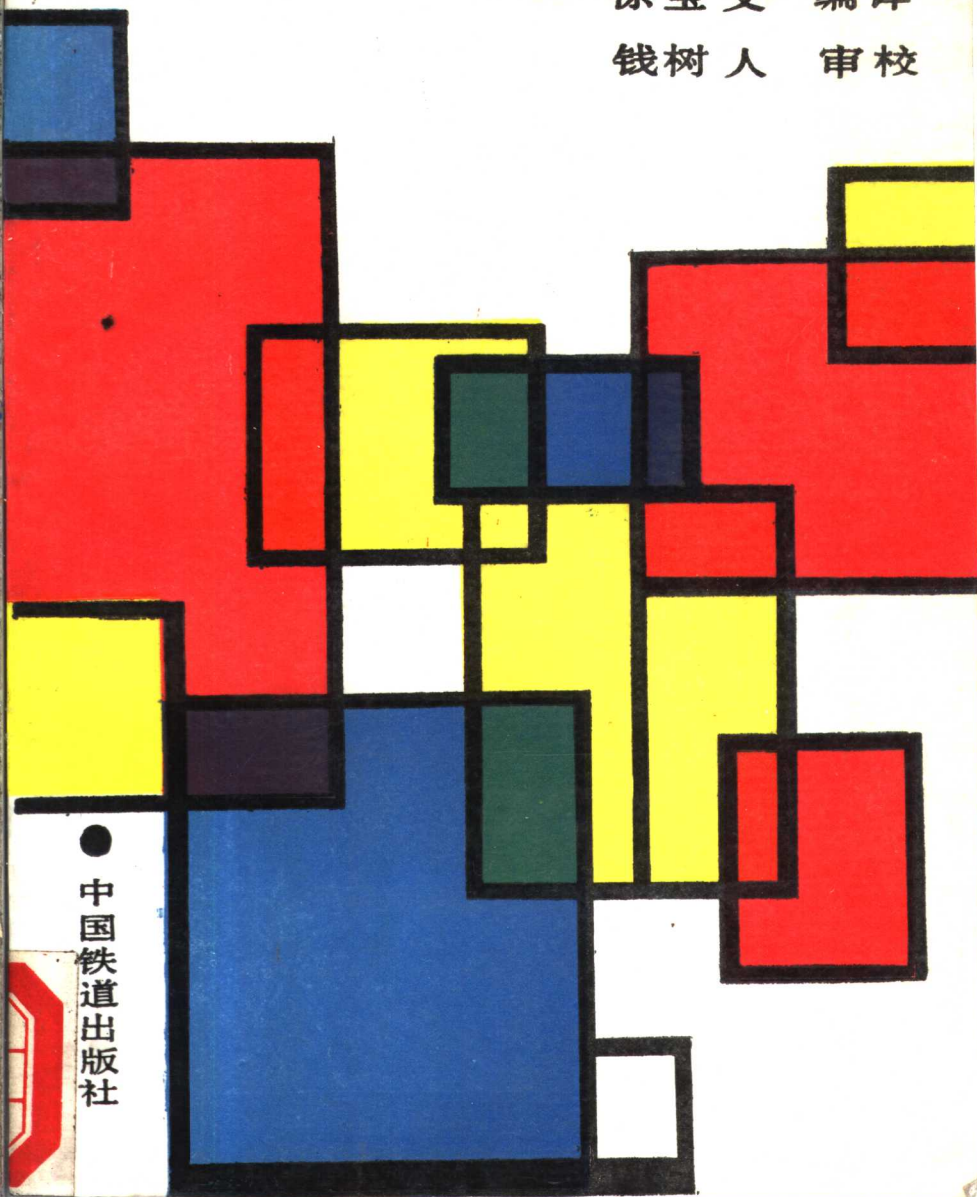


实用软件开发手册

徐宝文 编译

钱树人 审校

●
中国铁道出版社



实用软件开发手册

徐宝文 编译

钱树人 审校

中国铁道出版社

1990年·北京

内 容 简 介

本书是国内第一本介绍软件开发过程、方法、技术与工具的工具书，内容涉及软件开发各阶段数十种比较有名的软件开发方法、技术以及若干有关的工具。本书没有一一罗列它们，而是详细描述它们在软件开发各阶段的应用方法与过程。对软件开发每一阶段，本书首先指出该阶段的任务、目标与要求，然后针对它们介绍各种方法与技术的应用，最后给出十分详尽的用于检查该阶段任务完成情况的检查清单。

