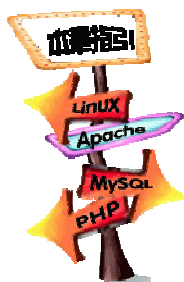


第27章

B/S 结构软件开发流程



软件开发流程即软件设计思路和方法的一般过程，包括设计软件的功能和实现的算法和方法、软件的总体结构设计、数据库设计和模块设计、编程和调试、程序联调和测试以及编写、提交程序。提高软件开发能力没有捷径可走，唯有走“规范化”之路。即：制定适合于本企业的软件过程规范，并按照此规范执行。本规定对软件开发各个过程中的目的、要求、人员和职责、工作的内容以及输入输出、评审等进行规范。本规定主要的约束对象是 B/S 结构的应用软件开发，涉及的开发部仅指软件开发部，产品仅指狭义范围内的 B/S 结构程序或应用软件程序。

27.1 软件开发过程的划分

本规定对一个完整的开发过程按“软件过程改进方法和规范”把产品生命周期划分为 6 个阶段：包括产品概念阶段（记为 PH0）；产品定义阶段（记为 PH1）；产品开发阶段（记为 PH2）；产品测试阶段（记为 PH3）；用户验收阶段（记为 PH4）；产品维护阶段（记为 PH5）。软件项目的过程有三大类：项目管理过程、项目研发过程和机构支持过程。而这三类过程可以细分为 19 个主要过程域，分布在 PH0 到 PH5 的各个阶段。项目管理过程包含 6 个过程域，分别为：立项管理、结项管理、项目规划、项目监控、风险管理、需求管理。项目研发过程包含 8 个过程域，分别为：需求开发、技术预研、系统设计、实现与测试、系统测试、Beta 测试、客户验收、技术评审。机构支撑过程包含 5 个过程域，分别为：配置管理、质量保证、培训管理、外包与采购管理、服务与维护。建议用户（企业）根据自身情况（如发展战略、研发实力等）适当地修改使用。如图 27-1 所示：

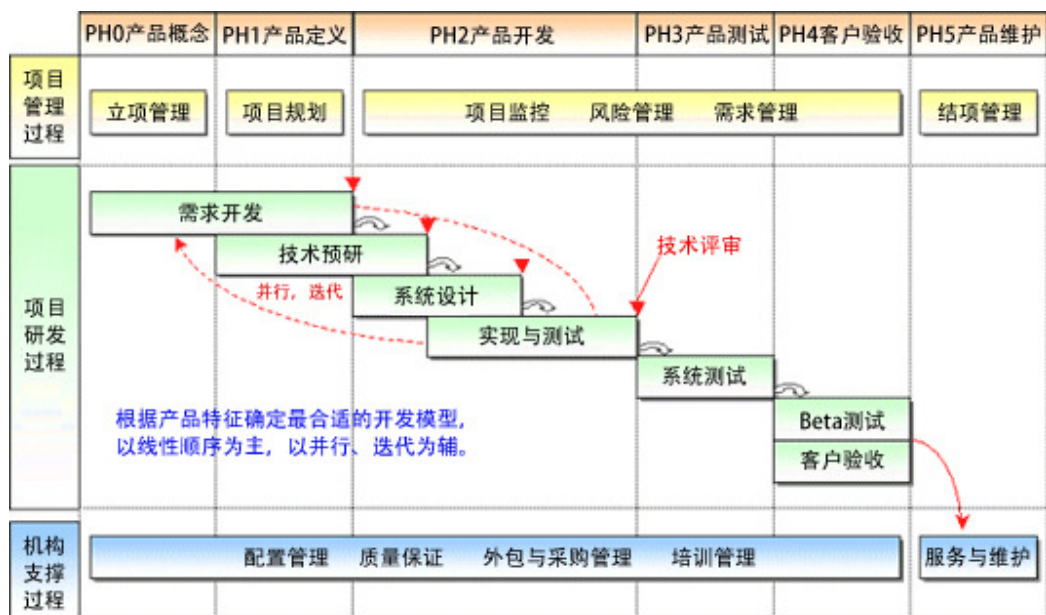


图 27-1 软件开发过程划分流程图

按照软件开发的流程规范，一个项目从策划到完成是由众多的过程规范和文档模板组成，如表 27-1 所示。主要用于指导国内互联网企业持续地改进其软件过程能力。在项目经理领导的团队开发小组，在各种过程开发基础上，接受迭代开发和敏捷软件开发过程的思想，要不拘泥与传统的开发过程，建立公司的快速开发过程，随需而变，及时满足客户需求的变更。

表 27-1 软件开发过程域遵循的标准文档

需求开发	《用户需求说明书》 《软件需求规格说明书》	技术预研	《技术预研计划》 《技术预研报告》
系统设计	《体系结构设计报告》 《用户界面设计报告》 《数据库设计报告》 《模块设计报告》	实现与测试	《实现与测试计划》 《编程文档》
系统测试	《系统测试计划》 《测试用例》 《测试报告》	Beta 测试	《Beta 测试协议》 《Beta 测试报告》
客户验收	《客户验收计划》 《客户验收报告》	技术评审	《技术评审计划》 《技术评审报告》 《技术评审检查表》

由于软件开发过程域遵循的标准文档众多，本书在后面几个章节中只提供了最常用的 3 个文档说明，供大读者学习参考使用，包括《项目需求说明书》、《数据库设计说明书》及《程序设计说明书》。因为这 3 个文档是和我们的项目设计有直接联系，也是项目开发流程中最主要的说明文档，其它的文档则需要在使用时自行整理。

27.2 需求开发

需求分析说明书的形成需要市场调研，技术和市场要结合才能体现最大价值。这个阶段需要出三样东西，用户视图，数据词典和用户操作手册。用户视图是该软件用户（包括终端用户和管理用户）所能看到的页面样式，这里面包含了很多操作方面的流程和条件。数据词典是指明数据逻辑关系并加以整理的东西，完成了数据词典，数据库的设计就完成了一半多。用户操作手册是指明了操作流程的说明书。请注意，用户操作流程和用户视图是由需求决定的，因此应该在软件设计之前完成，完成这些，就为程序研发提供了约束和准绳。很遗憾太多公司都不是这样做的，因果颠倒，顺序不分，开发工作和实际需求往往因此产生隔阂脱节的现象。需求分析，除了以上工作，作为项目设计者应当完整的做出项目的性能需求说明书，因为往往性能需求只有懂技术的人才可能理解，这就需要技术专家和需求方（客户或公司市场部门）能够有真正的沟通和了解。

