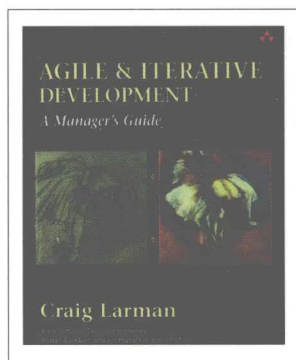


PEARSON



软件开发方法学
精选系列



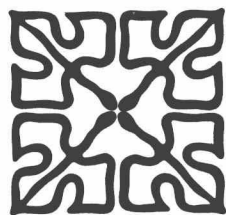
Agile and Iterative Development:
A Manager's Guide

[美] Craig Larman 著

张晓坤 译

敏捷迭代开发

管理者指南



软件开发方法学精选系列

Agile and Iterative Development:
A Manager's Guide

[美] Craig Larman 著

张晓坤 译

敏捷迭代开发

管理者指南

人民邮电出版社
北京

图书在版编目 (C I P) 数据

敏捷迭代开发 : 管理者指南 / (美) 拉尔曼
(Larman, C.) 著 ; 张晓坤译. — 北京 : 人民邮电出版
社, 2013. 7

(软件开发方法学精选系列)

书名原文: Agile and iterative development: A
manager' s guide

ISBN 978-7-115-31642-4

I. ①敏… II. ①拉… ②张… III. ①软件开发
IV. ①TP311.52

中国版本图书馆CIP数据核字 (2013) 第074906号

内 容 提 要

本书是敏捷和迭代开发方法的权威指南。著名软件方法大师 Craig Larman 在书中不但说明什么是敏捷/迭代方法, 其运作机制、实施策略以及原因, 而且通过具有统计意义的重要研究数据, 以及大规模的项目案例分析, 为读者呈现了最具有说服力的采用迭代开发的有力证据。本书主要内容包括: 大量实用的敏捷和迭代技巧, 面向敏捷/迭代项目主管的新管理技能, 敏捷与迭代的价值与实践, Scrum、XP、UP 和 Evo 的关键实践, 以及常见问题的问答。

无论是对 IT 主管、项目经理, 还是对软件开发人员, 这都是了解敏捷和迭代开发最理想的一本书。

-
- ◆ 著 [美] Craig Larman
 - 译 张晓坤
 - 责任编辑 杨海玲
 - 责任印制 程彦红 焦志炜
 - ◆ 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 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 700×1000 1/16
印张: 15.5
字数: 290 千字 2013 年 7 月第 1 版
印数: 1—3 000 册 2013 年 7 月河北第 1 次印刷

著作权合同登记号 图字: 01-2012-6494 号

定价: 49.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

版权声明

Authorized translation from the English language edition, entitled: *Agile and Iterative Development: A Manager's Guide*, 9780131111554 by Graig Larman, published by Pearson Education, Inc., publishing as Addison-Wesley Professional, Copyright © 2004 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2013.

本书中文简体字版由 Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

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。

版权所有，侵权必究。

译者序

“过去几十年的大型系统开发就犹如这样一个焦油坑，很多大型和强壮的动物在其中剧烈地挣扎。他们中大多数开发出了可运行的系统——不过，其中只有非常少数的项目满足了目标、时间进度和预算的要求。各种团队，大型的和小型的，庞杂的和精干的，一个接一个淹没在了焦油坑中。”^①传统的瀑布型软件工程方法将软件业等同于传统的大规模制造业，试图回避软件领域固有的复杂性以及变化性，不顾及实际情况而套用一些所谓“银弹”的开发方法，以至于削足适履、南辕北辙，陷入泥潭中而不能自拔。事实上，软件开发属于新产品研发的范畴，你所面对的犹如一片混沌的海洋，需要因地制宜，根据实际情况对各种条件进行权衡与博弈，从而探索出一种行之有效的解决之道。

面对复杂与变化的软件开发，“敢问路在何方？”

UP、XP、水晶、Scrum、DSDM、FDD、ASD 以及精益开发等都给出了相应的答案。然而，“忽如一夜春风来，千树成树梨花开”，一下子涌现出这么多琳琅满目的软件工程方法，让人眼花缭乱，无所适从。需要说明的是，许多标榜自己采用“敏捷与迭代方法”的组织，其实是在追逐时髦，并没有真正领会其要旨，甚至断章取义，陷入教条主义，甚至重蹈瀑布型的覆辙。

