 应用（WebApp）。当一个 WebApp 和客户端与服务器端的硬件、操作系统、网络软件以及浏览器结合起来的时候，一个基于 Web 的系统就出现了。

1.2 Web 应用

在万维网 (World Wide Web) 的早期 (大约从 1990 年到 1995 年), “Web 站点” 仅包含链接在一起的少量超文本文件, 这些文件使用文本和有限的图标来表示信息。随着时间的推移, 一些开发工具和技术 (如扩展标记语言 XML、Java) 使 HTML (Hypertext Markup Language, 超文本标记语言) 得到增强, 它使 Web 工程师能够在客户端和服务端的内容中提供计算能力。基于 Web 的系统和应用^①应运而生。今天, WebApp 已经发展为尖端的计算工具, 不仅可以为最终用户提供独立的功能, 而且也已经在公司以及政府的数据库和应用进行了集成。

1.2.1 一个学习案例

CPI 公司是一个虚构的公司, 为家庭和小型企业来构建、买卖、销售、监控安全系统。CPI 公司还没有 Web 的展示, 现在它希望推出一个 “重磅” 的网站, 来介绍它新的安全传感器生产线和一系列全新的基于 Web 的服务。他们希望能在名为 SafeHomeAssured.com 的 WebApp 的开发中得到你的帮助, 同时希望你能够在他们创建增加市场份额的新 Web 服务时提供支持。

你被通知去参加一个讨论基本想法的会议。在会议中, 你得知 CPI 已经设计出了一个简洁的无线传感器控制器, 这个控制器将成为被称为 SafeHome 的商业和住宅安全系统的一个新生产线的核心部件。以下是会议谈话的一个片段。

SafeHome	
	一个项目开始了 场景: CPI 公司的会议室里。 参与者: 一个高级业务经理, 一个产品开发经理, 一个营销经理, 一个工程经理, 还有你, Web 工程的专家。 会话: 业务经理 (对产品经理说): 我听说你们正在开发一个产品? 一个通用的无线盒? 产品经理: 它是相当酷的产品, 只有一个小火柴盒的大小。我们可以把它与各种只要使用了 IEEE 的任何无线协议的传感器 (比如一个数码相机) 相连接。它允许我们通过无线的方式访问设备。我们认为它将带来一代全新的产品。 业务经理 (望着营销经理): 你同意吗? 营销经理: 我同意。事实上, 从今年趋缓的销售业绩来看, 我们需要一些新的产品。我
	们已经做了一些市场调查, 同时已经有有了一个可能很大的产品和服务线。 业务经理: 多大? 底线是多少? 营销经理: 这是全新一代的 “住宅管理系统”, 我们称之为 SafeHome。产品采用新的无线接口, 为家庭和小型企业使用者提供一个由电脑通过因特网控制的系统 (家庭安全、监控、应用和设备控制)。当你在开车回家的路上的时候, 可以打开家里的空调, 或者诸如此类的事情。我们同时也考虑了住宅和企业内部的视频监控和控制。另外很重要的一点是, 我们试图把产品与监控服务直接集成起来, 允许客户通过 Web 去访问他们的账户, 做出一些决定, 比如什么时候对系统进行监控, 什么时候撤销监控, 以及在一个特定的时间段内发生了什么事情。我们还想通过 Web 来完成维护诊断。

① 在本书中, Web Application (WebApp) 这个术语涵盖的范围甚广, 小到可以帮助用户计算租车费用的简单 Web 页面, 大到为商务人士和度假者提供全面的旅行服务的站点。这类应用包含驻留在因特网、企业内部和外部网络上的完整的网站、网站中的特定的功能以及信息处理应用。

产品经理：工程部已经对这些想法进行了技术可行性分析。结论是，这个项目可行，而且可以做到低成本。绝大多数的硬件都是现货供应的。基于 Web 的软件是个问题，但也并非是我们做不了的事情。我们已经注册了一个域名：SafeHomeAssured.com。
[所有 CPI 经理们都微笑着看着你。]
业务经理：有意思。现在，我来问一下底线。

营销经理：在美国，有 PC 机的家庭占有很大的比例。如果我们定价合适，它将会是一个非常成功的应用。只有我们才拥有这一无线盒的专利（所有权）。我们将在这个方面拥有两年的竞争优势。至于收入？在第二年大概会有三千万到四千万美元。
业务经理（笑容满面）：让我们进行更进一步的讨论。我很感兴趣。

就这样，一个项目开始了。你会注意到现阶段很少涉及细节。有许多事情需要被定义、具体化，然后实现。对产品的内在理解和支持它的基于 Web 的系统会发生变化。但是在现在早期阶段这确实没有什么关系。SafeHome 有高级管理者（可以预见到巨大的潜在利益的人）的支持，而且你有机会成为完成这一工作的一员。

在本书中，我们会不停地回到 SafeHome 和 SafeHomeAssured.com 的 WebApp 上来，把这个项目作为一个案例，来描述 Web 工程中的许多方面。但是现在回到对 WebApp 的介绍性讨论上来，并研究一下它们和传统的计算机软件之间的相似性。

1.2.2 WebApp 就是计算机软件吗

如果把 WebApp 看作是一组指令和数据，可为最终用户提供信息和功能，那么毫无争议，WebApp 就是计算机软件。因此，也就是说，我们有理由设想，可以借鉴在过去几十年构建传统的基于计算机的系统时所学到的一些（如果不是全部）知识来构建 WebApp。同时也有理由认为，我们将会遇到很多（如果不是全部）在早期曾经经历过的问题（包括文化上的和技术上的）。本书的后面将对其进行讨论。