本书内容清楚、层次分明，既有一定的理论性，又有较高的实用价值，可作为大学有关专业高年级本科生和研究生的教材与主要教学参考书，也可供从事软件开发方法技术研究与从事各类实际软件开发工作的人员参考、使用。

实用软件开发手册

徐宝文 编译

钱树人 审校

中国铁道出版社出版、发行

责任编辑 郭 宇 封面设计 刘景山

各地新华书店经销

中国铁道出版社印刷厂印

开本：787×1092毫米 $\frac{1}{32}$ 印张：16 字数：364千

1990年10月第1版 第1次印刷

印数：0001-3100册 定价：7.60元

ISBN7-113-00812-7/TP·81

前言

软件开发方法与技术是现代软件开发理论与实践的一个重要组成部分。几十年来的软件理论研究与实践经验都表明，程序设计（或广义地讲，软件开发）不只是技巧问题，虽然程序质量与开发者的开发经验与技巧有一定的关系，但要开发一个高质量的程序，光有经验与技巧是远远不够的，还需要正确地掌握与运用合适的软件开发方法与技术。这点已受到越来越多的软件理论与实际工作者重视，软件方法论已被公认为软件工程的两大支柱之一。美国的 STARS 计划、英国的 Alvey 计划，日本的 Σ 计划等都把软件方法论当作有关研究的重要组成部分。

目前，国内一些从事软件方法论研究的同仁为介绍推广各种软件开发方法做了不少工作，发表了一定数量的文章并在有关书籍中介绍了一些比较有名的方法与技术，如结构程序设计、SA、SD、SADT、JSD、PSL/PSA、HOS 等。但是，国内尚未出版过一本软件开发方法与技术手册，有感于此，在有关同志的关心与帮助下，我们编译了这本书。

本书是一本介绍软件开发过程、方法、技术与工具的工具书，同时也可作为具有一定层次的学生的教材或教学参考书。本书内容涉及项目开发生存期中项目初始、系统定义、系统设计、系统生成、系统验收与验收后开发以及项目汇报等各个阶段与步骤中数十种比较有名的软件开发方法与技术以及若干有关的工具，本书没有简单地罗列各种方法与技术，而是详细描述了各有关方法与技术软件开发各阶段

(步骤)的应用方法与过程。对每一阶段，本书首先指出该阶段的任务、目标与要求，然后针对它们介绍各种方法与技术的应用，最后给出十分详尽的用于检查该阶段任务完成情况的检查清单。

由于各种新的方法、技术不断地出现，本书不可能把它们都罗列进去。有些方法（如OOD），虽然问世已有一段时间，但尚处于发展、完善阶段，本书也未作介绍。

本书是在 N. D. Birrell 等的著作（见参考文献[Birrell 1985]）的基础上编译而成的，南京大学钱树人副教授在为本书审校中作了大量的工作，中国铁道出版社编辑部郭宇老师也为此书的出版费了很多心血，我所在的南京航空学院计算机科学与工程系的有关领导与同事也给予了关心与帮助，此外，本书在编译、抄写过程中还得到了其他一些同志的帮助。可以说，没有他们的支持、关心与帮助，本书是难以问世的，为此，谨向他们致以深深的谢意。

徐宝文

1988年秋于南京鸡鸣山庄

目 录

1	引言	1
2	软件开发过程	4
2.1	阶段与步骤	7
2.1.1	阶段的描述——可交付性	8
2.1.2	阶段轮廓	10
2.1.3	项目初始阶段	10
2.1.4	系统定义阶段	11
2.1.5	系统设计阶段	12
2.1.6	系统生成阶段	13
2.1.7	系统验收阶段	14
2.1.8	验收后开发阶段	15
2.1.9	项目汇报阶段	16
2.1.10	工作任务分解结构	16
2.1.11	原型制作	21
2.2	定量软件开发模型	23
2.2.1	引语	23
2.2.2	估算途径	25
2.2.3	算法估算方法	30
2.2.4	各阶段工作量分配	32
2.2.5	产生开发计划进度表	33
2.2.6	可靠性的量化	34
2.2.7	历史数据的使用	36
2.3	示踪器	37