本书列举了大量的证据，溯根求源，将各种方法的缘由与适用性娓娓道来，令人茅塞顿开，许多积压在心里的困惑一下子释然。全书共分 12 章：在前 6 章的描述中，作者列举了大量的证据和文献，揭示了瀑布型的症结，以及敏捷与迭代开发的动机与成功案例；第 7 章到第 10 章，则具体讨论了 Scrum、XP、UP、Evo 等 4 种敏捷方法，它们的原理与适用性，以及成功的案例与历史，还在章节后面列出大量的参考文献。这 4 章内容的组织形式相仿，便于读者进行比较学习；第 11 章更具有操作性，介绍了许多行之有效的敏捷与迭代开发的实践技巧；第 12 章是 FAQ，读者可以将它作为独立的章节，对实践中遇到的问题进行查询。

我们还注意到在本书英文版的封面上，是一名芭蕾舞演员。其寓意在于软件开

^① 摘自 Frederick P. Brooks, Jr. 著，UMLChina 翻译组汪颖译，清华大学出版社出版的《人月神话》。

发不仅是一门技术，同时也是一门艺术。作者希望通过他的努力，开发出漂亮的软件来。随便提一句，这本书在亚马逊网站上被评为五星^①。

本书的作者 Craig Larman 先生相信大家不会陌生。他的《UML 和模式应用：面向对象与设计导论》深受中国读者欢迎。Philippe Kruchten 博士（国际著名过程大师，IBM Rational 名士，前 RUP 产品总监）这样评价道：“Craig 是一位杰出的授业者，一位卓越的方法学家，一位对象技术的精神导师(guru)。” Martin Fowler（国际著名 OO 大师，ThoughtWorks 首席科学家）曾说过：“人们常常问我，哪本书是引领他们进入 OO(面向对象)设计壮丽殿堂的最佳著作。自从幸获 Craig 的《UML 和模式应用》，它就成为了我的倾力之荐。”。

阅读 Larman 先生的书可以说一种享受，而翻译他的书则是一种压力。2004 年我有幸接手这部书的翻译工作，并期望向读者呈现原汁原味的译著。翻译过程中，对于书中的一些疑惑之处，Larman 先生给予了热情的帮助和鼓励，并欣然为本书的中文版作序。事隔 8 年，也就是 2012 年，人民邮电出版社期望再版，并请了冯春丽协助进行全文的审校。尽管大家都非常努力，也难免有疏漏之处，希望大家批评指正！此外，我要感谢我在北京魔笛创新科技发展有限公司的团队，并给我试验敏捷方法的机会。最主要的还是要感谢读者朋友，只有你们的认同，才是这项工作最终成功的标志。

张晓坤

2013 年 5 月于北京师范大学辅仁校区

^① 参见：Larman 的个人网站 <http://www.craiglarman.com> 以及亚马逊网站 <http://www.amazon.com/exec/obidos/ASIN/0131111558/qid%3D1063687928/sr%3D11-1/ref%3Dsr%5F11%5F1/103-3958085-8199045>。

前言

谢谢你阅读这本书！我真心希望这本书对你有用——优质信息，易于理解。

此外，访问 www.craiglarman.com 可以找到一些相关的文章和链接。如果你有什么问题，请与我联系，我的电子邮箱是 craig@craiglarman.com。

排版约定

使用楷体排版的是强调的基本观点和需要引起重点关注的地方，我希望读者比较容易看到它，这样，就可以快速浏览，挑选关键思路。

在一句话中排成**黑体**的部分指的是术语。

类似于 [Bob67] 的形式，是指在参考文献中列出的某本书。

作者简介

Craig Larman 是 Valtech 公司的首席科学家。Valtech 公司是一家国际化的咨询和技能转移公司，在欧洲、亚洲和北美洲都设有分支机构。同时，他还在全球范围内担任独立顾问、团队教练和演讲人。

Craig 是 *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* (《UML 和模式应用：面向对象分析与设计导论》) 的作者。此书是 OOA/D 和迭代开发方面全球最畅销的教科书，被译成多种语言，在全球业界和大学中被广泛使用。

