

软件

创世纪工商管理译库
主 编：廖 理

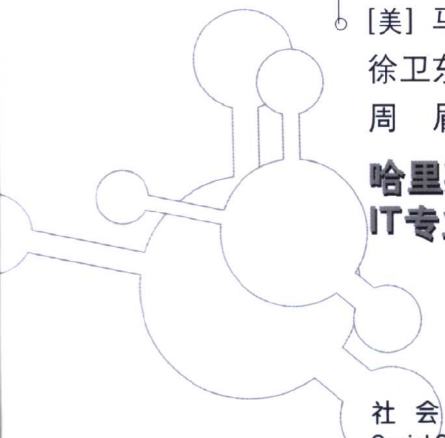
开 发

建立可靠的系统

Building Reliable Systems

○ [美] 马克·汉密尔顿(Marc Hamilton) 著
徐卫东 译
周 眉 审校

哈里斯·科恩企业计算技术协会
IT专业人士解决方案



社 会 科 学 文 献 出 版 社
Social Sciences Documentation Publishing House

创世世纪工商管理译库

软件开发

建立可靠的系统

Software Development
Building Reliable Systems

[美] 马克·汉密尔顿(Marc Hamilton) 著



创世纪工商管理译库

软件开发

建立可靠的系统

Software Development
Building Reliable Systems

[美] 马克·汉密尔顿 (Marc Hamilton) 著

徐卫东 / 译

周 眉 / 审校

软件开发：建立可靠的系统

·创世纪工商管理译库·

著 者 / [美] 马克·汉密尔顿

译 者 / 徐卫东

审 校 / 周 眉

出 版 人 / 谢寿光

出 版 者 / 社会科学文献出版社

地 址 / 北京市东城区先晓胡同 10 号

邮政编码 / 100005

网 址 / <http://www.ssdph.com.cn>

责任部门 / 财经与管理图书事业部

(010)65286768

责任编辑 / 屠敏珠

文稿编辑 / 冯韵文

责任校对 / 邵鸣军

责任印制 / 同 非

总 经 销 / 社会科学文献出版社发行部

(010)65139961 65139963

经 销 / 各地书店

读者服务 / 客户服务中心

(010)65285539

法律顾问 / 北京建元律师事务所

排 版 / 北京中文天地文化艺术有限公司

印 刷 / 北京智力达印刷有限公司

开 本 / 787×1092 毫米 1/16 开

印 张 / 18.5

字 数 / 323 千字

版 次 / 2003 年 12 月第 1 版

印 次 / 2003 年 12 月第 1 次印刷

书 号 / ISBN 7-80190-002-2/F·002

著作权合同 / 图字 01-2002-6259

登 记 号

定 价 / 35.00 元

本书如有破损、缺页、装订错误，
请与本社客户服务中心联系更换



版权所有 翻印必究

版权声明

Software Development: Building Reliable Systems

Marc Hamilton

ISBN 0-13-081246-3

Copyright © 1999 by the Prentice Hall, Inc.

All Rights reserved. No part of this book may be reproduced, in any form or any means, without permission in writing from the publisher.

Simplified Chinese translation edition jointly published by Prentice Hall, Inc. and Social Science Documentation Publishing House, CASS.

This edition is authorized for sale only in People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书仅限在中华人民共和国境内（不包括香港和澳门特别行政区）销售。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签。无标签者不得销售。

主 编：廖 理
译 者：徐卫东
审 校：周 眉
出版人：谢寿光

策 划：路卫军 周 丽 王莉莉
责 编：屠敏珠

图书在版编目 (CIP) 数据

软件开发：建立可靠的系统/[美] 马克·汉密尔顿著；徐卫东译. —北京：社会科学文献出版社，2003.12

(创世纪工商管理译库)

书名原文：Software Development

ISBN 7-80190-002-2

I. 软... II. ①马... ②徐... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2003) 第 042153 号

献 词

在我的工作生涯中，已经编写了相当于几千页的代码，以及大量技术文章和其他新闻方面的尝试，因此，编写一本 350 页的有关软件开发方面的书在我开始动笔时似乎并没有太大的困难。大约一年之后，我必须说我过分低估了所需时间及其中的挑战。不过，这本书并不是我个人努力的结果，而是许多帮助过我的人的共同努力的结果，包括在过去的一年中直接或间接地帮助过我的人。最大的帮助来自我的家庭，他们在寒假、暑假、周末以及任何空余的时间里忍受着我在深夜打字。因此，我要把这本书献给我结婚 10 年的妻子，罗伯塔（Roberta）和我的两个可爱的孩子卡米尔（Camille）和埃文（Evan）。

