

第1章

软件开发与管理概述

➔ 学习目标

[知识目标]

- 理解程序、软件、系统 3 个层次概念及其区别。
- 理解程序设计、软件开发、系统应用 3 个层次的活动。
- 了解软件开发过程主要阶段的概念与任务，包括需求分析、软件设计、编码实现、软件测试和软件维护等阶段。
- 了解几种常见的软件开发模型。
- 了解软件的结构化开发方法、面向对象开发方法。
- 了解软件开发辅助工具的相关概念。
- 了解软件项目管理内容和制作项目进度计划。
- 了解软件质量、质量管理及质量保证体系相关内容。

[能力目标]

- 能将软件项目按软件开发过程分解任务，并进行任务管理安排。
- 能用 Microsoft Project 制作项目开发进度计划，并打印出进度表。

1.1 软件与软件开发概述

1.1.1 程序、软件与系统

刚开始学习软件时，一般先是学“程序”的设计。但程序与软件是一回事吗？

程序是计算机执行代码组成的指令集。读者在刚开始学习程序设计时，往往是先学习程序设计基础知识，了解程序设计思想及逻辑算法的概念。但随着学习的深入，逐步到了具有一定“实用”价值的软件开发阶段，这时需要进一步学习软件开发的理论、过程规范等。

从学习阶段来讲，程序设计是软件设计学习的初期阶段；从软件组成来讲，程序设计是软件开发的组成部分之一。软件中的各个程序，是整个软件系统的有机组成部分，在软件的运行过程中，它们之间可能需要进行复杂的交互。这也说明了软件的结构复杂，软件开发与管理难度大。

1. 软件相关概念

计算机软件又称“软件”，是相对于计算机“硬件”的概念，它是具有可用性的逻辑“物品”，也是一种“产品”。计算机软件是具有使用价值的计算机（软件）产品的一个完整配置。

从产品构成上来说，软件由下面3个部分组成：

- (1) 程序代码。
- (2) 软件文档。
- (3) 数据结构和数据。

这3个组成部分，构成了完整的软件，程序代码只是软件的部分组成。而程序代码又是由各个部分的程序代码组成的一个整体，人们很容易将这个程序代码与软件概念等同。程序代码和软件的关系如图1.1所示。

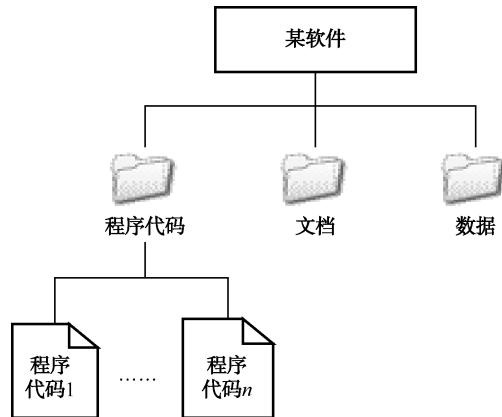


图 1.1 软件与程序代码的关系

这些程序代码整体集合，再加上软件文档、数据结构与数据等构成了软件的配置，也就是该软件。软件的开发，需要基于这些配置才能有效顺利地进行。软件开发的理论与方法是人们经历了长期的实践与探索才形成的一个较完整的体系。即从早期的程序设计的五花八门，到现在的软件工程思想、技术方法，再到软件开发标准与规范，逐步形成了现代的软件开发与管理理论。

系统是指为某个目标而有机地结合的一个较完整的整体，如硬件系统、软件系统、应用软件系统、财务软件系统等。系统具有“生命”与“边界”；系统的结构是由各相互合作的子系统组成的。系统的运行需要硬件条件与软件条件，是在人的操作下进行的。软件系统是计算机系统的一个必要组成部分。

程序、软件、系统（软件）是既有区别又有联系的概念，有着各自不同的范畴与知识领域，且理解角度不同。在学习软件开发时，要立足软件的范畴与体系，围绕“软件产品”的结构、功能、设计与实现过程等各方面的知识与技能。

2. 软件的特点

软件相对于硬件来说，具有以下特点：

- (1) 软件是逻辑的，而不是物理的产品。因为逻辑实际只存在于人的头脑当中，所以软件

的开发过程极难控制。

(2) 软件是由人开发形成的，没有明显的制造过程。软件成本集中于“开”上，因而软件项目不能完全像硬件制造项目那样来管理。

(3) 软件由人的“开发”劳动而成，到目前为止，软件开发尚未完全摆脱人的手工方式。所以，个人因素在软件开发过程中所占的重要性比重很大。

(4) 软件成本相当高，具体包括开发费用和维护费用。

(5) 软件本身是复杂的，维护困难且维护成本高。

软件的特点决定软件的开发与管理复杂。

3. 程序设计、软件开发及系统应用与支持

软件设计开发的学习是从程序设计开始的。这时程序设计的规模较小，实现的目标少、实用性不高。随着软件开发知识的深入，进入软件设计与开发领域知识与技能的学习。

程序设计是程序级代码的设计与实现，主要体现在程序处理过程中逻辑的设计与实现上，其相当于软件设计中的“详细设计”。而软件设计不是简单的程序设计的叠加，各个程序之间是有机联系的。由于软件设计复杂、规模大，一般软件设计需要一个宏观蓝图的规划，即所谓的“概要设计”。通过所谓概要设计将复杂的大问题分解成一些小问题，直到软件中程序代码的设计（即详细设计）。软件开发还包括需求分析、编码实现与软件测试等阶段。

所以，程序设计是在软件的“整体”结构下的各有机组成部分的程序代码的设计。

例如：一个财务管理软件，它包括该软件的程序代码、使用文档、数据与数据库等；而程序则是该软件中可以执行的程序文件、源程序文件等；而安装使用后，便与计算机硬件、网络等构成一个财务管理系统，而运行的财务软件部分则称为财务系统软件。

软件开发的最终产品是可供应用的软件系统。软件系统能为用户提供一定的应用服务。系统运行与应用对应的工作岗位有：软件系统实施、用户培训、系统操作、系统维护、系统技术支持等。