27.2.1 需求分析流程

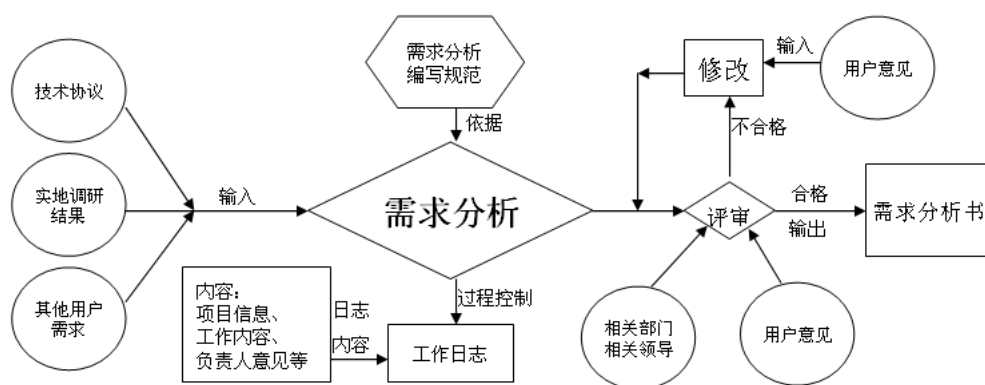


图 27-2 需求分析流程图

工作流程：市场部签定软件开发合同后，向开发部移交与之相关的资料，如：合同书、技术协议等；开发部组织人员根据相关资料进行需求分析，并且要与用户进行技术交流，充分获取用户对软件开发的边界等具体问题的确认；需求分析编制完成后，经相关部门评审合格即付诸实施。

责任部门：开发部

相关部门：市场部、主管副总、用户

相关资料：软件合同、技术协议、需求分析书、用户确认单、评审记录、日志。

相关规范：系统总体设计编制规范、系统详细设计编制规范。

27.2.2 需求分析说明

如果是公司开发的产品软件，则由公司领导根据市场的需求、技术的趋势及员工、经销商和用户的建议，提出新产品的概念及方向性意见，形成新产品概念书，并委托专人负责策划的前期工作，从而启动后续的工作。受委托者随后组织有关人员对新产品的市场前景、顾客需求、技术可行性、对手状态等进行调研，形成调研报告。需求调研的主要收集方式有以下方面：

- 与用户交谈，向用户提问题。
- 参观用户的工作流程，观察用户的操作。
- 向用户群体发调查问卷。
- 与同行、专家交谈，听取他们的意见。
- 分析已经存在的同类产品，提取需求。
- 从行业标准、规则中提取需求。
- 从 Internet 上搜查相关资料。

受委托人完成调研报告后，经公司领导或部门领导批准，并组织开发部对调研报告进行需求分析、研讨，确定产品的设计方案，形成《需求说明书》及项目建议书。项目建议书应阐述产品的背景、市场前景、产品的定位、特点、卖点以及功能和性能的基本要求、可行性和风险评估等。需求分析的目的是对各种需求信息进行分析、消除错误、刻画细节等，进一步定义准确无误、没有二议的软件需求。

《需求说明书》的评审由开发部主持，可邀请市场部、测试组等相关人员参加，需求分析人员根据评审意见，完善需求说明。对于客户提出的需求开发，还需与最终用户代表一起评审，原则上通过双方确认之后，需与客户方负责人做书面承诺，使之具有商业合同效力。项目建议书经过公司领导批准后，即可进入立项程序并成立项目组。立项应详细定义产品的功能和性能，确定项目的正式负责人和其他责任人以及完成任务的时间和占用的资源，说明实现的大体方案和要求，以及确定对各个过程是否合并和省略的输出文件等。

立项通过评审并获得总工程师批准之后，项目负责人应组织项目组的人员制订计划，编写计划书和计划表，经评审通过后即可进入后续的过程。关于计划的制订做法，请参照各公司的《工作计划管理规定》执行。

27.2.1 输出

这一过程的输出主要包括以下方面：

- 新产品概念书
- 调研报告
- 《需求说明书》
- 《项目建议书》
- 计划书和计划表

27.3 系统设计

系统设计主要分为数据库设计，程序的概要设计及详细设计。概要设计是将系统功能模块初步划分，并给出合理的研发流程和资源要求。作为快速原型设计方法，完成概要设计就可以进入编码阶段了，通常采用这种方法是因为涉及的研发任务属于新领域，技术主管人员一上来无法给出明确的详细设计说明书，但是并不是说详细设计说明书不重要，事实上快速原型法在完成原型代码后，根据评测结果和经验教训的总结，还要重新进行详细设计的步骤。详细设计则是考验技术专家设计思维的重要关卡，详细设计说明书应当把具体的模块以最‘干净’的方式(黑箱结构)提供给编码者，使得系统整体模块化达到最大；一份好的详细设计说明书，可以使编码的复杂性减低到最低，实际上，严格的讲详细设计说明书应当把每个函数的每个参数的定义都精精细细的提供出来，从需求分析到概要设计到完成详细设计说明书，一个软件项目就应当说完成了一半了。换言之，一个大型软件系统在完成了一半的时候，其实还没有开始一行代码工作。那些把作软件的程序员简单理解为写代码的，就从根子上犯了错误了。

27.3.1 系统设计流程

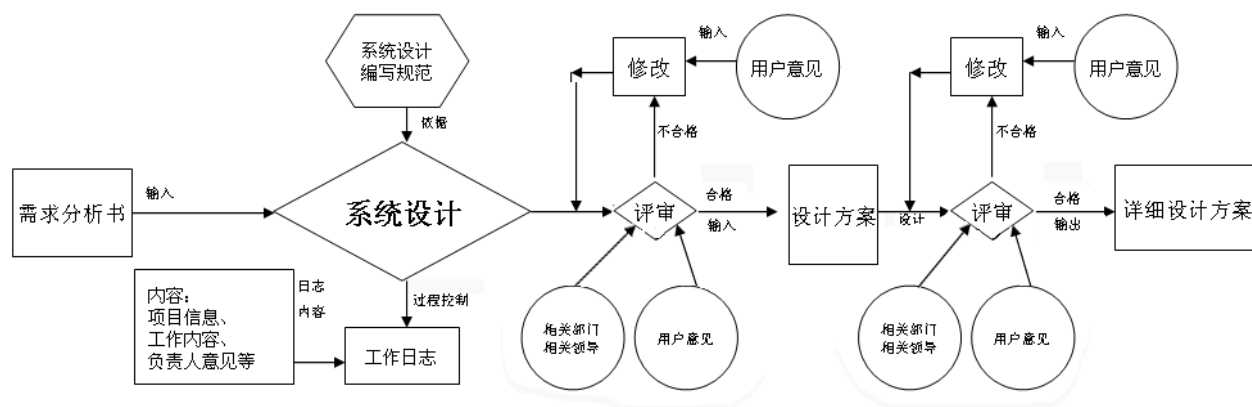


图 27-3 系统设计流程图

工作流程：需求分析经评审通过后，开发部组织人员进行系统设计；系统设计完成后，开发部组织相关专业部门进行评审并获得用户的确认。系统设计和系统详细设计均属于评审范围。