鸣 谢

感谢哈里斯·科恩 (Harris Kern)，是他首先说服我编写这本书。无论他在世界任何地方旅行，我总是能得到他的帮助。他帮我审查新写的一段内容以及整本书的草稿。哈里斯在此期间一直与我共同工作，从最初的轮廓到最后的定稿，没有他的帮助和支持，这本书永远也不可能完成。

感谢罗耀拉·玛丽芒特 (Loyola Marymount) 大学的雷·托尔 (Ray Toal) 博士。除了编写了前言外，他还编写了本书的众多草稿，为本书的定稿做出很大贡献。托尔博士，就像一位真正的教授能够做的那样，详细阅读了本书的许多技术细节和实例，力求做到准确无误。另外，由于我的工作经历，我有时会把我自己对一些语言或供应商的偏见带入书中，每当出现这种情况时，他总是能及时指出。

感谢过去一年我在太阳微系统公司 (Sun Microsystems) 的同事，包括史蒂夫·巴兰坦 (Steve Ballantine)、卡罗琳·布马泰 (Carolyn Bumatay)、史蒂文·程 (Steven Cheng)、戴维·楚 (David Chu)、约翰·弗雷格拉 (John Fragalla)、芭芭拉·赫尔 (Barbara Herr)、阿尼塔·詹森 (Anita Jensen)、克雷格·莱曼 (Craig Lehman)、布鲁斯·马修斯 (Bruce Mathews)、托尼·曼苏尔 (Tony Mansour)、尼尔·帕森斯 (Neil Parsons)、哈曼特·帕特尔 (Hamant Patel)、安德鲁·施莫勒 (Andrew Schmoller)、斯蒂夫·斯塔索 (Steve Staso) 以及杰夫·韦斯 (Jeff Weiss)，感谢他们对我的始终信任和支持。

感谢过去 4 年我在太阳微系统公司的经理汤姆·斯威斯古德 (Tom Swysgood)，他不仅支持我写此书，而且在那段时间，为了管理位于洛杉矶的 Sun 公司所发生的显著增长和变动提供了推动力和新方法。汤姆或许不记得了，但正是从哈里斯那里转发过来的他的电子邮件首先激发了我写这本书的兴趣。

感谢多年来在 Sun 公司中我有幸共同工作的所有人，包括：

- 恺撒·巴雷 (Cesar Baray)，他教我如何管理销售部门。
- 杰伊·春 (Jae Chung)，他教给我这样一种观念，当你真正想达到某种目的时就决不要放弃。
- 吉姆·伊斯特曼 (Jim Eastman)，他尽力教我有关销售的知识。
- 丹尼斯·赫尔甘 (Dennis Horgan)，在我写这本书时，他对新闻工作的努力对我

也有很大激励，并且他丰富的信息技术知识不仅丰富了这本书的内容，而且对 Sun 公司中他一起工作过的所有客户都有帮助。

- 恰克·梅耶尔 (Chuck Meyer)，经过 5 年的时间，他最终说服我加入 Sun 公司，并且一直是我的老师和好朋友。
- 彼德·斯达福 (Pete Stafford)，他教会我如何突破任何障碍。

最后，感谢多年来和我一起工作的 TRW 的所有出色的开发人员，以及真正教会了我有关成功的软件开发内容的人们。

前 言

现在的软件企业所面对的世界和几年前的大不相同。现在整个地球都被线路连接在一起。人们在家里和经纪人的房间里都可以进行购物和储蓄。对于企业来说，大型机继续让位给客户/服务器以及多层系统。台式个人电脑已变得很普通，基本上是一次性的了。网络电脑的新应用也将很快普及。不久，我们现在的家庭用具甚至也将联网。

