

新工科建设之路·软件工程规划教材



# 软件过程： 原理、方法与工具



张剑波 方芳 周顺平 编著

 中国工信出版集团

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

新工科建设之路·软件工程规划教材

# 软件工程过程： 原理、方法与工具

张剑波 方 芳 周顺平 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书以 IEEE 计算机协会 2014 年 2 月发布的 SWEBOK V3 为蓝本, 以软件工程过程、模型与方法为主线, 围绕其中 8 个主要的软件工程实践活动, 详细介绍了在软件工程领域被广泛接受的知识域。全书分 10 章, 主要内容包括: 软件工程过程、软件工程模型与方法、软件需求、软件设计、软件构造、软件测试、软件维护、软件配置管理、软件项目管理、软件质量等。

本书适合作为高等学校软件工程及计算机相关专业的研究生、高年级本科生教材, 也适合软件工程专业人员及希望从事软件工程相关工作的其他专业人员阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

### 图书在版编目 (CIP) 数据

软件工程过程: 原理、方法与工具 / 张剑波, 方芳, 周顺平编著. —北京: 电子工业出版社, 2019.8  
ISBN 978-7-121-36661-1

I. ①软… II. ①张… ②方… ③周… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2019) 第 100394 号

策划编辑: 冉 哲

责任编辑: 底 波

印 刷: 北京盛通商印快线网络科技有限公司

装 订: 北京盛通商印快线网络科技有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 16.75 字数: 422.4 千字

版 次: 2019 年 8 月第 1 版

印 次: 2019 年 8 月第 1 次印刷

定 价: 49.80 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 [zltts@phei.com.cn](mailto:zltts@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式: [ran@phei.com.cn](mailto:ran@phei.com.cn)。

# 前 言

软件是可以运行在计算机及电子设备中的指令和数据的有序集合。随着人类社会的发展和技术的进步，现在的软件具有产品和产品生产载体的双重作用。作为产品，软件在不同硬件环境（移动电话、手持平板、台式机或者大型计算机）中扮演着信息转换的角色；作为产品生产的载体，软件提供了计算机控制、信息通信、应用程序开发与控制的基础平台。相比计算机硬件能力以每两年提高一倍的速度发展，软件开发则面临着复杂性（Complexity）、不可见性（Invisibility）、易变性（Changeability）、服从性（Conformity）、非连续性（Discontinuity）等亟待解决的难题。

软件工程是指将系统化、规范化和可量化的工程化方法应用于软件的开发、运行和维护，包括过程、一系列实践方法和大量工具，用于帮助专业人员构建高质量的计算机软件。同时，软件工程也是一门独立的学科，有自己的课程教育体系。为了规范和推动软件工程理论研究、工程实践和教育的发展，2014年2月20日，IEEE计算机协会发布了软件工程知识体系（Software Engineering Body of Knowledge, SWEBOK）指南第3版，简称SWEBOK V3。该指南将SWEBOK2004第2版（软件工程本科课程大纲）、GSWE2009（软件工程硕士课程大纲）、CSDP（软件开发工程师认证）和CSDA（初级软件工程师认证）、SEVOCAB（系统与软件工程术语）等标准进行了统一，同时补充了近10年来软件工程研究与实践的新成果，包括11个软件工程实践知识域和4个软件工程教育基础知识域（软件工程经济学、计算基础、数学基础和工程基础）。该指南描述了软件工程学科内容的特征，明确定义了软件工程相对于计算机科学、项目管理、计算机工程和数学等其他学科的范围。本书内容的组织正是以该指南为参考依据的。

与国内外已经出版的同类书籍相比较，本书围绕“软件工程过程、模型与方法”，针对软件需求、软件设计、软件构造、软件测试、软件维护、软件配置管理、软件项目管理和软件质量这8个主要软件工程实践活动（见图1），通过原理、方法与工具的介绍，为读者构建了一个完整的软件工程过程的知识体系，帮助读者更好地将所学知识应用于实践，并运用软件工程方法解决实际问题。