1.2.3 WebApp 的属性有别于传统软件的属性吗

人们在这个问题的回答上存在争议。有些人认为一个 WebApp 不过是一个高度重视美学表示（例如，布局、图像、音频和视频元素）和功能的客户端 - 服务器应用而已，认为 WebApp 和传统客户端 - 服务器应用有着相同的属性。但是另一些人（包括本书的作者）认为，当考虑它们的整体性时，一个 WebApp 的完整的特征集确实使基于 Web 的系统有别于更传统的基于计算机的系统。绝大多数 WebApp 都会具有如下属性。

网络密集性（network intensiveness）。每一个 WebApp 都驻留在网络上，并且必须为大量不同客户群体的需要提供服务。就 SafeHome[⊖] 的产品而言，许多即将由 CPI 实现的新特征都将通过 Web 进行初始化、控制和监控。网络使 SafeHomeAssured.com 这个 WebApp 的基于客户的特征与由 CPI 建立的服务器之间建立了通信。

并发性（concurrency）。大量的用户可能会同时访问某个 WebApp。在很多情况下，最终用户的使用方式差异非常大。有时，一个或一组用户的行为可能会影响到其他用户的行为或者展示给其他用户的信息。就 SafeHomeAssured.com 而言，数以万计的住宅将被并发地监控；几百或几千个客户可能在任意时间访问 WebApp；几十个服务技术人员也可能同时在线。

⊖ SafeHome 是本章前面介绍过的一个基于 Web 的系统所支持的一个安全系统，将用作全书中连续的例子。

不可预测的负载 (unpredictable load)。访问 WebApp 的用户数目每天都可能会有数量级的变化。至于 SafeHomeAssured.com, 被监控的住宅和业务的数量变化将很缓慢。但是 WebApp 必须能够处理同时发生的数目不可预测的事件 (例如, 防窃报警器、火警装置和一氧化碳检测装置)。周一, 也许每个小时会报告 10 个事件; 周二, 也许会记录 100 个事件; 周三 (在一个区域大规模断电之后) 每分钟就可能报告几千个事件。

性能敏感性 (performance sensitive)。如果一位 WebApp 用户 (对于访问、服务器端的处理、客户端的格式化和显示) 必须要等很长时间, 该用户就可能会转到其他地方。具体到 SafeHomeAssured.com, 它的性能是至关重要的, 因为人们的生命可能受到威胁。如果 WebApp 对事件的响应太慢, 诉讼就可能发生。

高可用性 (high availability)。虽然说期待百分之百的可用性是不现实的, 但是流行的 WebApp 的用户通常要求基本的“24/7/365” (全天候) 的可访问性。就 SafeHomeAssured.com 而言, 其目标是百分之百的可用性 (已经说过, 这一系统是关于住宅安全的), 因此 WebApp 必须被设计成满足这个理想状况 (或者非常接近它)。

数据驱动 (data driven)。很多 WebApp 的主要功能是借助于超媒体来向最终用户展示文本、图像、音频和视频内容。另外, WebApp 通常用于访问那些存在于数据库中的信息, 而这些数据库并非基于 Web 的环境的整体的一部分 (例如, 电子商务或金融应用)。就 SafeHomeAssured.com 而言, 所有这些属性都将是明显的。而且, 这个 WebApp 必须访问存储了每个用户信息的数据库; 用户具有的系统配置; 以及给定系统的监控需求、一个事件日志和一个维护日志。

内容敏感性 (content sensitive)。内容的质量和美学性依然是 WebApp 质量的重要决定因素。就 SafeHomeAssured.com 而言, WebApp 的一个重要的用户类将是“平民”, 即那些要求简单而有意义的内容展示的非技术人群。

持续演化 (continuous evolution)。传统应用软件随一系列计划好的时间间隔发布而演化, 而 WebApp 是持续地演化。对某些 WebApp (尤其是 WebApp 的内容) 而言, 根据以分钟计的进度表进行更新, 或者针对每个请求进行独立的内容计算是司空见惯的事。在本书的后面, 我们将会看到, 随着时间的推移, SafeHomeAssured.com 这个 WebApp 将随着对系统变化的理解而演化。WebApp 的演化将需要一种“增量”的开发方法。

即时性 (immediacy)。虽然说即时性 (也就是把软件尽快推向市场的强制性要求) 是很多应用领域的特点, 但是将 WebApp 投向市场常常是几天或者几周的事情^①。Web 工程师们必须使用为满足 WebApp 开发所要求的紧迫时间进度而修改过的方法来进行计划、分析、设计、实现以及测试。就 SafeHomeAssured.com 而言, CPI 管理的焦点在于短期的收入增加和中期的显著收入。当这发生时, 就体现出“昨天”需要 WebApp 了。

安全性 (Security)。由于 WebApp 是通过网络进行访问的, 所以即使可能也很难限制访问这个应用的最终用户的数量。为了保护敏感的内容, 并提供安全的数据传输模式, 在支持 WebApp 的整个基础架构以及应用本身内部都必须实现强有力的安全措施。就 SafeHomeAssured.com 而言, 信息从人们的住宅和商务中流入流出, 使这个 WebApp 成为那些有犯罪意图的人的绝好的目标。它必须是安全的。

美学性 (aesthetics)。WebApp 吸引力的一个不可否认的部分是它的外观和感觉。当设计好一个应用要推向市场、出售产品或者想法、提供服务来产生收入时, 美学与技术设计同样关系到该应用的成功。就 SafeHomeAssured.com 而言, WebApp 将提供的内容和功能的多样性 (将在第 4

^① 有了现代的工具, 复杂的 Web 页面在几个小时之内就可以完成。