表 1.1 为软件开发各阶段特征及工作任务类型。

表 1.1 程序设计、软件开发和系统支持的区别

类 型	阶段特征及工作任务类型
程序设计	程序过程级的算法分析与设计、程序设计与实现
软件开发	包括软件的总体结构与程序过程级。任务有需求分析、软件设计、数据库设计、编码实现、软件测试等
系统支持	包括计算机硬件系统与软件系统的运行与支持。对应的任务有以系统的观点进行的系统认知、系统设计、系统开发、系统应用

1.1.2 软件开发的特征

1. 软件的设计与开发的复杂性特征

如前所述，软件本身是个复杂体，软件的设计与开发也是个复杂的过程。

软件设计与开发的复杂性特征如下：

(1) 软件要满足用户的使用需求，软件的设计要从用户需求开始。

(2) 软件设计开发常常是多人组成的团队进行的复杂工作，需要团队成员之间进行交流与合作。

(3) 软件开发“文档”在项目的开发与管理中是一个基础性内容，文档是团队成员交流与

合作的基础，也是管理与规范的基础。

(4) 软件的开发需要通过项目的管理与控制，使各方面工作秩序化。

(5) 某个软件开发团队的模式相对稳定，是由该团队的成员特点共同决定的，但团队的管理水平和能力有不同的层次，需要不断地改进。

(6) 软件的质量是软件设计与开发的生命。

2. 软件文档及其作用

软件开发知识：软件开发文档

在软件开发中，团队成员之间进行技术交流、分析与设计的表达、管理计划与过程控制、技术资料归档等，均需要软件文档。软件文档在软件开发过程中起着非常重要的作用。初学者开始时感觉不到软件文档的重要性，觉得它是可有可无的。但随着软件开发的深入，特别是团队形式的开发，文档的重要性就会逐步体现出来。

软件文档一般包括开发文档、产品文档、管理文档3类。

(1) 开发文档是描述开发过程本身的文档，如需求分析文档、软件设计文档、软件测试文档等。

(2) 产品文档是描述开发过程的产物，如培训手册、用户指南、产品手册、产品宣传册或广告等。

(3) 管理文档是记录项目管理的过程信息，如开发过程的每个阶段的进度和进度变更的记录、软件变更情况的记录、开发阶段评审记录、职责定义等。

软件文档具有以下作用：

(1) 软件文档是软件项目管理的依据。

(2) 软件文档是软件开发过程中各任务之间联系的凭证。

(3) 软件文档是软件质量的保证。

(4) 软件文档是用户手册、使用手册的参考。

(5) 软件文档是软件维护的重要支持。

(6) 软件文档是重要的历史档案。

软件开发所涉及的文档主要有以下几种。

(1) 可行性研究报告。

可行性研究报告说明该软件在技术上、应用上、经济上是否可行，即是否值得开发、是否开发得出、是否违背社会法律和人们的道德规范，如果可行则应该怎样开发等，供决策者参考。

(2) 项目开发计划。

项目开发计划是指为软件项目实施制订出具体的计划，该计划包括各部分工作任务的负责人员、开发进度、开发经费预算、所需要的硬件和软件资源等。

(3) 软件需求说明书。

软件需求说明书也称软件规格说明书，是对所开发软件的功能、性能、用户界面、运行环境等做出的详细说明。它是用户与开发人员双方对软件需求取得的一致理解和协议，通过该文档确定下来并作为今后开发工作的基础。

(4) 数据要求说明书。

数据要求说明书对数据的逻辑结构及各数据项的描述，以及对数据采集、数据约束的各项要求的说明，为今后生成和维护数据库、数据文档做准备。

(5) 概要设计说明书。

概要设计说明书是软件概要设计阶段的工作成果,是软件总体设计内容,它说明功能分配、模块划分、程序总体结构、输入/输出以及接口设计、运行设计、数据结构设计、出错处理设计等,为详细设计奠定基础。

(6) 详细设计说明书。

详细设计说明书描述每一模块内部是如何实现的,包括实现算法、逻辑处理流程等。

(7) 用户手册。

用户手册详细描述使软件正常运行的步骤,以及软件的功能、性能、用户操作界面,使用户了解如何使用该软件。

(8) 操作手册。

操作手册为操作人员提供了该软件各种运行情况的有关知识,特别是操作方法的具体细节、注意事项等。

(9) 测试计划。

为做好各种测试,需要为如何组织测试制订实施计划。计划应包括测试的内容、进度、条件、人员、测试用例等,并且包括各测试用例的执行步骤及预期结果和运行的偏差范围等。

(10) 测试分析报告。

测试用例执行完后,通过编写测试计划执行情况的说明,对测试结果予以分析,并提出测试结论意见,形成测试分析报告。

(11) 项目开发总结报告。

软件项目开发完成之后,通过总结软件开发执行情况,如进度、成果、资源利用、成本和投入的人力等进行总结,为今后的开发提供借鉴。同时还对整个开发工作进行评价,总结出经验和教训,以不断提高团队的能力。

1.2 软件开发

软件开发是个复杂的工作任务,且其中包括多个领域的知识,主要有以下几个方面:软件开发过程、软件开发方法、软件开发工具、软件项目管理等。

1.2.1 软件开发过程

软件的开发不同于程序阶段的开发,它是个复杂的系统工程。虽然软件开发是逻辑产品的创造过程,没有一个明显制造过程的特点,但在开发工作过程中的主要工作任务是明确的,即包括:需求分析、软件设计、编码实现、测试、项目管理等。这些任务有着各自明确的内容,它们之间内容虽不同但却相互联系与衔接,形成复杂的工作流程,这个复杂的工作流程就是所谓的软件开发过程。软件开发过程一般不是一个固定的模式。不同的相互衔接模式,构成了软件开发的工作过程的不同模型。