责任部门：开发部

相关部门：市场部、主管副总、用户

相关资料：需求分析书、系统总体设计规范、系统详细设计规范、数据字典、用户确认单、数据流定义、编码规范、日志。

相关规范：系统总体设计编制规范、系统详细设计编制规范。

27.3.2 系统设计说明

设计过程是指根据《项目建议书》、《需求说明书》的内容设计软件的系统结构、用户界面、数据库、功能模块，并编写《使用说明书初稿》等，从而在需求与代码之间建立桥梁，指导开发人员的工作，开发出能够满足用户需求的产品。

- 软件系统设计的好坏，可以用“可靠性、可维护性、结构稳定性、可扩展性、可复用性、软件执行效率”等因素来评估。
- 原则上软件系统设计、数据库设计应全局考虑，需同步编写文档，轻易不要简化。
- 界面的设计只要大多数用户、开发人员理解并认同界面原型即可，细节可在编程时实现。
- 功能模块主要是确定每个模块的主要接口、数据结构与算法，能够清楚地指导模块编程即可，可不必花太多时间设计模块细节，如开发压力较大时可在编程工作结束后安排编写。

27.3.2.1 输入

这一过程的输入包括以下部分：

- 《项目建议书》
- 《需求说明书》
- 软件设计过程中的标准与规范。
- 软、硬件开发环境。

27.3.2.2 人员与职责

项目组组长应根据开发人员的实际能力，合理的分配设计任务，包括系统设计、用户界面设计、数据库设计、模块设计，编写《使用说明书初稿》等。开发人员需详细阅读《项

目建议书》、《需求说明书》，明确设计任务并准备相关的设计工具和资料。

27.3.2.3 确定约束

开发人员从《项目建议书》、《需求说明书》中提取需求约束，例如：

- 本系统应当遵循的标准或规范。
- 软件、硬件环境（包括运行环境和开发环境）的约束。
- 接口/协议的约束。
- 用户界面的约束。
- 软件质量的约束，如可靠性、可维护性、结构稳定性、可扩展性、可复用性、软件执行效率等。

27.3.2.4 确定设计

开发人员根据产品的需求，以及产品的发展战略，确定设计策略，例如：

- 根据产品的功能性、非功能性需求，确定某些设计模式。
- 说明为了方便本系统在将来的扩展功能，现在有什么措施。
- 说明本系统在当前“复用什么”以及将来“如何被复用”。
- 说明当两个标准同时被优化时如何折中，例如时间性与空间性之间的折中，复杂性与实用性之间的折中。

27.3.2.5 系统结构设计

- 将系统分解为若干子系统，确定每个子系统的功能以及子系统之间的关系。绘制系统的总体结构图。
- 将子系统分解为若干模块，确定每个模块的功能以及模块之间的关系，绘制子系统的结构图。
- 确定系统开发、测试、所需的软硬件环境。

27.3.2.6 撰写文档

这个过程主要是将设计过程中确定约束、确定设计、系统结构设计这几个阶段中涉及的内容编写出《系统设计说明》。

27.3.2.7 用户界面设计

界面设计人员应详细阅读《需求说明书》，了解用户的需求，与用户交流，了解用户的工作习惯和他们对界面的看法，同时应多收集意见，及时改进。

27.3.2.8 数据库设计

- 数据库设计人员应详细阅读《需求说明书》和《系统设计说明》，明确数据库设计任务。
- 数据库设计人员准备相关的设计工具和资料。
- 数据库设计人员确定本软件的数据库规则，例如数据库命名等。

27.3.2.9 功能模块设计

- 模块设计的核心工作是“接口设计”和“数据结构与算法设计”。
- 模块设计人员详细阅读《需求说明书》和《系统设计说明》，明确模块设计任务。
- 模块设计人员需准备相关的设计工具和资料。
- 模块设计人员需确定软件的编程规范，确保模块设计文档的风格与代码的风格保持一致。

27.3.2.10 输出

- 《系统设计说明》
- 用户界面原型
- 《数据库设计说明》

- 《功能模块设计说明》
- 《使用说明书初稿》

27.3.2.11 评审

项目组负责人可邀请公司领导、市场部（用户）、测试组等相关人员对系统设计进行技术评审。评审小组需考察系统设计的综合能力，最后由总工程师确定可否进入软件实现过程并给出评审意见。

27.4 编码测试

在规范化的研发流程中，编码工作在整个项目流程里最多不会超过 1/2，通常在 1/3 的时间，所谓磨刀不误砍柴功，设计过程完成的好，编码效率就会极大提高，编码时不同模块之间的进度协调和协作是最需要小心的，也许一个小模块的问题就可能影响了整体进度，让很多程序员因此被迫停下工作等待，这种问题在很多研发过程中都出现过。编码时的相互沟通和应急的解决手段都是相当重要的，对于程序员而言“bug”永远存在。按照测试执行方，可以分为内部测试和外部测试；按照测试范围，可以分为模块测试和整体联调；按照测试条件，可以分为正常操作情况测试和异常情况测试；按照测试的输入范围，可以分为全覆盖测试和抽样测试。总之，测试同样是项目研发中一个相当重要的步骤，对于一个大型软件，3 个月到 1 年的外部测试都是正常的，因为永远都会又不可预料的问题存在。完成测试后，完成验收并完成最后的一些帮助文档，整体项目才算告一段落，当然日后少不了升级，修补等工作，只要不是想通过一锤子买卖骗钱，就要不停的跟踪软件的运营状况并持续修补升级，直到这个软件被彻底淘汰为止。

