

21世纪

高等院校计算机系列教材

软件过程管理

龚 波 主 编

刘宝军 刘卫宏 刘宪军 副主编



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机系列教材

软件过程管理

龚 波 主编

刘宝军 刘卫宏 刘宪军 副主编

中国水利水电出版社

内 容 提 要

本书比较系统地讨论了 CMM、TSP、PSP 等模型的基本材料。本书包含 10 章和 3 个附录。第 1 章介绍了传统软件工程。第 2 章介绍软件能力成熟度模型的基本思想和结构。第 3 章讨论了软件过程改进和实践。第 4 章讨论了团队软件过程的原则、流程和质量管。第 5 章讨论了软件能力成熟度模型集成 CMMI。在附录部分, 提供本书涉及的术语解释和 CMM 评估提问单。

本书全面讨论了软件组织过程改进和实施中用到的最基础知识和方法。本书可作为高等院校计算机专业及相关专业工程专业的教师、项目经理、系统和软件工程师以及过程改进实践人员、工程专业的高年级学生、研究生和老师使用。

图书在版编目 (CIP) 数据

软件过程管理 / 龚波主编. —北京: 中国水利水电出版社, 2003
(21 世纪高等院校计算机系列教材)

ISBN 7-5084-1550-7

I. 软… II. 龚… III. 软件工程—高等学校—教材

中国版本图书馆 CIP 数据核字 (2003) 第 043116 号

书 名	软件过程管理
主 编	龚波
副 主 编	刘宝军 刘卫宏 刘先军
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 1 层) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销) 全国各地新华书店和相关出版物销售网点
经 售	
排 版	北京万水电子信息有限公司
印 刷	北京北医印刷厂
规 格	787×1092 毫米 16 开本 20.625 印张 46
版 次	2003 年 6 月第一版 2003 年 6 月北京第一次印
印 数	0001—5000 册
定 价	29.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社负责调换。

版权所有·侵权必究

前 言

在计算机渗透人类生活的几十年来，一直到现在仍然困扰着绝大多数软件开发和管理机构的问题是：工程组织怎样开发符合预算和进度要求的高可靠性和可用性软件。

许多年以来，工程开发和研究机构一直在寻找解决这个问题的答案。在通过提供新方法和新技术使得软件生产率和质量得到提高的许诺落空了 20 年以后，软件机构意识到实质性是缺乏管理软件过程的能力。在许多机构中，软件项目常常严重超期或超出预算，处于混乱状态，根本不能体现新方法、新技术和新工具所带来的优势。

一个研究小组在分析“软件危机”时写道，“很少有这样的领域，在理想与实践之间有如此巨大的鸿沟”，并进一步得出结论，“软件开发中当前最主要的问题不是技术问题，而是管理问题”。

现在，与软件过程管理相关的研究项目在逐年增加，希望能够有充足的手段以改进软件开发过程，实现开发过程的可控性，将软件失效限制在萌芽中。软件过程是指软件开发人员开发和维护软件及相关产品（如项目计划、设计文档、代码、测试用例和用户手册）的一套行为、方法、实践及变换过程。

现在业界最成熟的软件过程化评估和改进方法是美国卡内基·梅隆大学的软件工程研究所（SEI）提出的软件过程能力成熟度模型。该模型描述了有效的软件过程单元的框架，为软件机构描述了从混乱的、不成熟的软件过程向成熟的、有纪律的软件过程改进的一条途径。软件过程成熟度指针对具体的软件过程进行明确定义、管理、度量、控制的程度。成熟度表明软件工程能力增长的潜力，表明机构软件过程的丰富性及其应用于项目的一致性。软件过程成熟度意味着必须改进软件过程能力。改进软件过程要求软件机构加强管理机制，并能长期、始终地关注软件过程。对于成熟的软件机构，需要机构的基础设施和机构文化支持软件开发中使用的方法、实践和规程，这样在最初定义的人员离开之后，仍能持续进行。