软件开发过程一般从问题定义,再到需求分析、软件设计、编码实现、软件测试,直到交付使用,最简单的过程如图 1.2 所示。

软件开发过程中的各个任务阶段之间的关系,不是一种简单的流程,如有线性的瀑布式、有螺旋上升的迭代式等。

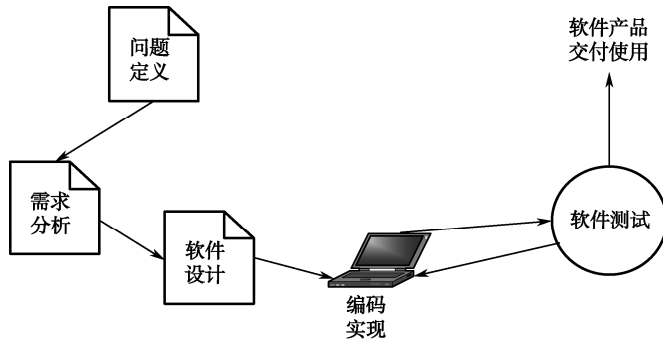


图 1.2 简单的软件开发过程示意

1. 软件开发过程的主要任务活动

软件开发知识：软件开发的工作任务

软件开发不只是程序编码工作，它还包括需求分析、软件设计、编码、软件测试、软件维护等工作任务。

软件开发过程一般包括的任务活动有：问题定义、需求分析、软件设计、编码实现、软件测试、软件维护等。

（1）问题定义。

从高层次了解“用户要计算机和软件做什么”，确定软件的功能及边界。只有了解了计算机和软件要干什么，才能安排好下一步的工作。该工作一般由系统分析师根据调研现实情况，通过精确的文字陈述出来。

（2）需求分析。

需求分析的任务是精确地描述软件系统必须“做什么”，确定系统具有哪些功能。该任务是由需求分析师通过分析得到软件的需求，并通过需求分析文档精确地表达出来。该文档是下一步软件设计的基础。

（3）软件设计。

软件设计的任务是软件设计师将软件要做的功能，即上一步的软件需求转化为要做的内容与规划蓝图，以设计文档的形式表现出来。软件设计回答“怎么做”的问题，包括宏观与结构层面的设计，以及各程序内部处理过程的设计，即所谓的概要设计和详细设计。

（4）编码实现。

编码实现是程序员将上一步的软件设计蓝图，通过某种程序设计语言一个一个地完成软件的程序代码的编写，然后将它们集成起来形成一个可使用的完整软件。

（5）软件测试。

软件测试是测试员将已经编写好的软件进行操作以发现问题。测试员测试的主要依据是软件需求、软件设计等。软件测试是软件质量保证的重要手段，测试要尽可能地发现软件中的问题，这是个非常复杂的工作，测试一般包括单元测试、集成测试、验收测试等。

（6）软件维护。

软件维护是软件交付使用后，为了保证软件正常的使用，以及满足用户使用时的各项要求而进行的维护工作。一般来说，如果软件在没被废弃前，都需要进行软件维护。

其实，上述各阶段任务是对要开发的“软件”的一个层次的抽象与描述。每个阶段需要上一阶段作为输入，再加上本阶段任务的“工作”，便是后续阶段的输入。软件开发各个阶段的

任务，需要一直做到用户使用软件满意为止。

软件开发各阶段中的任务和要完成的工作内容如表 1.2 所示。

表 1.2 软件开发任务对应的角色、职责

任 务	职 责	开发人员角色
需求分析	了解、分析用户需求，编写需求分析说明书	需求分析师
软件设计	根据系统要求设计软件总体架构；根据功能需求分析说明书设计软件功能模块并细化，编写软件设计说明书	软件设计师
编码实现	根据软件设计说明书及编码规范编写代码，并进行单元测试	程序员
软件测试	根据软件需求、设计文档设计软件测试用例、制定测试计划，并对软件进行集成测试和系统测试，编写测试报告。配合程序员修改完善代码并进行回归测试	测试员
软件维护	负责维护用户使用过程中出现的各种问题	维护人员
项目管理	制定开发计划，组织团队并制定开发规则，领导各开发人员顺利工作并进行开发过程控制（监督、检查成员工作情况，控制质量等），成员工作评估及激励等	项目经理

2. 软件开发过程模型

软件开发知识：软件开发过程模型

不同的软件开发企业可能采取不同的开发过程方式，并形成了相对固定的开发模式，这些相对固定的开发模式称为软件开发过程模型。这些过程模型有：瀑布模型、快速原型模型、增量模型、螺旋模型、统一软件开发过程模型等。

软件开发各个阶段的任务，贯穿于从产品的提出到开发完成并进行应用的各个阶段。但各工作阶段任务之间的连接不是完全相同的，有线性的、递增的、迭代的、螺旋上升的等不同情况，即所谓的不同软件开发过程模型，如图 1.3~图 1.6 所示。

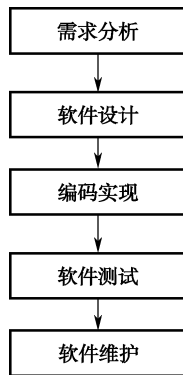


图 1.3 线性的瀑布模型

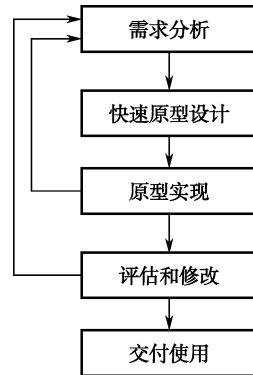


图 1.4 快速原型模型

(1) 瀑布模型。

瀑布模型是将软件开发过程的各个阶段任务规定为自上而下的线性过程，按照这种线性顺序连接的软件开发过程模型。该模型利于传统的结构化开发方法，但缺乏灵活性，用于解决由于需求的不确定性和不准确性带来的困难。

(2) 快速原型模型。

通过先快速建立反映用户需求的原型，以便与用户更好地沟通，通过它可以快速、全面地

和增量方法。每次迭代都具有需求、分析、设计、实现、测试等核心工作。强健的软件架构和设计，能避免源程序的无序堆砌；迭代和增量方法能避免一些风险。每一次迭代都通过任务驱动，如需求、风险等。UP 模型综合应用各种方法、技术、过程，以统一模型语言（UML）为整个开发过程的视图工具，将整个过程视为一个统一综合体。