通过本书的学习，读者可以掌握软件生命周期、软件工程过程中包含的主要活动和软件工程模型；理解常规软件生命周期框架中所包含的各阶段和活动；理解软件开发中的8个主要活动间的相互关系；掌握几种主要的软件工程模型的主要特征及适用范围；理解为了在进度和预算内获取高质量的软件，选择合适的软件工程模型的重要性；全面掌握软件项目的开发和管理技术，能运用软件工程方法鉴别和解决大型软件项目开发过程中出现的主要问题。

在章节安排上，本书首先为读者建立完整的软件工程过程的概念体系，接着详尽地给出各种常用的软件生命周期模型、建模过程与方法；然后以软件开发过程中工作产品构建时所执行的一系列活动、动作和任务的集合为参考，对其中软件需求、软件设计、软件构造、软件测试和软件维护这5个框架活动进行详细说明，帮助读者掌握软件工程过程从沟通、策划、建模、构造到部署这5个环节涵盖的基本原理、方法与工具；最后，对软件配

置管理、软件项目管理和软件质量这三个普适性活动进行详细介绍，它们包含软件工作产品的准备和生产过程中所必需的活动（建模、文档、日志等）集合，为读者深入学习软件工程过程管理、过程监控及过程改进等软件管理知识打下了良好的基础。

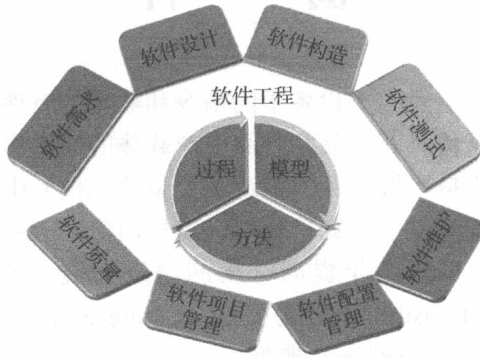


图 1

本书的编写得到中国地质大学（武汉）2018 年研究生精品教材建设项目的资助。

本书在编写过程中得到了中国地质大学（武汉）软件工程系老师们的指导和帮助，研究生赵素彬和刘明焱参加了资料整理、绘图和排版等工作，在此一并致谢。

由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

编者

# 目 录

第 1 章 软件工程过程 .....	1
1.1 软件过程定义 .....	1
1.1.1 软件过程管理 .....	2
1.1.2 软件过程框架 .....	2
1.2 软件生命周期 .....	5
1.2.1 软件过程分类 .....	6
1.2.2 软件生命周期模型 .....	8
1.2.3 软件过程适应 .....	18
1.2.4 实践考虑 .....	18
1.3 软件过程评估与改进 .....	18
1.3.1 软件过程评估与改进模型 .....	19
1.3.2 软件过程评估方法 .....	19
1.3.3 连续式和阶段式软件过程评估 .....	19
1.4 软件过程工具 .....	32
习题 1 .....	33
第 2 章 软件工程模型与方法 .....	34
2.1 建模 .....	34
2.1.1 建模的原则 .....	34
2.1.2 模型的性质与表达 .....	35
2.1.3 语法、语义和语用 .....	35
2.1.4 前置条件、后置条件和不变量 .....	36
2.2 模型的类型 .....	36
2.3 模型分析 .....	37
2.4 软件工程方法 .....	38
2.4.1 启发式方法 .....	38
2.4.2 形式化方法 .....	39
2.4.3 原型方法 .....	39
2.4.4 敏捷方法 .....	40
习题 2 .....	40
第 3 章 软件需求 .....	41
3.1 基本概念 .....	41
3.1.1 软件需求定义 .....	41
3.1.2 软件需求层次 .....	42
3.1.3 软件需求分类 .....	43