Craig 有过一段做街头音乐家的失败经历，从那之后，也就是 20 世纪 70 年代，他开始用 APL 语言、PL/I 语言和第四代语言构建软件。20 世纪 80 年代初期，经过全面的调整之后，他开始对人工智能和知识表示产生了浓厚的兴趣，并用 Lisp 机器、Lisp、Prolog 和 Smalltalk 构建了知识系统。他业余时间还在一支名为 Changing Requirements 的业余乐队（这个乐队曾被称为 Requirements，但有一些乐队成员已经有了变动……）中担任主奏吉他手。

Craig 毕业于加拿大温哥华美丽的西蒙弗雷泽大学 (Simon Fraser University)，获得了计算机科学学士和硕士学位。

致谢

特别感谢我在 Valtech 的朋友和同事，他们都是世界级的迭代开发者，尤其是 Tim Snyder。

非常感谢审读这本书的专家们，包括 Alistair Cockburn、Claudia Frers、Tom Gilb、Jim Highsmith、Ron Jeffries、Philippe Kruchten、Niels Maloteaux、Gary Pollice、Ken Schwaber 和 Jeff Sutherland。

还要感谢 Paul Petralia 和 Patti Guerrieri 两位编辑的引导。

目 录

第1章 概述	1
1.1 软件是新产品开发	2
1.2 后续内容预告	4
1.3 Web 资源	4
1.3.1 主要的链接或文章网站	4
1.3.2 更多的特别网站	5
第2章 迭代和渐进	7
2.1 迭代开发	7
2.2 风险驱动和客户驱动的迭代计划	9
2.3 时间箱迭代开发	10
2.4 迭代期间, 外部利益相关者不能变更迭代内容	11
2.5 渐进开发和自适应开发	11
2.6 渐进需求分析	12
2.7 早期排名前十的高级需求和技能分析	13
2.8 渐进和自适应计划	13
2.9 增量交付	15
2.10 渐进交付	15
2.11 最常见的错误	16
2.12 特定的迭代和渐进方法	16
2.12.1 Evo	16
2.12.2 UP	16
2.12.3 其他方法	17
2.13 后续内容预告	17
2.14 推荐读物	17
第3章 敏捷	18
3.1 敏捷开发	18
3.2 方法的分类	19
3.3 敏捷宣言和原则	20
3.3.1 敏捷宣言	20

3.3.2 敏捷原则	20
3.4 敏捷项目管理	21
3.5 拥抱沟通和反馈	22
3.6 以人为本的编程	22
3.7 简单的实践和项目工具	23
3.8 经验型过程与规定型过程	23
3.9 基于原则与基于规则	24
3.10 可持续规程——人员接触	24
3.11 团队是一个复杂的自适应系统	24
3.12 敏捷是在夸大其词吗	25
3.13 特定的敏捷方法	25
3.13.1 Scrum	26
3.13.2 XP	26
3.13.3 Crystal 方法	26
3.13.4 敏捷建模	27
3.13.5 其他方法和实践	28
3.14 后续内容预告	28
3.15 推荐读物	29
第4章 故事	30
第5章 动机	35
5.1 如果问题不是太多，就不要修复它	35
5.2 软件项目中变化的事实	36
5.3 迭代开发的关键动机	37
5.4 迭代地迎接需求挑战	39
5.5 瀑布型的问题	41
5.5.1 问题：“完成”前期的需求规格说明就万事大吉	43
5.5.2 问题：迟到的集成和测试	43
5.5.3 问题：“可靠的”前期进度和估算	43
5.5.4 问题：“计划工作，按计划工作”的价值	44
5.6 后续内容预告	44
第6章 证据	45
6.1 概要	46
6.2 研究的证据	47
6.2.1 迭代与渐进研究	47
6.2.2 规模研究	50
6.2.3 有关变化的研究	51

6.2.4	瀑布型失败的研究	53
6.2.5	生产力的研究	54
6.2.6	质量和缺陷的研究	55
6.3	早期历史项目的证据	56
6.3.1	1970 年以前	57
6.3.2	20 世纪 70 年代	58
6.3.3	20 世纪 80 年代之后	61
6.4	标准团体的证据	62
6.5	专家和思想领袖的证据	65
6.5.1	Harlan Mills	66
6.5.2	Tom Gilb	66
6.5.3	Frederick Brooks	67
6.5.4	Barry Boehm	68
6.5.5	James Martin	68
6.5.6	Tom DeMarco	69
6.5.7	Ed Yourdon	70
6.6	迭代开发的商业案例	70
6.7	瀑布型有效是历史的偶然	72
6.8	后续内容预告	75
6.9	推荐读物	75
第 7 章	Scrum	76
7.1	方法概览	77
7.1.1	分类	77
7.1.2	介绍	78
7.2	生命周期	79
7.3	工件、角色和实践	80
7.3.1	角色	81
7.3.2	实践	82
7.3.3	核心实践	82
7.3.4	Scrum 会议：细节	84
7.3.5	Scrum 会议的价值	85
7.3.6	工件	86
7.3.7	其他实践和价值观	88
7.4	价值观	88
7.5	常见错误和误解	89
7.5.1	Scrum 项目失败是怎么回事	89