中国工程院院士何新贵教授领导的一个小组根据中国的国情，提出了 CSCMM，并在某大型工程中付诸实施。东大阿尔派、联想、鼎新、亚信等国内企业的部分部门也通过了 CMM 一定级别的认证。2000 年 6 月国务院颁发了“鼓励软件产业和集成电路产业发展的若干政策”，该文件的第五章第十七条明确提出“鼓励软件出口型企业通过 GB/T19000—ISO9000 系列质量保证体系认证和 CMM（能力成熟度模型）认证。其认证费用通过中央外贸发展基金适当予以支持。”从此可见我国科研机构、企业和政府对软件过程化改进的态度和认识。

个体软件过程（PSP）是为帮助软件工程师完成出色的工作而设计的。它告诉工程师如何把这些先进的工程方法应用到日常的任务中去。它提供详细的估计和计划方法，告诉他们如何按照计划去跟踪任务的执行情况，向他们详细解释如何利用已定义的过程来指导

他们的工作。遵循个体软件过程 PSP 开发软件不是一件容易的事,需要人们改变自己的工作习惯。PSP 可以帮助学生认识到规范化方法对开发软件的重要性,也为将来进一步介绍高级个体软件和群组过程奠定更严谨的基础。

团队软件过程(TSP)提供了开发过程、产品和小组协同工作之间平衡的重点,并且在规划和管理软件工程中利用了广泛的工业经验基础。TSP 指导你逐步通过团队软件过程的各个步骤,阐述在协同软件开发环境中如何应用软件工程原理。严格遵守 TSP 七条主要的设计原则:提供 PSP 基础之上的简单框架;把产品研发划分成多个周期;建立标准的质量和效率评测机制;提供关于小组和组员的准确评价;采用角色和小组评估;开发过程中需要纪律;提供关于团队协同工作的问题指导。

在软件能力成熟度模型的 18 个关键过程域中,有 12 个与个体软件过程 PSP 紧密相关,16 个与团队软件过程 TSP 紧密相关。3 者是软件过程改进共同的主体,3 者互相配合,各有侧重,形成一个不可分割的整体,以共同提高软件过程化管理水平,增强软件工作产品质量。

全书从内容组织方面可以分为 3 部分:第一部分是软件工程基础,引出了软件工程的观念,并从软件计划、需求分析、分析建模、软件设计、软件编码、软件测试和软件维护等软件整个生命期角度介绍各个开发阶段所实施的传统开发方法。

第二部分是软件过程管理。介绍了软件过程管理的概念和最新的软件管理理论。本部分主要从 CMM 模型,以及个体软件过程(PSP)和团队软件过程(TSP)等 3 个方面介绍软件过程化实践过程、角色,以及个人和团队在软件过程中的管理和实践行为。本部分的最后一章介绍了 CMMI 模型及其最新发展。

第三部分是附录。本部分包括术语表、CMM 评估过程介绍、CMM 成熟度提问单等。“术语表”部分给出了本书所涉及基本概念的简单解释。“CMM 评估过程介绍”简单介绍当前 CMM 成熟度基本评估流程。CMM 评估中一项非常重要的过程是参与人员填写针对本组织的不同软件处理过程,也就是 CMM 成熟度提问单,本附录给出常见的提问单样式。

本书是作者多年从事软件工程教学和研究的总结,以装备指挥技术学院的研究生课程《软件过程管理》和《软件质量管理》为模板。在教学过程中,得到专家和领导的好评。经验证明,这对培养学生成为合格的软件专业人员很有帮助。每章前面都有本章目标,介绍本章所讲授的重点知识;每章最后有习题,借助于习题读者可以进一步掌握该章的知识点。

本书实用性和可读性较强,不仅可作为高等院校计算机专业本科以上软件工程课程的教材,也可作为工程技术人员自学材料,是软件机构进行软件过程改进和能力成熟度模型评估的参考资料。本书同样适用于软件开发项目经理、程序员和一般编程爱好者在开发软件时参考。

本书是集体合作的结果。第 1 章由刘宪军编写;第 2 章到第 9 章由刘卫宏编写;龚波负责全书的策划、统稿和审校,并负责第 10 章的编写;田丽韞同志负责本书的资料整理和收集;附录由刘卫宏和龚波共同整理。工程院院士、北京大学计算机学院院长何新贵老师