3.1.4	需求工程	45
3.1.5	启动步骤	46
3.2	需求获取	48
3.2.1	软件需求来源	49
3.2.2	需求获取技术	50
3.3	软件需求分析	54
3.3.1	分析模型概述	55
3.3.2	建立分析模型	57
3.3.3	分析技术	59
3.3.4	架构设计	60
3.3.5	需求协商	61
3.4	软件需求规格说明	61
3.4.1	意义	62
3.4.2	分类	62
3.4.3	描述方法	63
3.5	软件需求确认	65
3.5.1	软件需求评审	65
3.5.2	原型法	67
3.5.3	软件需求测试	68
3.5.4	验收测试	68
3.6	软件需求管理	69
3.6.1	需求基线	70
3.6.2	需求跟踪	72
3.6.3	需求变更	74
3.7	软件需求工具	76
	习题 3	77
<b>第 4 章</b>	<b>软件设计</b>	<b>78</b>
4.1	软件设计基础	78
4.1.1	软件设计过程	81
4.1.2	软件设计原则	82
4.2	软件架构设计	86
4.2.1	软件架构风格	87
4.2.2	软件架构设计方法	88
4.2.3	软件架构设计步骤	92
4.3	用户界面设计	95
4.3.1	通用用户界面设计原则	96
4.3.2	用户交互模式设计	98
4.3.3	用户界面设计流程	99
4.3.4	用户界面设计方法	100
4.4	软件设计质量	102

4.4.1	软件设计质量的意义 .....	102
4.4.2	软件设计质量的评估 .....	104
4.5	软件设计符号 .....	105
4.5.1	结构描述 .....	106
4.5.2	行为描述 .....	115
4.6	软件设计策略和方法 .....	124
4.7	软件设计工具 .....	125
习题 4	.....	126
<b>第 5 章</b>	<b>软件构造</b> .....	<b>127</b>
5.1	软件构造基础 .....	127
5.1.1	复杂性最小化 .....	127
5.1.2	多维视角的软件构造 .....	127
5.2	软件构造过程 .....	128
5.2.1	生命周期模型 .....	128
5.2.2	构造语言 .....	129
5.2.3	开发者测试 .....	131
5.2.4	重构 .....	132
5.3	软件构造管理 .....	133
5.3.1	变更管理 .....	133
5.3.2	版本控制 .....	133
5.4	软件构造技术 .....	134
5.5	软件构造工具 .....	136
习题 5	.....	137
<b>第 6 章</b>	<b>软件测试</b> .....	<b>138</b>
6.1	软件测试基础 .....	139
6.1.1	软件测试目的 .....	139
6.1.2	软件测试定义 .....	139
6.2	软件测试级别 .....	140
6.2.1	测试阶段级别 .....	140
6.2.2	测试对象级别 .....	144
6.3	软件测试技术 .....	147
6.3.1	静态测试 .....	147
6.3.2	动态测试 .....	150
6.3.3	白盒测试 .....	150
6.3.4	黑盒测试 .....	153
6.3.5	自动化测试 .....	158
6.4	软件测试过程 .....	159
6.4.1	测试计划阶段 .....	160
6.4.2	测试设计阶段 .....	163

6.4.3	测试执行阶段	165
6.4.4	测试监控阶段	166
6.4.5	测试结束阶段	167
6.5	软件测试工具	168
6.5.1	静态分析工具	168
6.5.2	黑盒测试工具	168
6.5.3	单元测试工具	169
6.5.4	负载测试工具	169
	习题 6	170
<b>第 7 章</b>	<b>软件维护</b>	<b>171</b>
7.1	软件维护基本概念	171
7.1.1	软件维护定义	172
7.1.2	软件维护特点	172
7.1.3	软件维护目的	173
7.1.4	软件维护组织	173
7.2	软件维护关键问题	174
7.2.1	软件维护技术问题	174
7.2.2	软件维护管理问题	175
7.2.3	软件维护成本预算	176
7.2.4	软件的可维护性	177
7.3	软件维护过程	179
7.3.1	软件维护过程概述	179
7.3.2	软件维护活动	180
7.4	软件维护技术	181
7.4.1	程序理解	181
7.4.2	再工程	182
7.4.3	逆向工程	184
7.4.4	迁移	185
7.4.5	退役	186
7.5	软件维护工具	186
	习题 7	187
<b>第 8 章</b>	<b>软件配置管理</b>	<b>188</b>
8.1	软件配置管理的过程管理	189
8.1.1	软件配置管理的组织背景	189
8.1.2	软件配置管理涉及的人员	189
8.1.3	软件配置管理计划	190
8.1.4	软件配置管理的监管	191
8.2	软件配置标识	192
8.2.1	被管控项目的识别	192