7.5.2	什么时候你知道自己并不了解 Scrum	90
7.6	样板项目	90
7.7	过程混合	91
7.7.1	Scrum + Evo	91
7.7.2	Scrum + UP	91
7.7.3	Scrum + XP	92
7.8	采用的策略	92
7.9	现实与幻想	93
7.10	优势与其他	93
7.10.1	优势	93
7.10.2	其他	94
7.11	历史	94
7.12	后续内容预告	95
7.13	推荐读物	95
第 8 章	极限编程	96
8.1	方法概览	96
8.1.1	分类	96
8.1.2	介绍	98
8.2	生命周期	100
8.3	工件、角色和实践	101
8.3.1	角色	102
8.3.2	实践	103
8.3.3	核心实践	104
8.3.4	工件	106
8.3.5	其他实践和价值观	106
8.4	价值观	108
8.5	常见错误和误解	109
8.5.1	极限编程项目失败是怎么回事	109
8.5.2	什么时候你知道自己并不了解 XP	112
8.6	样板项目	112
8.7	过程混合	113
8.7.1	XP + Evo	113
8.7.2	XP + Scrum	113
8.7.3	XP + UP	114
8.8	采用的策略	114
8.9	现实与幻想	116

8.10 优势与其他	116
8.10.1 优势	116
8.10.2 其他	117
8.11 历史	118
8.12 后续内容预告	118
8.13 推荐读物	119
第9章 统一过程	120
9.1 方法概览	121
9.1.1 UP	121
9.1.2 介绍	122
9.2 生命周期	125
9.3 工件、角色和实践	128
9.3.1 角色	129
9.3.2 实践	130
9.3.3 核心实践	130
9.3.4 UP 指南	131
9.3.5 6 个最佳实践	131
9.3.6 工件	132
9.3.7 其他实践和价值观	133
9.4 价值观	133
9.5 常见错误和误解	135
9.5.1 统一过程失败是怎么回事	135
9.5.2 双重瀑布思想	135
9.5.3 其他的常见错误	136
9.5.4 什么时候你知道自己并不了解 UP	137
9.5.5 UP“专家”不是迭代专家	138
9.6 样板项目	138
9.7 过程混合	139
9.7.1 UP + Evo	139
9.7.2 UP + Scrum	140
9.7.3 UP + XP	140
9.8 采用的策略	141
9.9 现实与幻想	142
9.10 优势与其他	142
9.10.1 优势	142
9.10.2 其他	143

9.11 历史	143
9.12 后续内容预告	144
9.13 推荐读物	144
第10章 Evo	146
10.1 方法概览	146
10.1.1 分类	146
10.1.2 介绍	148
10.2 生命周期	150
10.3 工件、角色和实践	152
10.3.1 角色	153
10.3.2 实践	154
10.3.3 核心实践	154
10.3.4 工件	160
10.3.5 其他实践和价值观	163
10.4 价值观	164
10.5 常见错误和误解	164
10.6 样板项目	165
10.7 过程混合	165
10.7.1 Evo + Scrum	166
10.7.2 Evo + UP	166
10.7.3 Evo + XP	166
10.8 采用的策略	167
10.9 现实与幻想	167
10.10 优势与其他	167
10.10.1 优势	167
10.10.2 其他	168
10.11 历史	168
10.12 后续内容预告	169
10.13 推荐读物	169
第11章 实践技巧	170
11.1 项目管理	170
11.1.1 多团队或多现场的早期开发	171
11.1.2 艰难的多团队或多现场迭代计划	171
11.1.3 跨多个迭代的重叠或者“流水线”活动	172
11.1.4 滚动波浪式自适应计划与预见性计划	174
11.1.5 计划：考虑在“星期三”结束	177