幸运的是，软件开发人员有了很多新的软件工具和技术，因此在这不断变化的环境中仍然能繁荣发展。现在已经有了新的编程语言和原有语言的新版本。C++（终于！）成为了国际标准，老的数组和“星号字符”只有在低级的系统和嵌入式软件上才可以继续使用。由于Java具有很高的安全性、可移植性（一旦写成便可在任何地方运行）和丰富的Core APIs，因此这种新型语言获得了无可比拟的成功。Visual Basic被证明是真正有效的软件组件，在写客户程序，甚至是中间件业务对象时都非常受欢迎（FORTRAN和COBOL仍然存在，甚至可能会被用到下一次出现Y10K问题上）。

开发人员的工作方式也发生了变化。他们已经有了编制器和可视环境。他们现在不写那么多“单块”的应用了，相反，他们更多地采用构架，插件程序，并编写越来越多的组件。无论这些组件是JavaBeans，还是Active X控件或是在连接CORBA的企业中，这些可重复使用的二进制组件已经并将继续对软件的开发、市场和销售产生深远的影响。组件产品超越了面向对象的编程；基于组件的编程使我们更注意可重复使用的结构和设计方式，更重视利用界面，并减少了对继承性的依靠，增加了对组件的依靠。

这些迅猛的技术进步在人员和程序方面造成了新的和非常重要的问题。在人员方面，出现了一些新的职位，例如工具专家（Toolsmith）和网管（Webmaster）。在现在这个工资逐步上涨的时候，留住人才对许多企业来说也成为重要的问题。管理人员的工作环境中充斥着许多高傲、衣着随便、薪酬过高的程序员，这些程序员以超过自己的同事为荣，而他们互相攀比的因素可能在其他行业甚至是没有听说过的。

在程序方面，对大型机的放弃使人们担心会同时放弃以数据为中心的规程，因此诞生了新的开发方式。生产所面临的基本挑战仍然存在：软件生产和其他商品的生产

不同，软件太“灵活”了，无法在繁忙的日程中增加对其重复使用，等等。验证大量分布式系统的挑战使人们需要新的验收方法。很明显的一点是，这些人员和程序的挑战是普通的管理原理所无法解决的。

《软件开发》是所有软件业人士都可以、应该、并且肯定想读的一本书。它为人员、程序和技术问题提供了统一的解决办法，同时，其涵盖的范围又很适度。该书的作者在管理和软件工程方面都有多年的经验，在两方面都很有权威。管理人员会在本书中发现有关招聘、组织、保留人才、员工介绍、如何测量软件质量、如何使用各个领域的专家以及如何回答一些常见的问题，如“自己制作还是购买”等方面的有用的信息。开发人员毫无疑问将对关于程序语言、工具、硬件、多线程、组件、分布式结构和 Jini 等技术部分感兴趣。但这本书真正的优势之一是管理人员可以读懂本书的技术部分，而开发人员也能理解人员 and 程序的问题。管理人员需要了解技术，开发人员也不能不清楚程序和管理问题。设计、开发和质量保证等互不干涉的流水线型的企业是注定要失败的。组织中的每个人都应该参与软件的整个生命周期的工作。开发人员需要和用户沟通，以了解其要求。每个人都应该参与测试和使用。这是一本向每个人介绍宏观概况的书。

现在大多数的软件开发仍然被拖延和超出预算。这就意味着使一个项目成功将非常艰难〔这并不奇怪，因为，就像布奇（Booch）所指出的那样，软件从本质上来说就是复杂的〕。很多人就如何使软件成功这一问题提出过很多建议：一些人用学术原理、一些人用管理原理、一些人用新技术、一些人用程序模型的蓝图，还有一些用案例分析。而《软件开发》这本书是通过对人员、程序和技术问题的完美结合，告诉你需要知道的东西。通过这本书，你会了解到成功的企业之成功或失败的原因、怎样建立和保留可以胜利的队伍，以及目前最新和最有前途的技术的全面信息。你会在这本书中发现你可以具体运用并保证你开发成功的信息。