8.2.2 软件库 .....	199
8.3 软件配置控制 .....	201
8.3.1 软件变更请求 .....	201
8.3.2 跟踪并控制变更 .....	202
8.3.3 软件配置偏差和弃用 .....	203
8.4 软件配置状态统计 .....	203
8.5 软件配置审计 .....	204
8.5.1 配置库审计 .....	205
8.5.2 基线审计 .....	205
8.6 软件构建和发布管理 .....	206
8.6.1 软件构建 .....	206
8.6.2 软件发布 .....	207
8.7 软件配置管理工具 .....	208
习题 8 .....	208
<b>第 9 章 软件项目管理</b> .....	<b>209</b>
9.1 软件项目管理概述 .....	210
9.1.1 项目与软件项目 .....	210
9.1.2 项目管理 .....	210
9.1.3 软件生命周期与项目管理 .....	211
9.2 软件项目启动 .....	212
9.2.1 软件项目启动任务 .....	212
9.2.2 软件项目可行性分析 .....	214
9.2.3 制订项目任务书 .....	215
9.3 软件项目计划 .....	216
9.3.1 软件范围计划 .....	217
9.3.2 项目进度计划 .....	217
9.3.3 项目成本计划 .....	222
9.3.4 项目风险计划 .....	228
9.3.5 项目合同计划 .....	231
9.3.6 人员与沟通计划 .....	232
9.4 软件项目执行控制 .....	235
9.4.1 软件项目控制方法 .....	235
9.4.2 软件项目控制过程 .....	236
9.5 软件项目收尾 .....	237
9.5.1 软件项目收尾概述 .....	237
9.5.2 软件项目收尾过程 .....	237
9.5.3 软件项目验收 .....	238
9.6 软件项目管理工具 .....	238
习题 9 .....	239

第 10 章 软件质量 .....	240
10.1 软件质量概述 .....	241
10.1.1 软件质量概念 .....	241
10.1.2 软件质量成本 .....	242
10.1.3 软件质量模型 .....	242
10.2 软件质量管理过程 .....	245
10.2.1 软件质量计划 .....	245
10.2.2 软件质量保证 .....	246
10.2.3 软件质量控制 .....	248
10.2.4 软件过程改进 .....	249
10.3 软件质量度量 .....	249
10.3.1 软件质量度量概述 .....	250
10.3.2 软件项目质量度量 .....	251
10.3.3 软件产品质量度量 .....	252
10.3.4 软件过程质量度量 .....	253
10.3.5 软件缺陷度量 .....	255
10.4 软件质量工具 .....	256
习题 10 .....	256
参考文献 .....	257

# 第 1 章 软件工程过程

软件工程过程由一组相互关联的活动组成。这些活动将一个或多个输入转化为输出，同时消耗资源来完成转换。传统工程领域（如电子、机械、化学）的许多过程涉及将能源和物理实体从一种形式转变为另一种形式，例如，水力发电的大坝将势能转化为电能，石油精炼厂利用化学过程将原油转化为汽油。

软件工程过程关心的是软件工程师对软件的开发、维护和操作如何完成工作活动，如需求、设计、结构、测试、配置管理和其他软件工程过程。为了便于阅读，本书将“软件工程过程”称为“软件过程”。需要注意的是，“软件过程”表示工作活动，而不是实现软件的执行过程。