27.4.1 编码与测试流程

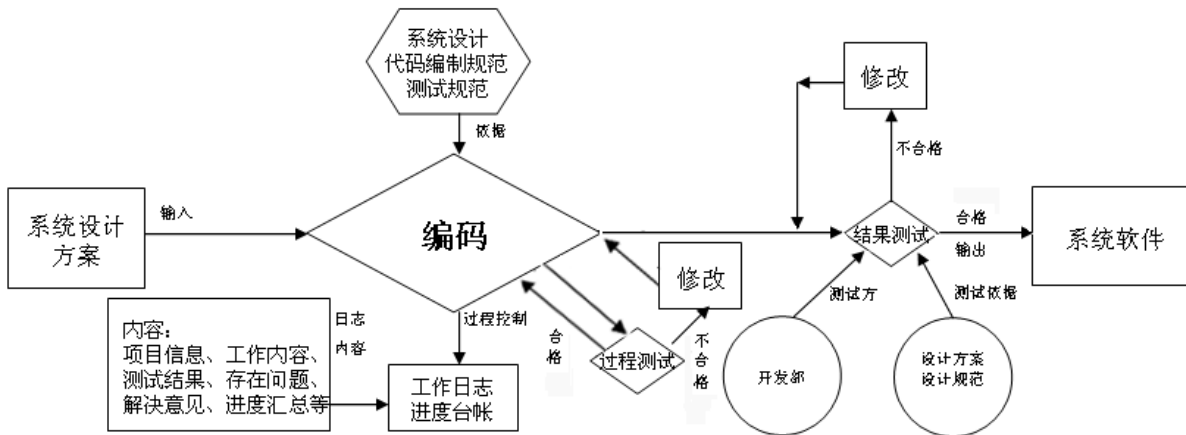


图 27-4 编码与测试流程图

工作流程：系统设计完成并经评审通过后，开发部组织人员进行代码编制（如采用外包方式编码，开发部要组织专门人员为外包单位提供代码编制规范和技术文档要求，并负责监控整个编码过程）。编码过程中，开发部相关人员应对完成后的每一模块组织进行过程测试；编码完成后，开发部组织相关人员对系统进行测试。测试分符合性测试和功能性测试两步进行，测试完成后，开发部组织相关专业部门对系统进行整体测评。

责任部门：开发部

相关部门：主管副总 、代码编制部门（外包）

相关资料：系统详细设计、数据字典、编程记录；测试记录、测试报告、数据流定义、编码规范、代码描述、程序源代码及相关文档。

相关规范：软件设计代码编制规范、软件测试标准。

27.4.2 编码说明

软件实现是指通过编程、调试、优化、内部测试和代码审查等活动，开发出符合用户需求、质量合格的产品。

27.4.2.1 输入

这一过程的输入包括以下内容：

- 《项目建议书》
- 《需求说明书》
- 《使用说明书初稿》
- 《系统设计说明》
- 用户界面原型
- 《数据库/数据结构设计说明书》

- 《功能模块设计说明》
- 编程规范

27.4.2.2 人员

软件实现过程主要由开发部负责。

- 项目组长应合理分配开发任务，确定何人何时完成哪些模块。
- 开发人员需构建编程与测试环境，例如软件开发工具的选择、硬件开发环境的选择等。
- 项目组长应确定编程规范，统一编程风格、提高代码质量。参考本书《附录 A》

27.4.2.3 调试

开发人员如果仅认为程序通过了连接，就认为程序基本上没有问题交由测试组测试，这是一种不良的做法。所谓编程调试指的是：当开发人员编写完一个或几个相关程序之后，不必等别人测试，自己马上对代码进行单步跟踪调试。调试带来的好处是：

- 大大减少后继的测试和改错代价。
- 单步跟踪调试能够发现数据溢出、内存泄漏等仅靠黑盒测试难以察觉的软件缺陷，提高了程序的质量和测试效率。

27.4.2.4 优化

软件的优化指的是提高软件的运行速度、提高对内存资源的利用率、加强用户界面的友好化等方面。

27.4.2.5 内部测试和代码审查

软件实现阶段的测试在开发部小组内部展开，主要的检查方式是白盒测试，由开发人员进行单步跟踪、代码审查、软件功能测试、软件性能测试等。代码审查通常可在开发人员之间展开，主要是检查代码是否符合编程规范，并发现代码编写的缺陷等，如人员及时间不充分时至少也要由开发人员自行进行代码审查。

27.4.2.6 输出

- 改进后的《需求说明书》
- 改进后的《使用说明书初稿》
- 改进后的《系统设计说明》
- 改进后的《数据库设计说明书》
- 改进后的《功能模块设计说明》
- 改进后的用户界面
- 带注解的源程序清单

27.4.3 结果测试说明

软件测试与改错的目的是依据测试计划和测试用例，在给定的项目条件下（人员、时间、工具等）尽可能地找出软件中的缺陷，并及时消除这些缺陷。这一过程可划分为以下几个步骤：制定测试计划→设计测试用例→执行测试→消除缺陷→审核确认。如果没有通过审核则继续执行测试。

27.4.3.1 输入

这一过程的输入有：

- 《项目建议书》
- 《需求说明书》
- 《使用说明书初稿》
- 《系统设计说明》
- 用户界面说明

27.4.3.2 人员责任

制定测试计划、设计测试用例、执行测试由测试组负责，消除缺陷由项目组开发人员负责。

27.4.3.3 制定测试计划

项目组组长应组织项目组人员讨论测试工作，然后由测试组长组织测试人员撰写《测试计划》。《测试计划》需通过开发部经理审核，经总工程师批准，测试人员即可按计划执行测试工作。

27.4.3.4 设计测试用例

测试用例是用于检验目标软件是否符合要求的一种“示例”，其基本要素有：前提条件、输入数据或动作、期望的响应。《测试用例》主要是描述各种测试操作的手册。测试用例需经过项目组组长审核，总工程师批准。

- 设计测试用例的目的是找出需求、设计、代码中的缺陷。
- 不同的测试用例其用途应当不一样，不要累赘。
- 显而易见的测试用例不必完整地用文字描述，可采用图表法等简化。

27.4.3.5 执行测试

测试人员按照《测试计划》及《测试用例》执行测试，如果发现缺陷，则记录在缺陷跟踪工具中，并及时通知项目组开发人员。测试人员完成测试后应撰写《测试报告》，总结测试工作，分析问题并给出建议及测试结论。

27.4.3.6 测试的优先级

有些时候因人员及时间限制，不能对软件做全面的测试，那么测试人员应集中力量测试优先级较高的内容，放弃优先级较低的内容。下表 27-2 所示列出部份测试优先级供测试人员参考。

表 27-2 测试优先级

测试内容	测试优先级	测试内容
软件特色功能	高于	非特色功能
用户常用功能	高于	非常用功能
功能出错将导致用户索赔的模块	高于	不会索赔的模块
系统性能瓶颈所在的模块	高于	不是性能瓶颈所在的模块

最复杂、最容易出错的模块	高于	不复杂、一般不会出错的模块
开发者没有信心的模块	高于	开发者自信的模块

27.4.3.7 消除缺陷

如果在测试时发现了软件缺陷，项目组开发人员应当尽早消除缺陷，在改错时需注意以下事项：

- 找到错误代码时，不要急于修改，先思考：修改此代码是否会引发其它问题。
- 有些时候，软件中可能潜伏同类型的错误，应当一一改正。
- 改错之后一定要马上重新测试，以免引入新的错误。