UP 是通用的软件开发过程，在使用时必须为软件开发组织定制，然后为每个项目定制。因为所有的项目都不同，如果教条地使用 UP 方法并不能很好地工作。

1.2.2 软件开发方法学和软件工程

软件开发知识：软件开发方法

软件开发方法可以认为是用一套已定义好的某种技术集和工具来组织软件开发的过程。人们在软件开发中创建了很多种方法，并逐步形成了两种方法体系，即传统的结构化软件开发方法和面向对象的软件开发方法。

1. 软件开发方法学

软件开发领域除了编码，以及分析、设计、测试等技术方面外还有管理，所以其体系非常复杂、庞大。为了保证软件开发的成功，人们探索了一些软件开发的方法，它们对软件产业的发展起到了不可估量的作用。

软件开发方法是人们用已经定义好的技术集表示符号来组织软件生产过程的方法。其方法一般表示一系列的步骤，每个步骤都与相应的技术和符号相关。软件开发方法又称软件工程方法学，它是通过某种工程步骤、方法、工具形成的一种相对固定的工程与方法体系。

软件开发方法学包括两大类：传统的软件开发方法、面向对象的开发方法。

传统的软件开发方法又称结构化开发方法、面向过程的软件开发方法，它用结构化方法进行分析、设计、实现、测试、维护，并分别用结构化模型建立分析模型、设计模型。

面向对象的开发方法是以对象作为基本元素构建系统的方法，如面向对象方法分析阶段建立面向对象分析模型，设计阶段建立面向对象设计模型，开发语言采用面向对象程序设计语言。

采用面向对象程序设计语言，如 Java 语言编写程序，并不能说明采用的是面向对象的方法，因为完全可能采用传统的结构化方法的需求分析模型、软件设计模型，实现阶段采用面向对象语言。由于传统方法的各个过程的元素（以模块为单位）相关性不强、差异性大，这也是其软件开发成功率不高的原因；而面向对象的各个过程都是以“对象”的概念相关联，具有许多优点，利于修改与维护，软件开发成功率高。

传统的软件设计以“模块”为单位，常以面向对象的编码语言实现这些“模块”及“模块”之间的交互关系。面向对象的软件设计主要以“系统”与“对象”为单元，以供面向对象的编码语言实现。

两种方法的优缺点：模块与编码语言的元素没有对应，各个阶段的过渡难，各阶段之间容易“失真”。类与对象在面向对象的各个阶段都是一致的概念，所以面向对象开发的各个阶段过渡容易。

(1) 传统的软件开发方法采用结构化模型，即结构化分析模型、结构化设计模型工具来给软件建模。

(2) 面向对象软件开发方法采用面向对象模型，即面向对象分析模型、面向对象设计模型工具给软件建模。

2. 软件工程相关概念

软件开发方法学中，除技术方面（如分析技术、设计技术、编码技术、测试技术等）外，还有软件开发管理。而软件开发管理就是通过计划、组织和控制等一系列活动，合理地配置和使用各种资源（如人力资源、软硬件资源等），使工作有序地进行，以逐步达到既定开发目标。

软件工程的定义：采用工程的概念、原理和方法来开发和维护软件，把经过时间考验而证明的正确的管理技术和当前能够得到的最好的技术方法结合起来，这就是软件工程。

在软件开发过程中，技术和管理都非常重要，二者缺一不可。但长期以来，在实际工作中，重视技术的多，对过程管理重视的少。特别是对于大型的软件开发，没有好的过程管理，团队的开发容易处于无序状态，很难保证开发的进度与质量。

1.3 软件项目管理内容

通过软件分析与设计，从技术上将要做的任务搞清楚后，下面就是执行的问题了。技术和管理是软件开发过程中不可或缺的两个方面。

技术是软件开发过程中分析、设计、实现、测试、维护等采取的各种技术、方法，以完成所分配的任务。但这些工作任务能否有条不紊地顺利进行，还需要管理来加以控制。通过软件项目管理能保证在软件开发过程中有秩序地达到目标。软件项目管理贯穿于软件开发的始终。

例如通过软件设计，绘制了工作蓝图与要做的任务，技术上指导了开发的秩序化，但实际工作过程中，需要管理进行控制与落实。

所谓“项目”，是指在一定约束条件下（如人力、时间、资金、环境等）具有特定目标的一项要完成的一次性任务。而“项目管理”则可认为是在一个确定的时间范围内，为了完成一个既定目标，通过特定的临时性运行组织，有效地计划、组织、领导和控制，充分利用既有资源的一种系统管理方法。通过项目管理，使得该组织能有序地完成任

软件开发知识：项目的要素

根据“项目”的定义，项目主要包含要完成的任务、限定的时间、需要的成本、利用的资源等几个方面要素。

所以，软件项目管理则是在软件开发过程中，将项目进行分解、细化成不同的相互衔接的工作任务，通过制定计划、分配相应的人员和其他资源，进行监督、跟踪、协调与控制，使项目组织有序地工作，直至项目完成的过程活动。

因此，项目管理有：任务活动、时间、资源、成本等要素。其中，定义与分解任务活动、安排活动、估算活动资源是进行有效项目管理的关键。

引导案例

确定软件项目管理内容

软件项目管理内容主要包括：

- (1) 制定项目进度计划。
- (2) 人员的组织与安排。
- (3) 项目过程管理与控制。
- (4) 软件质量管理。

1.3.1 软件项目管理

1. 制定项目进度计划

项目开发是一个总任务,它是由许多子任务组成的,各个子任务还可以分解为更小的任务。这些任务如何合理地进行安排、分配资源、控制进度等,是项目管理者(如项目经理)要考虑的问题。项目管理者通过制定项目进度计划,为后续的工作进行安排,以使项目开发团队能顺利、协调地进行。

合理的项目计划,能合理地配置资源,发挥成员的工作能力,有利于协调工作,提高工作效率,使开发工作顺利进行。否则,会浪费资源,时间安排不准确,工作混乱,效率低。

案例实现

制定项目进度计划

制定项目计划是项目管理的重要一步,下面用 Microsoft Project 作为工作制定项目进度计划。制定项目计划需要确定以下几个方面的内容:

- (1) 确定工作总任务与各子任务。
- (2) 确定完成各任务的起止时间。
- (3) 确定完成任务需要的条件与资源。
- (4) 确定完成任务需要的具体人员。

一个项目进度计划有如下几个因素:工作任务、时间(起止时间)、人力资源与其他资源与条件等。例如,某个项目的计划安排,有需求分析、软件设计、编码、测试、验收等5个任务。完成的起止日期为2012年2月27日到2012年4月4日。目前的人员有5人,可以承担这些任务,并且要添置5台计算机。图1.7就是利用 Project 制定的该项目的进度计划。



标识号	任务名称	工期	开始时间	完成时间	前置任务	资源名称
1	需求分析	3 工作日	2012年2月27日	2012年2月29日		张学兵, 电脑6台
2	软件设计	6 工作日	2012年3月2日	2012年3月9日		李峰
3	编码	6 工作日	2012年3月12日	2012年3月19日	2	王刚
4	测试	8 工作日	2012年3月20日	2012年3月29日		曾小强
5	验收	2 工作日	2012年4月3日	2012年4月4日		吴强

图 1.7 某项目的进度计划

制定好 Project 项目后,选择“文件”/“打印预览”或“打印”命令,就可以进行项目计划的预览和打印了(见图1.8,当然还可以选择不同的格式打印)。

项目管理者要制定合理、准确的项目进度计划,对需要做的任务、时间、资源进行合理的安排。

项目总体任务要求有一个总体计划,各个阶段要有阶段计划。制定计划时,管理人员要根

据任务的性质及要求，合理安排人力和物力等资源进行制定，制定好的工作计划也是项目管理者能力之一。

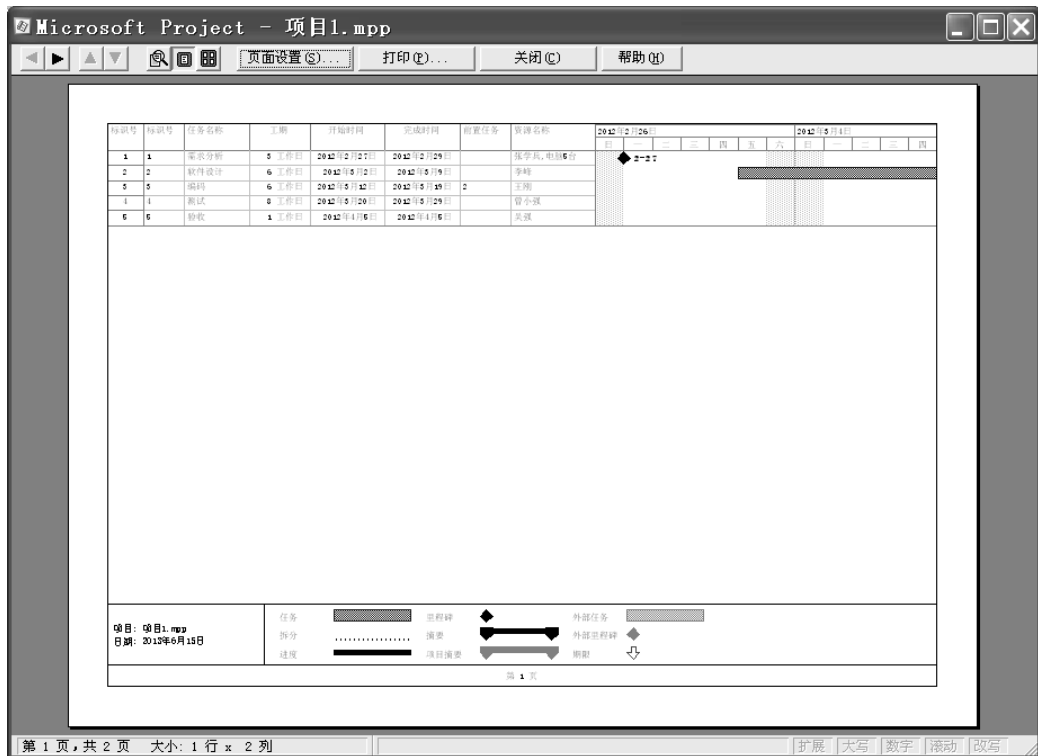


图 1.8 项目计划表的打印（多页）

项目的任务有串行进行的，有并行进行的，还可能有子任务，各子任务之间可能有衔接关系。管理者在安排计划前要对任务的规模和自己能调配的人力、物力、财力及解决的方案等心中有数，才能制定好的计划。

2. 人员的组织

软件开发项目中主要是人的开发，人是第一资源。保证软件开发进度与质量的关键是人。有效管理的关键是对人员的管理。日常工作时安排、监督、跟踪、考核评价员工的工作，平时培训员工、激励员工，提高他们的能力与素质等，都是对人员及其工作的管理。

在安排人员过程中，要根据工作任务的性质、人员的岗位能力，合理地配置。软件项目的人员有：管理者、分析员、设计者、程序员、测试员、客户等。项目人员的组织结构与工作安排会根据项目开发管理模式的不同而不同。

一般的软件开发项目模式有：主程序员式、民主分权式、层次式等。项目管理者要根据自己项目成员的特点，选择好适当的开发模式，安排好成员的职责与工作。

（1）人员的组织。

新项目组创建后，需要安排任务给相应人员，如果人员不够则要进行新成员的招聘。软件项目的组成人员一般有：管理者、分析员、设计员、程序员、测试员等，有时一个人可兼多职。一个项目组一般会有一位经验丰富、善于管理的组织者——项目经理，以及承担各个职责的项目成员角色。

(2) 项目人员的任务分工。

当项目小组成立后,就可以进行任务分工,正式进行开发。任务分工要根据各个成员的能力安排任务,并落实到项目进度计划中。项目进度计划是项目管理的重要管理文档,通过该文档,可以监督与跟踪日常项目的进展情况。