软件过程的具体目的是：促进人们之间的理解、沟通和协调；协助管理软件项目；以有效的方式衡量和提高软件产品的质量；支持改进过程；为过程执行的自动化提供基础。

本章的章节结构如图 1-1 所示，将从软件过程定义、软件生命周期、软件过程评估与改进和软件过程工具 4 个方面介绍软件过程。

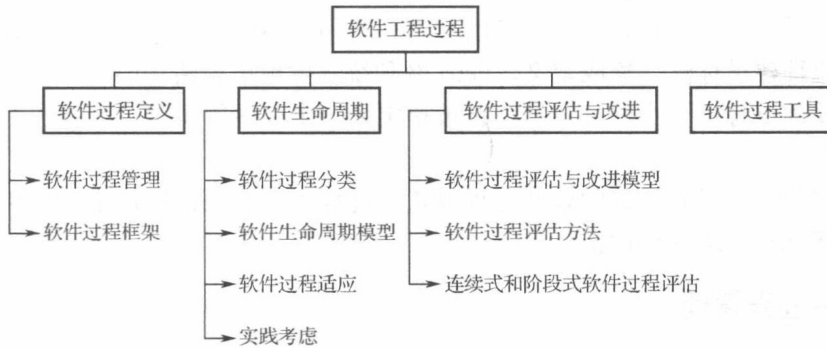


图 1-1 章节结构图

## 1.1 软件过程定义

如前所述，软件过程是一组相互关联的活动，将输入工作产品转换为输出工作产品。至少，软件过程的描述包括所需的输入、转换工作活动和生成的输出。如图 1-2 所示，软件过程可能还包括其输入和输出标准，并将工作活动分解为任务，这是软件过程管理的最小单位。软件过程输入可能是触发事件或另一个软件过程的输出。输入标准应该在一个软件过程开始之前得到满足。在成功地完成一个软件过程之前，必须满足所有指定的条件，包括输出工作产品或工作产品的验收标准。

软件过程可能包括子过程。例如，软件需求验证是一个过程，用来确定需求是否为软件开发提供了足够的基础，它是软件需求过程的一个子过程。软件需求验证的输入通常是软件需求规范和执行验证所需的资源（人员、验证工具、足够的时间）。软件需求验证的任务可能包括需求审查、原型和模型验证。这些任务包括个人和团队的工作任务。软件需求

验证的输出通常是一个经过验证的软件需求规范，它为软件设计和软件测试过程提供了输入。软件需求验证与软件需求过程的其他子过程通常会以不同的方式进行交叉和迭代。软件需求过程及其子过程可以在软件开发或修改过程中多次输入或输出。

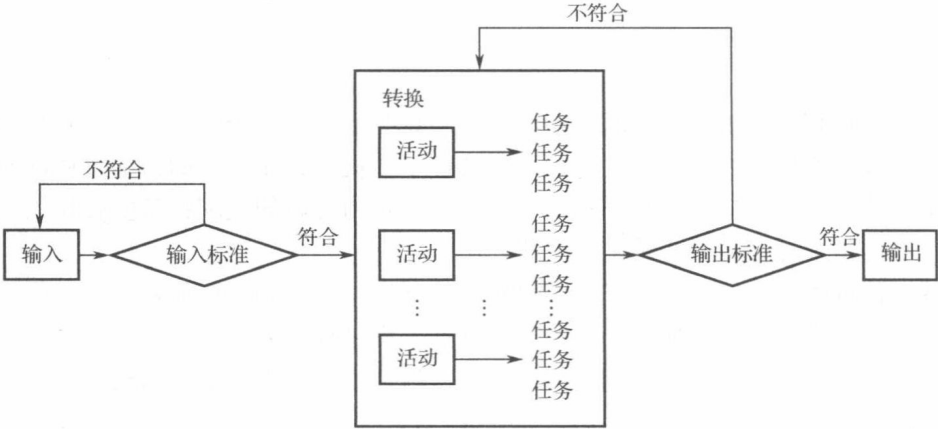


图 1-2 软件过程流程图

软件过程的完整定义可能还包括角色和能力、IT 支持、软件工程技术和工具、执行过程所需的工作环境、用于确定执行过程的效率和有效性的方法与度量。

此外，软件过程可能包括交叉技术、协作和管理活动。

定义软件过程的符号包括组织活动的文本列表和自然语言描述的任务、数据流图、状态图、业务流程建模标注、集成定义、Petri 网和统一建模语言活动图。流程中的转换任务可以定义为软件过程：一个软件过程可以被指定为一个有序的步骤，或者作为待完成任务的工作检查表。