在百忙之中通读了本书稿，对本书组织结构和内容等提出很有价值的意见。其他参与本书策划、评审、编写、录入、审校和排版的还有：刘向军、龚皓天、张文敏、田军、赵军锁、田蕴哲、牛志奇、丁天、龚志翔、李红玲、白红利、金荣学、薛彪、叶哲等。最后，何新贵老师对本书稿进行了审核，并提出了许多宝贵的意见。

龚波

2003年2月于北京

目 录

前言

第 1 章 软件工程基础	1
1.1 软件工程的概​​念	1
1.2 软件策划	3
1.2.1 问题描述	3
1.2.2 可行性分析	3
1.2.3 软件策划的内容	4
1.2.4 软件需求描述	5
1.3 需求分析	6
1.4 分析建模	7
1.4.1 数据流图与数据字典	7
1.4.2 结构化分析方法	10
1.4.3 快速原型法	13
1.5 软件设计	14
1.5.1 模块化设计原则	15
1.5.2 总体设计	18
1.5.3 详细设计	20
1.5.4 结构化设计方法	22
1.5.5 其他设计方法	24
1.6 软件编码	26
1.6.1 结构化程序设计	26
1.6.2 编码风格	27
1.7 软件测试	29
1.7.1 软件测试方法	29
1.7.2 黑盒测试法	30
1.7.3 白盒测试法	32
1.7.4 软件测试过程	34
1.8 软件维护	36
1.8.1 软件维护的内容	36
1.8.2 软件维护的过程	37
1.9 面向对象技术	40
1.9.1 面向对象的基本概念	40

1.9.2 面向对象分析	41
1.9.3 面向对象设计	42
1.9.4 面向对象程序设计	43
1.10 总结	43
1.11 习题	43
第 2 章 软件过程管理	44
2.1 软件过程管理的基本概念	44
2.2 软件过程改进	45
2.3 软件能力成熟度模型	47
2.3.1 CMM 概述	47
2.3.2 CMM 的理解	48
2.3.3 CMM 的发展	48
2.3.4 CMM 的其他产品	49
2.4 个体软件过程	50
2.4.1 PSP 提出的背景	50
2.4.2 PSP 概述	50
2.4.3 PSP 的内容	50
2.4.4 基于 PSP 的软件过程改进	51
2.4.5 PSP 的现状	52
2.5 团队软件过程	52
2.5.1 TSP 概述	52
2.5.2 实施 TSP 的条件	52
2.5.3 TSP 的管理原则	53
2.5.4 TSP 流程	53
2.5.5 TSP 的早期应用	54
2.6 CMM、PSP 和 TSP 的关系	54
2.7 总结	55
2.8 习题	56
第 3 章 软件能力成熟度模型	57
3.1 CMM 介绍	57
3.2 CMM 的 5 级成熟度模型	58
3.3 CMM 体系结构	59
3.3.1 成熟度等级的内部结构	60
3.3.2 成熟度等级	60
3.3.3 关键过程域 KPA	60
3.3.4 共同特性	61
3.3.5 关键实践	61

3.4	成熟度等级的行为特征	61
3.4.1	初始级的行为特征	61
3.4.2	可重复级的行为特征	62
3.4.3	已定义级的行为特征	62
3.4.4	已管理级的行为特征	63
3.4.5	优化级的行为特征	63
3.5	软件过程的可视性	63
3.6	过程能力和效能预测	65
3.7	成熟度等级的跳跃	66
3.8	总结	67
3.9	习题	67
第 4 章	CMM 初始级和可重复级	68
4.1	初始级	68
4.2	需求管理	68
4.3	软件项目计划	72
4.4	软件项目跟踪和监督	80
4.5	软件子合同管理	85
4.6	软件质量保证	91
4.7	软件配置管理	95
4.8	总结	100
4.9	习题	101
第 5 章	CMM 已定义级	102
5.1	组织过程焦点	102
5.2	组织过程定义	106
5.3	培训大纲	111
5.4	集成软件管理	116
5.5	软件产品工程	125
5.6	组间协调	135
5.7	同行评审	139
5.8	总结	142
5.9	习题	143
第 6 章	CMM 已管理级	144
6.1	定量过程管理	144
6.2	软件质量管理	151
6.3	总结	156
6.4	习题	156
第 7 章	CMM 优化级	158