(3) 制定项目计划并执行。

将项目任务分工通过制定项目计划落实下来,然后各个成员就可以根据这些计划进行工作,按计划中的要求执行相应的任务。管理者则通过该计划进行项目监督与控制。

3. 项目过程管理

项目进度计划已经制定了,各成员的任务也分工了,他们现在已经在按照计划执行各自的任務,但是否意味着管理者的任务就完成,只等最后的结果了呢?项目成员的各项工 作是否在按计划顺利进行,是否出现问题,是否需要调整等,都需要管理者在过程中进行跟踪、协调、控制。同时,管理者也需要对员工进行考核与激励,以利于团队更紧密和谐地工作,不断改进工作过程。

项目过程是对软件开发项目进行执行的过程。该过程首先是对项目计划的执行,然后进行任务执行情况记录、工作协调、任务检查与跟踪、调整计划、阶段任务的完成及新任务安排与衔接、工作的评价与考核等,直到整个工作完成。项目过程管理的主要工作内容包括:

- (1) 工作监督、检查、跟踪。
- (2) 计划调整与工作协调。
- (3) 员工考核与激励。
- (4) 软件质量管理等。

软件过程管理也需要文档支持,成为管理文档,如工作计划、过程日志及工作报告等。

1.3.2 软件质量管理

软件开发知识: 软件质量

什么是软件质量?按照 GB/T16260—1996“信息技术软件产品评价、质量特性及其使用指南”标准,软件质量是与软件产品满足明确或隐含需求的能力有关的特征和特性的总和。简而言之,软件质量是软件一些特性的组合,它仅依赖于软件本身。

质量是软件的固有属性,是软件的生命,如图 1.9 所示。软件的质量包括:正确性、健壮性、效率、完整性、可用性、可维护性、可移植性等,它们均难以用定量方式度量。就像其他产品可以通过质量管理来保证质量一样,软件质量也是通过软件质量管理来保证的,如图 1.10 所示。

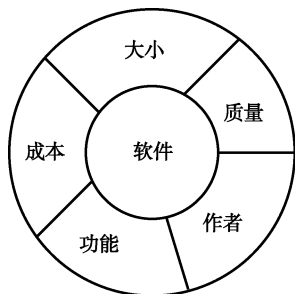


图 1.9 质量是软件的固有属性

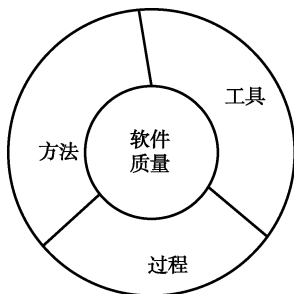


图 1.10 软件质量管理

软件开发知识：软件过程与软件的质量

软件是软件开发出来的成果，但软件的质量与软件开发过程的规范化有直接关系，即规范化的软件开发过程能得到较好质量的软件；否则，开发过程混乱的软件质量难以得到保证。

软件质量与软件开发的什么因素有关系？人们通过长期实践发现，高质量的生产过程能生产高质量的产品，它们之间有一定的关联关系。所以，软件质量管理内容包括事后的产品质量检测，以及软件生产全过程的规范化管理。

软件质量管理需要通过建立质量保证体系予以保证。质量保证（QA）就是建立起机构质量规程和标准的整体框架，这是生产高质量软件的保证。QA 过程包括对软件开发过程标准和软件产品标准的制定。质量管理包括 3 大方面的内容：质量保证、质量规划、质量控制。

（1）软件生产过程中：制定过程标准，贯彻标准，执行标准，过程控制，审查、复查。包括建立基本的质量标准体系，并贯彻执行；日常生产过程中按照过程标准执行，并进行监督检查与控制。

（2）软件成品：软件测试，软件评审。其中需要对产品标准进行定义，并通过测试与评审来确定是否达到质量要求。

软件开发知识：确定软件开发质量管理内容

软件项目质量管理的内容：

- （1）成立软件质量管理小组，制定项目开发质量标准或项目小组内的开发标准。
- （2）制作软件开发文档编写标准，并根据该标准制定软件开发文档模板与要求。
- （3）项目开发过程的评审、复审和软件测试，并进行质量汇报。
- （4）监督跟踪开发部门发现的质量问题及其解决情况。
- （5）对影响软件质量的开发过程标准提出修改与完善意见。

1. 制定并执行保证软件质量的规范

软件企业有自己的软件开发质量规范，具体的软件开发项目也有项目的质量规范，如设计标准、编码规范、测试计划、文档编写规范等。项目质量管理者在软件开发前先制定项目应遵循的保证软件质量的规范，然后在软件开发过程中进行贯彻执行。在实际的软件开发过程中，质量经理负责审查标准的实施情况，及时发现质量问题，督促小组其他成员严格按照标准来操作，并进行指导与调整完善。

2. 制作规范的软件开发文档

软件开发文档是对整个软件开发过程的记录和说明。如同程序代码，软件文档也是软件中不可缺少的重要组成部分。软件开发文档对今后软件使用过程中的维护、软件版本的升级等都有着非常重要的意义。软件是否有完整与规范的软件开发文档已经被看作一个衡量软件过程质量的重要标准。软件开发文档编写格式可参考本书附录 A “计算机软件开发文件编制指南 GB/T8567—1988”。

3. 软件评审

在软件开发的各个阶段都可能产生错误，如果不及时发现并纠正这些错误，会不断地扩大影响，最后可能造成巨大的损失甚至失败。软件评审是在软件开发的各个阶段结束前，对该阶段产出的软件配置成分进行严格的技术审查，以及时发现错误。软件评审是一项非常重要的工作。

4. 复查和管理复审

软件开发每个阶段开始前，对前期工作进行复查，看是否满足当前工作所必需的材料。管理复审是从管理的角度对开发工作进行复审，可根据情况对计划做进一步的调整，并对可能出

现的突发事件和风险做好安排与准备。

5. 软件测试

软件测试是软件开发的一个重要的环节，同时也是软件质量保证的一个重要环节。软件的测试一般有单元测试、集成测试、系统测试等。如果测试的结果与预期结果不一致，则很可能是发现了软件中的错误。

在全面质量管理阶段，质量管理和质量保证是同一概念。20世纪90年代以来，全面质量认证逐渐在企业流行。质量保证也扩展到整个企业的质量认证。现在软件质量管理的内容也包括质量保证和质量认证。