必须强调的是，没有最好的软件过程或软件过程集，必须根据每个项目和每个组织的内容选择、调整和应用软件过程。

### 1.1.1 软件过程管理

软件过程管理的两个目标是，实现软件过程的效率与有效性和生产工作产品的系统方法的效率与有效性，无论是在个人、项目或组织层面，还是在引入新的或改进的过程方面。

过程随期望而改变，一个新的或修改后的过程将提高过程的效率与有效性，同时提高所生产工作产品的质量。引入一个新的过程、改进现有的过程或者改变现有的组织和框架（技术插入或工具的改变）是密切相关的，因为所有这些的目的通常都是为了改进软件产品的成本、开发进度或质量。过程的改变不仅对软件产品有影响，还经常会导致结构的改变。改变一个过程或引入一个新过程会在整个组织结构中产生连锁反应。例如，对 IT 基础设施、工具和技术的更改通常需要过程改变。

在第一次部署新过程时，可能会修改现有的过程（例如，在软件开发项目中引入检查活动可能会影响软件测试过程——参见第 6 章和第 10 章）。这些情况也可以称为“过程演化”。如果修改是广泛的，那么可能需要在组织培养和业务模型中进行更改以适应过程改变。

### 1.1.2 软件过程框架

建立、实施、管理软件过程和软件生命周期模型通常发生在单个软件项目的层次上。

然而，在组织中系统地应用软件过程和软件生命周期模型将有益于组织中的所有软件工作，尽管它需要在组织层面上的协议。软件过程框架可以提供过程定义、解释和应用过程的策略，以及用于实现过程的描述。此外，软件过程框架还可以提供资金、工具、培训和工作人员，这些人员被分配负责建立和维护软件过程框架。

软件过程框架因组织的规模和复杂性及组织内的项目而异。小型、简单的组织和项目有小而简单的框架需求，大型、复杂的组织和项目必须有更大、更复杂的软件过程框架。在后一种情况下，可以建立各种组织单元（如软件工程过程组或指导委员会）来监督软件过程的实现和改进。

一个常见的误解是，建立一个软件过程框架和实现可重复的软件过程将增加软件开发和维护的时间及成本。当然，引入或改进软件过程需要付出一定的代价。然而，经验表明，通过提高效率、避免返工，以及使用更可靠和可负担得起的软件，以系统地改进软件过程，往往会降低成本。因此，软件过程性能影响软件质量。

软件过程框架定义了若干框架活动，为实现完整的软件工程过程建立了基础。这些活动可广泛应用于所有软件开发项目，无论项目的规模和复杂性如何。此外，软件过程框架还包含一些适用于整个软件过程的普适性活动。一个通用的软件过程框架通常包含以下 5 个活动。

#### （1）沟通

在技术工作开始之前，和客户（及其他干系人）的沟通与协作是极其重要的。其目的是理解干系人的项目目标，并收集需求以定义软件特性和功能。

#### （2）策划

如果有地图，那么任何复杂的旅程都可以变得简单。软件项目好比是一个复杂的旅程，策划活动就是创建一张“地图”，以指导团队的项目旅程。这张地图称为软件项目计划。它定义和描述了软件工程师工作，包括需要执行的技术任务、可能的风险、资源的需求、工作产品和工作进度计划。

#### （3）建模

无论你是工程师、建筑师还是木匠，每天的工作都离不开模型。你会画一张草图来帮助理解整个项目大的构想——体系结构、不同的构件如何结合，以及其他一些特性。如果需要，可以把草图不断细化，以便更好地理解问题并找到解决方案。软件工程师也是如此，需要利用模型来更好地理解软件需求，并完成符合这些需求的软件设计。

#### （4）构建

必须要对所做的设计进行构建，包括编码（手写的或者自动生成的）和测试。后者用于发现编码中的错误。

#### （5）部署

部署是指软件（全部或者部分增量）交付给用户，由用户对其进行评测并给出反馈意见。

上述 5 个通用软件框架活动既适用于简单小程序的开发，也可用于 Web APP 的建造，以及用于基于计算机的大型复杂系统工程的实现。在不同的应用案例中，软件过程的细节可能差别很大，但是软件框架活动类型都是相同的。