2.3.1	为什么需要示踪器?	37
2.3.2	示踪器的管理	38
2.4	配置管理	40
2.4.1	问题的引入	40
2.4.2	基线	41
2.4.3	可交付项的发布	43
2.4.4	配置控制机制	45
3	开发技术	48
3.1	引语	48
3.1.1	技术特性与问题特性的匹配	53
3.1.2	项目特性检查清单	55
3.1.3	问题特性检查清单	56
3.1.4	技术特性检查清单	58
3.1.5	关于技术与工具的资料来源	59
3.2	成本估算模型	61
3.2.1	IBM 模型	61
3.2.2	SLIM 模型	61
3.2.3	PRICE-S 模型	62
3.2.4	COCOMO 模型	62
3.2.5	Bailey—Basili 元模型	63
3.2.6	Schneider 模型	63
3.3	结构英语	64
3.4	PSL / PSA	64
3.5	结构分析——SA	65
3.6	结构分析与设计技术——SADT	67
3.7	受控需求表示式——CORE	68
3.8	软件需求工程方法——SREM	69
3.9	有限状态机——FSM	72

3.10	Petri 网	73
3.11	Jackson 系统开发——JSD	74
3.12	软件开发系统——SDS / RSRE.....	76
3.13	结构设计——SD	77
3.14	Jackson 结构程序设计——JSP	78
3.15	系统设计师的助手	79
3.16	MASCOT	80
3.17	形式开发方法——FDM	82
3.18	层次开发方法——HDM.....	83
3.19	高级软件——HOS	85
3.20	结构程序设计	86
3.21	数据控制	87
3.22	编码技术	88
3.23	单元测试技术	89
3.24	系统综合	90
3.25	结构走查	91
3.26	Fagan 审查法	92
4	项目初始	93
4.1	项目初始阶段目标	93
4.2	初始阶段起点	94
4.3	初始阶段终点	95
4.4	项目初始期间可用技术	98
4.5	预测	98
4.5.1	招标评估	98
4.5.2	招标准备	98
4.6	项目初始阶段检查清单	99
4.7	计划初始期间要点	101
4.7.1	成败记录	101

4.7.2	如果开始没有成功.....	102
4.7.3	全面协商——成功的关键	102
4.8	典型样例——VISTA	103
4.8.1	引言	103
4.8.2	图象处理背景	104
4.8.3	VISTA 系统功能与硬件配置	106
5	系统定义	110
5.1	系统定义阶段的目标	110
5.2	系统定义起点	110
5.3	系统定义终点	111
5.3.1	功能规格说明	111
5.3.2	项目计划	112
5.3.3	质量管理计划	113
5.3.4	验收测试规格说明	115
5.4	系统定义期间可用技术	115
5.4.1	系统定义技术分类	117
5.4.2	结构英语	119
5.4.3	PSL / PSA	123
5.4.4	结构分析——SA	127
5.4.5	结构分析与设计工具——SADT.....	134
5.4.6	受控需求表示式——CORE.....	144
5.4.7	软件需求工程方法——SREM	153
5.4.8	有限状态机——PSM	167
5.4.9	Petri 网	174
5.4.10	Jackson 系统开发——JSD	184
5.4.11	软件开发系统——SDS / RSRE	192
5.4.12	结构走查	198
5.5	系统定义期间监督管理的示踪器	201

5.5.1	源代码大小估算	201
5.5.2	系统大小示踪器	202
5.5.3	需求规格说明质量示踪器	203
5.6	系统定义阶段检查清单	205
5.6.1	功能规格说明的质量准则	205
5.6.2	功能规格说明的内容	207
5.6.3	验收测试实施方法	212
5.6.4	系统定义与可维护性	212
5.6.5	质量管理计划内容	213
5.7	系统定义阶段要点	215
5.7.1	系统定义阶段人员配备	215
5.7.2	信息流维护	215
5.7.3	保持质量目标	216
5.7.4	为各种对象选取合适的名字	216
5.7.5	确定较低层的功能	216
5.7.6	限制功能规格说明范围	216
5.7.7	更新项目汇报报告	217
5.7.8	记录定义过程	217
5.7.9	将系统中的设施通用化	218
5.7.10	英语非常适用于功能规格说明吗?.....	218
5.7.11	对模型的期望	218
5.7.12	Boehm 关于软件工程的七条基本原理	219
5.7.13	编制模拟程序前的计划	220
5.7.14	与标准程序包的接口	220
5.7.15	错误出现得越早发现得越晚	221
5.7.16	你的系统能用吗?.....	221
6	系统设计	222