目前，比较流行的软件开发过程质量保证体系有 ISO9000 标准、CMM（软件开发能力成熟模型）标准等。

1.3.3 软件过程质量保证体系

软件开发知识：软件过程质量保证体系

所谓的质量保证体系是指企业以提高和保证产品质量为目标，运用系统方法，依靠必要的组织结构，把组织内各部门、各环节的质量管理活动严密组织起来，将在产品研制、设计制造、销售服务和情报反馈的整个过程中影响产品质量的一切因素统统控制起来，形成的一个有明确任务、职责、权限，相互协调、相互促进的质量管理的有机整体。质量保证体系分为内部质量保证体系和外部质量保证体系。

可以通过第三方认证确定企业的生产过程是否已达到了某种质量标准。软件的质量管理标准可以采用 ISO9000 相关的要求，也可以采用 CMM 进行评价。

质量保证体系是指企业为了提高和保证产品质量，制订与执行的一套企业内各部门、各环节相关的质量管理体系；它使企业的质量管理形成各职责明确、工作相互协调的有机体系。但确定是否满足某种质量管理要求，需要通过专业机构认证。质量体系的认证是指对产品与服务的供方的质量体系进行第三方的评定或注册活动。其目的在于，通过评定和事后监督来证明供方的质量体系符合并满足需方对该体系规定的要求。

软件质量保证体系的标准根据建立机构与适用范围不同而不同，比较流行的有国际标准化组织 ISO 推出的 ISO9000 标准化体系，以及美国国防部委托卡内基·梅隆大学软件工程研究所（SEI）推出的评估软件能力与成熟度等级的一套标准 CMM。

1. ISO9000 体系

ISO 是国际标准化组织（International Organization for Standardization）的简称，它的前身是国际标准化协会，即国际联合会。ISO 的宗旨是在世界范围内促进标准化的工作及其有关活动的开展，以有利于国际间的物资交流和相互服务与合作。ISO 的工作领域涉及除电工、电子以外的所有学科。

ISO9000 是国际标准化组织于 1987 年公布的第一套质量管理和质量保证标准，其主要用途是为企业建立质量体系，并提供质量保证的模式。ISO9000 标准系列是一个大的家族。ISO9000 以一般术语描述了能够适用于任何行业的质量保证系统的要素，这些要素包括用于实现质量计划、质量控制、质量保证和质量改进所需的组织结构、程序、过程和资源。软件企业贯彻实施 ISO9000 认证应选择 ISO9001 标准。ISO9001 规定了设计、开发、生产、安装和服务中的质量保证模式。

软件企业要获取 ISO9000 认证一般需要经过以下流程：

- (1) 企业申请。
- (2) 认证机构审核受理。
- (3) 文件材料审查。
- (4) 审核准备。
- (5) 现场审核。
- (6) 跟踪、验证，纠正不合格项。
- (7) 审核合格后颁发证书。
- (8) 证后监督。

在我国，ISO9000 工作开展得较早，1988 年开始推行 ISO9000，且国内有认证机构。ISO9000 是基础性的质量保证体系，在我国通过 ISO9000 认证的企业相对于 CMM 要多。

2. 软件配置管理

软件配置管理首先要了解什么是软件配置项。软件配置项（Software Configuration Items, SCI）是软件配置管理的对象，它包括软件生存周期内产生的所有信息项。软件配置项一般包括：

- (1) 与合同、源代码、过程、计划和产品有关的文档及数据。
- (2) 目标代码和可执行代码。
- (3) 相关产品，如软件工具、库内可复用件、外购软件等。

软件配置就是软件配置项在不同时期按不同要求进行的组合。例如：Java JDK 有 JDK 5.0、JDK 6.0 等不同版本。实际工作中，一般用“版本”来表示配置项的演化阶段。

随着软件开发过程的进展，软件的配置项也在迅速增长，并且发生不断的变化。这些变化的原因主要表现在：新的商业或市场环境的变化，引起产品需求或业务规则变化；用户有新的要求以及企业结构发生变化等。

软件配置管理就是对计算机软件在整个生命周期内各个阶段管理变化的活动。软件配置管理也是一个管理学科，它对配置项的开发和支持生存周期给予技术上和管理上的指导。实施软件配置管理主要有以下任务：

- (1) 制订配置管理计划。
- (2) 确定配置标识。
- (3) 进行配置控制，实施变更管理。
- (4) 配置审计。
- (5) 记录并报告配置状态。
- (6) 版本控制。
- (7) 发行管理和交付。

3. 能力成熟度模型

能力成熟度模型（Capability Maturity Model for Software, CMM），是对于软件组织在定义、实施、度量、控制和改善其软件过程的各个发展阶段的描述。CMM 是一种用于评价软件承包能力并帮助其改善软件质量的方法，侧重于软件开发过程的管理及工程能力的提高与评估。

所谓的软件过程能力，是描述开发组织或项目组通过执行其软件过程能实现预期结果的程度。而软件过程成熟度，是指一个特定软件过程被明确和有效地定义、管理、测量和控制的程度。成熟度可以指明开发组织软件过程能力的增长潜力。CMM1.1 将软件过程的进化步骤分成以下 5 个等级。

- (1) 初始级（混沌的软件过程）。
- (2) 可重复级（经过训练的软件过程）。
- (3) 已定义级（标准一致的软件过程）。
- (4) 定量管理级（可预测的软件过程）。
- (5) 优化级（能持续改善的软件过程）。

其实，CMM 还为软件企业的过程能力提供了一个阶梯式的改进框架，CMM 这 5 个等级的特征是：

(1) 一级（初始级 CMM1）。这时软件生产过程的特征是随机的，有时甚至是混乱的。很少过程被定义，一般达不到进度和成本的目标，软件产品的质量具有不可预见性。成功依赖于个人的技术与努力。

(2) 二级（可重复级 CMM2）。建立基本的项目管理过程，以跟踪费用、进度和功能。设定必要的过程纪律以重复以往相同应用项目的成功。以前的开发经验成为开发新产品能否成功的关键因素。