27.4.3.8 完善使用说明书

测试人员在软件测试过程中逐步熟悉产品功能，在接受开发人员的委托后对使用说明书进行完善及排版工作。

27.4.3.9 评审

测试组经过严格测试并确认的程序，经开发部经理审核，总工程师批准，颁布试用版本后，可进入试用过程。

27.4.3.10 输出

- 改进后的《需求说明书》
- 改进后的《系统设计说明》
- 改进后的《数据库设计说明书》
- 改进后的《功能模块设计说明》
- 改进后的用户界面
- 带注解的源程序清单
- 《测试计划》
- 《测试用例》

- 《测试报告》
- 《使用说明书》
- 《试用版颁布命令》

27.5 试运行

软件试运行是将产品交给最终用户，由其按实际业务流程对产品进行使用，一般试用三个月或更长时间没问题后即可正常投产，试运行过程中，一般要进行双线运行，即同时按产品上线试运行前的操作和在试运行的软件中操作。

27.5.1 软件试运行流程

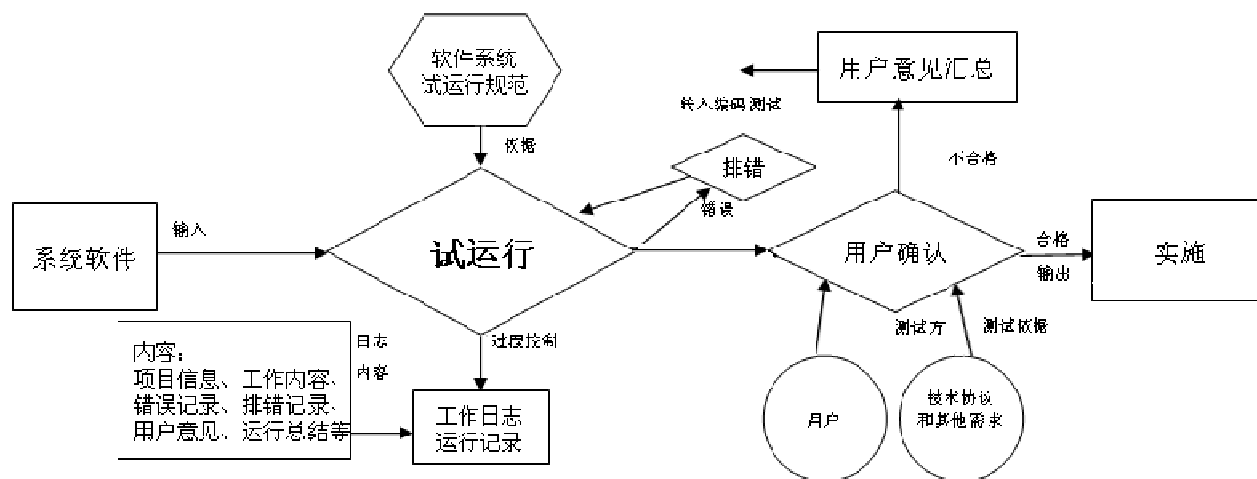


图 27-5 软件试运行流程图

工作流程：编码测试完成后经相关部门同意后开发部组织系统试运行，试运行过程中要对系统所产生的问题详细记录并马上解决。

责任部门：开发部

相关部门：用户、主管副总、代码编制部门（外包）

相关资料：试运行记录、错误和排错记录、试运行总结报告。

相关规范：软件系统试运行规范、技术协议。

27.5.2 软件试运行说明

对于试用版本软件需经过用户试用，由用户提交《试用报告》，并按试用的情况对于用户发现的软件缺陷开发人员应当及时纠正，对于一些难以马上实现的有益建议由项目组长决

定如何处理。对于合同项目的软件产品，需交客户验收，客户要对开发人员交付的产品进行审查和测试，确保产品满足客户需求，由客户提交《客户验收报告》。

27.6 实施

软件项目实施方案概述软件产品，特别是行业解决方案软件产品不同于一般的商品，用户购买软件产品之后，不能立即进行使用，需要软件公司的技术人员在软件技术、软件功能、软件操作等方面进行系统调试、软件功能实现、人员培训、软件上线使用、后期维护等一系列的工作，我们将这一系列的工作称为软件项目实施。大量的软件公司项目实施案例证明，软件项目是否成功、用户的软件使用情况是否顺利、是否提高了用户的工作效率和管理水平，不仅取决于软件产品本身的质量，软件项目实施的质量效果也对后期用户应用的情况起到非常重要的影响。