11.1.6	计划: 整个团队的计划会议	178
11.1.7	计划: 工作者评估	178
11.1.8	计划: 用 Wideband Delphi 改善评估	178
11.1.9	计划: 多次迭代计划会议	179
11.1.10	计划: 敏捷的任务制定	180
11.1.11	计划: 不要忘记为迭代工作的开销做预算	180
11.1.12	任务: 每次迭代都要进行时间估算	180
11.1.13	计划: 自愿	180
11.1.14	可视化的项目计划	181
11.1.15	迭代目标: 风险、覆盖面、紧迫性、技能性开发	181
11.1.16	迭代目标: 将什么分级	181
11.1.17	迭代目标: 如何分级? 打点投票	182
11.1.18	迭代目标: 如何分级? 定量方法	182
11.1.19	迭代目标: 相关的迭代长度	183
11.1.20	迭代目标: 在第一次开发迭代之前	184
11.1.21	迭代目标: 第一次开发迭代	184
11.1.22	迭代目标: 用例和场景	184
11.1.23	迭代目标: 主要需求和次要需求	185
11.1.24	迭代目标: 不要向一个迭代中添加需求	185
11.1.25	跟踪迭代进度	186
11.1.26	跟踪迭代过程——跟踪什么	186
11.1.27	跟踪和计划: XPlanner	187
11.1.28	IID 项目中的挣值跟踪	187
11.1.29	对风险评级	187
11.1.30	管理风险	188
11.2	环境	188
11.2.1	持续集成	188
11.2.2	项目的 Wiki 网	190
11.2.3	CASE 工具和逆向工程	190
11.2.4	考虑使用绘图仪	191
11.2.5	单间和公共工作室	191
11.2.6	解放墙壁	191
11.2.7	贴纸或者白板漆	192
11.2.8	数码照相机	192
11.3	需求	192

11.3.1	敏捷建模	193
11.3.2	定义并遵循愿景规划	193
11.3.3	产品愿景箱	193
11.3.4	Moore 风格的愿景声明	194
11.3.5	产品表单	194
11.3.6	渐进的需求研讨	194
11.3.7	跨迭代跟踪需求	196
11.3.8	直接用户介入需求和产品	196
11.3.9	可以使用用例	196
11.3.10	可以量化	197
11.3.11	将 GUI 粘在一起	197
11.3.12	头脑风暴	198
11.3.13	灵感记录	198
11.3.14	类聚	198
11.3.15	思维导图	198
11.3.16	团队轮流编写	199
11.4	测试	199
11.4.1	测试驱动开发	199
11.4.2	用于验收测试的 Fit 和 Fitness	201
第 12 章	常见问题解答	202
12.1	问题清单	202
12.2	问题和解答	204
参考文献		222

第 1 章

概述

逻辑是一种用信心面对错误的艺术。

——约瑟夫·伍德·克鲁奇 (Joseph Wood Krutch)

概述

- 本书中有哪些内容?
- 预见性开发与新产品开发。

本书介绍了迭代 (iterative) 和敏捷 (agile) 方法, 你能从中获得哪些有用的知识呢?

第一, 你会知道 4 种著名方法的关键实践: **Scrum**、**极限编程** (Extreme Programming, XP)、**统一过程** (Unified Process, UP) 和 **Evo** (一种早期的迭代方法)。本书提供了一些软件开发方法的思路和实践的总结, 对于使用开发方法的管理人员、开发人员以及学生来说, 每一章都会有一些有用的内容。(Scrum 参见第 7 章, 极限编程参见第 8 章, 统一过程参见第 9 章, Evo 参见第 10 章)。

第二, 因为本书能够提供很好的帮助, 所以可以缩短你的学习曲线。阐述这 4 种方法的章节都具有相同的结构, 便于快速理解与比较。此外, 还有一章 FAQ (见第 12 章) 和一章常见的实践技巧 (见第 11 章)。

第三, 你将了解到敏捷与迭代开发的动机 (见第 5 章) 和证据 (见第 6 章)。有些组织重视迭代开发的价值, 但是有一些组织则不以为然。如果你需要为一次迭代项目实验找到充分的理由, 那么你会发现本书提供了迭代开发的关键原因、研究成果、大型项目示例、标准团体认可、商业案例, 以及著名编程思想领袖们几十年以来的推动历程。研究成果与历史阐述部分也适合学习软件工程方法的学生进行研读。