雷·托尔博士

加利福尼亚州洛杉矶罗耀拉·玛丽芒特（Loyola Marymount）大学
计算机科学教授

简 介

如今，人们生活的方方面面几乎都涉及了软件。从《财富》500强企业玻璃暖房数据中心的重要商务应用软件，到汽车的防抱死刹车系统的实时软件，到家庭电脑的网络浏览器，软件无所不在。由于传统软件开发的复杂性，据估计现在使用的一半以上的代码都至少是在5年前设计的。但是，从1995年起，互联网技术和Java平台的广泛应用从根本上改变了软件开发的性质。到2000年，除了很多永远无法排除的Y2K故障外，软件工程师可以为所有东西编写代码，包括从企业配有JavaBeans的应用服务器，到可以和Jini技术实时联网的消费者设备安装驱动程序。毫无疑问，像Intel这样的公司将能够继续开发功能越来越强大的微处理器，来执行所有这些新的代码。自从Intel的创始人之一戈登·摩尔（Gordon Moore）1965年得出其著名的观察结果以来，摩尔法则，即集成电路处理器的功能每12~18个月翻一番，在过去的30年中都是正确的。很多专家，包括摩尔本人，都认为这种趋势至少在未来的20年中仍将保持。但遗憾的是，软件开发没有这样持久的趋势。

第一个描写软件开发的困难的作者是弗里德里克·布鲁克斯（Frederick Brooks）。他在1975年写的《神秘人月》（The Mythical Man Month）一书描述了为什么为一个已经被拖延了的软件项目增加人员只会更加推迟项目的进程。12年后，在题为《没有最有效的方法：软件工程的精髓和故障》的IEEE计算机文章中，布鲁克斯写下了他的一句经典名言：“软件工程没有最有效的方法”。实际上，很多软件项目之所以还算成功只不过是出于软件开发的延误，硬件速度的大量提升掩盖了糟糕的软件设计带来的问题。很多其他的软件项目，由于缺乏成熟的开发程序而不幸完全失败。在很多公司，工程师仍然在使用文本编辑器以及跟20年前差不多的编译器一行一行地编写软件。当然，也有很多新程序员可以使用编写简单程序的图形数据开发工具。但如果想要成功完成一个大型软件项目的话，你所需要的不仅是一个好的开发工具。

最成功的、按时并在预算内完成的软件项目来源于遵守严格规程的技术高超的程序员、结构设计人员和其他专家。尽管有关软件开发的书有几百本，据估计80%的软件项目都没有按照预定的时间和预算完成。造成这种状况的原因是有技术的程序员或功能强大的开发工具本身并不能保证软件开发的成功。《软件开发》一书将具体阐述

使那 20% 的项目成功的人员、程序和技术方面的因素。

谁应当读这本书

这本书是为专业软件工程师、软件结构设计师、软件工程管理人员和 MIS 管理人员写的。典型的读者可能是一个大公司的信息技术部门的应用开发组成员，如系统积分器的开发人员或商业软件生产商的内部开发人员。本书中的大多数概念都是在涉及 5~50 名软件工程师的中型到大型软件项目中开发出来的。但这并不是说这些概念就不适用于更小型的项目。只是这样一来你可能会太关心高层管理问题而没有太多时间来处理实际的软件工程程序，从而把时间都花在人员和方法问题上。如果你的项目只需要不到 5 名软件工程师，你还是应该读这本书，因为本书中的技术概念也适用于最小型的、甚至只需 1 名开发人员的开发项目。此外，一旦你在小项目上证明了你的技术，你肯定就会有参与大型项目，而对大型项目来说，人员和程序的问题就变得更为重要了。以下是各种不同的人能分别从这本书中得到的东西。

对于软件工程师来说，描述这本书最好先说说它不包括的内容。基本上来说，这本书不是为新程序员所写的讲“如何编程”的书。它也不是讲程序语言和具体硬件的书。我们在书中使用 Java 和 C 来说明一些编程的例子只是因为大多数程序员都很熟悉这两种语言。但是，其中所说的原理适用于用任何语言进行的软件开发。

将从这本书中获益的是那些希望了解正在开发中型到大型项目的技术领先的公司所使用的最新技术的专业软件工程师。这本书对于从来没机会和一组开发人员一起做项目，而又想了解在那种环境中有助成功的技术的开发人员来说也很有用。这本书主要着重于软件开发的设计阶段，其中介绍的概念和技术可以使最有经验的开发人员进一步提高工作效率。由于这本书着重于软件开发的设计阶段，所以对想转变为有更多设计责任的软件结构设计师的那些软件工程师来说也很有价值。

除了给想成为软件设计师的软件工程师提供帮助之外，它还向软件结构设计师介绍了最新的软件工程技术，如基于组件的软件开发、多层网络中心设计、面向对象的技术和 Java 领域的令人激动的新发展，如 Jini 技术。由于新的软件技术演变得非常快，大多数软件结构设计师可能会发现本书的内容有很多领域都是他们没有亲身实践过的。除了软件技术之外，本书还给软件结构设计师提供了如何用最熟练的人和编程技术设计软件系统的信息。这对于想涉足软件工程管理的结构设计师来说非常有用。