27.6.1 软件实施流程

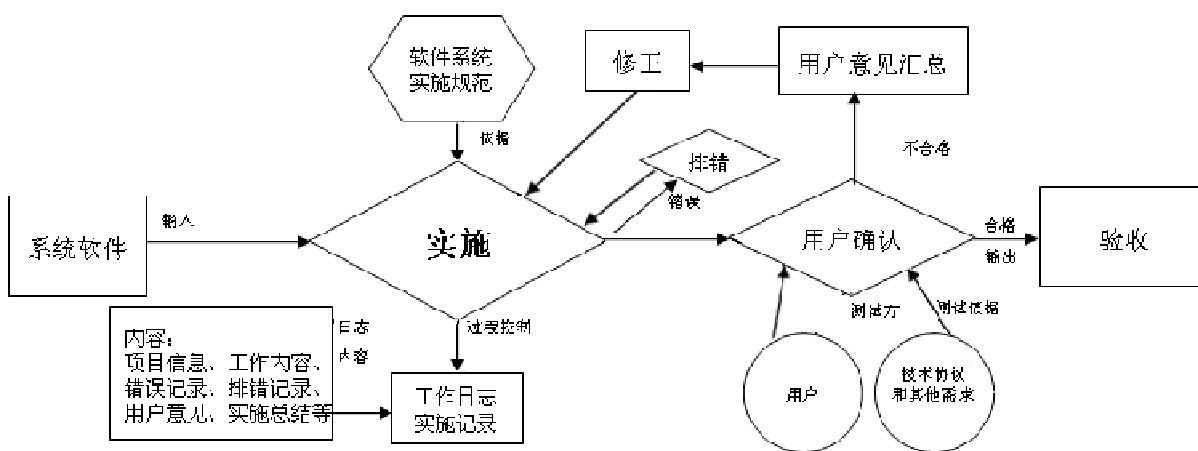


图 27-6 软件实施流程图

工作流程：试运行完成后，由开发部组织软件的实施（如由外包单位实施，开发部应该负责整个实施过程的监控、管理和协调）。

责任部门：开发部

相关部门：用户、主管副总、代码编制部门（外包）

相关资料：实施记录、用户意见表、用户意见反馈表、系统实施总结报告。

相关规范：软件系统实施规范、技术协议。

27.6.2 软件实施说明

当软件产品经过一段时间的试用基本稳定之后，项目组长需整理软件的相关文档（保

密性文件需按公司保密条款执行)，交资料室备案，同时项目组开发人员及测试组人员负责颁布软件正式版本，一般来说软件正式版本的颁布由总工程师评审，总经理批准。

27.6.2.1 责任人

项目组组长完成相关文档的整理工作，开发人员及测试组人员负责软件版本的颁布工作。

27.6.2.2 输出

一套完整的文档可包括以下文件：

- 新产品概念书
- 《项目建议书》
- 《需求说明书》
- 《系统设计说明》
- 《数据库设计说明书》
- 《功能模块设计说明》
- 用户界面说明
- 带注解的源程序清单
- 《测试计划》
- 《测试用例》
- 《测试报告》
- 《使用说明书》
- 《用户试用报告》
- 《颁布命令》

27.7 验收

在软件开发合同的签订阶段就提出软件验收项目和验收通过标准的意见；在软件的需求评审阶段，仔细审阅软件的需求规格说明书，指出不利于测试和可能存在歧义的描述；在开发方开发完软件并经过开发方内部仔细的测试后，对完成的软件进行评审或第三方的验收测试，提供完整的错误报告提交给用户方，由用

户方根据之前签订的开发合同中相应的验收标准判断是否进行验收。

27.7.1 软件验收流程

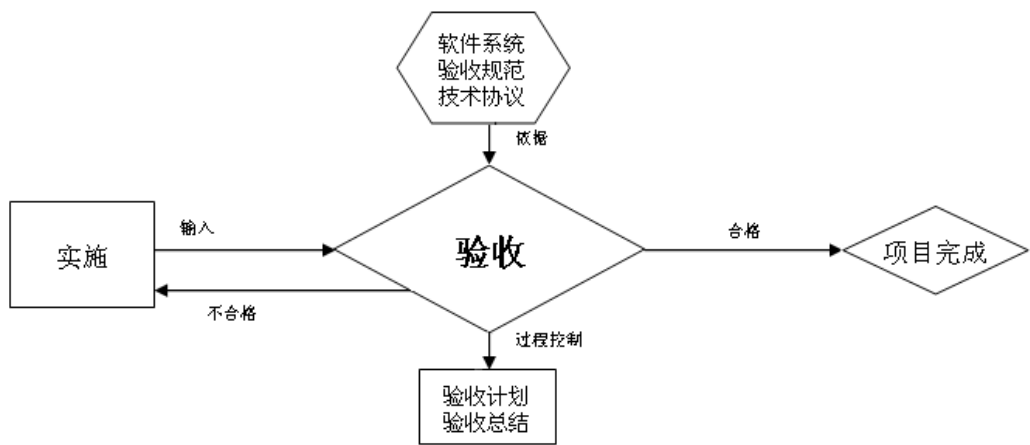


图 27-7 软件验收流程图

工作流程：实施完成后经用户确认，由开发部提交验收计划，并会同质量保证部、市场部和用户进行验收；验收完成后提交验收报告，软件开发及实施全部完成。

责任部门：开发部

相关部门：用户、质量保证部、市场部

相关资料：系统实施总结报告、用户意见表、验收计划、验收报告。

相关规范：软件系统验收标准、技术协议。

27.7.2 软件验收说明

27.7.2.1 功能项测试

对软件需求规格说明书中的所有功能项进行测试。

27.7.2.2 业务流程测试

对软件项目的典型业务流程进行测试。

27.7.2.3 容错测试

容错测试的检查内容包括：

- 软件对用户常见的误操作是否能进行提示；
- 软件对用户的操作错误和软件错误，是否有准确、清晰的提示；
- 软件对重要数据的删除是否有警告和确认提示；
- 软件是否能判断数据的有效性，屏蔽用户的错误输入，识别非法值，并有相应的错误提示。

27.7.2.4 安全性测试

安全性测试的检查内容包括：

- 软件中的密钥是否以密文方式存储；
- 软件是否有留痕功能，即是否保存有用户的操作日志；
- 软件中各种用户的权限分配是否合理。

27.7.2.5 性能测试

对软件需求规格说明书中明确的软件性能进行测试。测试的准则是要满足规格说明书中的各项性能指标。

27.7.2.6 易用性测试

易用性测试的内容包括：

- 软件的用户界面是否友好，是否出现中英文混杂的界面；
- 软件中的提示信息是否清楚、易理解，是否存在原始的英文提示；
- 软件中各个模块的界面风格是否一致；
- 软件中的查询结果的输出方式是否比较直观、合理。

27.7.2.7 适应性测试

参照用户的软、硬件使用环境和需求规格说明书中的规定，列出开发的软件需要满足的软、硬件环境。对每个环境进行测试。

27.7.2.8 文档测试

用户文档包括：安装手册、操作手册和维护手册。对用户文档测试的内容包括：

- 操作、维护文档是否齐全、是否包含产品使用所需的信息和所有的功能模块；
- 用户文档描述的信息是否正确，是否没有歧义和错误的表达；

- 用户文档是否容易理解，是否通过使用适当的术语、图形表示、详细的解释来表达；
- 用户文档对主要功能和关键操作是否提供应用实例；
- 用户文档是否有详细的目录表和索引表。
- 用户有特别要求的测试。

27.7.3 验收标准

- 测试用例不通过数的比例 $< 3\%$ ；
- 不存在错误等级为1 的错误；
- 不存在错误等级为2 的错误；
- 错误等级为3 的错误数量 ≤ 10 ；
- 所有提交的错误都已得到更正。

27.8 服务与维护

维护是指软件产品交付给客户之后的客户服务和产品维护。客户服务和产品维护的宗旨是提高客户对产品的满意度。项目组在完成产品开发的过程后，应根据实际情况组织编写培训大纲，同时对市场部技术人员做好培训工作。市场部技术人员接受培训之后，按培训大纲的内容整理并编写完整的培训手册，同时完成对客户的培训。用户在使用产品的过程中如对产品有何疑问或意见可咨询市场部，市场部服务人员有义务帮助客户解决问题。客户在使用产品的过程中如发现软件缺陷，原产品开发人员、维护人员有义务帮助客户解决缺陷。

27.8.1 责任人

开发部组织编写培训大纲，完成对市场部技术人员的培训工作。市场部技术人员经过培训组织编写培训手册，并完成客户的服务和培训。市场部应对客户在使用产品的过程中提出的问题进行收集，给予详细的解答，开发人员应对客户提出的软件缺陷给予解决。