7.1	缺陷预防	158
7.2	技术革新管理	164
7.3	过程更改管理	170
7.4	总结	176
7.5	习题	177
第 8 章	个体软件过程	178
8.1	PSP 与 CMM 的关系	178
8.2	软件质量	179
8.3	PSP 原理	179
8.4	PSP 进化框架	180
8.5	PSP 过程结构	182
8.5.1	PSP 过程	182
8.5.2	PSP 项目计划总结表	183
8.5.3	PSP 脚本	186
8.5.4	PSP 计划	186
8.6	PSP 数据收集	188
8.6.1	时间测量	188
8.6.2	规模测量	189
8.6.3	质量测量	190
8.7	PSP 质量管理	192
8.7.1	缺陷和质量	192
8.7.2	工程师的职责	193
8.7.3	早期缺陷改正	193
8.7.4	缺陷预防	193
8.8	PSP 设计和实践	194
8.8.1	PSP 设计	194
8.8.2	PSP 实践	194
8.9	现状和将来的发展趋势	195
8.10	总结	195
8.11	习题	196
第 9 章	团队软件过程	197
9.1	TSP 开发的背景	197
9.2	TSP 小组的建立	198
9.2.1	TSP 是如何创建小组的	198
9.2.2	小组的发展	198
9.2.3	TSP 是如何创建小组的	199
9.3	TSP 的原则和设计	200

9.3.1	TSP 原则	200
9.3.2	TSP 的设计原则	201
9.4	TSP 操作过程	202
9.5	TSP 的结构	202
9.6	启动 TSP 组	204
9.7	TSP 的小组工作流程	206
9.7.1	领导小组	207
9.7.2	交流	207
9.7.3	维护计划	208
9.7.4	平衡小组工作负担	208
9.8	TSP 的质量管理	209
9.8.1	质量计划	209
9.8.2	确认质量问题	213
9.8.3	发现和预防质量问题	213
9.9	总结	214
9.10	习题	214
第 10 章	CMMI 模型简介	216
10.1	CMMI 模型引入	216
10.1.1	CMMI 出现背景	216
10.1.2	CMMI 源模型	219
10.1.3	CMMI 模型部件	220
10.1.4	CMMI 表示方法	221
10.2	CMMI 过程域	225
10.2.1	过程管理类过程域	227
10.2.2	项目管理类	230
10.2.3	工程类	233
10.2.4	支持类	234
10.3	CMMI 阶段式表示	237
10.3.1	成熟度等级 1: 初始级	237
10.3.2	成熟度等级 2: 已管理级	238
10.3.3	成熟度等级 3: 已定义级	244
10.3.4	成熟度等级 4: 定量管理级	254
10.3.5	成熟度等级 5: 优化级	256
10.4	CMMI 连续式表示	259
10.4.1	能力等级 0: 未执行级	262
10.4.2	能力等级 1: 已执行级	262
10.4.3	能力等级 2: 已管理级	263

10.4.4 能力等级 3: 已定义级	264
10.4.5 能力等级 4: 定量管理级	264
10.4.6 能力等级 5: 优化级	265
10.5 IDEAL 模型	265
10.6 总结	267
10.7 习题	267
附录 A 术语表	268
附录 B 基于 CMM 的评估过程	290
附录 C CMM 提问单	295
参考文献	316

第 1 章 软件工程基础

本章目标

本章对传统的软件工程方法做了概括性的描述，目的是使读者对软件的传统开发方法有一个完整的了解，从而指导他们在软件开发过程中自觉运用工程化的方法，保证软件有合理的结构、良好的编程风格，以及很好的可测试性和可维护性。本章中以一些实际的系统为例来帮助用户理解抽象的理论，并在本章最后留了部分习题，帮助读者加深对所学内容的理解。