对许多软件项目来说，随着项目的开展，软件框架活动可以迭代应用。也就是说，在项目的多次迭代过程中，沟通、策划、建模、构建、部署等活动不断重复。每次项目迭代

都会产生一个软件增量，每个软件增量都实现了软件的部分特性和功能。随着每次增量的产生，软件将逐渐完善。

同时，软件过程框架活动由很多普适性活动来补充实现。通常，这些普适性活动贯穿软件项目始终，以帮助软件团队管理与控制项目的进度、质量、变更和风险。典型的普适性活动包括如下活动。

(1) 软件项目跟踪和控制：项目组根据计划来评估项目进度，并且采取必要的措施来保证项目按进度计划进行。

(2) 风险管理：对可能影响项目成果或者产品质量的风险进行评估。

(3) 软件质量保证：确定和执行保证软件质量的活动。

(4) 技术评审：评估软件工程产品，尽量在错误传播到下一个活动之前发现并清除它。

(5) 测量、定义和收集软件过程、项目及产品的度量：帮助团队在发布软件时满足干系人的要求。同时，测量还可与其他软件框架活动和普适性活动配合使用。

(6) 软件配置管理：在整个软件过程中管理变更所带来的影响。

(7) 可复用管理：定义工作产品复用的标准（包括软件构件），并且建立构件复用机制。

(8) 工作产品的准备和生产：包括生产工作产品（如建模、文档、日志、表格和列表等）所必需的活动。

软件过程框架如图 1-3 所示。可以看出，每个框架活动都由一系列软件工程动作构成：每个软件工程动作都要由一个任务集来定义，这个任务集明确了将要完成的工作任务、将要生产的工作产品、所需要的质量保证点，以及用于表明过程状态的项目里程碑。



图 1-3 软件过程框架

在软件过程中，使用过程流来描述对软件过程框架中的活动、动作和任务如何在执行顺序和执行时间上进行组织。常用的过程流包括线性过程流、迭代过程流、演化过程流和并行过程流。线性过程流从沟通到部署顺序执行 5 个活动，如图 1-4 所示。迭代过程流在执行下一个活动前重复执行之前的一个或多个活动，如图 1-5 所示。演化过程流采用循环的方式执行各个活动，每次循环都能产生更完善的软件版本，如图 1-6 所示。并行过程流将一个或多个活动与其他活动并行执行，如图 1-7 所示。

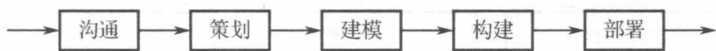


图 1-4 线性过程流

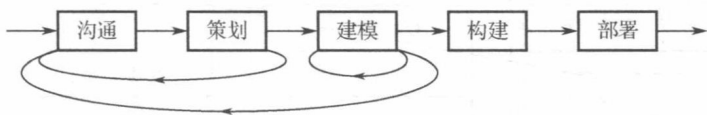


图 1-5 迭代过程流

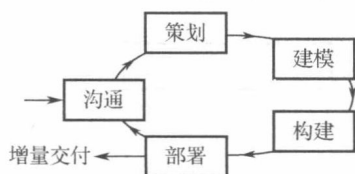


图 1-6 演化过程流

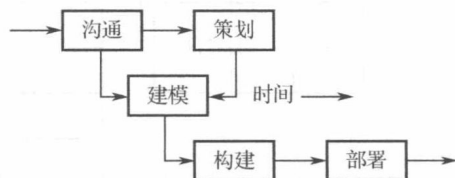


图 1-7 并行过程流

## 1.2 软件生命周期

软件开发生命周期（Software Development Life Cycle, SDLC）是软件的产生直到报废的生命周期。软件产品生命周期（Software Product Life Cycle, SPLC）包括软件开发生命周期，再加上为软件产品的部署、维护、支持、演化、退役和所有其他从开始到退役过程提供服务的软件过程，以及应用于整个软件产品生命周期的软件配置管理、软件质量保证过程等。一个软件产品生命周期可以包括用于演化和增强软件的多个软件开发生命周期。

单个软件过程没有时间顺序。软件过程之间的时间关系是由软件生命周期模型提供的：要么是软件开发生命周期，要么是软件产品生命周期。软件生命周期模型通常强调模型内的关键软件过程，以及它们在时间和逻辑上的相互依赖关系。在软件生命周期模型中，软件过程的详细定义可以直接提供，也可以参考其他文档。