6.1	系统设计阶段目的	222
6.1.1	引言	222
6.1.2	易实现性	222
6.1.3	功能完整性	224
6.1.4	遵循限制	224
6.1.5	可维护性	225
6.2	系统设计起点	227
6.3	系统设计终点	228
6.3.1	结构	229
6.3.2	细节	229
6.3.3	有效性	230
6.3.4	可交付项	230
6.4	系统设计期间所用技术	230
6.4.1	结构设计——SD	233
6.4.2	Jackson 结构程序设计——JSP	242
6.4.3	Jackson 系统开发——JSD	254
6.4.4	有限状态机——FSM	262
6.4.5	Petri 网	265
6.4.6	系统设计师的助手——SARA	269
6.4.7	MASCOT	280
6.4.8	形式开发方法——FDM	288
6.4.9	层次开发方法——HDM	296
6.4.10	高级软件——HOS	307
6.4.11	结构走查	317
6.4.12	Fagan 审查法	319
6.5	系统设计阶段监督管理的示踪器	321
6.5.1	目标	321
6.5.2	互连性变量	322

6.5.3	Myers 模块耦合度度量	323
6.5.4	Myers 模块内聚度度量	324
6.5.5	调用图度量	325
6.5.6	错误检测度量	326
6.5.7	错误校正时间变量	327
6.5.8	文件大小度量	328
6.5.9	源代码长度估算	328
6.5.10	系统大小示踪器	329
6.6	系统设计阶段检查清单	329
6.6.1	设计过程中的课题	329
6.6.2	设计文件的一般性质	331
6.6.3	系统设计与可维护性	332
6.6.4	潜在设计错误类型	334
6.7	系统设计要点	335
6.7.1	增加项目汇报报告内容	335
6.7.2	设计目标定量化	335
6.7.3	明显错误的消除	335
6.7.4	暴露错误	336
6.7.5	系统情况	336
6.7.6	初步测试	336
6.7.7	人尽其用	336
6.7.8	随时向委托人通报情况	336
6.7.9	草案设计工具	337
7	系统生成	338
7.1	系统生成阶段目标	338
7.1.1	系统生成背景	338
7.1.2	完善总体设计	340
7.1.3	将正确设计变换成正确代码	340

7.1.4	代码变换的正确性检查	341
7.1.5	系统综合	342
7.1.6	系统生成期间的辅助步骤	343
7.2	系统生成起点	344
7.3	系统生成终点	345
7.4	系统生成期间所用技术	345
7.4.1	Jackson 结构程序设计——JSP	347
7.4.2	Jackson 系统开发——JSD	349
7.4.3	有限状态机——FSM	355
7.4.4	系统设计师的助手——SARA	356
7.4.5	MASCOT	357
7.4.6	层次开发方法——HDM	364
7.4.7	结构程序设计	367
7.4.8	数据控制	375
7.4.9	程序设计语言	384
7.4.10	宏处理程序	387
7.4.11	程序库与可重用性	389
7.4.12	走查与审查	392
7.4.13	静态分析	394
7.4.14	符号执行	396
7.4.15	断言检查	399
7.4.16	测试数据选择	401
7.4.17	测试覆盖范围分析与软件度量	406
7.4.18	开发环境	408
7.4.19	系统综合技术	416
7.4.20	系统开发人员心理学	422
7.5	系统生成阶段监督管理的示踪器	427
7.5.1	详细设计示踪器	427

7.5.2	编码示踪器	428
7.5.3	测试示踪器	434
7.6	系统生成阶段检查清单	435
7.6.1	详细设计检查清单	436
7.6.2	编码检查清单	437
7.6.3	单元测试检查清单	438
7.6.4	用户文件检查清单	439
7.6.5	系统转换进度表检查清单	440
7.6.6	培训进度表检查清单	441
7.6.7	验收测试说明书检查清单	442
7.6.8	潜在编码错误类型	445
7.7	生成阶段要点	446
7.7.1	维护信息流	446
7.7.2	生成阶段的人员配备	446
7.7.3	系统设计规格说明的控制变化	446
7.7.4	估算测试时间	447
7.7.5	采用与改编	447
7.7.6	增加 PDF 内容	448
7.7.7	扔掉坏苹果	448
7.7.8	人工错误播种	448
7.7.9	系统的独立测试	449
7.7.10	保持软件在测试期间的静态性	449
7.7.11	程序设计风格	449
8	系统验收与验收后开发	451
8.1	系统验收目标	451
8.2	系统验收起点	452
8.3	系统验收终点	452
8.4	验收测试的实施	453