对于软件工程经理来说，本书为软件开发的人员和程序问题提供了一个独特的视角。本书的部分章节介绍如何发现、招聘、聘用和保留最好的软件工程师和结构设计师。对软件开发不熟悉的管理人员将会了解，为什么他们的软件工程师和软件结构设

计师的技术和经验对软件项目是否成功的影响超过他们以前管理的其他信息技术领域。不过，本书中并没有具体的“管理”章节。如果你想要成为成功的软件工程经理，你需要理解本书中介绍的所有概念。很多其他的书只着重于管理技巧，而忘记了最好的经理不仅应该理解他们的软件工程师实际正在做什么，而且还应该了解他们的开发人员真正想要做什么。如果软件工程师正在做的事就是他们真正想要做的事，那么他们的生产力将会提高。你一旦理解了这些概念，你就能更好地为软件开发项目制定时间计划和预算，从而保证各个层次的成功。

最后，本书对从管理信息系统（MIS）经理到首席信息官（CIO）的高层管理人员都很有用。这些管理人员并不一定认为自己需要具备软件技术，但本书对于成功的软件开发需要哪些东西提供了独特的见解。首先，要想使生产力最大化，软件开发团队需要有好的软件开发基础设施。软件开发环境对于开发人员的桌面、服务器和相关网络基础设施都有特殊的要求。而这些都是由组织中的管理信息系统经理控制的。此外，一旦软件开始使用，这可能是最终测试项目是否成功的时候，管理信息系统经理就需要提供适合的生产基础设施。通过了解软件项目的基本细节，例如是以网络为中心的还是基于主机的，管理信息系统经理就可以提供更合适的基础设施。反过来，软件结构设计师和开发人员了解了软件的使用环境后，就可以在早期制定正确的设计决策，这也会大为简化他们的管理信息系统经理的工作。

怎样阅读这本书

《软件开发》共分为 4 部分，分别介绍成功的软件开发的背景、人员、程序和技术问题。我们希望你能把这本书从头读到尾，但我们在每章的开头还是做了一个简短的、不到一页的简介。任何真正对软件开发感兴趣的人，包括新程序员到最有经验的结构设计师和经理，都至少应该阅读每章的简介。此外，应该把第 1 章“软件开发成功的十戒律”全部阅读一遍，因为这部分总结了我们认为所有涉及软件开发项目的人都应该熟悉的 10 个最重要的概念。这 10 条戒律将在本书的其他部分一遍又一遍地提到。对于其他章节，软件结构设计师、开发人员和经理可能会根据自己的兴趣来挑选。以下是对这 4 个部分的进一步介绍。

第一部分包括第 1~4 章，这部分概括介绍了软件开发概念的背景。软件开发人员和结构设计师只要大致看一下第 2~4 章的内容就可以，因为他们对这部分可能都非常熟悉。软件开发经理和管理信息系统经理，特别是没有太多开发背景的人，应该仔细阅读这一部分，并确保理解其中的概念，因为这部分是理解本书其余部分的基础。第 5~9 章是本书的第二部分，主要是介绍和软件开发相关的人员问题。对于软



件开发经理来说，这可能是本书最重要的部分。第三部分包括第 10~13 章，主要是介绍软件开发的程序。所有的读者都应该对这部分感兴趣。虽然有人认为软件开发更像是一门艺术，而不是一门科学，但具有成熟和规范的开发方法的项目的成功率总是更高一些。第 14~22 章是本书的第四部分，这部分介绍了软件开发的技术。经理至少应该了解这部分介绍的概念，而软件结构设计师或开发人员可能会详细阅读这部分。

关于作者

马克·汉密尔顿从 1975 年开始开发软件，当时他还在上中学，他使用的是 FORTRAN 语言，学校的电脑是 IBM1602，有 8K 内存。文本编辑器是 Key punch，编译器是一张 3 英寸厚的计算机穿孔卡。后来汉密尔顿先生进入 UCLA 学习计算机科学，并继续对软件充满兴趣。虽然那时 UCLA 的 IBM 大型机比 IBM1620 要先进，但大学生仍然使用穿孔卡。只有研究生才能使用大型机的绿屏 3270 终端及其 TSO（IBM 的“时间共享权”终端控制软件）环境。但幸运的是，汉密尔顿先生的第一门编程课没有助教，当时他自告奋勇做班上的系统管理员，于是可以使用一个 TSO 终端。汉密尔顿记得，正是在那个时候，他意识到自己不会再改变对计算机的兴趣了。