27.8.2 收集信息

市场部在产品销售过程中定期通过上门或电话等方式回访客户，了解客户在使用产品的过程中碰到的问题，同时收集客户意见。客户通过各种渠道（如电话、Internet 等）向市场部客服人员提出服务请求。市场部客服人员应对客户提出的问题给予答复，如果自己可以解决的，那么立即给以解答，如果自己不能解决的，可以请相关的开发人员或测试组人员协助解决问题，同时做好服务与维护记录。如果客户提出软件维护，则转维护分析；提出需求

变更，则转入定制开发。

27.8.3 维护分析

一般来说，开发人员既要开发新的软件项目，又要维护老的软件项目（兼任维护人员），当维护人员接到客户或者市场部客服人员的维护请求时，需先进行维护分析。

- 如果用户因为不会使用产品的某些功能而请求帮助，那么维护人员应当尽快通过电话或 Internet 指导用户操作。
- 如果用户发现了产品的缺陷，维护人员有义务尽快消除该缺陷，转入软件维护。
- 如果用户希望开发人员修改或者增加产品的功能，此时维护人员不可轻易答应客户，需按照《定制产品的开发规定》执行。

27.8.4 软件维护

- 维护人员和客户协商维护方式：如果可以通过 Internet 执行远程维护的话，那么采用远程维护以降低维护的成本。
- 维护人员在修改产品时，必须严格遵循《软件版本管理规定》来操作，避免工作成果的版本发生混乱。
- 维护人员必须对修改后的产品进行相关的测试，以免引入新的错误。
- 维护人员应填写《维护记录》。

27.8.5 改进

开发人员应定期对客户提出的需求改进和意见进行整理，同时收集市场的需求，通过对比分析，将比较通用的功能加入软件当中，从而提升软件的性能。

27.8.6 输出

- 培训大纲
- 培训手册

27.9 项目管理

同其他任何工程项目一样，软件项目同样存在一个非常重要的问题，这就是项目管理的

问题，而这一问题通常容易被一般的软件开发人员所忽视。在一般的软件工程资料中所讨论的重点也只是软件开发方法，对软件管理问题大多一笔带过。在一个小的软件开发项目中也许还无所谓，但一个大型的软件开发项目如果没有优秀的软件管理人员来领导和协调整个项目，其失败的可能性就很大了。因此有必要引起大家对此问题的重视，这也是本节的目的所在。项目管理工作涉及到软件开发工作的方方面面，其直接对象包括人、财、物，简单地说，人就是指软件开发人员，财就是指项目经费，物就是指软件项目。软件项目管理常见的管理内容如图 27-9 所示。

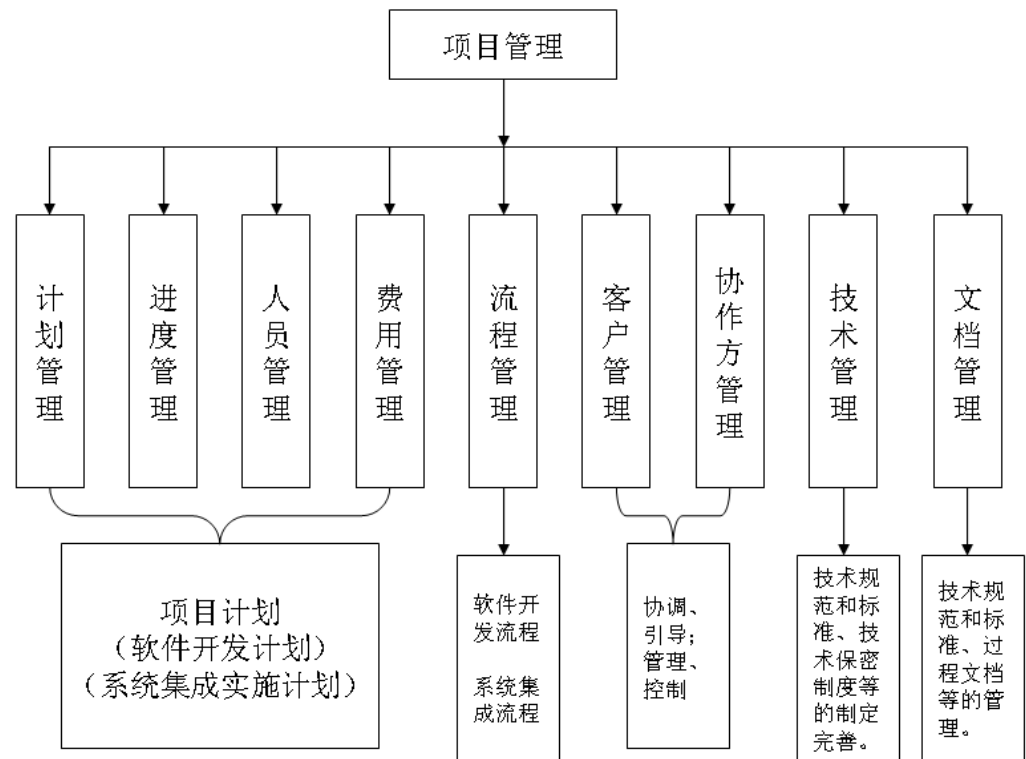


图 27-8 软件项目管理结构图

作为软件管理人员，应该站在高处来俯瞰整个项目，如果有不识庐山真面目的感觉就不太好了。有了俯瞰全局的意识这一前提，采用适当的管理技术，项目开展就容易了。软件项目的管理工作可以分位四个方面：软件项目的计划、软件项目的流程管理、软件项目协调和软件项目的控制。

27.9.1 软件项目的计划

软件开发项目的计划包括定义项目的目标，以及达到目标的方法。他涉及到项目实施的各个环节，带有全局的性质，是战略性的。计划应力求完备，要考虑到一些未知因素和不确定因素，考虑到可能的修改。计划应力求准确，尽可能提高所依据的数据的可靠程度。主要工作集中在软件项目的估算、软件开发成本的估算和软件项目进度安排。软件项目计划的目标是提供一个能使项目管理人员对资源、成本和进度做出合理估算的框架。这些估算应在软件项目开始时的一段有限时间内作出，并随着项目的进展进行更新。

27.9.1.1 软件项目的估算

软件项目管理过程开始于项目的计划，在做项目计划时，第一项活动是估算。现在已经使用的技术是时间和工作量的估算。因为估算是其他项目计划活动的基石，而且项目计划又为软件工程过程提供了工作方向，所以我们不能没有计划就着手开发，否则就会陷入盲目性。