除在软件过程中传递时间和逻辑关系之外，软件开发生命周期模型（或组织中使用的模型）还包括应用输入和输出标准的控制机制（如项目评审、客户评估、软件测试、质量阈值、项目演示和团队共识）。一个软件过程的输出常常为其他软件过程提供输入（例如，软件需求为软件架构设计、软件构建和软件测试等过程提供了输入）。并发执行多个软件过程活动可能产生一个共享的输出（例如，不同团队开发的多个软件组件之间的接口规范）。一些软件过程可能被认为不那么有效，除非同时执行其他的软件过程（例如，软件需求分析过程中的软件测试计划可以提高软件需求质量）。

## 1.2.1 软件过程分类

在软件开发和软件维护生命周期的各个部分中，已经定义了许多不同的软件过程。ISO 和 IEC 联合推出了“ISO/IEC 12207 软件生命周期过程”标准，为开发和管理软件提供了标准公共框架，如图 1-8 所示。

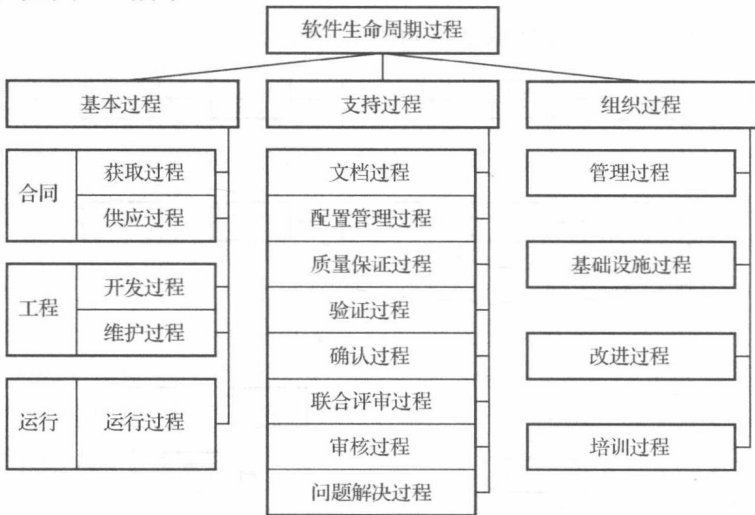


图 1-8 “ISO/IEC 12207 软件生命周期过程”标准公共框架

### 1. 基本过程

基本过程定义了与软件生产直接相关的过程，即软件从无（或原有）到（新）有到运营的过程，包括软件的获取、供应、开发、运行和维护。

(1) 获取过程，是指获取方为得到一个软件产品所进行的一系列活动，包括确定获取产品的需求定义、投标准备、合同准备和修改、对供应方的监督及验收完成和结束。

(2) 供应过程，是指为获取方提供软件产品所进行的一系列活动，包括理解产品需求、应标准备、合同签订、计划制订、实施和控制、评价及交付完成。

(3) 开发过程，是指组织开发软件所从事的一系列活动，包括需求分析、系统设计、编码、测试、安装及验收。在开发过程中还贯穿了其他软件过程的实施。

(4) 运行过程，是指操作人员日常使用软件的过程。该过程是指用户和操作人员为用户业务运行环境中，使用软件投入运行所进行的一系列活动。其目的是在软件开发过程完成后，将软件从开发环境转移到用户的业务环境中运行时，对用户的要求提供咨询和帮助，并对运行效果做出评价。这个过程为开发过程和维护过程提供反馈信息。运行管理方可根据对软件项目的总体要求，按照软件管理过程的内容对运行过程进行管理。

(5) 维护过程，是指维护人员所从事的一系列活动。其目的是在保持软件整体性能的同时进行修改，使其达到某一需求，直至其被废止。维护包括改正性、适应性和完善性维护。维护过程包括过程实现、问题分析与修改、修改实施、维护评审和验收、移植和软件退役等。在维护过程中还贯穿了其他软件过程的实施。

### 2. 支持过程

支持过程是指为了保证基本过程的正常运行、目标的实现和质量的提高所从事的一系列过程。它们可被基本过程的各个过程部分或全部采用，并由它们自己的组织或一个独立