汉密尔顿先生在 UCLA 读三、四年级的时候，在 TRW 的航天飞机工程部做暑期工，开发软件。那时南加利福尼亚的航空业正在开始进入里根时代的全盛期，因此他很容易就在 TRW 找到了一份兼职工作。TRW 的航空部门是世界上最大的软件开发机构之一。因此他毕业后自然也选择继续到那里工作。

在 20 世纪 80 年代早期，TRW 的软件开发通常意味着在 DEC 机上工作。VAX11-780 是航空业广泛使用的计算机。VMS 是行业级操作系统。但 DEC 的销售代表可能不知道的是，每天晚上 5 点钟，当计划生产结束之后，像汉密尔顿这样的新员工就会装上有伯克利（Berkeley）Unix 操作系统〔伯克利的比尔·乔伊（Bill Joy）在其共同创立太阳微系统公司之前开发的系统〕的磁盘，然后一直工作到深夜，用 C 语言开发新的软件原型。新的计算机公司，例如加利福尼亚的太阳微系统公司和马萨诸塞的阿波罗（阿波罗后来被惠普收购了），都在开发他们的第一个工作站，所以除了 DEC 之外，大家都知道这个情况。

汉密尔顿先生全职工作一年后，认为 TRW 的研究生课程太好了，不容错过。于是在他就读于 USC 的研究生院的时候，第一次接触了面向对象的编程，拥有 Symbolics Lisp 工作站和被称为“Flavors”的对象系统。Flavors 和另一个叫做 Common Loops 的对象系统后来成为 Common Lisp Object System 的基础。除了一两台 Symbolics 机器

和一个使用遗留的 DEC 硬件的项目之外，汉密尔顿先生将他在 TRW 的 11 年时间中的大部分都花在了基于 Sun 的软件开发项目的后继工作中。

1993 年末，汉密尔顿先生离开了 TRW，加入了太阳微系统公司的系统工程组。现在，他是 Sun 公司位于洛杉矶办公地的系统项目经理。在 Sun 公司，他继续与终端用户软件的开发人员打交道。另外，他还密切地与内部的 Sun 软件开发部门、独立的软件供应商和商业系统集成器的开发人员一起工作。在最近 5 年，他和将近 100 家不同公司的软件开发部门和信息技术部门合作过，包括施乐（Xerox）PARC、Oracle、Mattel、UCLA、USC、波音（Boeing）、TRW、休斯（Hughes）、安德森咨询公司（Anderson Consulting）、丰田（Toyota）和 JPL 等，当然还有太阳微系统公司。另外，他还是 Sun 的 Java ACES 组的创始人之一，负责完善 Java 技术和对 Sun 的系统工程的现场力量进行培训。“我对软件总是充满激情，”哈密尔顿说，“并且我在 Sun 的工作使我能够接触到世界上一些最好的软件。”哈密尔顿先生几乎把他接下来在 TRW 的 11 年的时间都用来做一系列基于 Sun 的软件开发项目。

在 1993 年底，哈密尔顿先生离开了 TRW，加入了太阳微系统公司的系统工程队伍。现在，他是 Sun 在洛杉矶的办事处的系统工程经理。他在 Sun 继续进行终端用户软件开发。此外，他还和 Sun 的内部软件开发机构、独立软件制造商以及商业系统集成器的开发人员紧密合作。在过去的 5 年中，他和几乎 100 家不同的公司，包括 Xerox PARC, Oracle, Mattel, UCLA, USC, Boeing, TRW, Hughes Anderson Consulting, Toyota, JPL, 当然也包括太阳微系统公司的软件开发机构和信息技术部一起工作。此外，他还是 Sun 的 Java ACES 小组的发起人之一，主要负责 Java 技术的转出和 Sun 的系统工程人员。“我对软件总是充满热情，”哈密尔顿说，“我在 Sun 的工作使我接触到了一些世界上最优秀的软件技术。”