估算本身带有风险，估算资源、成本和项目进度时需要经验、有用的历史信息、足够的定量数据和作定量度量的勇气。估算的精确程度受到多方面的影响。首先，项目的复杂性对于增加软件计划的不确定性影响很大，复杂性越高，估算的风险就越高。复杂性是相对度量的，它与项目参加人员的经验有关，比如如果让搞电子商城的项目组去搞 SNS 设计显然增加了复杂性。其次，项目的规模对于估算的精确性和功效的影响也比较大，因为随着软件规模的扩大，软件相同元素之间的相互依赖、相互影响也迅速增加，因而估算时进行问题分解也会变得更加困难。还有项目的结构化程度也影响项目估算的风险，这里的结构性是指功能分解的简便性和处理信息的层次性，结构化程度提高，进行精确估算的能力就提高，相应风险将减少。此外，历史信息的有效性也影响估算的风险，在对过去的项目进行这综合的软件度量之后，就可以借用来比较准确地进行估算。影响估算的因素远不止这些，比如用户需求的频繁变更给估算带来非常大的影响。

估算的依据是软件的范围，包括功能、性能、限制、接口和可靠性。在估算开始之前，应对软件的功能进行评价，并对其进行适当的细化以便提供更详细的细节。由于成本和进度的估算都与功能有关，因此常常采用功能分解的办法。性能的考虑主要包括处理和响应时间的需求。约束条件则标识外部硬件、可用存储和其他现有系统对软件的限制。

另外软件项目计划还要完成资源估算，包括人力资源、硬件资源和软件资源。在考虑各种软件开发资源时最重要的是人，必须考虑人员的技术水平、专业、人数以及在开发过程各阶段对各种人员的需要。硬件资源作为一种工具投入。软件资源包括各种帮助开发的软件工具，比如某某数据库等。

工作量估算是最普遍使用的技术。经过功能分解之后，可以估计出每一个项目任务的分解都需要花费若干“人日”，总计之后就知道了软件项目总体工作量。下表 27-3 就是一个示意性工作量估算表。

表 27-3 某软件系统工作量估算表（单位：人日）

任务	需求分析	设计	编码	测试	小计
用户定义	2	5	1	0.5	8.5
系统定义	2	5	1	0.5	8.5
广告预定	4	10	2	0.5	16.5
划版	5	20	10	0.5	35.5
制作和组版	3	5	3	1	12
总计	16	45	17	3	81

27.9.1.2 软件开发成本的估算

软件开发成本主要是指软件开发过程所花费的工作量及其相应的代价。它不同于其他物

理产品的成本，它主要包括人的劳动的消耗，人的劳动的消耗所需的代价就是软件产品的开发成本。开发成本的估算方法有很多种，象简单的代码行技术，任务分解技术，自动估计成本技术，专家判定技术，还有参数方程法，标准值法，以及 COCOMO 模型法。其中 COCOMO (Constructive Cost Model)模型法是一种精确、易于使用的成本估算方法。

27.9.1.3 软件项目进度安排

软件项目的进度安排主要是考虑软件交付用户使用前的这一段开发时间的安排。进度安排的准确程度可能比成本估计的准确程度更重要。软件产品可以靠重新定价或者靠大量的销售来弥补成本的增加，但进度安排的落空会导致市场机会的丧失或者用户不满意，而且也会导致成本的增加。因此在考虑进度安排时要把人员的工作量与花费的时间联系起来，合理分配工作量，利用进度安排的有效分析方法严密监视软件开发的进展情况，以使得软件开发的进度不致被拖延。

在进行进度安排时要考虑的一个主要问题是任务的并行性问题。当参加项目的人数不止一人是软件开发工作就会出现并行情况。因为并行任务是同时发生的所以进度计划表必须决定任务之间的从属关系，确定各个任务的先后次序和衔接，确定各个任务完成的持续时间。另外还应注意关键路径的任务，这样可以确定在进度安排中应保证的重点。常用的进度安排方法有两种，即甘特图法和工程网络法。

27.9.2 软件项目的组织

参加软件开发的人员如何组织起来，使他们发挥最大的工作效率，对成功地完成软件项目极为重要。开发组织采用什么形式由软件项目的特点决定，同时也与参加人员的素质有关。通常有三种组织结构模式：

➤ 按课题组划分的模式

把开发人员按课题组成小组，小组成员自始至终承担课题的各项任务。该模式适用于规模不大的项目，并且要求小组成员在各方面有技术专长。

➤ 按职能划分的模式

把开发项目的软件人员按任务的工作阶段划分为若干工作小组。要开发的软件在每个专业小组完成阶段加工后沿工序流水线向下传递。这种流水作业的方式使用于多项目并行的情况。

➤ 矩阵形模型

这种模式是以上两种模式的复合。一方面按工作性质成立一些专门小组，另一方面每一个项目都有它的经理人员负责。每一个软件开发人员属于某一个专门小组，有参加某一个项目的工作。该模式的优点有一方面参加专门组的成员可以在组内交流在各个项目中取得的经验，这更有利于发挥专业人员的作用；另一方面，各个项目有专门的人员负责，有利于软件项目的完成。这种模式比较适合于规模比较大的项目。

组织结构的最后一层是程序设计小组的组织形式。通常认为程序设计工作是按独立的方式进行的，程序人员独立地完成任务。但这并不意味着相互之间没有联系。一般在人数比较少时组员之间的联系比较简单，但随着人数的增加，相互之间的联系变得负责起来。小组内

部人员的组织形式对生产率有着十分重要的影响。

27.9.3 项目小组组织形式

常见的小组组织形式有三种，这三种形式可以灵活使用。

1. 主程序员制小组：相当于组长负责制，小组的核心由一位主程序员，另外配备两到三位技术员、一位后援工程师组成。这种组织结构突出主程序员的领导，强调主程序员与其他技术人员的联系。

2. 民主制小组：在民主制小组中，遇到问题可以在组员之间平等地交换意见，工作组目标的制定以及决定的作出都由全体人员参加。这种组织形式强调发挥每个成员的积极性，并要求每个成员发挥主动精神和协作精神。

3. 层次式小组：在层次式小组中，组内人员分位三级：组长（项目负责人）一人负责全组工作，他直接领导两到三名高级程序员，每位高级程序员通过基层小组，管理若干位程序员。这种结构比较适合于项目本身就是层次结构的课题。

合理地配备人员是成功地完成软件项目的切实保证。所谓合理地配备人员应包括按不同阶段适时运用人员，恰当掌握用人标准。一般来说，软件项目不同阶段不同层次技术人员的参与情况是不一样的。在人力配备问题上，由于配置不当，很容易造成人力资源的浪费，并延误工期。特别是采用恒定人员配备方案时在项目的开始和最后都会出现人力过剩，而在中期又会出现人力不足的情况。

27.10 项目参考