1.1 软件工程的观念

软件工程是以工程化的思想和原理进行软件的开发和维护，从而保证软件开发的进度、经费以及软件可靠性的方法。

软件工程最早是由北大西洋公约组织（NATO）于 20 世纪 60 年代提出的。当时计算机的应用处在急剧膨胀时期，软件需求量增加，软件的规模也大大增加。对于一个普通的软件来说，随着规模的增加，其复杂度将大大增加，即使是最高明的程序员，也难免在编程中出现错误。这样软件开发的费用和周期将会超出计划，软件的可靠性也难以保证。更为糟糕的是，程序员往往没有很好地了解用户的意图就匆匆进行编码，因而在编码过程中要不断修改代码以适应用户的需求。这种修改在越接近开发后期时，付出的代价就越高。即使是已经完成、交付用户使用的软件，也会存在各种隐藏的误差和缺陷。于是程序员就像陷入了可怕的泥沼中不能自拔，这就是当时出现的所谓“软件危机”。

对于一个软件，如果我们把开发和维护任务分成若干个相对独立的阶段，每个阶段由不同的人员完成相对独立的工作，并且作为下一阶段的基础和依据，这样就可以降低软件开发的难度，减少隐藏问题的继承，这就是软件工程的基本思想。

在软件开发和维护的每一个阶段，开发人员都要进行扎实、详尽的工作，并形成详细的文档。只有当本阶段的工作通过严格的审查后，才可以进行下一阶段的工作。如果发现问题，可以回归到上一阶段，找到问题的根源，改正后才能继续。按照软件工程的思想，软件的生存周期可以分为以下几个阶段：

- 软件策划 对软件进行可行性分析，估算开发费用和时间等。
- 需求分析 向用户了解需求，解决“做什么”的问题。

- 软件分析 分为概要设计和详细设计。对需要解决的问题进行分析，提出最佳的解决方案，解决“怎么做”的问题。
- 编码 按照软件分析的结果，采用某种编程语言加以实现。
- 测试 编码结束后，要用大量的数据进行测试，找出其中隐藏的错误。
- 维护 软件交付用户使用后，在运行过程中仍有可能出错，而且用户的需求也有可能改变，因此，软件需要继续修改。

软件的生存周期模型如图 1-1 所示。

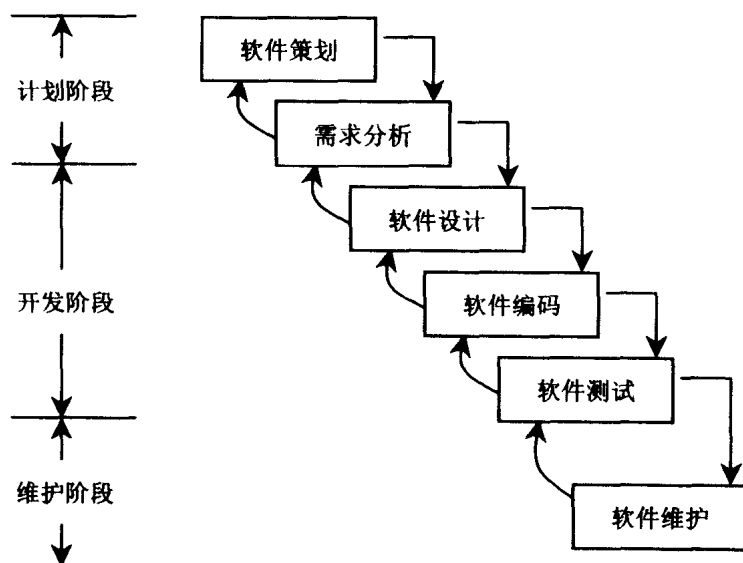


图 1-1 软件生存周期

软件生存周期中的几个步骤并不是毫无关系，而是紧密相关的，它们共同组成一个有机的整体，共同完成一个复杂的工作。例如，软件调试是配合软件编码的一个过程，目的是为了找出软件编码中的错误，软件测试是为了找出需求分析和软件设计阶段中的错误，而在软件运行过程中，可以检验软件是否与可行性分析阶段的成果相符合，即是否满足用户的需求。软件测试分为单元测试、整体测试和有效性测试等过程，分别用来测试详细设计、概要设计、需求分析的工作。这几个步骤的对应关系如图 1-2 所示。这个关系图称为软件生存周期的“V”字模型。