8.5	验收后开发的管理	453
8.5.1	为变化作准备	453
8.5.2	变化的监督管理与评估	454
8.5.3	变化的规格说明、设计与实现	456
9	项目汇报	458
9.1	项目汇报的目的	458
9.2	项目汇报的结构	459
9.3	项目汇报的内容	460
9.3.1	项目说明书	460
9.3.2	项目属性历史	460
9.3.3	示踪器历史	461
9.3.4	技术与系统评价	461
9.3.5	潜在可重用项	461
9.3.6	补遗	462
9.3.7	关键词摘要	462
9.4	项目汇报报告检查清单	462
9.5	项目汇报报告样例	463
	参考文献	472
	英汉名词对照	482

1 引言

如果给我们一个开发软件系统的任务，我们怎样去完成它？一开始就建立一个有一千行甚至一百万行源代码的系统会使人们对前景不寒而栗(即使是激动人心的)。我们绝不应该在还没有非常清楚地计划出开发工作如何进行以及其产品质量如何评估的情况下就开始工作。

把这一方法改变一下，我们可以说，不管是开发何种规模软件的软件开发人员都必须确立开发软件的方法——一种或多种通过仔细地综合与控制使次序与方向进入生产过程的技术。

每一个软件开发组织都有某种建立软件系统的方法。但是，按现代软件开发实践开发某种软件时，其中绝大部分仍以特定的方式建立。因此，最好根据当前实践的变化来讨论软件开发的技术与方法，用那些能改进软件开发过程、提高最终产品质量的新技术取代它们。

作为系统开发人员，我们在这里自然会进退维谷：X技术有可能帮助我们减小开发时间、降低成本、提高可靠性与质量等，但是，如果采用它，我们也会冒着因不熟悉X技术所带来的危险，它可能不适合于我们要开发的系统，我们的工作人员及其技能也可能与之不相适应。但是，如果我们要提高开发效率，提高产品质量，那么必须摆脱这一困境。

我们必须经常提出的问题是：有些什么技术？哪些技术适合于我们的问题？我们怎样才能监督管理产品质量？正如飞机制造商总在不断地寻找新材料与新的生产过程一样，作为软件系统制造人员，我们应不断地寻找、不断地选择、不断地改进新技术与新方法。

本书的第一个任务是：考察你可以用在开发过程中的各种不同技术（及某些例子），作出符合你自己情况的选择，作出果断的判断来支持你所做的选择。

第二个任务是：建立一个管理体系，在这个体系中，你可以用可控的方式使用你所选择的技术，这是必要的。你能确定你将如何用明确的里程碑去标记进度。

第三个任务是：技术开发管理要求不仅仅要计划并管理传统的可管理示踪器——预算与时间量。技术示踪器（特别是各种复杂性因素）必须要在开发的所有阶段都得到管理，以便保证用正确的方式来使用正确的技术。

所以，为了改进软件开发，既要仔细选择合适的技术，也要适当的开发体系与可管理的技术示踪器。

但是，引入新技术不仅仅是一个技术问题，它也有社会影响，也需要软件工程师们改变他们的工作习惯。更新、更形式化、更机械的技术可能会使人觉得软件工程师们的技能失去价值。对此，Michael Jackson 认为：“关于开发方法对技能的影响人们仍存在着误解。如果开发方法肯定有某种优点，则它必定会提高你的技能，它必定使你能做在此之前不能做的事，它不仅不会降低工作技能反而会使技能得到提高。”我们相信，即使把下面几章浏览一遍，持怀疑态度的软件工程师也会相信他们的技能与能力得到了很大的提高。

我们在各个部分都为读者准备了练习题，这些习题既不是理解或记忆性测验，也没有诸如“42”这样简单的答案，它