(3) 三级（已定义级 CMM3）。形成了管理软件开发和维护活动的组织标准软件过程，包括软件工程过程和软件管理过程。项目依据标准，定义了自己的软件过程，并且能进行管理和控制。组织的软件过程能力已描述为标准的和一致的，过程是稳定和可重复的，并且高度可视。

(4) 四级（定量管理级 CMM4）。详细的软件过程和产品质量的特征已被收集。软件过程和产品已被定量管理和控制，软件产品具有可预测的高质量。

(5) 五级（优化级 CMM5）。能自觉利用各种经验和来自新技术、新思想的先导实验的定量反馈信息，不断改进和优化组织统一的标准软件过程。整个软件机构的重心转移到优化软件过程中。

软件过程能力成熟度模型有两个基本用途：软件过程评估和软件能力评价。软件过程评估用于确定一个组织当前软件过程的状态，找出组织所面临的急需解决的与软件过程有关的问题，进而有步骤地实施软件过程改进，使组织的软件过程能力不断提高。软件能力评价可以识别合格的能完成软件工作的承制方，或者监控承制方现有软件开发工作中软件过程的状态，进而指出承制方应改进之处。

从当今整个软件行业现状来看，最多的成熟度为一级，多数成熟度为二级，少数成熟度为三级，极少数成熟度为四级，成熟度为五级的更是凤毛麟角。

CMM 是目前国际上最流行、最实用的一种软件生产过程标准，已经得到了众多国家以及国际软件产业界的认可，成为当今企业从事规模软件生产不可缺少的一项内容。

CMM 为软件企业的过程能力提供了一个阶梯式的进化框架，阶梯共有 5 级。第一级实际上是一个起点，任何准备按 CMM 体系进化的企业都自然处于这个起点上，并通过这个起点向第二级迈进。除第一级外，每一级都设定了一组目标，如果达到了这组目标，则表明达到了这个成熟级别，可以向下一个级别迈进。CMM 体系不主张跨越级别的进化，因为从第二级起，每一个低的级别实现均是高的级别实现的基础。

CMM 是专门针对软件产品开发和服务的，而 ISO9000 涉及的范围则相当宽。CMM 强调软件开发过程的成熟度，即过程的不断改进和提高，而 ISO9000 则强调可接收的质量体系的最低标准。

小 结

本章从阐述程序、软件、系统 3 个不同层次的概念，进而导出程序设计、软件开发、系统应用支持 3 个不同类型的活动，使读者了解软件开发的任务是什么，正确地对待“软件”开发与管理过程的理论、方法、技术。

针对软件、软件开发的特点，阐述了软件开发模型、软件开发方法学、软件开发管理领域的知识体系。软件开发一般经历需求分析、软件设计、编码实现、软件测试、软件维护等任务过程，但因这些任务的完成过程不同而形成了不同的软件开发过程模型；完成这些任务可采用不同的方法与工具，如传统的方法、面向对象方法；对每个过程的任务的规范要求与管理形成了软件项目管理体系。

软件项目管理包括：任务、时间、成本、资源等。如何将这些因素分配与控制好便是项目管理的内容，而制定项目进度计划是项目管理的具体落实与体现。

软件的质量是软件的生命，一个质量有问题的软件难以被用户认可。本章后半部分介绍了软件质量、软件质量管理等概念，以及常见的软件开发质量保证体系 ISO9000 和 CMM（软件开发能力成熟度模型）。另外，本章多次强调软件文档的重要性，以及在软件开发与管理过程中的作用。

习 题

一、填空题

1. 软件由_____、_____、_____3 部分组成。
2. 软件作为人工生产的_____产品，主要由人进行“开发”而来，与硬件比较起来没有明显的制造过程。
3. 软件开发过程复杂，但一般包括_____、_____、_____、_____等主要阶段。
4. 软件的_____是软件的生命。
5. 在软件团队开发中，由于人数的增加、成员间的交互与合作，所以除技术问题外，还存在_____问题。
6. 软件开发方法学包括_____和_____两种开发方法。
7. 软件项目进度计划主要包括_____、_____、_____和_____。

二、思考与简答题

1. 举例说明程序、软件、系统的区别和联系。
2. 软件有什么特点？软件开发有什么特点？
3. 简述程序、软件、系统的设计与应用的不同。
4. 软件开发过程主要有哪几个典型任务活动？请简述它们各自的内容。
5. 软件开发过程模型是什么含义？一般有哪几种开发过程模型？
6. 软件开发方法有哪两类？它们各自的优缺点是什么？

7. 请简述管理在软件开发中的重要作用。软件项目管理有哪些内容?
8. 软件质量是如何管理的?

试一试：制定项目进度计划

实训：制定某项目进度计划

(一) 实训内容与实验环境

用 Microsoft Project 项目管理工具制作一个简单的项目进度计划。该计划包括：任务、时间、人员的安排，并通过不同的视图打印或显示出来。

实验环境：

- Microsoft Word。
- Microsoft Project。

(二) 实训目的与要求

(1) 理解项目计划，会制定项目进度计划（包括总任务、子任务、前置任务与后续任务、资源、起止时间等的配置）。

(2) 会操作 Microsoft Project 2003 工具制定项目计划，并进行报表输出。

(三) 实训方案与步骤

1. 实训方案

一个项目的开发有一个过程与步骤，包括分析任务、设计任务、编码实现、测试、编写操作手册等任务。要求制定一个项目计划，适当地安排任务、工期、前置任务和后续任务、子任务、资源（主要是各类人力资源或设备等）等，并进行报表输出。

2. 实训步骤

(1) 先手工制定一个项目开发计划，然后用 Microsoft Project 制作该项目计划并生成打印报表。

(2) 过程与步骤：先创建一个“项目”文件，再在“项目”菜单选择“项目信息”输入项目的开始日期及完成日期等输入项；然后再输入任务、日期、资源等；最后以浏览的形式将计划显示出来（可以尝试不同的输出形式）。

(3) 用 Microsoft Project 以不同的形式输出该项目。

(4) 设置计划的每个任务的前置任务和后续任务、子任务、资源、起止时间、报表打印格式等，并将此项目文件进行保存与打印。