实际上，在软件开发的整个过程中，编码只是极小的一部分工作，大约占整个工作量的 10%~20%，更重要的工作是需求分析和软件设计。只要解决了“做什么”和“怎么做”的问题，编码是件很轻松的事，任何程序员都可以胜任。

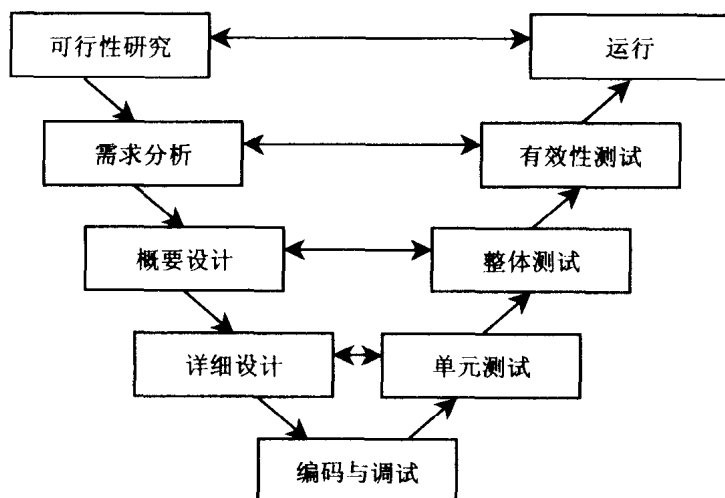


图 1-2 软件生存周期的“V”字模型

1.2 软件策划

开发软件的目的是为了帮助用户解决某个具体的问题。在解决问题之前必须搞清楚问题是什么，了解问题的性质、目标和大概的规模，然后确定这个问题是否有可行的解决方案，是否值得解决，估计解决问题的成本，最后制定大概的开发计划，并粗略地描述用户的需求，这就是软件策划的主要任务。

1.2.1 问题描述

问题描述的任务是对用户要求解决的问题做准确的定义，也就是确定问题的性质、目标和规模。

问题描述通常由系统分析员完成。了解问题的途径是向用户询问，仔细研究用户口头的或书面的陈述，确定用户的隐含要求，从中抽象出问题的定义。对于用户提出的解决方法，可以作为参考意见，但不能为其所左右。

系统分析员在了解清楚问题的本质后，应该写出书面报告，把自己对问题的认识返回给用户审查。如果有理解错误或理解不清楚的地方，要立即改正，然后重新提交用户审查。这一步工作要扎实细致，因为系统分析员对问题的描述确定了以后开发工作的方向，任何对问题的不清楚描述、错误描述或遗漏的描述都会导致软件不可用，不能解决实际问题。

系统分析员书写书面报告的目的是让用户检查开发人员对问题的理解是否与他们完全一致。如果确实达成了一致，系统分析员就可以着手进行可行性分析了。

1.2.2 可行性分析

用户提出的问题是否有可行的方案？能不能在预定的规模内解决？是否值得去做？

这是可行性分析的内容。

可行性分析不是研究如何解决问题，而是研究“值不值得做”的问题。系统分析员首先要对提出的各种方案进行研究，比较其优劣，确定是否有可行的方案，粗略估计需要的人力、时间和财力，比较开发的投入和随之带来的效益，以决定是否可以去。做。

可行性分析决不是可有可无的，它对软件开发过程中可能遇到的问题进行预测，尽量避免开发中的盲目性，减少开发过程中可能会陷入的经济、技术上的困境。

一般来说，可行性分析主要关心以下几个方面的问题：

- 经济可行性 估计开发软件所需要的投入和该软件所能带来的效益，比较投入与效益，确定该软件是否值得开发。
- 技术可行性 分析用现有的技术能不能解决这个问题，公司的开发人员能否胜任这项开发任务，开发所用的软件、硬件条件是不是能够达到要求。
- 社会可行性 分析软件的开发会不会影响他人的利益，是否会给社会带来不利的影响，会不会违反国家的有关法律法规。

如果问题没有可行的解决方法，系统分析员应建议终止开发工作，以免浪费人力、物力和时间资源，带来不必要的损失。可行性分析的结果应该由系统分析员写成书面报告，作为以后开发工作的依据。可行性分析报告大致可以包括以下内容：对要解决问题的描述、国内外相同、相似项目的发展现状以及软件在经济、技术、社会等方面的可行性等。

1.2.3 软件策划的内容

如果一个软件的开发经过分析后认为是可行的，那么现在就可以确定开发的工作范围，估算工作量，安排大致的开发进度了。软件策划以可行性分析报告为依据，由用户、系统分析员和开发人员共同确定软件功能，最终形成软件策划书。

软件策划书应当用通俗易懂的自然语言描述，要求简洁清晰。一个标准的软件策划应当至少包括以下内容：

- 软件指标 软件策划书首要的任务是确定软件所要达到的指标，如具有的功能，要达到的性能和可靠性等。功能描述指出软件的整体功能。如果有必要，可以进一步说明每一个子功能。性能描述指出软件的处理时间、响应时间、存储要求以及对计算机硬件的要求等。
- 资源描述 包括人力资源，计算机软、硬件资源。其中人力资源是最宝贵、最难以估计的资源，尤其是在大型软件的开发过程中，人力资源在不同阶段的变化是很大的。计算机软、硬件资源主要是指软件将要运行的环境，包括基本硬件系统、外设、操作系统和其他支撑软件等。
- 大致的开发进度 一般来说，软件在开发之前最后的交付日期就已经确定了。进度安排的内容是如何在交付日期之前合理分配资源，合理安排任务之间的协调关系，考虑意外因素的干扰，并把最后的复审工作作为一项重要的任务考虑。
- 估算开发成本 软件作为一种高智力劳动的成果，它的开发成本是很难精确计算的。影响开发成本的因素很复杂，所以只能进行大概估算。开发成本主要与人员

因素有关。开发人员的熟练程度和素质对单位时间内生产软件的数量起决定作用。另外,问题解决的难易程度、软件所要求达到的性能以及软件、硬件资源的可用性也对开发成本有重要影响。

常用的一种开发成本估算方法是按每项任务的工作量价格进行估算。这种方法首先对系统的功能进行分解,每项功能按照需求分析、概要设计、详细设计、编码、测试等几个阶段估计工作量(以人月为单位),然后规定每个工作量的价格。当然,不同阶段、不同人员的价格是不一样的。系统分析员的价格就比较高,初级程序员的价格就比较低。开发早期的需求分析、软件设计的价格比较高,而后期编码的价格比较低。由工作量和工作量的价格就得到每个阶段的价格。最后,各个阶段的成本之和就是开发总成本。

1.2.4 软件需求描述

软件需求描述用来说明软件的总体功能以及应当受到的限制。对软件需求描述的基本要求是要用通俗易懂的自然语言描述,不要出现专业术语,也不要有具体的实现细节。如果必要,可以附图说明。

软件需求描述通常包括功能和性能两方面的要求。功能要求指出系统应当提供的服务,一般由用户提出,并在用户和开发人员之间达成一致。

例如,图 1-3 列出了某电子邮件系统的功能描述。

- 邮件系统为用户提供收发电子邮件的功能
- 用户可以通过 POP3、IMAP 两种方式访问邮件系统
- 邮件系统允许用户申请帐号、注销帐号、修改帐号
- 邮件系统允许用户以附件的形式发送文件

:

图 1-3 某电子邮件系统的功能描述

性能要求指出软件应达到的性能指标和受到的限制,如系统响应时间、存储空间、可靠性等。图 1-4 列出了上述邮件系统的部分性能要求。

- 每个用户的邮箱容量限制在 10MB
- 拒绝垃圾邮件
- 用户只能查看、删除自己的邮件,不能访问别人的邮件,系统管理员有一切权限

:

图 1-4 某电子邮件系统的性能描述

由于软件需求描述是用自然语言书写的,不涉及任何技术细节,所以仅有软件需求描述作为软件开发的依据是不够的,还必须进行下一步的工作,即需求分析。